Security Check-In Station

A Major Qualifying Project
Submitted to the faculty of Worcester Polytechnic Institute
In partial fulfillment of the requirements for the Degree of Bachelor of Science

William Ross Blackmar

Robert Weir

Chatura Weliwetigoda

Submitted to Professor Robert Labonte,
Department of Electrical and Computer Engineering

July 28, 2006

Abstract

This major qualifying project is a culmination of lab and course work that has been done over four years. The Security Check-In Station is a device which communicates with a central server to give access to guards based on RFID badge verification and voice authentication. The device is designed to have guards check in with the central server showing the patrolled area. By using RFID tags and scanners, and using signal analysis techniques like frequency comparing and signal covariance, the device is able to distinguish guards from imposters. The client helps all the components to communicate with one another.

Executive Summary

1. Introduction and Purpose

Security has always been the front runner nearly everything is this new day and age. It dictates the well being of large businesses, small business and even homeowners. Security technology will always have to be redefined and revamped as time goes on. There are many different devices and systems out in the market that deal with security. Though all of them achieve the same result, their methods of achieving it are for the most part very different.

The security check-in station is a wall-mounted device that will identify a guard from an imposter by implementing RFID badge verification and voice authentication. The purpose for designing and implementing the device was so it could compete with similar products out in the market today. The cost of the security check-in station will be much less than most of the products already out on the market, which will make it a sought after product. Though the product costs less that does not mean that its reliability can be put into question; the reliability of the device is comparable to already existing products which are on sale for much more.

Though the security check-in station might seem like a cheaper version of already existing products, the thought process for its design was very different. The device was intended for patrolling guards. Multiple units would be placed in a secure area, and each guard will "check-in" at each station when he or she approaches it. This will give the central server all sorts of information like which guard it specifically is, whether or not he or she passed the voice authentication section, how long will it take the guard to get to the next station. This device was designed to keep track of guards and the areas of patrol, so that the customer can make sure that all areas are being looked after.

2. Methodology

There were three major sections that had to be developed for the security check-in station to be fully functional; the RFID verification portion, the voice authentication portion, and the client and server portion. Each of these three sections play an important role for the functionality of the device. The three sections were as follows: RFID authentication, voice authentication, client and server.

The RFID portion was required to read in a specified number of characters to the screen for comparison against numbers stored on the server. After finding a reader that would do as such, the focus of the portion was converting from the reader output to a computer. This was done with RS 232 conversion and use of a serial port. Within the computer, the number is read from the port and then compared to existing values stored on it. Much of this portion was done on a discrete hardware level.

The voice authentication section uses two signal analysis techniques which can differentiate between two different voices. The first part of the voice authentication process was the addition of the frequencies. Through research and testing it was found that different people output certain phrases and words at different frequencies. By taking the FFT of a normalized speech signal, a number of frequencies at different points could be found. By finding the sum of all of the un-interpolated speech signals, we were able to record that each individual had a unique sum of frequencies. When this sum of frequencies was compared to another individual, we would notice that there would be a considerable difference at times. When the individual was compared to him or herself, the percent difference between the two sums of frequencies was much lower. This was the primary signal analysis technique used for the voice authentication. The sum of frequencies was not able to produce a substantial percent difference between two individuals that sounded alike.

The secondary technique used for voice authentication was the signal covariance technique. This technique allowed us to see the number of variances of both the normalized raw signal and the FFT signal. Again, through research and testing we noticed that when an individual was compared to him or herself, he or she will have a greater percent match with themselves; but when two different individuals were compared it was noticed that the percent matches were much lower.

By combining these two techniques, we were able to come up with a solid voice authentication system. Taking the percent match from the sum of frequencies and the signal covariance and averaging them achieved a final percent match. This average produced results that were very favorable for the voice authentication portion of the project. The cutoffs were set as the following:

- 79% or lower – Fail
- 80%-84% - Retry
- 85% - above – Pass

If a guard was prompted to retry, he or she would get another two chances to authenticate their voice correctly. If the guard fails after two more tries, the central server is notified.

Though all the different components wee designed for one system, the methodologies used to produce them were entirely different. The methodology for producing the client and server side of the project was quite different from the RFID portion and the voice authentication portion. There were four major sections that had to be developed for the client and server component. These four sections are as follows:

- User Management
- Client Management
- Event Management
- Diagnostics

User management was used to add and maintain users in the system. A new guard could be added into the system through the server. The server keeps a database of every guard that is registered in the system; therefore, a new guard would have to register before he or she can actually start patrolling. Information like the guard's name, badge number, age, eye color, and hair color would all be inputted in this stage. A picture of the guard along with two pre recorded sound files will also be uploaded into their profile. The user management section can also be used to update information, like deleting guards no longer working for the company or updating a guard's age.

In order to have sufficient security for a pre-designated area, companies will have to purchase more than one security check-in station. This will give the added security that they are looking for. The Client ID and Key are used to authenticate each Client (Check-In Station) into the system. When a guard checks into a station the client will identify itself to the server and then show who is trying to access it, and whether or not the particular guard passed.

Event management works hand in hand with the client. If the guard does not reach a particular station in the time allotted, an email will be sent to the security supervisor.

This feature was added to make sure that guards have checked into a station by a specific time. This allowed security managers to focus on other tasks knowing that they would apprised of any problems with their patrols.

The diagnostic section was included for mainly maintenance issues. Diagnostic tests can be run in order to test the system and checking it for any problems. By testing the client interface, any problems that the check-in has on the server can be viewed and also rectified. Diagnostics add an extra layer of security into the system; it allows the system to be checked for any weaknesses, and once the weaknesses are identified they can be fixed.

3. Results

After considerable testing and recalibration, the RFID portion worked as expected for extended periods of time. All data input to the system through a card was verified in different operating systems to ensure there was no mixed data. The product also did well in longevity testing, ably working for over a week with out power issues.

There were five experiments that were conducted for the voice authentication portion of the project. Each one of these experiments tested different aspects of the voice authentication system. The reason that each test was different was to see which method of implementation would yield the most desirable results. The first experiment consisted of three test subjects; these subjects were asked to stand about half a meter away from the microphone, while the microphone was about two feet below them. The subjects had to talk down into the microphone. The results produced from this experiment were undesirable because the percent matches were all too low. The average percent match was about 45%.

Experiment two had the three subject sitting down facing the microphone, and also about five inches away from it. This experiment yielded better results but they were still too low. The average percent match was around 60%-65%. The results from this experiment helped us find the flaw with experiment one. It was concluded that the guards should be able to speak directly into the microphone. This would allow the speech signal to be inputted much more clearly.

Experiment three consisted of three subjects once again, but this time the signal was normalized. After normalizing the signal, the results for percent matching shot up drastically. The average results were now about 80%-85%. After reviewing the results of this experiment it was concluded that normalizing the speech signal was another step that had to be added into the algorithm.

Experiments four and five, were experiments dealing with sum of frequencies and signal covariance. By running may tests it was concluded that the sum or frequencies and the signal covariance comparisons were the best way to achieve a reliable and accurate voice authentication system.

4. Conclusion

Through research and testing, all the sections of the project were completed to the design's specifications. The RFID scanner takes the ID Tag number from a guard and it is output into the server. The client and server verify this number and then prompt the voice authentication section to start. If the guard passes he or she will be allowed to move onto the next section. If they fail however, the central server will notify the authorities as a possible intruder. Thought he project was successful there are a number of recommendations that can be made for future groups that want to embark ona similar project.

5. Recommendations

- **RFID should be cased**

  With a more portable case, the unit will be able to withstand considerable damage and still function normally. With wiring simplified, the unit can also be as compact as its largest item, the RFID antenna.

- **Experiments should always be performed in a controlled environment**

  At times the results of experiments were very unfavorable due to all the background noise that was present in the labs. If conducting tests like the ones performed for voice authentication, make sure you do it in a place where background noise is low.

- **Make sure to order multiple parts**

  The new RFID scanner that was bought for the project malfunctioned after a week of using it. Another RFID scanner had to be purchased, but it wasted precious time. If ordering a part, try to order at least one more for safety precautions.

Table of Contents

6.

1. Introduction

Security has been one of the major issues for big businesses, small businesses and home owners alike. As time goes on, new technology is developed to maintain security easily and efficiently. The project that has been presented is to design a new alternative to secure buildings. The design will be a security checkpoint for large buildings, areas etc, will be a security checkpoint for guards to check in to as they patrol their respective areas. The security checkpoint will consist of a Radio Frequency Identification (RFID) tag which will be given to each guard so that no unauthorized user may falsify the guards' identity. The security check-in station will require the guard to first have to run the RFID tag through the RFID scanner, and then the guard will be prompted to input a password into the device for voice authentication. Once the guard has passed the voice authentication, he/she will be allowed to proceed onto the next check-in station. Background research on already existing products is available in the proposal, which is located in Appendix A. The proposal highlighted all the features and components that the security check-in station would have, but for time and budget purposes there were changes that had to be made. The original device implementation is available on the proposal in Appendix A.

2. Project Changes

There were many changes that the project undertook because of time and budget constraints. The budget affected the project the most. The team was given $375 to complete the project, but we needed about $600 to purchase all the parts necessary to put the security check-in station together. The budget would did not leave any room to buy items like the embedded PC which around $350. Since the budget was a major issue, we got permission to do a proof-of-concept project. The proof-of-concept project gave the team the flexibility to use a regular PC instead of an embedded PC. When using a regular PC, we had no need for the LCD, the microphone and the speaker, because the PC would provide all those materials. The RFID tags and scanner was the only piece of equipment that had to be purchased.

The proof-of concept project also benefited the team because of time constraints. The team was given about seven weeks to complete the project, by the time that all of the separate components were functional; the team would not have had time to put all the components together and test the device. By making the project a proof-of-concept project, the team would be able to show complete functionality of the device, without having to worry about putting a physical device together. The team also had to purchase an alternate RFID scanner and RFID tags because of time constraints. Since the original RFID tags and scanner produced by ACG would not arrive till the end of July, a new RFID tags and scanner were bought from Parallax.

The voice authentication algorithm went through a few changes. In the proposal, it was stated that the team would try to find an already existing algorithm for the voice authentication portion of the device, but due to the lack of compatible systems out there, a voice authentication algorithm had to be created from scratch. Since the purchase of the microphone and speaker were also no longer needed, the budget that was specified in the proposal was not exceeded. When creating the voice authentication algorithm, there were many changes that it had to go through, this is documented later in this report. Though the voice authentication portion was more work that the team originally had planned, it did not take any extra time to implement. The time allotted to the project was enough to implement a fully functional voice authentication algorithm.

Although the project went through many changes, the functionality of the device is the same. The major difference that between the proposal and the final project is that a prototype security check-in station is not going to be built. If more money was allotted in the budget, and more time was available, the team would have definitely been able to produce a prototype.

3. RFID Overview

The option to add a radio frequency identification module added another level of security to the system intended to further deter anyone trying to break into the system. RFID, as it is more commonly known as, has been around since the 1950s, but common civilian uses for it have only been around for a couple decades. For this project, it was seen as a replacement for magnetic stripe cards, which were also considered. Other than this invisible reader, there will be no other way to begin the voice authentication and eventual security authorization.

RFID requires two units. The first is a receiver, normally a powered unit that remains stationary and is connected to a larger network or server. This serves as a compact intermediary so that the large server can be stored somewhere more secure. The other half of RFID communication is the transmitter, which comes in two different types. Active, which has its own power thus increasing range, and passive, which is powered by the signal sent out by the receiver. The power essentially wakes up the passive unit and has the information stored on it sent to the receiver.

4. RFID Design

4.1. Original Design

As laid out in the proposal, we were to use an RFID reader that was manufactured by ACG, a German wireless communications company. The software of the device was the determining factor in the purchase of the RFID reader, which included support for Linux and RF dump, a means by which the computer receives the information and can quickly use it. Since we would be getting an embedded computer with compact flash, or a CF slot, we chose this as an interface for the RFID reader.

The CF slot would have served dual purpose, the first would be load the operating system onto the embedded computer, and the second would serve as a permanent bay for RFID communication. The unit itself measures approximately 3x1.5x0.25 inches and only about half of it sticks out of the slot.

Communication with ACG began in April to get the unit best suited for the project by the time most work on the project was to be completed. After three weeks, we heard a response from one of their American sales representatives. We presented through e-mail our intentions with their product, but the module was put on hold pending fiscal limitations. When dialogue resumed toward the close of May, the product cost was justifiable as it was to be paid for outside of the MQP operating budget.

The product was selected and the choice sent back to ACG in an attempt to at least get the product by mid-June, this again ran into changes that had developed over the two months of communication between the project and the company.

4.2. Changes to Original Reader

Budget constraints played another role in the existence of a embedded computer for the project. Most embedded boards now come with compact flash slots for an easier means of loading software and any operating system onto the board without too much trouble. With a limited budget and some board prices exceeding $250, the portion was discarded in favor of using a standard size computer. This was seen as a fair cut because the project is more interested in functionality rather than form.

Because of this, it was determined by the team that a USB RFID reader would suffice and would not be extra since all the machines we were using had USB support and not CF support. Additionally, we were notified by ACG that the CF card reader was under development to be released a second time. For the project, this new release would not be on time and we would not be able to get the old version until the beginning of July. Because the project calendar does not include very much time past July, we could not use this.

4.3. Dissolution with ACG

From that point, we hoped that within two weeks, we would be able to receive the product and be working with it by the 21st of June. We were notified again that the product we were seeking would still not ship from Germany until the end of July. The USB model was out of stock so we asked to switch back to the older model of the CF reader. This was too no avail, the product was still not available and would not be able to ship until the same time in July as the USB product.

With such a limited schedule, we could no longer deal with holdups caused by shipping times and unavailable products. We began looking for new products from other companies that we had originally found. Also, we searched on Mouser and Digikey, the department electronics suppliers. The first find was in stock at Digikey.

Made by Texas Instruments, the model was S4100 Multi-Function Reader Module. It employs different frequencies from the reader to allow it to read more than one standard of RFID transmitter. Texas Instruments does not use a standard for their low frequency mode, instead employing a proprietary 134.2 kHz band. In addition to this band, they comply with the ISO 15693 and ISO 14443 RFID standards, both high frequency at 13.56 MHz. They have followed this standard due to the proliferation in the common market of already existing high frequency cards.

The Texas Instruments unit was only one of two considered, the price is $88.66, and has an allowable purchase quantity of one. It was, however, more expensive than the unit we ended up deciding on. The low frequency Parallax RFID reader cost is half of that of the Texas Instruments unit, and serves the same purpose. The reason we decided to simplify down to this model is because we are able to control the quantity of RFID cards. If the system that was going into affect had to make use of existing cards, we would have to suffice, by using a more general reader. The cost of the Parallax reader outweighed its lack of high frequency reception, at only $39.00; we were also able to furnish six cards for testing well under the cost of the TI unit.

4.4. Parallax RFID Reader

At a cheaper cost, it does not have the features of the Texas Instruments unit and also falls short of the ACG unit. In addition to the lower cost, this module will now require much work to gain results we are looking for. The complete functionality of the reader and accompanying cards can be found in the datasheet, located in Appendix C. Simplified, the reader is a low power unit that works with passive transmitters. When the unit is on, it interrogates its surroundings for any transmitters. If any are found, they are transferred via a single data line output.

Additional work that must be done with this unit that was not necessary with the ACG unit was designing the method of communication with the computer. This includes converting the signal of the output to a nine-pin serial port recognizable to the computer.

The output of the reader is a simple twelve byte signal. The first digit is a start, 0x0A, a line feed followed by the ten-digit number of the card. The final digit is a stop digit, signified by 0x0D, a carriage return. Unlike other systems we searched for, this uses proprietary cards in the 125 MHz low frequency band. Normally, we attempt to shy away from any systems that will limit the project, but at the cost of $2.25 per card, it was beneficial to use this system.

Since the unit only has four pins, high, low, signal and ground. There are no ports that this can simply be plugged into the computer and used. The card does not require a large amount of power or transfer speed. At 2400 bps, we are able to use the serial port of the computer using converters. The first is a R232 that the reader can plug into for proper power. The R232 also has a data line output that we will use for the next stage of conversion. The data will be put through a Maxim MAX232A which can be used to convert the single signal to 9-pin serial.

4.5. Converting TTL to RS 232 Protocol

The following portion discusses the method we converted the output of the reader to a signal that the computer could easily understand. TTL or Transistor-Transistor Logic is based on simple computing methods that used AND, NOT, and other Boolean logic. Since our device outputs on a single serial output in TTL, we had to convert to something more understandable.

RS 232 is a communications standard that came out over 40 years ago. It was mainly developed around communications devices like modems that have since gone out of primary use in favor of the far smaller Ethernet RJ-45. RS 232 continued to move into human interface devices like game controllers, keyboard and mice, but also have been replaced by new standards like universal serial bus.

Our selection of RS 232 was due to the fact there were DE-9 connectors on the computers we were using in development. Universal serial bus, or USB, the more common connector used today could have been used, however far more development would have been necessary and time-consuming. RS 232 uses DE-9 connectors which are cost-effective and simple to develop and implement.

Using the MAX 232A, we are able to make the proper conversions from TTL to RS 232. From that point, we are able to connect the MAX 232A to the computer using a DE-9 connector. Because the way the device will communicate, it will not necessarily need all nine pins of the DE-9 connector. The next section on implementation will indicate wire connections between the three devices used.

5. RFID Implementation

There are two types of implementation discussed in this report. The first is functional implementation, where necessary actual connections are made and documented for repeatability. Schematics and diagrams are in this section. The second is form implementation. Time permitting; this section discusses transitions from a functional device to a marketable device. Plans for miniaturization including a printed circuit board and containment for the device are in this section.

5.1. Functional Level

An important intermediate step from on-paper design to prototype is functional level implementation. Most of the work we will be completing is this stage will be on a protoboard. It requires little extra work, no cost and is easy to change if any problems are encountered. The main parts used on the protoboard will be the MAX 232A chip, the Parallax RFID reader, and necessary in-line components such as capacitors and resistors.

5.1.1. Parallax RFID Reader

The RFID reader, which will be used to receive data from various RFID card transmitters that were purchased for the purpose of module testing, has four connections. From top to bottom, in reference on the breadboard, these pins are high reference, enable, signal out and ground reference. High will be tied to the high rail of +5V, supplied by a power supply. Ground will be tied to an adjacent rail at 0V, or 5V below the high rail.

The two other pins are of greater importance, the enable will be attached to the ground rail. According to datasheet specifications, a low logic is within .7 V of the ground reference, meaning that it does not necessarily need to be attached to that rail, however, for ease, we will implement it as such. This tolerance will also be useful in case there is some unexpected voltage draw in testing. When this pin is logic low, it turns the device on, causing it to send out a signal interrogating for any proximate RFID cards.

The final pin of importance is the signal out pin. This sends out the specified number of the card or transmitter. This will be connected to pin 10 of the MAX 232A. Before the signal is understood by a computer, it is unspecified highs and lows. The code is actually transmitted in twelve byte length ASCII characters. Only ten of these are used to transmit the actual code.

5.1.2. MAX 232A Integrated Circuit

The MAX 232A has two variations of model that could be used for this project. The first is the MAX 232, an older model that uses larger capacitors of varying size. Since this has become unavailable, we are able to use the 232A, which uses all the same size capacitors, 0.1 μF. Another variation of the 232A is the 233. This model of the capacitor has a greater number of pins for increased use. However, the benefit of this model over the one we are using is that it requires no external capacitors.

The project group had both the 232A and 233 models available. The 232A was available in a more easily used plastic package. The 233 was more difficult because it was a surface mount chip. For the purposes of functional implementation, the 232A will be used so there is no surface mount requirement.

The 232A uses four capacitors, all sized 0.1 µF. Because of their size, these are commonly found as polarized capacitors and have specified pin assignments for each end. Pins 1 through 6 are assigned for gaining power to supply the RS 232 protocol, which requires 10 V. A more detailed and specific schematic is provided in Appendix B.

Power is provided through two pins. The V_{CC} of the chip is off pin 16 and requires +5V. The V_{SS} or GND pin is pin 15 and is tied to 0V. These levels are the same as required for the Parallax RFID Reader, simplifying the power circuit. Power draw and current draw will be discussed in a later section.

Finally, two pins of the greatest importance will be the data pins. Pin 11, signified as T_{1IN} is the input from the RFID reader in TTL. Through this device, it is converted to RS 232 and output to pin 14, T_{1OUT}. This pin will be connected to the DE9 connector.

There are several other pins on this device that will not be used. Two pins service T_2, another TTL to RS 232 line that serves the exact same purpose as the T1 line. Four other pins service the exact opposite purpose. These pins, are marked R_1 and R_2. They convert RS 232 protocol to TTL for easier logic manipulation. The MAX233 model also features four other pins that will not be used.

5.1.3. RS 232 and DE-9 Connector

This circuit will employ a DE-9 connector for serial data traffic. This 9-pin connector has multiple pins we will not use. Because of this, only the pins that we will use will be mentioned. The remainders are ignored.

Two pins of primary focus for us will be pins 2 and 5. Pin 5 is assigned as Receive Data Enable. This pin will be tied high as the device will require no other data to be output to it. Pin 2 is assigned as the Receive Data Input. This pin will be used to receive data output from pin 11 of the MAX232A. All other pins will be unused or tied low in case the serial port is confused.

5.1.4. Computer with DE-9 Port

With software already installed on the computer. We will connect the DE-9 to the computer and the program will read the number that is received on pin 3 of the DE-9 port. The number will be output into a file that will then be used by programs developed separately to verify that the device is one previously existent.

5.1.5. Power Supply

Both the RFID reader and the MAX 232A/233 require a +5 V and 0 V rail. For functionality, we will be using an in-lab power supply, GPS-3303. In other lab experiments, this supply has been reliable in keeping a constant supply at a desired voltage. To avoid spiking, there will be a small capacitor draining to the ground rail, as required in the specifications of the integrated circuit. The 5 V power supply will be attached to V_{CC} of the reader and pin 16 of the MAX232A chip. The ground will be connected to V_{SS}/GND of the reader and pin 15 of the chip.

5.2. Form Implementation

This portion of the implementation discusses changes to the functional design that will be incorporated into the product if it were to become marketable. This includes modifications to the prototype and integrated circuit.

5.2.1. Printed Circuit Board

If the product functions as desired, all components that are assembled on the protoboard will be incorporated and designed into a printed circuit board. This will require that all parts be surface mount or be soldered. The PCB would be placed parallel to the embedded computer with the reader pins bent to allow the reader to parallel the PCB. This compact design would allow for a simpler thin encasing.

5.2.2. MAX 233A

As an alternative for the MAX 232A, the MAX 233A is the chip we would use in the case that is product is placed on the market. The 233A is a 20-pin alternative that has integrated capacitors. The data pins change their assignments in the 233A. T_{1IN} is on pin 2 and T_{1OUT} is on pin 5. They would be connected the same way that the data pins on the 232A are. Power also works slightly differently. The 233A still requires external connections, even though there is no need for external capacitors. V_{CC} is on pin 7 and the two grounds are connected on pins 6 and 9. Pins 11 and 15, 10 and 16, and finally 12 and 17 are paired to build the external circuit. The only capacitor required is a 1.0 μF capacitor draining from V_{CC} to ground. All other pins are unattached or tied to ground.

5.2.3. Power Supply

The power supply in the actual unit will be a 300 W or less supply that uses 110/120 V AC from a standard circuit. Two connections will be used for the embedded computer and LCD screen. One of the +5 V connectors will be connected to the PCB along with its ground. From those pins, a voltage regulator will be placed to ensure an uninterrupted +5 V.

5.2.4. Other Module Parts

The DE-9 and RS 232 connections will remain the same, except that the DE-9 connector on the board will be complete and not partial like it is for the functional implementation. Connections on the reader and computer will remain the same.

6. Testing

Testing will be done in two different ways. There will be section testing and cascading testing. Each section has expected results, which we have gained from datasheets or part specifications. Each will be tested against these results to make sure they are working properly. The devices will then be tested connected through to the end, in stages.

6.1. Individual Tests

Only two parts of the module will be tested individually. The first is the RFID reader to make sure that it is in some way communicating with another device. The other will be for the form implementation, but will not be used in the functional implementation. It is testing the power circuit to ensure that there is an output of +5 V and ground of 0 V.

The other parts that will not be tested individually are the MAX 232A/233 and the computer. The 232A/233 cannot be tested without completing the connection, if successful; the number will invert the input. If the part is not successful, it will display a false number to the computer. If no number shows up, it is an indication that the computer connection through the DE-9 is not working.

6.2. Sequential Stage Tests

Another method the RFID module will be tested through is sequential stage testing. Each part of the module as it is added on is tested rigorously to assure that with added parts, the module still performs as desired. The stages begin with power, then reader, MAX 232 IC, DE-9 connection, and finally the completed module with correctly output number to the computer.

6.2.1. Power Module

Testing for this device will only take place if the form implementation is completed. Because there is no independent power supply except from a variable lab power supply, we cannot explicitly test our supply during functional implementation. The desired results for the power supply is that the output of +5V and 0V. If this value is not attained, or varies over a short period of time, a voltage regulator will be used in attempt to lock in that value.

6.2.2. RFID Reader

The reader's four pins were connected to their specified values, except for the signal out pin. The desired result of the reader is that the signal output will show a signal. Because we are not given the numbers of the cards, we were unable to decipher the digital signal easily.

6.2.2.1. Results

After connecting all four pins, V_{CC} and Enable to high, ground to low and signal out to an oscilloscope, we were able to attain some results. The test also helped us to find that the query rate of the RFID reader is fewer than two milliseconds. When displayed on the oscilloscope, there was a very distinct +5V high logic and 0V low logic. When this is passed through the next stage, the MAX232A, this will be inverted. This portion of the testing deemed the RFID reader functional and usable in further experiments

6.2.3. MAX 232A/233A

Of the sixteen pins, only ten will be used, most of them convert the +5V supply at V_{CC} to +10V, something required to convert to RS232. If the device works properly, results will show that the signal on one side of the converter, T_{1IN} will be inverted to its output, T_{1OUT}. This inversion is due to the device necessity to convert TTL to RS232.

6.2.3.1. Results

The reading of the scope following the MAX 232A was a +13V translation of the signal output from the RFID reader. This was the intended result, with +13 V as the logic high and -5V as the logic low. Although the result of +13 was higher than expected, the output is lower than the +15V maximum. This output will then be connected to a DE-9 I/O port on the computer. The +13V/-5V voltages are necessary to be read in RS232.

6.2.4. DE-9 Connection

Of the nine pins on the DE-9 connector, we will only be using one of them. Pin 2 is the receive data pin, normally marked by an orange wire. A raw wire connection on the protoboard and a DE-9 female connector on the computer will connect the two. The connection does not need to be powered and the result should be that the output is translated into something that can be read by the computer. Because the numbers of the cards were not provided, any number verifies that the system is communicating

6.2.4.1. Results

Using HyperTerminal, the communication settings were set to 2400 bps, the output rate of the RFID reader. After referencing the data settings, HyperTerminal was set 8-N-1, without any flow control. From here, we waved a card in front of the reader and the result immediately popped up on the screen. We then double-checked this by checking another 2.5x3.5 card, then a 50 mm round card. All of the cards checked out and the results were saved.

0F0296CD50
0F0296CD64
0F0296CF9E
04158DB7BA

04158DBE07
04158DC0E8

Each of the keys has a specific hexadecimal code that will be stored on the server and used to reference the card and identify its user. A similar test was done in Debian using ttysnoop.

7. Software Implementation

The final step in functional implementation is the program designed to read from the DE-9 port on the computer to the main program. There are two programs that have been developed for this module of the project. The polling program, the one that reads the port for an identification number, is inside the main program. This portion of the software implementation will only be discussing the polling program.

Polling the port requires a simple design. It is a three step process involving enabling a read to the port, a while loop which waits for an input from the port and a return of the result. Because, we are using Debian instead of windows for our server applications, the program must be written differently than how it was tested. In a windows platform, the port would be set and COM1 would be used. In Debian, the port is ttyS1.

In setting the port, we use all the same numbers that were given by the RFID datasheet. Because the settings call for 2400 bits per second and 8 bits, no parity and 1 stop bit, there’s no reason we should use anything else. The program primarily uses termios.h and strings.h for port controls. Designed using non-canonical input, which allows for a limit of characters, this program waits for 10 characters to be input. After that, all characters are ignored and the number is then printed out. For internal use in the larger program, it is not printed and is purely returned. The port settings are saved before being changed before the loop, then restored to their original state after the program completes.

7.1. Testing

The easiest way to see whether the program is working properly is to write it, compile it, and see if it does as intended. To test this, the program will be loaded onto the client computer and tested using the DE-9 connector to the back of the client. If it works properly, the program will compile, open and stay open until there is a card placed up against the RFID reader. At this point, the program will return a 10-digit number and will close.

A few errors could occur in testing. If not written correctly, the program could return a number but not close, holding the program in an infinite while loop. For our design, because we do not want a time-limit, this will be something to look out for, but not something we can remove.

7.1.1. Results

In the first round of testing, the output of the program was filled with extraneous data. When a timer was added by cutting the ground reference, the return contained a partial of the correct number from the beginning to the middle or the middle to the end. More tests will be done in an attempt to attain the desired result.

8. RFID Reader Reliability

The most likely place that the reader module would fail is in the software. Although the software is simple, the loop will not finish unless it gets exactly 10 characters as the program asks. This can be remedied, but we would not necessarily want it remedied. The second fault reducing reliability could come from the power supply. In testing we protected the system from reading a false logic low by pushing the voltage from 5.0 to 5.3V. In form implementation, the power supply would be a 5.0V, with some variability, primarily between 4.5 and 5.5V. Any negative variation would negatively affect the system possibly leading to false readings. In an attempt to remedy this, the MAX 232A pushes voltages beyond a necessary logic high of +10V to +15V. Due to the variance of the voltage, our system is pushed down to +13.8V, still good enough for the RS232 to read properly.

With these two precautions, the reliability will be very high contributing to a system that already has a high reliability.

9. Experimental Work

Because the RFID module of the project was almost entirely a discrete hardware design, there was a considerable amount of work done in lab. All of the equipment used was new and very reliable. If there was an error found, it could be quickly pinpointed and remedied. The following equipment was used in lab:

- Applied Technologies MSO6102A Mixed Signal Oscilloscope
- GW Instek GPS-3303 DC Power Supply
- GW Instek GDM-8304 Digital Multimeter

Other than those items, several items were required by the circuit to make all the necessary connections. The first of these items was a serial extension cable. This cable needed to be clipped at one end and the wires stripped and spread. This allowed us to only use two of the four wires that the cable contained. Ordinarily there are nine, but the cable we attained started with four. The two used were orange for communication and white for ground reference. Our project also required five capacitors. All hardware needed is outlined in the Appendix.

The majority of the experimental work began after the arrival of the RFID reader. Experiments were done as verification of functionality. This mainly involved connecting the power supply to its two wires and the oscilloscope to various locations on the proto board. To test the RFID, the oscilloscope was connected to the SOUT pin of the RFID reader. After this was verified, this was compared to the output of Pin 14 of the MAX 232A IC. These signals were then continually compared when the final connection to the computer was made to assure there was still a viable signal being communicated.

10. Voice Authentication Overview

The first major module that is being developed is the voice authentication module. It was selected first due to its complexity and its likely long development time. The parts used for this from a discrete standpoint are only a microphone and embedded computer. Although not needed in the final design, audio playback through speakers will also be used in development. The use of a speaker for the final design would be for calibration verification. This would be moved to a desktop computer to avoid the inflexibility of the unit.

Development of the module took place over several stages. The first included preliminary research. Using information found on the internet and other sources, we will be able to create our voice authentication module through successes and failures of other systems. The second stage is considered separation. After taking many samples, work will be done to ensure that voices of two different people do not match. Following separation, everything must be done to make sure two different samples from the same person cause a match. This part is known as verification and is the key to this module. Lastly, developments revolved around what the system does following verification.

11. Preliminary Research

In addition to prior art, research must be done into individual modules and they should be treated as if they were projects themselves. With that in mind, the team began developing the voice authentication module on systems that had succeeded before. For easier development, the team decided to use MATLAB as the tool of development. Packages and basic commands native to the program allow for easy comparison between two sound waves. From this, the choice was made to search for successful voice authentication programs in MATLAB. The following two examples were deemed feasible for application and are discussed.

The first was a simple code for comparing amplitudes of two input waves.[1] With this comparison, we would be able to develop the code further to allow for a success or failure type of verification. The code requires the wave to either be recorded into MATLAB or as into a .wav file using a different application. The resulting output of MATLAB is the percentage match of the two .wav files. A high number indicates that the two sound waves are close to being the same; a low number indicates that the .wav files are dissimilar or distinguishable.

The other example that was found is an actual program that utilizes other functions that the programmer has developed separately. When the program is run, using two .wav files as variables, and the output of the program is a percentage similarity between the two files, similar to the other one found. The downfall of this simple do-it-all program is the cost. The program itself is free, however, for development it would be

[1] http://www.contrib.andrew.cmu.edu/~jterlesk/robotics/voice/voice.html

prudent to view the code and be able to modify it to our specific requirements. The code costs $20 and was deemed an unnecessary cost to the project.

Using the code that was found in the first example, we were thinking of building a program in MATLAB that was of equal caliber to the program in the second example. Development of it will also be a good use of the skills learned in courses that used MATLAB.

Though the code that was found was useful, it did not fulfill all the functions that project would need it to perform. While doing more research into the subject, a paper was found done by students in the University of Pennsylvania[2]. This paper described their approach to voice recognition using MATLAB. Upon reading the paper, it was decided that trying to recreate the MATLAB code suggested in the paper would be a better plan of attack. The MATLAB code was not present in the paper, so it had to be recreated according to the description given in the paper. There were five subroutines two the whole program. These subroutines included:

- Wav_gather
- Wav_plot
- Wav_filter
- Peak_finder
- Peak_compare

These five subroutines were used to run the whole MATLAB algorithm. There were a few additions and deletions made to our algorithm, but it ended up being very similar. The peak_compare subroutine was taken out and combined with the peak_finder subroutine. The peak_finder subroutine contains a function called findpeaks.m3, which was created by Mike Brooks, a professor at the Imperial College in London, England. Though the peak_compare routine was taken out, it will be added towards the end of the project. This will be done to make the MATLAB algorithm easier to read and follow should another group try to look at our work.

[2] http://www.seas.upenn.edu/courses/belab/LabProjects/2001/be310s01t2.doc
[3] VOICEBOX: Speech Processing toolbox for MATLAB, http://www.ee.ic.ac.uk/hp/staff/dmb/voicebox/voicebox.html

11.1. Separation

11.1.1. Method 1

After complete development of the program, the design of the voice authentication module requires us to distinguish between the voices of two different people. Separation is being defined as voices between two people being less than 10% similar. This similarity will be the percentage found in the program. Results higher than 50% require more work to lower the number or the product will not be able to perform as intended.

Using multiple subjects, sound bytes were recorded of them saying key phrases that were also used in the development of the first example. The recordings have been made using the commands wavrecord, wavplay, and wavwrite in MATLAB. To begin, the recordings were monaural at a sampling rate of 22,050. To set a sound byte length, the number of 100,000 samples was selected. This number is used primarily for development, for application, we would not want to limit the number of samples. If the user were to speak slower, or be caught off-guard by the system, the sample would have to be rerecorded if there was an established limit.

In an effort to increase the separation between two voices, the method of recording was changed to stereo with a sampling rate of 44,100. This increased the amount of amplitudes will lower the amount of matches and in turn, lower the percentage. In addition to the sampling rate, we added a channel. This was done due to the fact that stereo recordings were used in the first example. In an effort to replicate their experiment, we attempted to match as many variables as we could.

Following testing using the code from the first example, we decided to begin anew using the code from the first example only as a source and not as a backbone to our code. Most of the variables remained the same, but it cleaned up the MATLAB code so that everything that was in it was something recognizable.

11.1.2. Method 2

The first method that was implemented did not work to the precision and success that the team would have liked it to. The results that were produced in the matches were too high; sometimes producing matches of 95% or higher with two different voice signatures.

The second method that was implemented was by directly following the paper written by the student of UPenn. A new method of recording the speech files was also found. By recording the speech signals with commands present in the audiorecorder block in MATLAB, we were able to attain a clearer speech signal. The audiorecorder function recorded the speech signals at 24 bits, using 100,000 samples, and it used a sampling frequency of 44100 Hz. These signals were also recorded in stereo instead of mono; the main reason for this was to get a more accurate speech signal. After the speech signals were recorded, the functions wavplay and wavwrite were used to save them onto the MATLAB directory. After the separation of these speech signals was competed, the raw signal was put into the MATLAB algorithm that had been created for voice authentication.

11.2. Matching

11.2.1. Method 1

Following the completion of separation, the field of testers for the voice authentication module will be narrowed and from the sample, multiple samples of the same phrases will be recorded. The same engine that was developed over the previous two stages will be used for this stage, except that we are looking for matching numbers in excess of 90%. This essential section must work in order for the module to succeed.

If the numbers are low, modifications will be made to lessen the amount of noise in the sound file. In essence, it is the opposite work that needs to be done in order to distinguish two different people's voices. Another difficult portion of the matching stage is that voices of the same person must match regardless of tempo. Although we have limited the length to 100000 samples, or 2.27 seconds, we have not created a minimum limit. If a user speaks too quickly, the design must account for it by either changing the tempo to something around 2.27 seconds or by having the user rerecord their voice.

11.2.2. Method 2

The implementation of the voice authentication program was put to the test once the separation of the speech signal was complete. The speech signal was first run through the wav_gather subroutine, which takes the signal and gathers different pieces of information about the signal. Some of the information that it looks for are the signal's bit rate, sampling rate, number of samples and the type of recording. From here it is sent to the wav_plot subroutine. The wav_plot subroutine sends the signal to the wav_filter subroutines where the signal is passed through a high pass filter. This filter attenuates all the high frequencies from the signal. After the signal is filtered, it is sent back to the wav_plot subroutine. The wav_plot subroutine plots the raw signal and filtered signal. This will give a visualization of the particular signal. Finally the signal will be sent to the peak_finder subroutine where it will run the filtered signal through the program. There will be twenty frequencies and twenty amplitudes that will be outputted. All the frequencies will be added together and they will be compared to the original recording that the guard has inputted. The same implementation will be done with amplitudes that are outputted. Once the percentages of the frequencies and amplitudes are found, they will both be averaged out.

11.2.3. Authentication

After the voice authentication sequence is completed, MATLAB must determine if the user that the file is being compared to is the user at the terminal. With a result of 85% or higher, MATLAB will output a success code to the operating system. From this, the user will then be told that they have successfully matched their voice.

For a result of less than 85%, but greater than 75% the user will be returned to the voice record prompt to try again. More than likely in this situation, a failure close to 90% would be caused by a severe noise disruption or by a poor recording. However, with a result of less than 75%, the user will be returned to the home screen and the result will be sent back to the server as a failure and a security warning. By having these percentage cutoffs, we are confident that the voice authentication part of the security check-in station will be more accurate. The voice authentication will take place after the guard has scanned his or her RFID tag.

12. Voice Authentication Experimentation

There were several experiments that were conducted to check the validity of the voice authentication algorithm. These experiments yielded varying results, which led to a change in the voice authentication algorithm. Each of the experiments were done with a minimum of three volunteers, and depending on the experiment, different aspects of the volunteers' speech was controlled to attain results.

12.1. Experiment 1

Experiment one used three subjects, and compared each one of their voices to themselves and one another. Since amplitude was also being taken to account at the time the subjects were directed to say two different phrases. Phrase one consisted of the subject saying, "This is what I sound like" and the second phrase consisted of the subject having to say "My name is 'respective name'". The results for experiment one can be seen in Appendix F.

The first trial was performed by having the subject stand up and talk down into the microphone from about half a meter away (about 20 inches). The trials yielded a lot of erroneous data. Sometimes the frequencies were close enough to each other to get a good match, and in other cases the frequencies were so far apart that the algorithm would not recognize a particular person; even when the subject was being matched to himself. Sometimes the frequencies would be too close for two different people and the algorithm would not be able to tell "who was who". Furthermore, the amplitudes were still too random, and controlling the output of the amplitudes was another aspect the team had to look into.

The second trial consisted of all the three subjects sitting down and saying each phrase from half a meter away (20 inches). The point of having the subject sit down was to see what matching changes would occur if the subject was in line with the microphone and speaking directly into it. This trail yielded more positive results. Instead of having matching percent of forty to forty-five percent; the second trial yielded matching of fifty to about sixty-five percent. The amplitudes were still uncontrollable; and at times were very close and other extremely distant.

The third trial was very similar to the second trial except for the fact that the microphone was now placed 5 inches away from the subject. The results from the third trial yielded higher matching percentages than the previous two trials. The matching percentages shot up to about seventy-five percent to about eighty-five percent. This trial produced more promising results.

The fourth trial conducted in experiment one consisted of taking the signals produced in trial three and normalizing them. When the signals were normalized, the results produced a solid eighty to about eighty-five percent matching. After the fourth trial was run, there were a few changes that were made to the voice authentication algorithm.

12.1.1. Algorithm Changes I

After obtaining the results from experiment one, the team decided to get rid of the amplitude component of the algorithm. It was concluded that the amplitude was a factor that was too hard to control. To control the amplitude, all the guards using the system would have to be trained to say their specific phrases with the exact same level of volume as they did with their original recordings. This not only proved to be inefficient but also unpractical. The team then decided on basing the matching using the frequencies alone.

12.2. Experiment 2

Experiment two was conducted very similarly to experiment one with only a few implemented changes. Experiment two used three different subjects, which were all male ages ranging from 20 to 22 years of age. Each of the subjects were given six phrases to record into the system. The phrases consisted of the following:

- Phrase 1 – Yellow
- Phrase 2 – 'Respective Subject Name'
- Phrase 3 - Sri Lanka
- Phrase 4 – 'Respective City of Birth'
- Phrase 5 – Worcester, Mass
- Phrase 6 – Land Rover

There phrases were reduced down to words. This was to check if the algorithm would recognize a particular subject based on only one or two words. Since amplitudes were no longer an issue, all this experiment measured was the frequencies on each individual subject. All the recordings for this experiment were done about 17 inches away from the microphone. This was done to keep the consistency of the experiment. Experiment number two can be seen in Appendix F.

The first trial results were high and very promising. The algorithm was able to recognize a certain subject when the subject was trying to authenticate himself. The percent matching ranged from 84% to about 95%. The second trial on the other hand, had varying results. The percent matching ranged from about 65% to about 98%. This did not cause too much concern as for only one of the subject yielded a percent match of 65%, the other two subjects yielded matching of 98 percent.

Throughout this experiment the algorithm had trouble authenticating the speech signals when the subject was required to say something with more than word. It was believed that this error was due to the number of frequencies that were being measured in the speech signals. It was after performing this experiment that the team decided to increase the number of points where the frequencies were measured. By doing so, all the percent matching increased dramatically. Out of the seventeen matches produced, ten of them were above 90%, five were above 80% and two were above 70%. These were the results that we needed to consistently match a subject with himself.

Although the results of this experiment were promising, there was one major problem. The problem was that the though the system was able to match an individual with just the frequencies of the speech signal, the algorithm was still not able to differentiate between two different types of people and differentiate between two different phrases. There were times where two completely different phrases would have similar frequencies, which would produce a high percent match. There were also times when two subjects generated a similar frequency which produced high levels of matching.

12.2.1. Algorithm Changes II

After obtaining the results from experiment two, there were more changes that were made to the algorithm. The first change was that the number of frequencies measured on a given speech signal increased from twenty measurements to thirty-two measurements. Another change that will be made is to find another method of comparing each of the speech signals. A method that has been experimented with is correlation, and comparing using MATLAB.

12.3. Experiment 3

Experiment number three was performed to check for any other weaknesses in the algorithm. All the subjects used in this experiment were male in their early twenties. The full results of experiment three can be seen in Appendix F. There were quite a few weaknesses even though the second experiment had desirable results. Once big weakness that was present was that the frequencies alone were not enough to solely identity one person. The algorithm was taking one frequency for every 50 Hz of the speech signal. This was done because the function used to find the frequencies and peaks of the signal (find_peaks), was set to interpolate all of the frequencies found in a certain range of signal. So every 50 Hz of signal, the function would take about 25 frequencies and interpolate them to make them one frequency. By disabling the interpolation that the function was doing, the function was able to display all of the frequencies. This new method was applied to the previous methods of experimentation. The results produced a higher percentage of matching when the subjects were compared to themselves. All the results that were yielded format his experiment were in the 90 percentile range, with only three matches in the 80 percentile range. When comparing the signals visually through graphs and charts, most signals matched up with the "counterparts" and mismatched with the other speech signals.

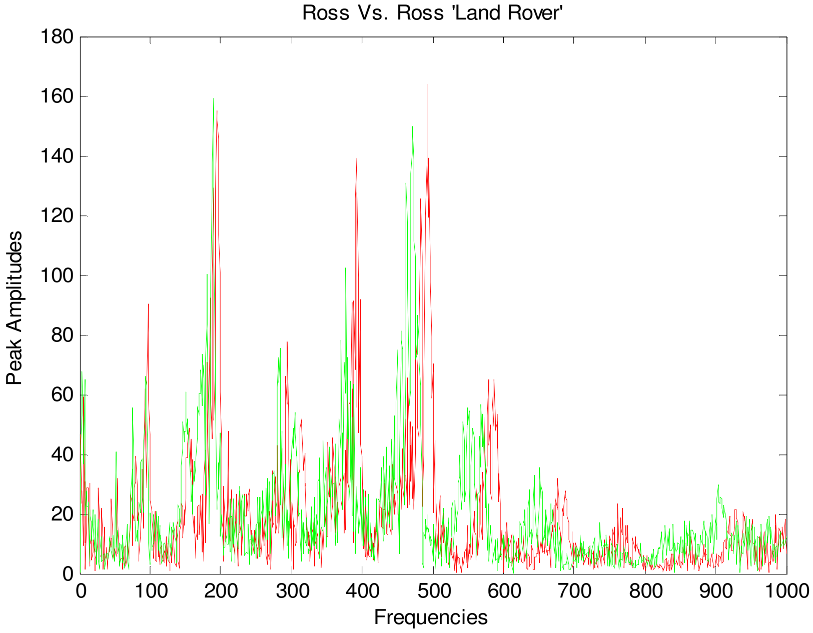


Figure 1 - Ross vs. Ross 'Land Rover'

As the above figure shows, there is a lot of matching between different speech signals. The two speech signals were taken from the same subject at two different instances; the subject was told to say the same phrase both times, in this case the phrase was ‘Land Rover’. The graph in figure 1 only shows the frequency spectrum to about 1000 Hz because it is easier to see the matching signals; but when two speech signals are compared, they are compared to about 30,000 Hz. The matching is not very similar when it comes to two different subjects saying the same phrase.

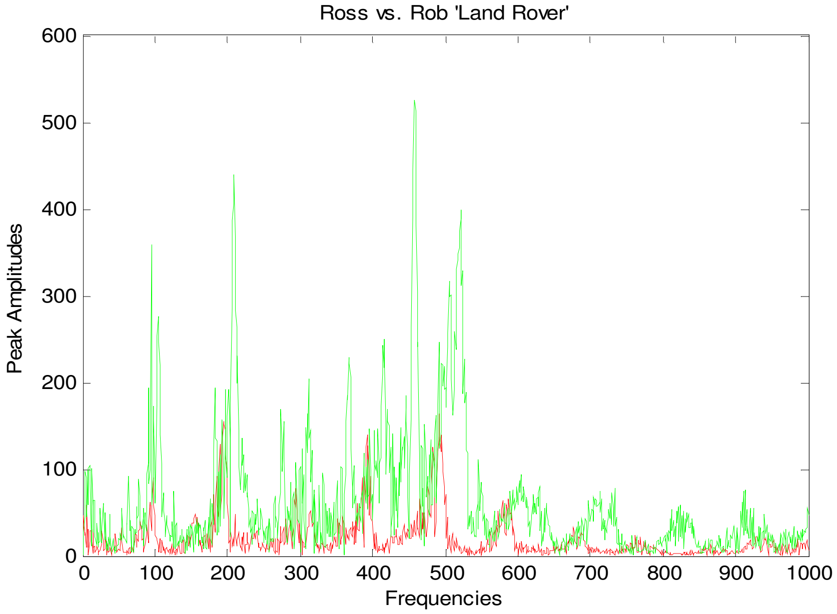


Figure 2 - Rob vs. Ross 'Land Rover'

Subject 1 which was used in figure 1 is in red while subject 2 is in green. As seen in Figure 2, the speech signals do not match to the degree that the previous signals did. The difference between the signals in the frequency spectrum proves that the algorithm can differentiate the voices between two people. Though the frequencies match at certain points, they do not share the same peak amplitudes and do not share as many common frequencies points as the first sample presented in Figure 1.

12.3.1. Algorithm Changes III

The algorithm went through another set of changes once experiment three was completed. One of the major changes it underwent was the deletion of the interpolation function in the find_peaks subroutine. This allowed the subroutine to find more frequencies throughout the entire speech signal. When the frequencies were found and outputted they were all added to produce a sum of frequencies. This is what was done to both of the speech signals. This new method produced better results when a subject's voice was being compared to himself. Another subroutine has also been added to the algorithm; compare_plot. Compare_plot takes the fast Fourier transforms from the wav_filter subroutines and plots them against each other.

12.4. Experiment 4

Experiment four was very similar to experiment three but covariance was added to the algorithm. By adding covariance to the signal comparison subroutine, the algorithm was partially able to start differentiating between two different subjects. The results of experiment four are available in Appendix F.

Experiment four used the frequency comparison method used in the previous experiments, while also adding the signal covariance. The signal covariance subroutine took the variances of two different signals, and compared the variances to get another percent difference. The percent difference produced by the covariance is added to the percent match produced by the frequencies and then averaged. The algorithm then seems to get good results with subjects being compared to themselves or subjects being compared other subjects. There were two cutoffs that were implemented in this experiment, 75% and 85%. When the 75% cutoff was implemented, the algorithm passed two different speech signals inputted by two different subjects more often than when the 85% cutoff was implemented.

Phrase 3 - 'Sri Lanka' Trial 1

	Chati	Rob	Ross	Mike
Chati	Pass	Pass	Fail	Fail
Rob	Pass	Pass	Fail	Fail
Ross	Fail	Fail	Pass	Fail
Mike	Fail	Fail	Fail	Pass

Table 1 - 75% and 85% Cutoff

As can be seen in Table 1, the algorithm still has a few glitches but for the most part it can differentiate between two voices. Though signal covariance might not be enough to fully implement the voice authentication system, it has shown a lot of promise and will be used along with the comparison of the frequencies.

While performing experiment four, other factors were also causing a few problems. By comparing only one or two words at a time, the system had a hard time differentiating between two subjects especially when they were saying the same phrase or two different phrases that were different but similar in sound. For example, the subjects were asked to input their cities of birth into the system, three of the subjects were born in cities that sounded very similar (Boston, Bolton, and Brockton). When comparing the speech signals of these three cities, there was a lot of confusing present in the algorithm and the algorithm was unable to differentiate effectively; thus, passing most of the signals that sounded the same.

Phrase 4 - 'City Of Birth' Trial 1

	Chati	Rob	Ross	Mike
Chati	Pass	Fail	Fail	Fail
Rob	Fail	Pass	Pass	Pass
Ross	Fail	Pass	Pass	Pass
Mike	Fail	Pass	Pass	Pass

Table 2 - 75% Cutoff

As can be seen from Table 2, the algorithm passed all of the subjects whose cities of birth sounded very similar and failed the city of birth which obviously sounded different. The city of birth which failed with the other three cities was Rome. When comparing Rome to Boston, Bolton, and Brockton, the algorithm was able to recognize the difference of the word.

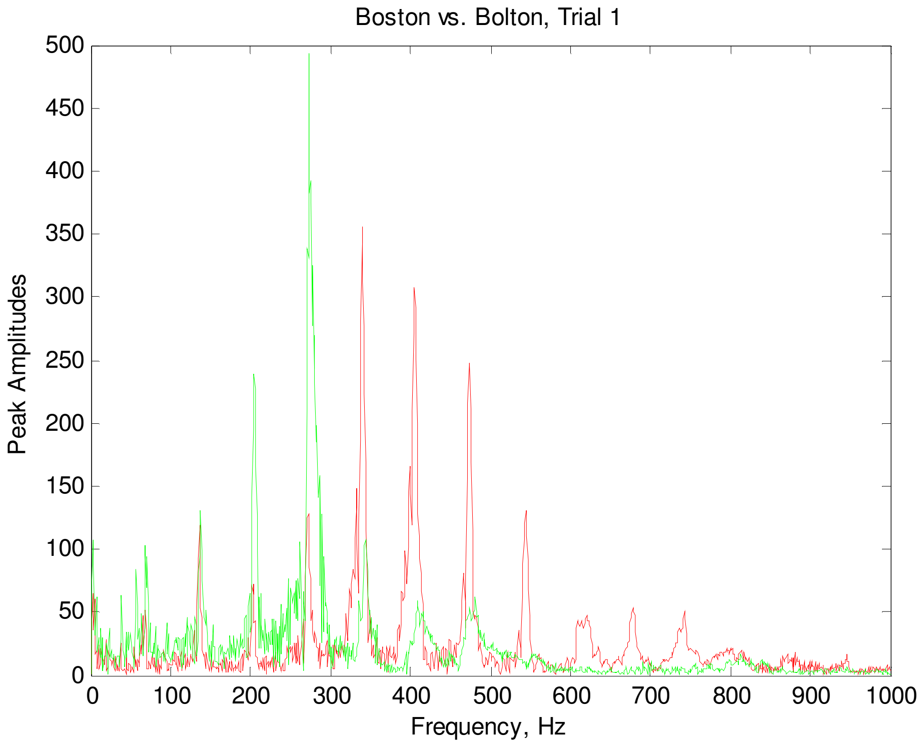


Figure 3 - Bolton vs. Boston Trial 1

As can be seen the frequencies of both of these speech signals are very similar. Though the amplitudes might vary, the frequencies present in the signal will confuse the algorithm to the point where it might pass two different signals that sound similar. The signals look quite different when Rome and Boston are compared.

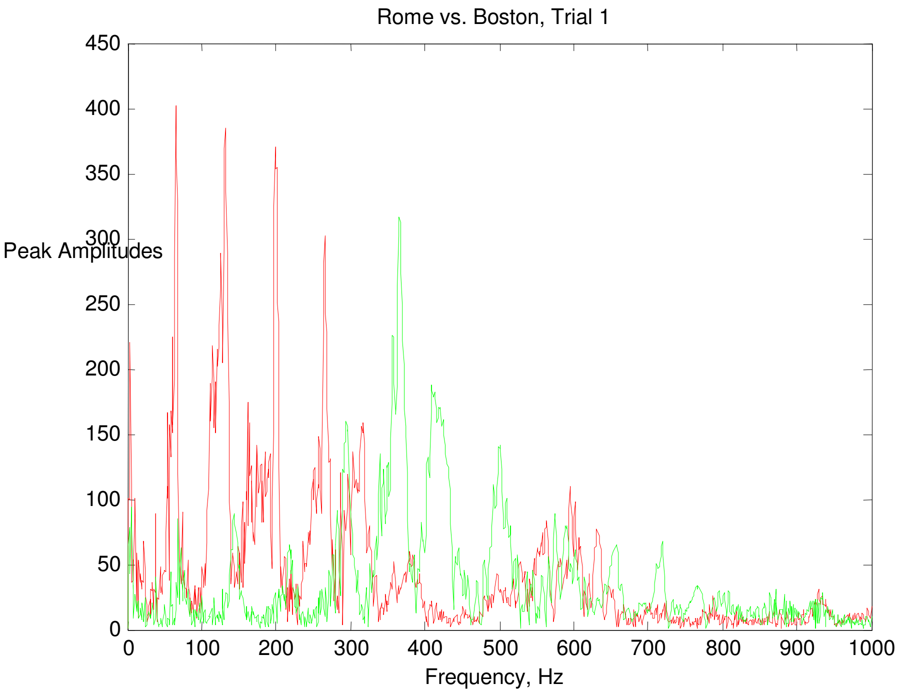


Figure 4 - Rome vs. Boston, Trial 1

Figure 4 shows the differences in frequency when comparing Rome and Boston. Rome is represented in red, while Boston represented in green. Though there are some frequencies that do overlap between the two signals, there aren't enough for the algorithm to mistake them as the same word. The following graph is of a zoomed in view of figure 4:

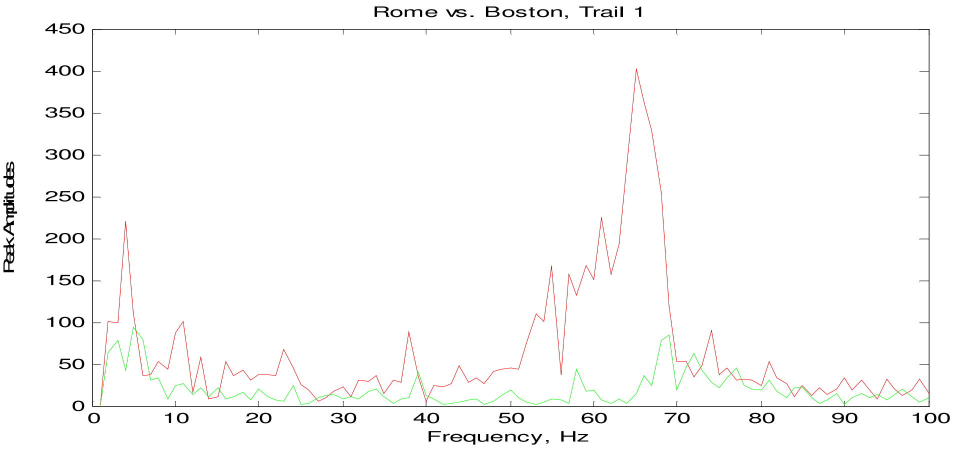


Figure 5 - Close up of Figure 4

12.4.1. Algorithm Changes IV

A new subroutine was added to the algorithm after experiment four. This subroutines was called signal_cov, which stands for signal covariance. This subroutine takes the fast Fourier transform of the signals produced in wav_filter and wav_filter2, and takes the variances of each of the signals. The once the variances of each signal is found, it compares them and gets the percent difference of those two variances. This percent difference is then sent to the peak_compare0 subroutine which adds it on to the percent match of the frequencies and the takes the average of the two percentages. By using this method, the algorithm is starting to differentiate individual voices.

12.5. Experiment 5

Many changes were made when conducting experiment five. The voice authentication algorithm consists of three main subroutines which work together to create a functional voice authentication module. The three main subroutines are voiceRecord, voiceCheck, Voiceauth2; each one of these subroutines contains smaller subroutines used in the previous experiments.

In experiment five, signal normalization using MATLAB was put to the test. The experiment wanted to measure the success rate of the voice authentication algorithm when using MATLAB to normalize the signals. So far, Audacity had been used to cut and normalize signals to get the positive. When using MATLAB to normalize the signals, almost the same success rates were recorded. The cutoffs for the pass and fail had to be recreated. After using MATLAB for signal normalization, 85% was considered a complete pass, 80%-84% was considered to be a partial pass and the guard would be asked to input another password and 79% and below was considered to be a complete fail. If a guard reached the 80%-84% range three times, the guard would automatically be passed; the probability of an imposter reaching that range three times on three different passwords is very low. The results produced in Experiment 5 can be viewed in Appendix F.

Phrase 1 - 'Yellow'

	Chati	Rob	Jeff	Kumari	Leslie
Chati	Pass	Fail	Fail	Fail	Fail
Rob	Fail	Pass	Fail	Fail	Fail
Jeff	Fail	Fail	Pass	Fail	Fail
Mom	Fail	Fail	Fail	Pass	Fail
Dad	Fail	Fail	Fail	Fail	Pass

Table 3 - 80% Cutoff, Phrase 1

The above table shows the results of five test subjects when inputting the word yellow into the voice authentication module. The results produced in this particular test were the most desirable, and if the algorithm worked perfectly all of the results would look like this. For the most part, most of the testing yielded the above results but there were some tests where the results were less desirable.

Phrase 6 - 'Land Rover'

	Chati	Rob	Jeff	Kumari	Leslie
Chati	Pass	Fail	Fail	Fail	Fail
Rob	Fail	Fail	Fail	Fail	Fail
Jeff	Fail	Fail	Pass	Fail	Fail
Mom	Fail	Fail	Fail	Pass	Fail
Dad	Fail	Fail	Fail	Fail	Pass

Table 4 - 80% Cutoff, Phrase 6

Table 4 shows results when they were less than desirable. It shows that the test subject Rob did not pass with himself. This was a problem that was present in a number of test runs, but this error did not occur often enough to pose a serious problem. One of the positive points about the undesirable results was that no one was passed when they were not supposed to. All of the subjects failed when they were compared with different test subjects.

12.5.1. Algorithm Changes V

The algorithm was changed in many ways after the completion of the experiment. All of the subroutines produced in the previous experiments were put into three bigger subroutines. The three main subroutines were created for the sole purpose of easing the compiling process when it was run through the client. The main subroutines, voiceRecord, voiceCheck, and Voiceauth2 performed different functions. The voiceRecord subroutine takes the password inputted by the guard and then normalizes he signal. The voiceCheck subroutine takes two previously recorded files and compares the two files to the file created in the voiceRecord subroutine. This comparison will then output a number which will let the system know if the particular guard will pass, have to repeat another password, or fail that particular test. If the guard fails the system will then alert central control about the fail. The Voiceauth2 subroutine puts voiceRecord and voiceCheck into one function so that the whole process will be fully automated once it ready to go into the client. Another routine was also created for the ease of recording and storing pass phrases. The routine, storepass, will take the recorded pass phrase of a particular guard and it will normalize the signal and store it in the database. This routine was created for the convenience of the customer; it will not communicate with the voice authentication subroutine. The complete code for the voice authentication algorithm can viewed in Appendix E.

13. Reliability Considerations

The security check-in station had to have a number of qualities that the customer could be confident in. Durability was one of the main qualities that the security check-in station had to possess; the device had to be durable enough to withstand possible tampering by unauthorized personnel. Another quality consideration that had to be taken to account was reliability. Reliability was one of the most important qualities that were considered while designing the security check-in station. It is pertinent that the device work exactly the way it was designed to; and while implementing the design of the device, the team devised different methods on making the security check-in station reliable.

Keeping reliability in mind, the team made the decision to make the device a "two part" security system. This is the main reason why the RFID portion was added into the device, to have an extra layer of security. In the case that an RFID tag went missing and ended up in the hands of unauthorized personnel, the voice authentication would be a back up security system. The RFID portion was very straightforward when it came to reliability, either the guards had the cards or they didn't. The difficult part was making the voice authentication portion of the device reliable.

The voice authentication portion had to go through a lot of testing to find the most reliable way of implementing it. By taking the frequencies at different pointas of a speech signal and then summing all those points together, it was apparent that each speech signal would have a different sum of frequencies. When a test subject recorded a phrase twice, it was noted that his/her sum of frequencies were very close to one another. When the sum of frequencies from one test subject was compared to a totally different sum of frequencies, it was noticed that they did not match as closely. This method was the building block of the MATLAB voice authentication algorithm. Although this method seemed to work on most speech signals, there were some test subjects with similar voices. This was when the sums of frequencies started to match closely, even though they were two different speech signals inputted by two different test subjects. To avoid a cross authentication between two people, the signal covariance of each signal had to be found. When implementing signal covariance, it was noticed that the number of variances between two speech signals inputted by the same test subject were very similar. When the signal variances were compared to a test subject of a similar voice, the voice authentication algorithm would generate a significant amount of change between the two speech signals. This change was enough to have a working voice authentication algorithm.

The cutoff, a match of eighty percent will separate the result from complete failure and for a retry. If a match is 82% or above the guard passes immediately. If there is a match between 75%-81%, the guard will be asked to input another password (which will be already pre recorded into the system). If the guard passes the second trail by 82% or more, he or she will be passed; in the case that the guard is still at the 75%-81% range, he/she will be asked to input another password, and if the match is still 75%-81% the guard will be passed. The probability of the algorithm passing an unauthorized person is very low, according to all the tests that have been conducted. The results from these tests are available in Appendix F.

We chose to use the LAMP (Linux, Apache, MySQL and PHP) architecture because it was inexpensive, stable, secure and reliable. Linux provided our Server and

Client with a reliable and secure operating system that would run Apache, MySQL and PHP. The Apache Web Server was chosen to provide the interface between PHP and the network because of its ease of use, reliability, security and popularity. Using Apache, we were able to provide a simple management interface that a security manager could access using their web browser and an interface for the Clients (Check-In Stations) to access the Server and database. PHP was chosen because of its speed, security, simplicity, and built-in support for Apache and MySQL. MySQL was chosen because of its security, speed, simplicity, and scalability. This system can scale from as few as three clients up to an almost unlimited number of clients using clusters of Servers tied to one Database.

14. Client

The Client acts as the gatekeeper in the system by providing the interface for the security personnel to authenticate themselves at each checkpoint. Our proposal indicated that the security check-in station would require two sets of credentials before allowing the user to pass through the system. This first credential is an RFID tag. Each guard is assigned a tag that contains a unique number that is stored in the Server/Database of the system. The second credential is a phrase spoken by the user that is also stored in the Server/Database. The Client interacts with the Server/Database using HTTP (hyper text transfer protocol), similar to what is used to deliver web content to computers on the internet. The client was written in pure C and uses a freely available program called Sound-Recorder to record the user’s voice and libcurl (http://curl.haxx.se/) to interact with the HTTP Server running on the Server/Database over a TCP/IP network.

The hardware of the Client was chosen because of its simplicity, cost and reliability and consists of the following components:

1. Embedded PC
2. Speaker
3. Microphone
4. Parallax RFID Reader

The Embedded PC acts as the heart of the system providing all of the connections for the Speaker, Microphone, RFID Reader and Network Adapter.

The authentication process was designed to be user-friendly and consists of a series of voice prompts that facilitate the authentication process. Initially, we hoped to use an LCD display to prompt the user, however our budget prevented us from doing so.

To authenticate into the system, the security staff member must:

1. Present their RFID security tag.
2. Say their authentication phrase.

As shown in Figure 6, the client waits for input from the RFID Reader.

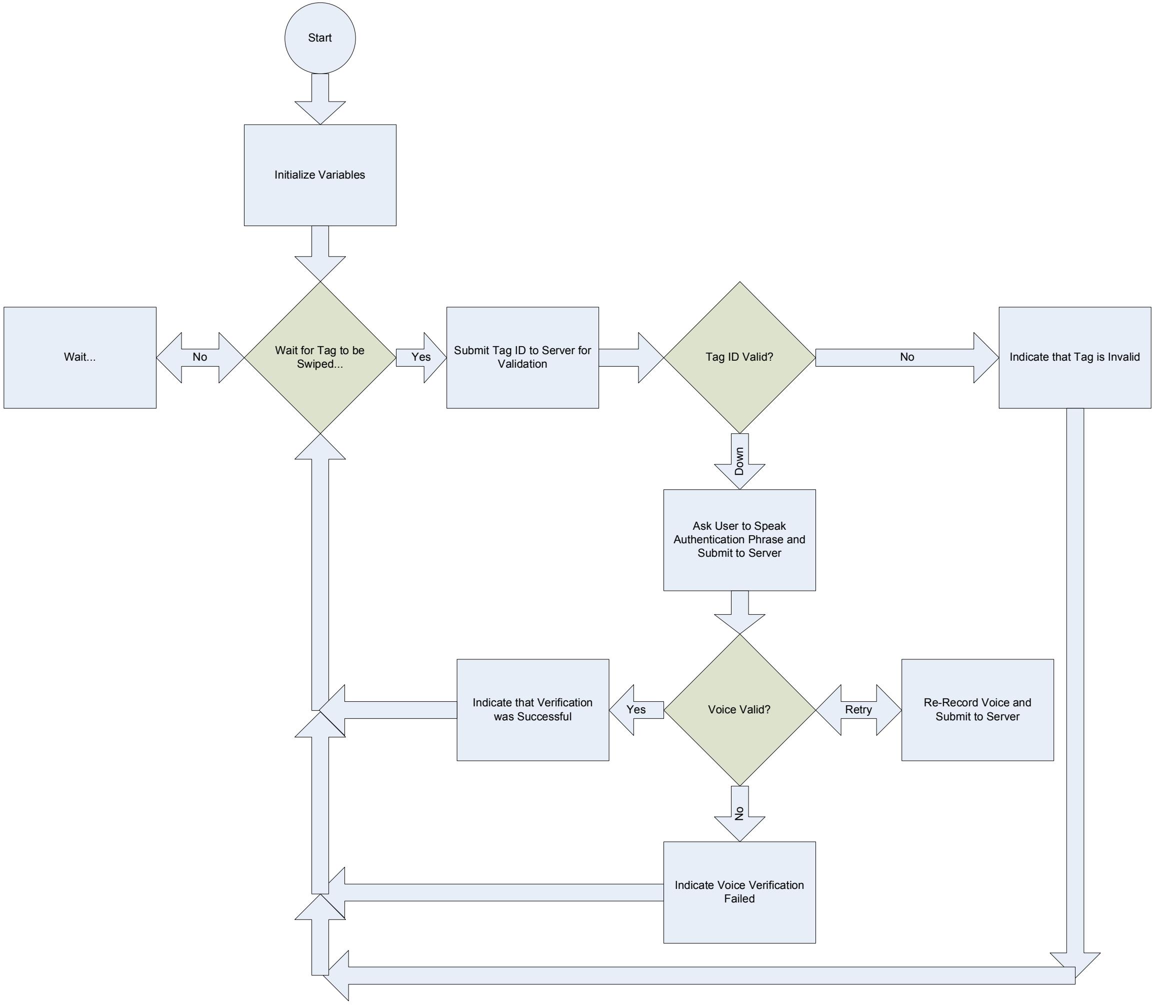

Start
Initialize Variables
Wait...
No
Wait for Tag to be Swiped...
Yes
Submit Tag ID to Server for Validation
Tag ID Valid?
No
Indicate that Tag is Invalid
Down
Ask User to Speak Authentication Phrase and Submit to Server
Indicate that Verification was Successful
Yes
Voice Valid?
Retry
Re-Record Voice and Submit to Server
No
Indicate Voice Verification Failed

Figure 6: Flow Diagram for Client (Check-In Station)

Once the user places their card within range of the RFID reader, the Client reads the tag and then sends it to the server for initial verification. If the tag fails the initial validation process, the check-in station does not allow the user to continue to the next step of the authentication process. After the tag has been validated, the system asks the user to say their authentication phrase. The system then sends the recording of the user's voice to the server where the file is then compared with the two voice files in the database. If the voice verification fails, the user is either asked to repeat their phrase or begin the authentication process again depending on the server's response.

15. Server

The server acts as a repository for all of the data in the security system that retains the master voice files, images and the handles the voice authentication for each of its connected clients. The Server is coded entirely in PHP, a free web scripting language and uses the MySQL database engine to store information about the security staff. To interact with the clients, we chose to use Apache, a scalable open-source web server that supports the PHP parsing engine used to process our code. We chose to use this combination because of its popularity, speed, flexibility, and security.

The database contains all of the information about the users in the system including their Name, Username, Password, Badge ID, Age, Height, Hair Color, Picture, and Voice Files. Some of these parameters did not affect the operation of this system but were added to help security managers maintain detailed profiles about their staff.

The server verifies the security staff member's voice using a custom voice authentication program. After creating the voice authentication algorithm in MATLAB, we used MATLAB's C Compiler and the GNU C Compiler to create a standalone executable that accepts three WAV files as parameters and returns a fraction between 0 and 1 that indicates the similarity between the recordings. To help the system to compensate for noisy areas we allowed the security manager select thresholds that would dictate a pass, retry, and fail in our system. Using PHPs built-in support for executing shell programs, we were able to develop a script that would execute our Voice Authentication Program and compare the recording taken at the check-in station with the master voice files stored on the server.

To ensure that guards are checking in at each station, we added a detailed access log that records every transaction made by the Clients (Check-In Stations). We also added a feature that lets managers specify when guards should check into a specific check-in station. If guards fail to check-in before a specified time, the system will send an e-mail to the security manager should a staff member fail to check-in.

Ideally, this system would operate on a computer network isolated from the companies existing network. However, some companies might opt to install this system on their existing computer network. To reduce the risk of these threats, we added several layers of protection to both the server and client to prevent unauthorized users from gaining access to the system

The first of these layers begins at management system. To access the administration interface of the server, a manager must provide a valid username and password before they can view any information about the system, its users, and data.

The next layer of security on server is the client identification system. Each Client (Check-In Station) on the network must be registered with the server before it can access the data on the server. This prevents other clients or systems acting as clients from accessing the data on the server. It also enables the system to identify specific clients in the log so that security managers can see where the activity occurred.

The final layer of security is a detailed logging system that records all of the queries made to the server by the connected Clients. Security managers can view all verification requests made by the different Clients and check to see if any unauthorized users have attempted to check into the system.

16. Synthesis

All portions came together in a matter of days. The first portion to be moved from a Windows environment to a UNIX environment was the voice authentication portion. With a couple issues that were resolved quickly, the portion worked properly and as intended. There were no disagreements between the code in MATLAB and the code above it, using it.

The RFID portion was added to the server and connected as hoped. Except for a couple of wiring issues, the portion functioned using the original code with a couple of adjustments made specifically to tailor it to this project. After testing, it was viewed that the portion was working in conjunction with the parent code.

The entire compiled project withstood reliability tests and worked for an extended period of time. Under reliability tests, multiple tests were done with different subjects. The verifications worked properly, rejecting and allowing correctly. Voice tests were also done, creating more failures than we had intended, but assuring that no one could pass off as a different person.

17. Aesthetics

The security check-in station will be a wall mounted device, so the aesthetics that it will contain will be minimal compared to similar products out in the market. The aesthetics of the security check-in station will also cut down the cost of production. The wall mounted design allows only half of the device to be seen by a particular user. The customer will be able to specify the color of the device; if no color is specified then a default color will be applied. The default color in this case will be black. Black is a color that will standout no matter what type of color the respective wall may be. The outer casing will be made out of a strong metallic substance like steel. This will prevent the device from being tampered with and broken into.

The layouts of the components in the security check-in station are placed for the convenience of the guards that will be using it. The microphone will be located in the upper right hand side of the device, while the speaker will be located directly below the microphone. The five inch LCD screen will be located on the top left hand side of the device, and the RFID will be located directly below it. An artist's rendition of the device is located in Appendix A. The physical interface of the device will go from left to right; a direction which all guards should be accustomed to. The reason the interface goes from left to right is because that is the same direction that English is read, so going from left to right will be something natural for the guards.

18. Culmination of Previous Courses

18.1. RFID

For the RFID portion of the project, there were two main focuses that helped develop the module. The first was microelectronics and the other was radio frequency design. Also as intended, the concepts learned in ECE 2799 were applied. Minimal amounts of programming knowledge from CS 2301 were used in the software implementation.

From microelectronics, lab concepts used in Microelectronics I and II and Analog Integrated Circuit Design were applied. This included component placement and circuit efficiency. Using more advanced concepts, found in ECE 4902, waste was reduced to a minimum allowing for a smooth transition from a proto board to a printed circuit board, had the project continued to that level. Knowledge gained in the lower level microelectronics courses made reading through data sheets with complicated integrated circuits easy.

Concepts that would have been learned in ECE 3113 may have helped the thought process around the RFID reader, but most knowledge was gained in preliminary research. A more serious look at how the RFID reader works would most likely be gained in Communications I, ECE 3403.

The project management type class that ECE 2799 intends to be is a good prerequisite for the MQP. Much of the proposal was written on a premise around the schedule of an ECE 2799 project, except without any actual synthesis. By the end of the proposal, high level design of the product was very easy to come up with.

18.2. Voice Authentication

Creating the MATLAB voice authentication algorithm required material learned from a number of classes. The whole algorithm contains seven subroutines which performs the voice authentication. The names of the subroutines are the following:

- Signal_norm
- Wav_filter
- Wav_filter2
- Compare_plot
- Peak_finder0
- Signal_cov
- Final

The first subroutine, signal_norm, an inputted signal is taken from the recorduser subroutine. The speech signal is then normalized so that the amplitudes are no greater than 1, or less than -1. Signal normalization was used to increase the volume of the speech signal which in turn gave a more accurate reading when the signal was run through the algorithm. Signal normalization was a subject that was talked about in ECE 2312 (Discrete-Time Signals).

The wav_filter subroutines combined two the concepts from two different classes. First, the fast Fourier transform (FFT), which was a concept first taught in ECE 2311 (Continuous-Time Signals). Once the speech signal was converted from the time domain to the frequency domain, it is then put through a high pass Butterworth filter, which was a concept taught in ECE 2312.

Once the signal is run through the wav_filter subroutine, the filtered signal is then sent to the compare_plot subroutine. This subroutine just takes the two filtered signals produced in the wav_filter subroutines and plots them to compare the similarities of the speech signals.

The two speech signals are then sent to the peak_finder0 subroutine, which takes each of the signals and finds the frequencies at different points of signals. Once the frequencies have been found they are summed up and compared to each other to produce a certain match. These concepts were taught in ECE 2311 and ECE 2312.

The same speech signals that were sent to peak_finder0 are also sent to the signal_cov subroutine. This subroutine finds the variances between the two signals. Once the number of variances has been found, the results are then compared to each other to produce another percent match. The concept of variance and covariance was first taught in MA 2621 (Probability). This concept could be applied into signal analysis. Once the percent matches are taken from both the peak_finder0 and signal_cov subroutines, they are sent to the final subroutine. This subroutine takes the percent matches and averages them out to make one final percent match. Depending on the percent match, it will determine the status of a particular guard, and will either pass or fail him/her.

18.3. Client-Server

The Client and Server portions of this project used concepts taken primarily from the computer science portions of our curriculum, specifically CS 1101 and CS 2102. Although the syntax for C and PHP languages differed slightly, we used many of the same programming concepts taken from our coursework to design and implement our code. In the logging section, we decided to create a log object that would contain all of the information about the logging entry. Using the object-oriented design principles taken from CS 2102, we created a "logEntry" object type to store information about each log entry before it was written to the log in our database. By creating this object type, we were able to save several lines of code and easily integrate the logging system into any of our other modules to help us identify bugs and record activity.

19. Economic Considerations

This project remained well within its bounds of time and money. When many of the expensive products that were discussed in the proposal were dropped, the budget fell from $600 to approximately $80. This drop did not lead to any particular drop in quality of product, only more time to develop. As for the end user, the product sale price, if ever one will be far cheaper than products already out there. Most of our economic considerations are explained in greater detail in our budget proposal and value analysis in the Appendix A.

20. Safety Considerations

For any marketable product, it is extremely important to consider the dangers involved in using the product. The hardware other than the computer in the client is low power, drawing +5.3V at 130 mA. Although this is considered high by today's integrated circuit standards, +5V is not dangerous enough to threaten a life.

The computer on the other hand runs on a +110/120V power supply unit. It is low compared to other contemporary power supplies, using only about 235 W peak, this however would require a warning label similar to those found on computers today. The unit, if brought to form implementation, would also need a static shock warning, as there are capacitors in the screen and computer that could cause electrocution.

This unit does not have any moving parts and relies entirely its power supply for its electricity. The unit must also stay plugged in for it to remain functional, as the final design did not include a backup battery. Due to the fact that the unit has an RFID unit and does transmit radio signals, there would be an FCC compliance label affirming the unit complies with the RFID 125 kHz standard.

21. Overall Conclusion

The project took a considerable amount of time and was well within its budget. Time allocated to it was appropriate and allowed for the extra time that we had not originally intended to use. Monetarily, the project did not overrun its budget and towards the end was able to order extra unintended parts without issue.

The project team views this as a proper design and application of a major qualifying project. Design took approximately one-third of the total time while implementation and testing each took approximately one-third each. With this balance, we were allowed ample time to write the project and retain all minimal project requirements, with extra work done. This project, if desired can be continued, but the basic functionality exists and more work, if done would not constitute a second major qualifying project. Another project would require considerable changes that would change the basic design of this project.

The following changes can be made as minor improvements to this project.

- **RFID should be cased**

  With a more portable case, the unit will be able to withstand considerable damage and still function normally. With wiring simplified, the unit can also be as compact as its largest item, the RFID antenna.
- **Experiments should always be performed in a controlled environment**

  At times the results of experiments were very unfavorable due to all the background noise that was present in the labs. If conducting tests like the ones performed for voice authentication, make sure you do it in a place where background noise is low.
- **Make sure to order multiple parts**

  The new RFID scanner that was bought for the project malfunctioned after a week of using it. Another RFID scanner had to be purchased, but it wasted precious time. If ordering a part, try to order at least one more for safety precautions.

22. Appendix A: Proposal

Table of Contents

Table of Figures

Table of Tables

Introduction

Security has been one of the major issues for big businesses, small businesses and home owners alike. As time goes on, new technology is developed to maintain security easily and efficiently. The project that has been presented is to design a new alternative to secure buildings. The design will be a security checkpoint for large buildings, areas etc, will be a security checkpoint for guards to check in to as they patrol their respective areas. This security checkpoint will consist of an Radio Frequency Identification (RFID) tag which will be given to each guard so that no unauthorized user may falsify the guards identity. In addition to these features there will be a timer that will be embedded in the checkpoint to see how long it takes the guard to reach the next checkpoint. This will help give businesses the added piece of mind as criminals become more technologically aware.

Market Research

There are many different types of security devices in the market today. Some are aimed at small businesses, others are aimed at homeowners etc, but were unable to find a complete security check-in station. Different components of it are in the market already but not the full product. RFID is used in many office buildings to grant employees access to secured areas. Voice authentication is also used in businesses and allows the device to grant access depending on who speaks into it

Internet Research

Two different types of internet searches were performed; a query of RFID systems and voice authentication systems. There were many different companies that sold RFID systems and this made it easier to identify our product's competition. One of the first RFID products found was the DS321 RFID Access Control system produced by Warwick Wireless Limited.

Figure 7: DS321 RFID Access Control

Some of the features of this system included, control and sensor units, transponder cards, user programmable relay etc. The cards used for this system are similar size to a credit card and need to be placed about four inches away from the system to function properly.

Another company that deals with RFID systems is ID Systems. The name of their product is called IT Tag and provides almost the same functionality as the DS321 RFID Access Control system. Some of its features include a 512 and 2048 bit EEPROM, a unique 64 bit serial number, and read/write protection.

The next step was to find companies that produced voice authentication systems. The AMBER system created by bioMetrics Technologies has an all inclusive package of voice authentication, facial picture authentication, fingerprint authentication, and iris authentication. All the systems work individually but utilize the same processing software making this product very efficient and easy to use.

Figure 8: IT Tag

The following (Figure 9: Amber SystemFigure 9) is a block diagram on how this system works.

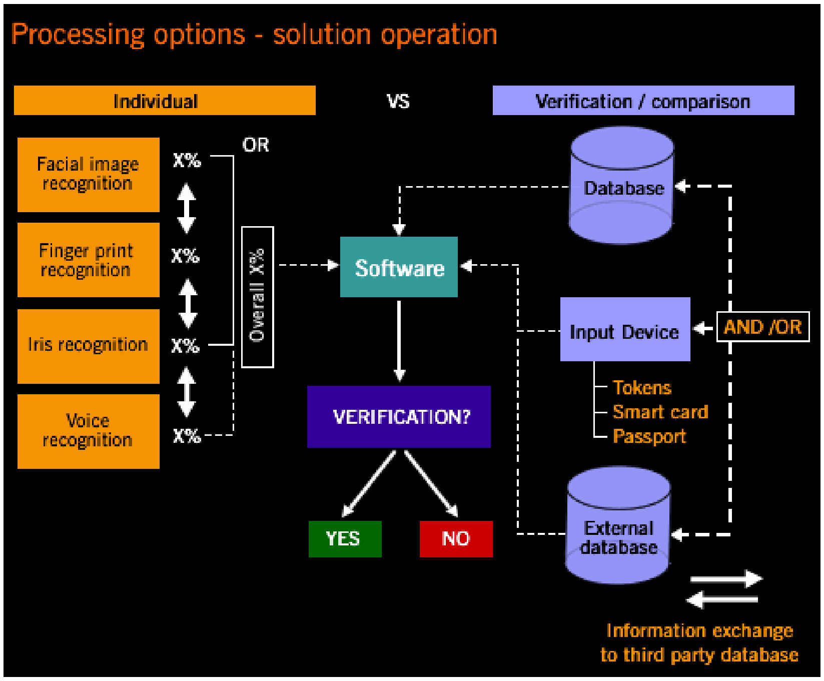


Figure 9: Amber System

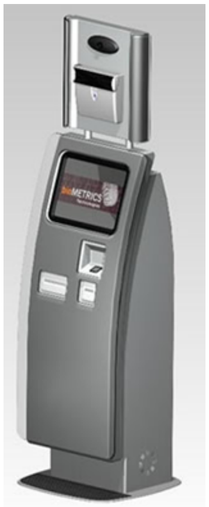


Figure 10: Authenti-Kate Kiosk

bioMetrics Technologies also has the Authenti-Kate Kiosk. This the only product that had met most of our product requirements. Some of its features include: voice authentication, RFID, finger-print scanner, and iris scanner. This system does not include voice authentication by default, but instead as an optional feature.

Potential Customers

Businesses and government agencies are some of the customers that will benefit from our product. Companies like Raytheon and Lockheed Martin manage several secret projects and would benefit from our product's ability to enhance their existing security policy.

Customer Requirements

Since our customers need this product for their high-level security systems, there a few customer requirements that have to be met for the product sell. The system will have to be easy to use but also provide a very powerful and effective level of security control. The check-in station will be compact in order to save space.

Durability is another specification that will be required. The product has to be durable enough so that it will not be tampered with.

Product Requirements

The product requirements for this device are very important for the design and marketability of the security check-in station. Security will be the main priority when designing this device. There are two components that need special care; these modules are the voice authentication module and the RFID module.

The voice authentication will have to be able to pick up the voice of a certain guard and accurately check and compare it with stored sound files, which will be stored in a database. This module should be accurate enough to deny access to any unauthorized personnel but at the same time, it can not be so precise that it won't recognize the voice of the actual guard.

The RFID module will detect the radio frequency transmitted from the card. This will ensure that a particular guard is at the right check-in station at any given moment. This module will then have to communicate with the embedded PC, which will in turn start up the voice authentication module.

Product Specifications

The product requirements are the following:

- Durability
- Reliability
- Ease of use
- Battery Backup

Durability

Durability is a very important specification which our product will have to possess. The product will have to be durable enough to withstand any punishment that anyone might want to give it. The device will have a metal casing, to protect it from immediate forceful damage. Durability should also ensure that the product will last for quite some time and will still work as time goes on.

Reliability

This product will have to be very reliable. The product will be expected to recognize the voice of each guard and also accept the RFID card that each guard will be wearing. If the security check-in station were not reliable, the market would reject it.

Ease of Use

The ease of use of the product will also play an important role in its marketability. The device should be functional enough to be easy to use and should limit the transaction between itself and the guard to only a few minutes. The more reliable the device is, the easier it will be to use and maintain.

Battery Backup

The battery backup will be used ~~to~~ as a safety measure. This is specification is going to be implemented to increase reliability and security. Security systems cannot be affected by power outages. To address this need, this system shall have a battery back up component that will allow it to function during power outages.

Conclusion

With a budget of roughly $375, the product specifications have been carefully selected to meet our budget's constraints. This will give the customer a good product at a very competitive price. The security check-in station will be used primarily by businesses and/or organizations that require guards patrolling the area for security. All the information will be logged by the device and sent to a central server.

By making our product marketable to big businesses and organizations, it will ensure that more than one unit will be bought at each purchase. Depending on the size of the respective facility three or more units will have to be bought. The security check-in station will be priced at a competitive price which will be appealing to potential customers.

The security check-in station will be a product that the customers will find durable and reliable. Along with reduced pricing, this product will be a formidable opponent to most of its competition. Competition will mainly come from biometric companies which sell a lot of similar products. With the RFID and voice authentication component both in place, the security check-in station will have as many features as the more expensive units out in the market today

Project Goals

The main objective that has been presented to us is to design and build a security check-in station that will provide an acceptable amount of security to the consumer but also can compete with the prices of comparable products. The purpose of this check-in station is to allow guards patrolling a certain area to log their current position to a central server, through voice authentication and RFID. This will allow companies to track their guards movement and alert security when guards are missing or delayed. The voice authentication and the RFID are the two ways that authentication of the guard will be identified.

Design Options

General Design Options

The implementation of the RFID and the voice authentication will be done separately but it will all run out of one device. The RFID portion will be added on to the security check-in station, so that when the guard reaches a particular check-in, he/she can just swipe or scan the RF transmitter provided to them. This will then access a database that will recognize the particular guard on duty and load their voice authentication data. Once this stage of the verification is complete, the voice authentication stage will take over. The system will ask the guard to repeat certain phrases or words. The guard will then repeat the phrases or words into the microphone. The voice authentication will then further identify the guard and log him in for that area. This process will be repeated through out the area of patrol.

Value Criteria for Design Options and Metrics for Value Analysis *(formatting)*(reformatted)

The value criteria that were chosen for the device depended on the individual component. Since not all the components had the same pertinent general characteristics, we chose different value criteria for each. The value criteria for the different components go as follows:

LCD Display

- Size
- Interface
- Resolution
- Pixels
- Price
- Touch screen capability

Microphone

- Size
- Input Frequency
- Price

Speaker

- Size
- Range Frequency
- Price

RFID

- Interface
- Transponder
- Receiver
- Price

Specific Design Options

Each of the components used in this project will need to have a certain number of options that will help the device come together efficiently. These components should also work the desired requirements that will make the product a reliable security device, but will also be cut down in price compared to the products already out on the market.

LCD Display

The LCD display will have to have a number of options that will have to fulfill our requirements. This module must be able to display the information that the client desires, while making the picture clear for the user. It must be compatible with the embedded computer that will be running the device.

Size

After some research on the type of LCD displays, we came up with some requirements for our LCD displays that would suit our project the best. The size that would best suit the device should be from five to eight inches. A large LCD display is not required since it will only be prompting the user with instructions to complete the identification process.

Interface

The interface type should preferably be TFT (Thin Film Transistor), but CSPN will work as well. The three types of TFT displays include TN (Twisted Nematic), IPS (In-plane switching), VA (Vertical Alignment). Based on our designs requirements, the TN display will not work, since the device will be wall-mounted; however, the IPS and VA will work great with our design.

Display Type

The twisted nematic TFT display is one of the most commonly used displays due to the low cost. One of the good points about this type of display is that the pixel response time is fast so it will avoid shadow trails. One of the disadvantages of this display is that it has very limited viewing angles. The in-plane switching TFT display has more viewing angles than the twisted nematic. The main problem with this type of display is that it the pixel response time is not as fast as the twisted nematic displays. Another problem with this type of display is that it is very expensive. The vertical alignment TFT display comes in two different packages; multi-domain vertical alignment

(MVA) and patterned vertical alignment (PVA). The MVA is a mixture of the TN and the IPS displays. It has a fast response and improved viewing angles. On the other hand the PVA has the best contrast ratios of any TFT display. The PVA is an improved version of the MVA. The problem with both of these displays is cost. The cost is too high for what we are trying to implement

Resolution & Pixels

There are also three types of resolutions available; the Video Graphics Array (VGA), Super Video Graphics Array (SVGA) and the Extended Graphics Array (XGA). Since the VGA is outdated, the Super VGA and XGA will perform to our desired requirements. The number of pixels in the Super VGA is 1024 × 768 8-bit pixels, while the number of pixels in the XGA is 800 × 600 pixels with high color and 1024 × 768 pixels with 256 colors. The resolution will also fit the pixel requirements that have been formulated for the device.

LCD Display Comparison

When doing the initial comparison of all the LCD displays that fit our design, there were many LCD displays that meet the criteria. There were only a few that however, met the price range requirement. The TOSHIBA P000236950 (6.1 Inch) LCD display met the size requirement quite well. It has a TFT interface, but had a resolution type of VGA. This is what drove the score for the value analysis down. The price was also about $200 which drove the score down even further. It came in a close second, compared to the number one choice.

The TOSHIBA P000249560 (7.1 inch) LCD display was very similar to its 6.1 inch counterpart. The only difference was the size of the display, which made the score just below the first Toshiba display.

The SONY 1-475-937-11 (8.9 inch) was reaching the limits on size. The security check-in station was supposed to have a nominal size of about 5 inches, and the fact that this particular LCD display was almost 9 inches would serve a problem. Another major problem that this display had was its price. The cost for this screen was about $350 which was way over the budget constraints designed for this component of the device.

The COMPAQ 147823-001 (8.4 inch) is the display that ended up winning the value analysis scoring. Though the display was a little larger than the nominal size, it contained many advantages. It had an interface that would fit perfectly with the design and an acceptable resolution type and pixel count. The price of this display was also only $129.

The LCD display considered was the Compaq 199232-001 CSTN (8.4 inch). This display was only $36, so it scored very high when it came to price *(?)*. The problem with this product was that it scored average on the rest of the requirements. The resolution type, pixel count, and interface were all average, which took away points from the overall scoring of the display.

(place criteria before comparison, also put weights in table) (eliminate touchscreen)

Table 5: LCD Chart

	TOSHIBA P000236950 (6.1 Inch)	TOSHIBA P000249560 (7.1 inch)	SONY 1-475-937-11 (8.9 inch)	COMPAQ 147823-001 (8.4 inch)	Compaq 199232-001 CSTN (8.4 inch)
Size	80	70	40	50	50
Color	8	8	8	8	8
Interface	70	70	70	70	14
Resolution	20	20	50	35	35
Pixels	20	20	50	35	35
Price	63	63	18	81	90
TOTAL	261	251	236	279	232

Value Analysis

Table 6: Value Criteria for LCD

Criterion	Rating (of 10)	Comments
Size:	10	Nominal size is about 5 inches
Color	8	Important so that display can be clearly scene
Interface	7	Interface had to be TFT
Resolution	5	Resolution had to be either Super VGA or XGA
Pixels	5	Pixels correlated with the resolution
Price	9	Second most important part. Drives the cost of device.

Conclusion

After considering the following criteria, the following value analysis has been formulated to show which product would be the best fit for our design. Price was also another aspect which was looked upon. Since the goal was to make a low budget security check-in station, the price of the display was cut down to lower costs so that it could compete with the products out on the market already.

Embedded PC

When considering hardware platform, we considered using one of three processing platforms: Peripheral Interface Controller, Single-Board Computer or an analog solution. We quickly eliminated the analog solution because it would be very hard to interface with any analog system because of the protocols the readers utilized and the inability for such a system to support a network interface. A peripheral interface

controller (PIC) although less costly and more efficient, could not interface with a RFID scanner, or handle voice authentication, our main objectives in this project. The single board computer although the most costly of the options, offered the most flexibility, could interface with an RFID scanner and could perform voice authentication.
(weights/use weighted system to keep in line with other modules)

Value Analysis

Table 7: Embedded PC Value Analysis

	PIC	**Single-Board Computer**	Analog
Cost	8	**6**	9
Easy to Interface	2	**9**	1
Power Efficiency	8	**6**	5
Network Interface	6	**10**	0
RFID Scanner Interface	5	**10**	0
TOTAL:	29	**41**	15

Microphone

The microphone will be used for voice authentication. This makes it important because the device must have proper directivity, and also requires that manufacturing on a scale of economy requires the same product for the same results. The microphone has four major requirements; size, frequency response, directional pattern and price.

Size

A basic requirement of the microphone is its size. It must stay in the space provided for it, next to the screen and above the speaker. The size of microphone has to be quite small, ideally about one centimeter. The microphone should be no larger than five centimeters to leave room for the other components. Microphones used for vocal input tend to be larger to properly receive all of the frequencies. This size limitation must be accounted for in the design.

Input Frequency

Because of the nature of the reception, the design must include a vocal microphone, in a dynamic range including 20 Hz to 20 kHz. A deviation from this ideal range would not affect the system detrimentally; however it would add unnecessary noise to the system, which would just have to be filtered out. Any range smaller than 20-20k would not be acceptable for this system.

Directional Pattern

There are two types of directional patterns that are appropriate for recording the human voice. The first is cardioid, which is unidirectional and inputs from a shape that resembles an apple. The other is binaural, or Omni directional. Binaural microphones receive sound from any direction, where as cardioid will not necessarily pick up as well. For this system, either directional pattern would be suitable, however prices may dictate that cardioid may be a better option.

Price

The price is the least important factor in our value analysis. Since it is important but the technology is prevalent, it will cost less than the other modules. The ideal price is lower than one dollar. Unless a particular microphone stand**s** out our maximum price would be approximately $5.

Value Analysis

The following products were matched for their characteristics based on the four main factors that were considered.

Table 8: Microphone Comparative Characteristics

	P11974-ND	P11975-ND	P9964-ND	P9958-ND
	WM-61B102B	WM-60AT	WM-65A	WM-64K
Size	6x3.4	6x5	6x5	6x2.2
Frequency	20-20k	20-20k	100-12k	20-16k
Shape	Omni	Omni	Uni	Omni with Cap
Price	$ 2.08	$ 1.71	$ 2.08	$ 2.03
Voltage	2	2	2	2
SN	62	58	55	58
Current	.5 mA	.5 mA	.5 mA	.5 mA
Sensitivity	-35 ±4dB	-44 ±5dB	-50 ±4dB	-45 ±4dB

Table 9: Microphone Value Analysis

		Microphone Value Analysis						Perfect Score: 290	
		WM-61B		WM-60AT		WM-65A		WM-64K	
	Wt	Score		Score		Score		Score	
Size	7	9	63	8	56	8	56	10	70
Response	10	10	100	10	100	6	60	8	80
Shape	7	10	70	10	70	8	56	10	70
Price	5	8	40	10	50	8	40	9	45
		Total:	*273*	Total:	**276**	Total:	212	Total:	265

Conclusion

Sticking with a single company and then comparing their products for the best matching attributes has been used because there are so many products available. Staying with one company, although sometimes more expensive initially, usually reduces the cost of maintenance and support in the future. From the over 200 microphones available, narrowing down to Panasonic's line of Electret Condenser Microphone Cartridges allowed us to only look through upwards of 50 microphones. From that selection, four were considered and are listed in the *above table*. It was discovered that the variety leaves many microphones around the same level of quality, especially in reference to this project. Both the WM-60AT and the WM-61B met our product requirements, scoring a

276 and 273 respectively. However, the WM-60AT cost 37 cents less than the WM-61B and edged ahead in the value analysis.

Speaker

The primary purpose of the speaker is to complement the LCD display to create an interactive system. It will play a role with the voice authentication part of the check-in station. The speaker and LCD will be used to prompt the user for input to complete the check-in process.

The requirements for the speaker are basic. It should be small, audible and have a wide enough frequency response to reproduce an understandable human voice.

Size

The ideal speaker should be about 5 centimeters in diameter with a maximum size of 10 centimeters. Size plays a role because of the limited space (available for the speaker. The driver must also be small enough to fit in the case. The depth of the machine depends mostly on the LCD display and the single board embedded system; the speaker cannot be deeper than the other main components, the screen and computer, combined.

Frequency Response

The output of the frequency for this system depends entirely on the user. Since many speakers are capable of outputting within the audible range, that leaves many speakers. The ideal range should be the audible range of 300 Hz to 8 kHz   If cost dictates, the output range can be reduced to only output what the computer will give it. For example, the system may not output a frequency higher than 5 kHz. If this is the case, the speaker's frequency response will not need to cover the 5-20 kHz range, which will also reduce the cost.

Power

Size limits the output of speakers quite a bit in comparison with power; however, it is necessary that the speaker be efficient. At most, the driver will probably need 8 W. This should suffice for anyone within a 5 foot range. Since the microphone's range is much smaller, the user will already be within this range. From an ideality of about .5 W, the price begins to rise because of miniaturization costs and also from consumption costs.

Price

Finally, the ideal price for the speakers should be about two dollars. The maximum price should about $5. This is reasonable for a product of this size and its applications. For the speaker, price will not be an important factor.

Value Analysis

Table 10: Speaker Compared Characteristics

	CUI GA0571	GA0771B	Lab Kit	GA0576
Size	57mm/Round	77m/Round	2"/Round	57mm/Rd/Framed
Impedance	8 ohm	8 ohm	8 ohm	8 ohm
Power	.25 W	3 W		.25 W
Response	400-4k	240-21k		400-4k
Price	$ 3.30	$ 2.04	$ -	$ 3.30
Magnet	Alnico	Alnico	Alnico	Alnico
Intensity	88 dB	92 dB		88 dB

Table 11: Speaker Value Analysis

	Speaker Value Analysis						Perfect Score: 310	
	GA0571		GA0771		Kit		GA0576	
	Score		Score		Score		Score	
Size	9	81	6	54	10	90	9	81
Power	6	42	6	42	6	42	6	42
Response	4	40	10	100	8	80	4	40
Price	7	35	8	40	10	50	7	35
	Total:	198	Total:	236	Total:	**262**	Total:	198

Conclusion

From research it was found the CUI Inc. is a leading retailer of speakers. Much of the available stock was from them via Digikey. Upon finding a variety, it was then narrowed to a model 77 mm in diameter and 57 mm in diameter. Because of its cost, the speaker found in our lab kits was also considered. Two models of the exact same speaker were compared, one with a round mount, the other with a square mount, more useful for our application. Although the speaker with the 57 mm diameter was nearly ideal in shape, it lacked the best frequency response, with only the range between 400 and 4000 Hz. The 77 mm made up for the lack of response with a lower cost. Finally, considering that we already had speaker in the kit, it was down to comparing it to the other speakers. It was the ideal size at 50.8 mm and the response proved versatile enough for our application. The microphone present in the lab kit was the microphone selected because of cost. This microphone will not have to be purchased since it already in the lab kits.

RFID Scanner

Radio frequency identification, or RFID, is a new technology that has developed primarily within the past decade. Advances in technology have allowed for the development of smaller transmitters and receivers. Many industries have migrated to security systems based on RFID where a false identification card can be made, however without the matching embedded identification, the system will fail the verification process. **(Are you saying that RFID tags can be forged?)*****(reword with two-stage authentication system)***

Our main purpose in using an RFID scanner is to double the verification process. Along with the voice authentication system, it will make the system harder to infiltrate. Because it is one of the more essential modules in the system, it is important that it interfaces with the rest of the components. Additionally, because the device includes a wireless transmission, it must not interfere with the operation of the rest of the system.

Computer Interface

One of the first components which had to be considered for this module was the type of interface that would best suit the design. There are four types of interfaces that had to be considered; serial, USB, Ethernet and parallel. The serial interface best suits our design because most RFID software supports it.

Size

Although size is a typical consideration, for the RFID module, we will be designing the casing around it. The two largest modules, the LCD monitor and computer board are large enough where the remaining modules can fit around them. So even if the module takes up a considerable amount of space in comparison to the microphone and speaker, it will likely not be comparable to the monitor.

Frequency and Range

The new ISO standard for RFID systems is the 13.56 MHz range. Because we are looking at developing a new product, there is no reason to design for applications outside of the 13.56 MHz standard. The system does not need an extremely large range, so the device will be selected as a passive transponder that requires no power of its own. Normally systems like these come in a variety of shapes like a button, our system will use a 2 1/8” by 3 1/8” transponder, similar to any identification card around today. At most the reader will have to read at 2”, any distance beyond that will cause the reader to return an error.

Price

Its ideal cost will be about $50, but after assessing the importance of the transponder, a budget of $150 will be allotted. $50 will keep the product within reasonable budget range. With an overall budget around $500, we are looking to keep any one module to about $250-350 or 25% overall.

Value Analysis

Table 12: RFID Compared Characteristics

	ACG	Socket Com	Wired	TI 5400 MFR
Frequency	13.56 M	13.56 M	NA	13.56 M/136k
Range	90 mm	50 mm	NA	90 mm
Tx Speed	9600-57.6k	N/A	400 M	N/A
OS	Windows CE	Windows CE	Any	Any
Interface	Compact Fl	Compact Fl	USB	
Price	$350	$240	$60	$600

Table 13: RFID Value Analysis

		RFID Value Analysis						Perfect Score: 230	
		ACG		Socket Comm		Magnetic Card		TI 5400 Multi	
	Wt	Score		Score		Score		Score	
Interface	8	10	80	10	80	6	48	10	80
Freq/Ran	6	10	60	8	48	0	0	10	60
Price	9	4	36	6	54	10	90	2	18
		Total:	*176*	Total:	**182**	Total:	138	Total:	158

Conclusion

For the RFID system, there are many available, but few have prices making preliminary research difficult. With some further in-depth research, we found pricing for modules made by ACG. At $350, this made it difficult to purchase, even though it had everything we were looking for. Similar options could be found with the Socket Communications device for only $240. To work from a variety of options, the wired reader was added. Having a wired reader would place it at a contemporary level rather than something more oriented towards the future, which is why it is a last resort.

Options about this product require more research because the funding necessary could make this happen, but focusing too much of the budget toward any one module could the project a failure. This is likely to be an option to be selected based on the decisions of other modules since it must be compliant with the rest of the system.

Power Supply

An essential part of any system is the power supply. Since much of the digital world operates at a clean 5V rail, it has become a simpler way to operate. The unfortunate thing about this new 5V world is the fact that the outlet from the wall is still 110/120 V AC. Getting from this voltage to something workable is where the work now lies.

Fixed Supply

There is great reliability in our power network as it stands. This means that any system can be designed to use a fixed supply. The fixed supplies that most North Americans sees is 110/120 V AC from the wall, but there's also 240 and 480 V. For our application, research will only be done using 110 VAC.

Using a standard wall outlet, the 110 VAC will be converted using a power supply. The power needed will be dictated by the most important module, the computer board. The power will be stepped down to 12 VDC and 5 VDC depending on the module. The larger module of the computer board will likely be the only 12 V module, the rest will need 5V.

Variable Supply

Even with the reliable system, battery backups are necessary for applications like security and data backup. Because this system will be needed 24 hours a day, no downtime will be accepted.

Battery Supply, 9 V

As a primary backup, a 9 V battery supply could run the system for a short period of time. Although the power consumption hasn't been determined yet, most 9 V batteries are 700-900 mAh. This means that a system running at 50 mA could last 14-18 hours, which as long as it's a short brown out, is plenty. Other batteries could last far longer, but their life would still be short. **(A 9V battery?)**

Rechargeable Battery

An alternative to a changeable battery supply is one that is rechargeable using the fixed supply already attached to the system. This would be done through a Lithium-Ion or Lithium-Polymer battery. Also dependent on system power consumption, it could range anywhere from 4 to 10 Ah. Depending on total capacity, the battery may eventually dominate the system because of space limitations.

Conclusion

The best system is one that uses the fixed system, but is capable of switching over to a battery system in case of network failure. Using a high capacity Lithium Ion battery with the capability of lasting 30 hours beyond power failure would allow the system very high reliability and near-100% uptime.

Competitor Analysis

One of the major competitors for our product is BioMetrics Technologies. They have a product called the "Authenti-Kate Kiosk". This kiosk has many features that our check-in station does not. Options like an RFID scanner, iris scanner, fingerprint scanner and facial camera are all included. As an extra option voice authentication can also be added. Though this product is not meant to be used for the exact same purpose that the security check-in station is, it serves as formidable competition because of the amount of security that can be implemented on one single machine. The only problem with this device is that the cost is very high. This is where the security check-in station might be able to excel.

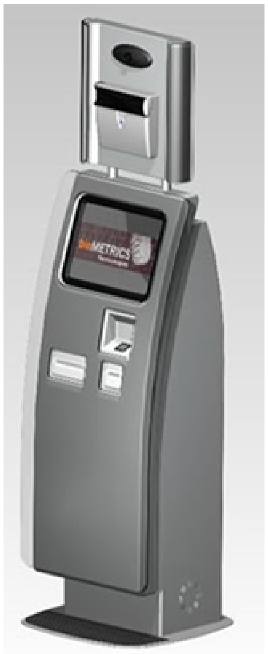


Figure 11: Authenti-Kate Kiosk

Another product in the market that can potentially be competition is DS321 RFID Access Control system produced by Warwick Wireless Limited. Some of the features of this product include: control and sensor units, transponder cards and user programmable relay. The cards have to be placed about 4 *(four)* inches away from the system for it to work function properly.

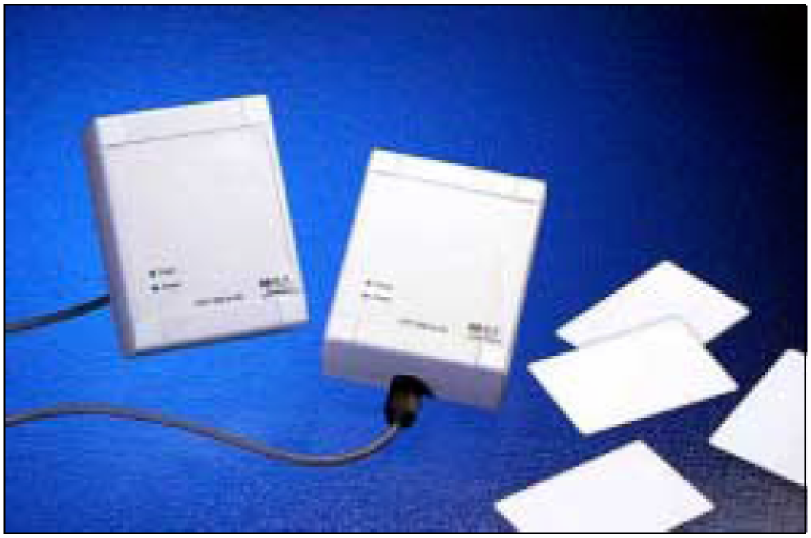


Figure 12: DS321 RFID Access Control

ID Systems is ~~also~~ another company that can be of some competition. Even though it only sells RFID scanners, the product, IT Tag, can stray potential customers away from the security check-in station. It performs some of the same functions as the DS321 RFID, it also has a few extra features which include; EEPROM of 512 bits and 2k bits, unique 64 bit serial number and read/write protection by security. *(weights)*

Value Analysis

Table 14: Competitor Value Analysis

	bioMetrics Technologies			Warwick Wireless Limited			ID Systems			Worcester Polytechnic	
	Authenti-kate Kiosk			DS321 RFID Access Control			IT Tag			Security Check-In Station	
	Value	Rating	Score	Rating	Score		Rating	Score		Rating	Score
Security	10	10	100	6	60		6	60		8	80
Durability	7	8	56	6	42		4	28		7	49
Reliability	8	9	72	8	64		8	64		7	56
Cost	9	3	27	5	45		7	63		8	72
RFID Scanner	5	5	25	5	25		5	25		5	25
Voice Recogniton	5	5	25	0	0		0	0		5	25
Other Options	5	5	25	0	0		0	0		0	0
Total			**305**		**236**			**240**			**307**

Conclusion

As can be seen by the value analysis done on each of the existing products on the market, the secrity check-in station is a cut above the rest. The only competitor that comes close is the Authenti-kate Kiosk by bioMetrics Technologies. The Authenti-kate kiosk should be more efficient than the security check-in station, and loses because of the mere fact that its price is higher than what the security check-in station would be. Since the price of the security check-in station is lower than most of the systems out there, it will be able to compete with other products already in the market.

Conclusion

By analyzing all the data obtained from the market research, we concluded that the security check-in station will be a very formidable product in the market. The voice authentication and the RFID scanner are two aspects of security that not many companies have put together. Though the security check-in station may not have the features of the other products already in the market; it more than makes up for the inadequacies when price is concerned. The lower price alone should attract customers.

Scheduling is also a big issue with the designing and building of this device. A strict schedule will have to be formulated in order to meet customer satisfaction. The prototype will have to built and ready by the end of August 2006. This next section will describe the scheduling of the development of the security check-in station.

Task Specific Gantt chart

Research

The research aspect of our project is the most important stage. It is here where all our planning and design ideas come together and formulate a prototype. We spent the first two weeks in this stage. We needed to decide on a target market and learn about it. Another part of this stage is researching prior art. Prior art is very important to learn about because it can end up making our job a lot easier. For example, we could get ideas from what others have done in our area of interest. This is useful when deciding what parts to use. There could be already made components that we could easily implement in our design that would make our job much easier. Learning about our competitors is also extremely important. We need to know what we are up against and how we will match up against them in the market. We want to be able to offer what they do and more to make our product more appealing. With all this information we can come up with the best solution for our design. Everyone participates in this stage because having all the background knowledge is essential when it comes to the latter stages of development.

During the beginning stages of the project, the amount major problems that could hinder it are minimal. This is deceiving though, as careless research and planning can lead to major problems down the road. This requires us though to be diligent and keep a strict schedule in the later stages because we will not be able to afford many delays with the very short building time given. Our schedule is laid out on the Gantt chart (Appendix C).

Design & Analysis

The design aspect of the project will consist of creating schematics and ordering the appropriate parts needed to complete the building of the prototype. This will help the team resolve a lot of the design flaws before building the prototype. During this phase, we will design the device and check the schematics for any errors. Should problems arise with our initial design, an alternate plan of action is being drafted. This plan will help us identify many of the major problems before the device is built.

Construction

The construction of the device is going to be divided into separate modules. Each module will be individually built and tested by the group ensuring that it is functional before work begins on the next module.

Microphone and Speaker

Both the microphone and speaker will function as one module. These two components will be fed into the embedded PC's audio card for the voice authentication module. This module has a relatively low level of risk, because the both the microphone and the speaker will be wired directly to the embedded PC. This step should take very little time, and the only delay that may occur is if either component is damaged or incompatible with the PC.

Voice Authentication Software

Since the team is allowed to take an already existing voice authentication algorithm, this will reduce the project's difficulty and allow the team to spend time on other modules. Otherwise, this module would take a significant amount of time to complete. The voice authentication module will require a significant amount of attention because of sophisticated nature of this system.

RFID Scanner

The RFID module will be one of the more troublesome modules. When an appropriate RFID scanner is selected, it will have to be able to interface with the software loaded onto the embedded PC. After this task is complete it will then access the database of users. After matching the RFID tag to the user in its database, the program will then start the voice authentication software so that the user can complete the check-in process. The risk of failure in this module is very high and a lot of other modules will depend on the RFID scanner.

Embedded PC

The embedded PC will be the most important module of the device. Every other module will function through it and it will play an integral role in the successful functionality of the device. Some risks that it might present are incompatibility with the other modules, drawing too much power, not having enough memory to perform all the desired tasks etc. This is why the embedded PC will have to be picked with the utmost care, and we will have to make sure that it will be able to perform all the tasks necessary.

LCD

The LCD will communicate directly with the embedded PC. Once the user has been identified, the LCD will display his/her information. After this process is over, the LCD will prompt the user to repeat his/her specific phrase. Risks for the LCD are fairly low, but some risks can arise. Problems like not communicating with the embedded PC, not being able to prompt the user for voice authentication, burning out and compatibility with the system all apply. Another major risk is the risk of the LCD getting damaged by an imposter of some sort. A plan for this problem is still being formulated.

Testing

Testing and debugging will take the most amount of time. The testing phase of this project will primarily include testing the embedded PC, voice authentication system and RFID scanner. Since these are the two important modules for the security check-in station, they will have to be tested the most.

RFID Tests

Some tests that will need to be run for the RFID portion of the device are the testing of the functionality of the RFID system, seeing if the software will access a database of preset users and testing to see if the software will then start the voice authentication module.

Voice Authentication Tests

Tests for the voice authentication portion of the device are as follows: testing to see if the software will recognize a particular voice, testing to see if the software can access the database of users to identify the voice, will lockdown if an unauthorized voice is detected. These tests are involved and have been allotted several weeks in the Gannt chart (Appendix C).

Conclusion

The Gantt chart provided should show the expected schedule for the entire project. This will serve as a milestone marker and a reminder to project deadlines. In Appendix C, a detailed Gantt chart is provided. Our current status is that the proposal for the project is being completed. Once the proposal is done, a parts list will be formulated. Once the parts list is formulated, the parts are then going to be ordered in the beginning of D term and so on. One of the goals for the team is to get as much of the project done before the E term begins which will give us more time for testing and debugging. Completion of the prototype in the expected time will be determined solely on the functionality of each module.

Architectural Description

This section will describe the modules that have been allotted to the security check-in station. It will show what the design approach for the device and how the implementation of the design will take place. Each module will have its own risks and issues to be dealt with, include the design, describe tests to verify functionality, and detail its cost.

Top Level Block Diagram

The top level diagram shown in Appendix A will give the general design approach of the security check-in station. It gives a small prelude to how the design was formulated and how each module will function and communicate. For the security check-in station there are eight modules that need to be dealt with, which include:

- Microphone
- Speaker
- Voice Authentication
- RFID + Software
- Embedded PC
- LCD
- Server
- Power

Modules

This project has a number of complicated modules. Putting all of the pieces together can easily create havoc without a proper level of design know-how and planning. This system carries seven different modules that must be designed and implemented, and an eighth that must be interfaced with.

LCD Module

The LCD module will serve only a few distinct purposes; which include displaying the information of each user once the RFID card is scanned and recognized. It will be connected only to the embedded PC and will take information provided by it. Once this process is complete the PC will send a photo image of the guard with the relevant information, which will indicate that the RFID has been verified. The LCD module will also play a role with the voice authentication. It will prompt the user to say his/her specific phrase.

Voice Authentication Module

The voice authentication module will consist of primarily the software which will analyze the voice of each user. The microphone and speaker modules will work hand in hand with the authentication module. The voice authentication module will then have to communicate with the server and match up the voice authentication data. Once the data is matched up, the embedded PC will then display the verification and then clear the user onto the next section. This module will consist of just the voice authentication software.

RFID and Software Module

The RFID module will be divided into two smaller subdivisions which include the RFID scanner and the software. The RFID scanner will be set to communicate with the embedded PC which will then communicate with the server. Once an ID is scanned, the embedded PC will send a request to the server to see if the RFID tag is in the database. This is where the software portion of the RFID will take over. Once the RFID tag is found in the database; the embedded PC to start the voice authentication section.

Input Module

The input module contains several of the top level modules within it. The system is complicated enough where it asks the user for multiple things to verify identity. The first of these is the RFID reader. This will allow the system to open the reference guide for the specific user. Also, because this serves as the first verification, it will lessen the load on the Voice Authentication module. The second module in the input module is the microphone, which will be used as the second verification tool.

Power Module

The purpose of the power module is to provide the other modules with the appropriate amount of power. The power module will have to make sure that the different outputs are isolated and stable at all times.

Information Output Module

Like the input module, the output module is made up of smaller modules. The first of these is the speaker. Using computer generated information, the speaker will output to the user, likely for more input. The same services will be generated for display by the display module. Only when the verification is complete, the output module returns the user information without expecting an input.

Embedded PC Module

The embedded PC will interface with the RFID reader, Speaker, Microphone, LCD and network. Once the user places their RFID card in range of the reader the PC will form an initial request with the RFID tag ID to the server for basic information about the user. Using the speaker, the PC and screen will prompt the user to speak to the microphone. Once the user speaks, the PC will send a second request to the server with the user's RFID tag ID and voice print encoded in a WAV file then wait for a response from the server. The server will reply by either returning either a valid or invalid response depending on whether the user's RFID tag ID and voice print matched those in its database.

Server Module

The server will maintain a database of the valid RFID tags and voice print files and provide an interface for the embedded PC to verify the end-user's identity. The server will also handle the voice authentication module since it will have the most computing power and can prevent the client PCs from having direct access to the voice print authentication files thereby adding an additional layer of security to the data on the server.

Hardware and Software Partitioning

Much of the project will be mixed using both software and hardware. The microphone and speaker will operate completely with hardware along with the power module. Because using hardware reduces debugging time, the remainder of the project will involve streamlining some of the software to hardware.

Module Descriptions

This section is intended to let the reader see the overall function of the system. These sections are split between the hardware, mixed and software modules.

These modules can be built on a prototyping board and transferred to a printed circuit board. These modules require no programming and testing can be done without software.

LCD Module

The LCD chosen for the device will be the Compaq 8.4 inch color LCD display. The LCD will be a TFT display, which will have a resolution type of XGA (640 x 480 pixels). The LCD module will be used to visually reinforce the transaction between the user and the security check-in station. The LCD will not drain too much power from the overall system. Since its uses are limited and there is no touch screen capability attached to it, the LCD will be simply an output to the system.

RFID Module

The RFID scanner module will consist of the HF dual ISO Reader produced by ACG. The RF transmit frequency is about 13.56 MHz. Its current consumption is relatively low, only needing about 200 mA to operate and 1 mA when in standby. Depending on the tag and the antenna it has a nominal reading distance of about 95 mm. The communications protocol that it uses is specific ASCII or binary protocol and its communication parameter is in between 9600 Bit/s to 460 Bit/s.

Microphone Module

As a subset of the input module, this will serve second in the line of verification tools for the system. The microphone will operate at the 2 V rated voltage and will have two connections. One of the wires will be connected to ground; the other will be the signal input. As the system requires there will be hardware based amplification and filtering before the system gets to the computer. The microphone input to the system will also be based on what the computer will accept as an input. If the system requires a 1/8" stereo input, it will be converted. This will be connected to the power module using a voltage regulator that will have been stepped down from 5 V.

Speaker Module

Being used as an output to the system the speaker has one main connection, to the output of the computer. All of the sounds for the speaker will be computer generated. This allows for versatility in the model of the speaker. Depending on whether the system needs a louder speaker, the system can move from using a .25 W to 1 or 3 W, without great space or power changes. The greatest size the speaker will be is a 77 mm diameter space, from a 1 or 3 W speaker.

Power Module

The power module will consist of a large step voltage transformer followed by a cascade of small step transformers. The first step will bring in the 110 V AC and transform it to two voltages in DC. The 19 V DC will be used to operate the system and charge the battery. 5 V will be output for the other components that require a lower voltage. To get proper operation using both the battery and power supply, the power will feed into a buck converter. From the battery, this will output 12 V for the computer and through another buck converter, the 2 V needed to operate the speaker and microphone. The display, which is the highest drain item, will be non-operational in the case of a power failure. The system will still be able to operate on the two steps, RFID-VR verification without the monitor. It has just been placed to ease the use for the user.

Testing

Testing the hardware modules will be considerably easier than the software modules; however it may require tedious amounts of hours if it is done incorrectly. Using a multimeter, the voltages will be verified to be the correct value. The current for the circuits will also be noted as it is necessary to keep an eye on the power consumed. For the majority of the circuits, the power will be low and require no active heat dissipation.

System Integration

With eight different modules, they must all be pieced together in a clear and logical manner. This section covers how each of the modules will be placed overall and how each will be tested together.

Hardware Integration and Testing

The embedded PC will be the heart of the device. Everything will come and go through it, so nothing will be totally independent from it. The input module which consists of the microphone and the RFID scanner will communicate with the embedded PC as inputs. Once the RFID card is scanned, the software will be run in the embedded PC. After the first part of authentication is complete, the voice authentication part will take over, which is where the microphone will start the voice authentication process.

The LCD and speaker will consist of the output module from the PC. Once the RFID card is scanned the LCD will display the particular guard's information. After this task is complete it will then show the guard the word he/she will have to speak into the microphone. The speaker will also tell the guard which words to say. This will ensure that the guard will say the correct word. Both the input and output modules will be tested separately at first to ensure functionality.

Software Integration and Testing

The software for the RFID scanner and the voice authentication will be accessed through a network. The network will be attached to a server which will access the database of guards, the RFID software and the voice authentication software. The reason all the necessary software is accessed through a network and sever is to prevent any possible tampering with the embedded PC.

Risks and Impending Issues

There are some major risks that can cause big problems in the future with the design and implementation of the security check-in device. The embedded PC will have to be compatible with all the other modules in the device. The device is centered on the embedded PC, and its ability to effectively communicate and function with the other modules. Other risks include the failure of the voice authentication and RFID modules. Should either of those modules not work properly the project will be set back and new options will have to be evaluated.

One of the issues affecting this project at the moment include the budget. The total cost of the parts needed to build the security check-in station exceeds the $375 allotted for the project. The budgeting issue can cause problems, and if more expenses can not be removed into the budget, then the design may need to be changed.

Emergency Options

Not everything goes according to plan, in the case that completion of the project is hampered by cost overruns, we have come up with options to reduce the cost of the project by increasing low level construction. Two modules can be converted to base level projects.

RFID Development

In the case that the RFID system that we are looking to purchase cannot be found at a reasonable price, the team will have to develop an RFID system from scratch. It will begin with preliminary research into existing RFID systems. Unfortunately, since most of the MQP development will work on not “reinventing the wheel”, the implementation of an “RFID from scratch” system will be used as a last resort.

Following preliminary research, the first step will be to develop the system to be compatible with the embedded PC. Because we’d be developing it for exclusive operation with our system, we could design it to be proprietary and could violate the ISO standards that dictate RFID to be 13.56 MHz. We could also use the other ISO standard that uses low frequency at 136 kHz.

The parts that we would have to look into purchasing would be the receiver and transmitter first; followed by the other pieces that would allow the receiver to wake up the transmitter. The module will likely be powered by the embedded PC.

Voice Authentication Development

Assuming we could not afford to license a professional voice authentication system, we have explored some different options for developing our own code. We concluded that the most practical systems would be MATLAB and C++. Since our system requires the application of filters and perform Fourier transforms on a sound files, MATLAB provides all the tools and functionality that we need to perform these operations. If we were to use a programming language such as C++, we would have to build these ourselves. Using the MATLAB compiler lets us develop C/C++ code from our MATLAB code so that we have the functionality of machine code, without having to develop the tools ourselves.

Budget Proposal

The overall budget we have been looking at has been approximately $500. Because the development of the product is similar to that of an automatic teller machine, many of the modules by themselves are expensive. They have been broken down in the following table into our original budget and the actual cost.

Table 15: Budget Proposal

Module	Price Range ($)	Price Found	Savings/Loss	Percent Overall
Embedded PC	100-200	150	Par	30%
LCD Monitor	50-100	30	$20	6%
Microphone	1-5	2	Par	.4%
Speaker	1-5	2	Par	.4%
RFID	100-200	350	-$150	70%
VR Engine	50-100	N/A		
Server	0	0	Par	0%
TOTAL		534	-$34	106.8%

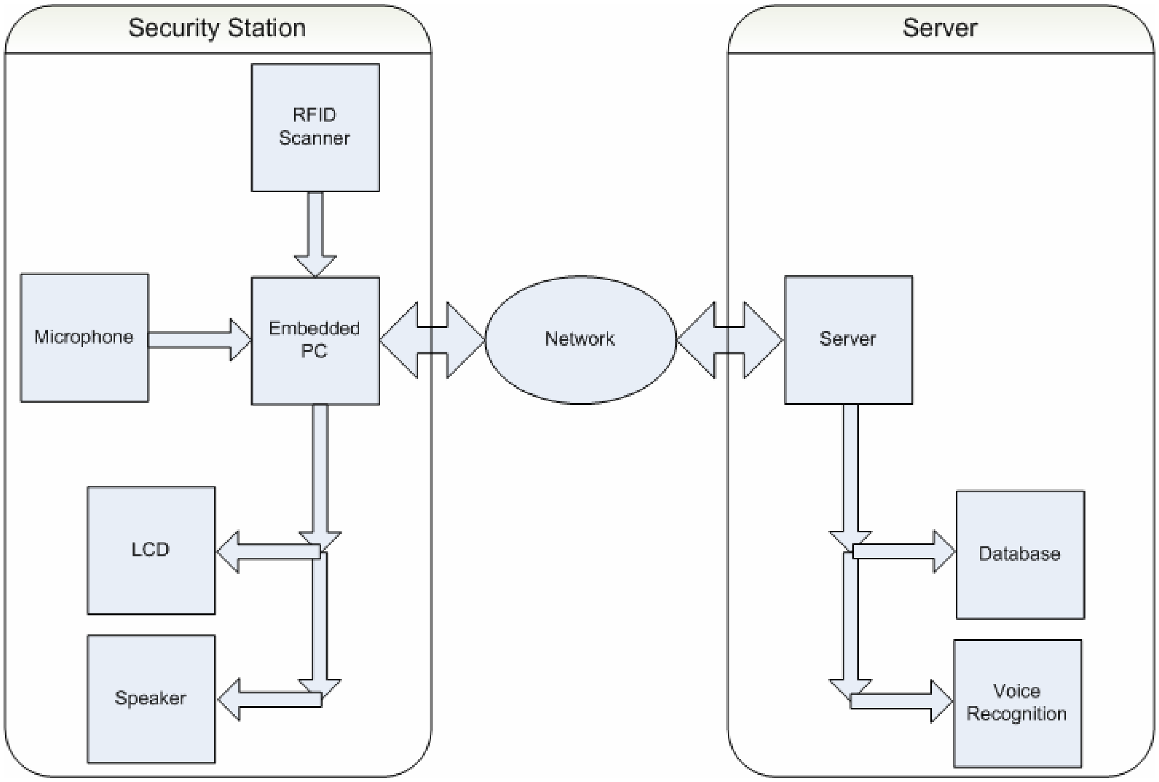
On top of the allotted $125 per person that the MQP budget allows, we have allotted an extra $125 total. Unfortunately, because of the cost of the RFID, we are still 6.8% over budget. Ideally, if we could find an RFID system under $200, we would fall below the $500 budget and even the $375 budget.

Further research must be done to find a reasonable RFID system. One alternative is to replace the RFID transceiver with a more cost-effective alternative. If we used a magnetic card identification system, we could save $330. However, this sacrifice would reduce the novelty of the project. Therefore, we hope to find a less expensive RFID system to maintain this project's newness.

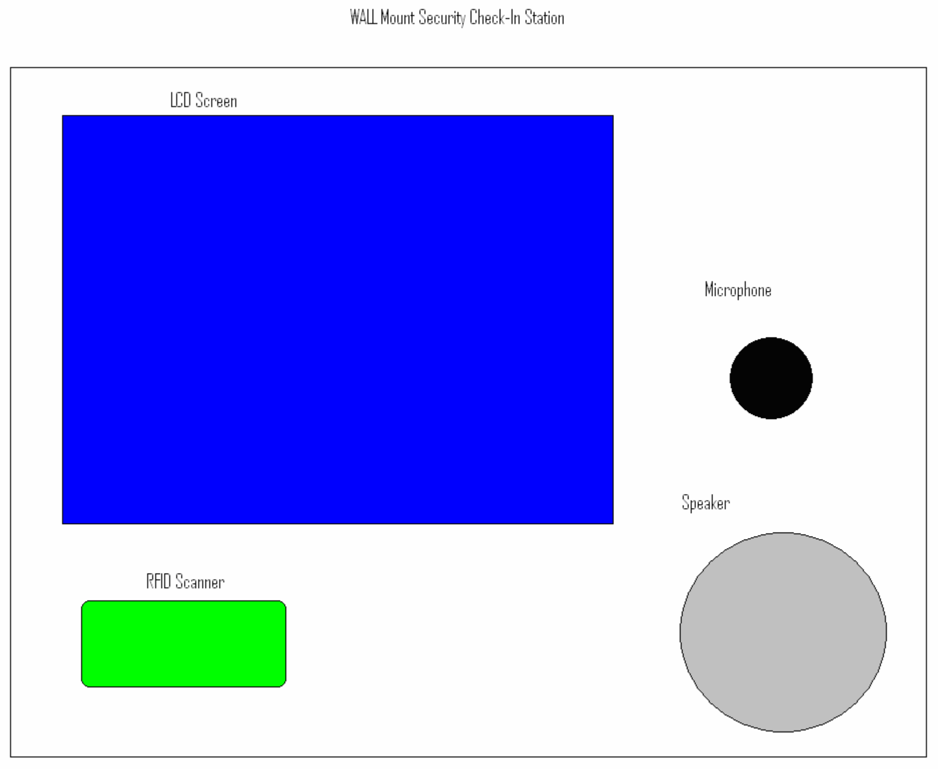
Overall Conclusion

As the criminal world continues to develop more sophisticated techniques for bypassing existing security systems, businesses need new tools to secure their sensitive facilities. Our project hopes to use RFID technology and Voice Authentication to authenticate users at our station. The hardware module will consist of an Embedded PC that shall interface with an LCD, RFID Scanner, network controller, Speaker and Microphone while the software module will consist of an RFID reader and a sound recording module on a Linux platform that uses an HTTP tunnel to interact with the server. The server will consist of a standard PC running web server software to interact with the check-in stations, a database of the valid RFID tags and voice signatures and a piece of software to analyze the user's voice. Using this combination of hardware and selected software packages, we hope to develop a system that will guarantee that only authorized users may verify their identity.

Appendix A - Functional Block Diagram



Appendix B - Artist’s Rendition



23. Appendix B: RFID Schematic

24. Appendix C: RFID Datasheets

Parallax RFID Datasheet: p86-91

MAX 232A/233A Datasheets p92-95

MAX232A/233A Datasheet (p17) p96

25. Appendix D: RFID Software Code

26. Appendix E: Voice Authentication Code

26.1. voiceRecord MATLAB Code

```
%% Record User

recorderObject = audiorecorder(44100,16,1);
fprintf('Recording Voice...\n');
record(recorderObject,3);
pause(3);
fprintf('Recording Complete!\n');
fprintf('Press any key to listen to recording\n');
pause;
audioData = getaudiodata(recorderObject);
wavplay(audioData,44100);
playCondition=input('Listen again? (y/n) ','s');

if playCondition == 'y'
   wavplay(audioData,44100);
end
writeCondition=input('Write to File? (y/n) ', 's');
if writeCondition == 'y'
   wavwrite(audioData,44100,16,name);
   fprintf('File written!\n');
end

%% Signal_norm
[y3,Fs] = wavread(name);
c       = y3(1:length(y3));

c       = c-mean(c);
SCALE   = max(abs(c));
S       = c/(SCALE);
time    = [1:length(y3)]*(1/Fs);



wavplay(S,44100);
wavwrite(S,44100,16,name);

%% Wav_filter
%Take in Signal and Plot Amplitude.

[S,Fs]=wavread(name);
d=1/Fs;
t=0:d:(length(S)-1)*d;
```

```
Z1=abs(fft(S));
S1=Z1(1:floor(length(S)/2));
df=Fs/length(Z1);
f=0:df:df*(length(S1)-1);

%% Create and plot high-pass filter to attenuate frequencies below 1650 Hz.

[B,A] = butter(5, 1650/44100,'high');


%% Filter signal and flip to decode secret message.

f4 = filter(B,A,S);


%% Compare_Plot

j=Z1;
h=f4;

%% Peak Finder
[k1b,v1b]=findpeaks(j(1:100),100000);
[k2b,v2b]=findpeaks(j(100:200),100000);
[k3b,v3b]=findpeaks(j(200:300),100000);
[k4b,v4b]=findpeaks(j(300:400),100000);
[k5b,v5b]=findpeaks(j(400:500),100000);
[k6b,v6b]=findpeaks(j(500:600),100000);
[k7b,v7b]=findpeaks(j(600:700),100000);
[k8b,v8b]=findpeaks(j(800:900),100000);
[k9b,v9b]=findpeaks(j(900:1000),100000);
[k10b,v10b]=findpeaks(j(1000:1100),100000);
[k11b,v11b]=findpeaks(j(1100:1200),100000);
[k12b,v12b]=findpeaks(j(1200:1300),100000);
[k13b,v13b]=findpeaks(j(1300:1400),100000);
[k14b,v14b]=findpeaks(j(1400:1500),100000);
[k15b,v15b]=findpeaks(j(1500:1600),100000);
[k16b,v16b]=findpeaks(j(1600:1700),100000);
[k17b,v17b]=findpeaks(j(1700:1800),100000);
[k18b,v18b]=findpeaks(j(1800:1900),100000);
[k19b,v19b]=findpeaks(j(2000:2100),100000);
[k20b,v20b]=findpeaks(j(2100:2200),100000);
[k21b,v21b]=findpeaks(j(2200:2300),100000);
[k22b,v22b]=findpeaks(j(2300:2400),100000);
[k23b,v23b]=findpeaks(j(2400:2500),100000);
[k24b,v24b]=findpeaks(j(2500:2600),100000);
[k25b,v25b]=findpeaks(j(2600:2700),100000);
```

```
[k26b,v26b]=findpeaks(j(2700:2800),100000);
[k27b,v27b]=findpeaks(j(2800:2900),100000);
[k28b,v28b]=findpeaks(j(2900:3000),100000);

 ans3 =
sum(k1b)+sum(k2b)+sum(k3b)+sum(k4b)+sum(k5b)+sum(k6b)+sum(k7b)+sum(k8b)+s
um(k9b)+sum(k10b)+sum(k11b)+sum(k12b)++sum(k13b)+sum(k14b)+sum(k15b)++su
m(k16b)++sum(k17b)+sum(k18b)+sum(k19b)+sum(k20b)+sum(k21b)+sum(k22b)+sum(
k23b)+sum(k24b)+sum(k25b)+sum(k26b)+sum(k27b)+sum(k28b);

%% Signal_Cov

w5 = cov(j,j);
x5 = w5(1,1);

w6 = cov(h,h);
x6 = w6(1,1);
```

26.2. voiceCheck MATLAB Code

```
%% Wav_filter
%%Take in Signal and Plot Amplitude.

[y1,Fs]=wavread(n1);
d=1/Fs;
t=0:d:(length(y1)-1)*d;
z1=abs(fft(y1));
s1=z1(1:floor(length(y1)/2));
df=Fs/length(z1);
f=0:df:df*(length(s1)-1);
%% Create and plot high-pass filter to attenuate frequencies below 1650 Hz.

[B,A] = butter(5, 1650/44100,'high');
%% Filter signal and flip to decode secret message.

f1 = filter(B,A,y1);

%% Wav_filter2
[y2,Fs]=wavread(n2);
d=1/Fs;
t=0:d:(length(y2)-1)*d;
z2=abs(fft(y2));
s2=z2(1:floor(length(y2)/2));
df=Fs/length(z2);
f=0:df:df*(length(s2)-1);
%% Create and plot high-pass filter to attenuate frequencies below 1650 Hz.

[B,A] = butter(5, 1650/44100,'high');
%% Filter signal and flip to decode secret message.

f2 = filter(B,A,y2);

%% Compare_plot
x=z1;
y=z2;
a=f1;
b=f2;

%% Peak_Compare

% Find Peaks of Signal X
[k1,v1]=findpeaks(x(1:100),100000);
[k2,v2]=findpeaks(x(100:200),100000);
```

```
[k3,v3]=findpeaks(x(200:300),100000);
[k4,v4]=findpeaks(x(300:400),100000);
[k5,v5]=findpeaks(x(400:500),100000);
[k6,v6]=findpeaks(x(500:600),100000);
[k7,v7]=findpeaks(x(600:700),100000);
[k8,v8]=findpeaks(x(800:900),100000);
[k9,v9]=findpeaks(x(900:1000),100000);
[k10,v10]=findpeaks(x(1000:1100),100000);
[k11,v11]=findpeaks(x(1100:1200),100000);
[k12,v12]=findpeaks(x(1200:1300),100000);
[k13,v13]=findpeaks(x(1300:1400),100000);
[k14,v14]=findpeaks(x(1400:1500),100000);
[k15,v15]=findpeaks(x(1500:1600),100000);
[k16,v16]=findpeaks(x(1600:1700),100000);
[k17,v17]=findpeaks(x(1700:1800),100000);
[k18,v18]=findpeaks(x(1800:1900),100000);
[k19,v19]=findpeaks(x(2000:2100),100000);
[k20,v20]=findpeaks(x(2100:2200),100000);
[k21,v21]=findpeaks(x(2200:2300),100000);
[k22,v22]=findpeaks(x(2300:2400),100000);
[k23,v23]=findpeaks(x(2400:2500),100000);
[k24,v24]=findpeaks(x(2500:2600),100000);
[k25,v25]=findpeaks(x(2600:2700),100000);
[k26,v26]=findpeaks(x(2700:2800),100000);
[k27,v27]=findpeaks(x(2800:2900),100000);
[k28,v28]=findpeaks(x(2900:3000),100000);

%%Find Peaks of Signal Y
[k1a,v1a]=findpeaks(y(1:100),100000);
[k2a,v2a]=findpeaks(y(100:200),100000);
[k3a,v3a]=findpeaks(y(200:300),100000);
[k4a,v4a]=findpeaks(y(300:400),100000);
[k5a,v5a]=findpeaks(y(400:500),100000);
[k6a,v6a]=findpeaks(y(500:600),100000);
[k7a,v7a]=findpeaks(y(600:700),100000);
[k8a,v8a]=findpeaks(y(800:900),100000);
[k9a,v9a]=findpeaks(y(900:1000),100000);
[k10a,v10a]=findpeaks(y(1000:1100),100000);
[k11a,v11a]=findpeaks(y(1100:1200),100000);
[k12a,v12a]=findpeaks(y(1200:1300),100000);
[k13a,v13a]=findpeaks(y(1300:1400),100000);
[k14a,v14a]=findpeaks(y(1400:1500),100000);
[k15a,v15a]=findpeaks(y(1500:1600),100000);
[k16a,v16a]=findpeaks(y(1600:1700),100000);
[k17a,v17a]=findpeaks(y(1700:1800),100000);
[k18a,v18a]=findpeaks(y(1800:1900),100000);
```

```
[k19a,v19a]=findpeaks(y(2000:2100),100000);
[k20a,v20a]=findpeaks(y(2100:2200),100000);
[k21a,v21a]=findpeaks(y(2200:2300),100000);
[k22a,v22a]=findpeaks(y(2300:2400),100000);
[k23a,v23a]=findpeaks(y(2400:2500),100000);
[k24a,v24a]=findpeaks(y(2500:2600),100000);
[k25a,v25a]=findpeaks(y(2600:2700),100000);
[k26a,v26a]=findpeaks(y(2700:2800),100000);
[k27a,v27a]=findpeaks(y(2800:2900),100000);
[k28a,v28a]=findpeaks(y(2900:3000),100000);

ans1 =
sum(k1)+sum(k2)+sum(k3)+sum(k4)+sum(k5)+sum(k6)+sum(k7)+sum(k8)+sum(k9)+su
m(k10)+sum(k11)+sum(k12)++sum(k13)+sum(k14)+sum(k15)++sum(k16)++sum(k17)+
sum(k18)+sum(k19)+sum(k20)+sum(k21)+sum(k22)+sum(k23)+sum(k24)+sum(k25)+s
um(k26)+sum(k27)+sum(k28);
ans2 =
sum(k1a)+sum(k2a)+sum(k3a)+sum(k4a)+sum(k5a)+sum(k6a)+sum(k7a)+sum(k8a)+su
m(k9a)+sum(k10a)+sum(k11a)+sum(k12a)++sum(k13a)+sum(k14a)+sum(k15a)++sum(
k16a)++sum(k17a)+sum(k18a)+sum(k19a)+sum(k20a)+sum(k21a)+sum(k22a)+sum(k23
a)+sum(k24a)+sum(k25a)+sum(k26a)+sum(k27a)+sum(k28a);

if (ans1<ans3)
   p1=ans1/ans3;
end
 if (ans3<ans1)
   p1=ans3/ans1;
 end

 if (ans2<ans3)
   p2=ans2/ans3;
 end

 if (ans3<ans2)
   p2=ans3/ans2;
 end
    fb1= p1;
    fb2= p2;
w3= cov(x,x);
 w4= cov(y,y);

x3 = w3(1,1);
x4 = w4(1,1);

if (x3<x5)
   p1a=x3/x5;
```

```
end
 if (x5<x3)
    p1a=x5/x3;
 end
if (x4<x5)
    p1b=x4/x5;
end
 if (x5<x4)
    p1b=x5/x4;
 end

%% Signal_cov
 w1= cov(a,a);
 w2= cov(b,b);

x1 = w1(1,1);
x2 = w2(1,1);

 if (x1<x6)
    p2a=x1/x6;
end
 if (x6<x1)
    p2a=x6/x1;
 end

 if (x2<x6)
     p2b=x2/x6;
 end
 if(x6<x2)
     p2b=x6/x2;
 end
 fa1=((p1a+p2a)/2);
 fa2=((p1b+p2b)/2);

%% Final

final1 = (fb1+fa1)/2;
final2 = (fb2+fa2)/2;

PercentMatch = (final1+final2)/2

if (PercentMatch<.8000)
    fprintf('Voice Authentication Failed\n')
end
 if (PercentMatch>=.8500)
    fprintf('Voice Authentication Passed\n')
 end
 if and((PercentMatch>=.8000),(PercentMatch<.8500))
    fprintf('Please Input Second Pass Phrase\n')
end
```

26.3. VoiceAuth2 MATLAB Code

```
function finalcheck = voiceauth2(name,n1,n2);
run voiceRecord
run voiceCheck
```

26.4. Storepass MATLAB Code

```
function storepassword = storepass(name);

%% Record User

recorderObject = audiorecorder(44100,16,1);
fprintf('Recording Voice...\n');
record(recorderObject,3);
pause(3);
fprintf('Recording Complete!\n');
fprintf('Press any key to listen to recording\n');
pause;
audioData = getaudiodata(recorderObject);
wavplay(audioData,44100);
playCondition=input('Listen again? (y/n) ','s');

if playCondition == 'y'
   wavplay(audioData,44100);
end
writeCondition=input('Write to File? (y/n) ', 's');
if writeCondition == 'y'
   wavwrite(audioData,44100,16,name);
   fprintf('File written!\n');
end



[y3,Fs] = wavread(name);
c       = y3(1:length(y3));

c       = c-mean(c);
SCALE   = max(abs(c));
S       = c/SCALE;
time    = [1:length(y3)]*(1/Fs);



wavplay(S,44100);
wavwrite(S,44100,16,name);
```

27. Appendix F: Voice Authentication Experiments

27.1. Voice Authentication Experiment 1

Subject	Distance	Recording #	Phrase #	Frequency	Amplitude	Match		
Chati	20 inches	1	1	396.46	1225.4			
Chati	20 inches	2	1	362.8694	39.9176	47%		
Rob	20 inches	1	1	352.6501	34.1685			
Rob	20 inches	2	1	205.8653	9.4985	42%		
Chati	20 inches	3	1	320.7425	28.7001			
Chati	20 inches	4	1	398.7042	13.7761	64%		
Rob	20 inches	3	1	274.0793	179.1989			
Rob	20 inches	4	1	281.2737	18.3914	54%		
Chati	20 inches	5	2	388.5119	90.1671			
Chati	20 inches	6	2	308.2943	13.398	47%		
Rob	20 inches	5	2	189.4241	13.7067			
Rob	20 inches	6	2	264.3847	7.8795	65%		
Chati	5 inches	7	2	315.4686	50.5886			
Chati	5 inches	8	2	288.847	39.4922	85%		
Rob	5 inches	7	2	194.4667	77.1037			
Rob	5 inches	8	2	275.2492	62.7272	76%		
Chati	5 inches	9	1	322.7197	16.0063			82%
Chati	5 inches	10	1	339.0089	21.8606	84%		
Chati	5 inches	11	1	365.3138	12.3138		75%	82%
Rob	5 inches	9	1	309.995	4.4041			
Rob	5 inches	10	1	275.3843	7.66	73%		
Chati	5 inches	8	1	322.9016	15.1678			85%
Chati	5 inches	9	1	342.206	16.3067	94%		
Chati	5 inches	10	1	359.8465	12.3908		85%	85%
Rob	5 inches	8	1	309.996	4.6231			
Rob	5 inches	9	1	271.1328	7.1328	76%		

Legend

Red	Standing
Blue	Sitting
Yellow	Normalized

27.2. Voice Authentication Experiment 2

Subject	Distance	Recording #	Phrase #	Frequency	Match
Chati	17 inches	1	1	350.5982	
Chati	17 inches	2	1	334.7753	95.486885
Rob	17 inches	1	1	323.576	
Rob	17 inches	2	1	288.1496	89.051598
Ross	17 inches	1	1	250.6169	
Ross	17 inches	2	1	296.6733	84.475718
Chati	17 inches	3	2	269.3784	
Chati	17 inches	4	2	266.1933	98.817611
Rob	17 inches	3	2	355.7773	
Rob	17 inches	4	2	227.8401	64.040089
Ross	17 inches	3	2	348.4621	
Ross	17 inches	4	2	354.3624	98.334953
Chati	17 inches	5	3	356.7124	
Chati	17 inches	6	3	342.1322	95.912618
Rob	17 inches	5	3	441.73	
Rob	17 inches	6	3	267.187	60.486496
Ross	17 inches	5	3	240.5654	
Ross	17 inches	6	3	369.8664	65.041161
Chati	17 inches	7	4	324.6606	
Chati	17 inches	8	4	295.6543	91.065654
Rob	17 inches	7	4	244.4157	
Rob	17 inches	8	4	348.2191	70.190205
Ross	17 inches	7	4	307.8437	
Ross	17 inches	8	4	325.4832	94.580519
Chati	17 inches	9	5	346.8141	
Chati	17 inches	10	5	272.6917	78.627628
Rob	17 inches	9	5	276.1322	
Rob	17 inches	10	5	327.0377	84.434363
Ross	17 inches	9	5	300.0709	
Ross	17 inches	10	5	246.9033	78.466225
Chati	17 inches	11	6	340.4716	
Chati	17 inches	12	6	327.5797	96.213517
Rob	17 inches	11	6	388.2522	
Rob	17 inches	12	6	382.806	98.597252
Ross	17 inches	11	6	261.1773	
Ross	17 inches	12	6	400.3252	65.241284

Phrase 1	Yellow
Phrase 2	Chati,Rob, Ross
Phrase 3	Sri lanka
Phrase 4	Rome,Bolton,Boston
Phrase 5	Worcester, Mass
Phrase 6	Land Rover

27.3. Voice Authentication Experiment 3

Subject	Distance	Recording #	Phrase #	Frequency	Match
Chati	17 inches	1	1	40202	
Chati	17 inches	2	1	44116	91.00%
Rob	17 inches	1	1	40044	
Rob	17 inches	2	1	33748	84%
Ross	17 inches	1	1	41502	
Ross	17 inches	2	1	41706	99%
Mike	17 inches	1	1	24451	
Mike	17 inches	2	1	28185	86%
Chati	17 inches	3	2	43794	
Chati	17 inches	4	2	41679	95%
Rob	17 inches	3	2	42665	
Rob	17 inches	4	2	44417	96%
Ross	17 inches	3	2	37074	
Ross	17 inches	4	2	34147	91%
Mike	17 inches	3	2	29028	
Mike	17 inches	4	2	34740	83%
Chati	17 inches	5	3	41915	
Chati	17 inches	6	3	44052	94%
Rob	17 inches	5	3	46219	
Rob	17 inches	6	3	45897	99%
Ross	17 inches	5	3	48908	
Ross	17 inches	6	3	46831	95%
Mike	17 inches	5	3	39641	
Mike	17 inches	6	3	36342	91%
Chati	17 inches	7	4	41977	
Chati	17 inches	8	4	43215	97%
Rob	17 inches	7	4	45434	
Rob	17 inches	8	4	48643	93%
Ross	17 inches	7	4	35899	
Ross	17 inches	8	4	37137	96%
Mike	17 inches	7	4	45888	
Mike	17 inches	8	4	48513	94%
Chati	17 inches	9	5	43994	
Chati	17 inches	10	5	46010	95%
Rob	17 inches	9	5	43767	
Rob	17 inches	10	5	46856	93%
Ross	17 inches	9	5	38795	
Ross	17 inches	10	5	42387	91%
Mike	17 inches	9	5	41876	
Mike	17 inches	10	5	44799	93%
Chati	17 inches	11	6	44354	
Chati	17 inches	12	6	42330	95%
Rob	17 inches	11	6	38324	
Rob	17 inches	12	6	39573	96%
Ross	17 inches	11	6	35652	
Ross	17 inches	12	6	33483	93%
Mike	17 inches	11	6	39702	90%

Mike	17 inches	12	6	43765
Phrase 1	Yellow			
Phrase 2	Chati,Rob, Ross			
Phrase 3	Sri lanka			
Phrase 4	Rome,Bolton,Boston			
Phrase 5	Worcester, Mass			
Phrase 6	Land Rover			

27.4. *Voice Authentication Experiment 4 Part 1*

Subject	Distance	Recording #	Phrase #	Frequency	Match	Signal Covariance Percentage	Percent Match
Chati	17 inches	1	1	40202			
Chati	17 inches	2	1	44116	91.00%	74.00%	82.50%
Rob	17 inches	1	1	40044			
Rob	17 inches	2	1	33748	84%	71.00%	77.50%
Ross	17 inches	1	1	41502			
Ross	17 inches	2	1	41706	99%	82.00%	90.50%
Mike	17 inches	1	1	24451			
Mike	17 inches	2	1	28185	86%	41.00%	63.50%
Chati	17 inches	3	2	43794			
Chati	17 inches	4	2	41679	95%	58.00%	76.50%
Rob	17 inches	3	2	42665			
Rob	17 inches	4	2	44417	96%	79.00%	87.50%
Ross	17 inches	3	2	37074			
Ross	17 inches	4	2	34147	91%	80.00%	85.50%
Mike	17 inches	3	2	29028			
Mike	17 inches	4	2	34740	83%	56.00%	69.50%
Chati	17 inches	5	3	41915			
Chati	17 inches	6	3	44052	94%	92.00%	93.00%
Rob	17 inches	5	3	46219			
Rob	17 inches	6	3	45897	99%	80.00%	89.50%
Ross	17 inches	5	3	48908			
Ross	17 inches	6	3	46831	95%	83.00%	89.00%
Mike	17 inches	5	3	39641			
Mike	17 inches	6	3	36342	91%	67.00%	79.00%
Chati	17 inches	7	4	41977			
Chati	17	8	4	43215	97%	86.00%	91.50%

	inches						
Rob	17 inches	7	4	45434			
Rob	17 inches	8	4	48643	93%	84.00%	88.50%
Ross	17 inches	7	4	35899			
Ross	17 inches	8	4	37137	96%	71.00%	83.50%
Mike	17 inches	7	4	45888			
Mike	17 inches	8	4	48513	94%	96.00%	95.00%
Chati	17 inches	9	5	43994			
Chati	17 inches	10	5	46010	95%	78.00%	86.50%
Rob	17 inches	9	5	43767			
Rob	17 inches	10	5	46856	93%	79.00%	86.00%
Ross	17 inches	9	5	38795			
Ross	17 inches	10	5	42387	91%	80.00%	85.50%
Mike	17 inches	9	5	41876			
Mike	17 inches	10	5	44799	93%	63.00%	78.00%
Chati	17 inches	11	6	44354			
Chati	17 inches	12	6	42330	95%	40.00%	67.50%
Rob	17 inches	11	6	38324			
Rob	17 inches	12	6	39573	96%	37.00%	66.50%
Ross	17 inches	11	6	35652			
Ross	17 inches	12	6	33483	93%	83.00%	88.00%
Mike	17 inches	11	6	39702			
Mike	17 inches	12	6	43765	90%	70.00%	80.00%
Phrase 1	**Yellow**						
Phrase 2	**Chati,Rob, Ross**						
Phrase 3	**Sri lanka**						
Phrase 4	**Rome,Bolton,Boston**						
Phrase 5	**Worcester, Mass**						
Phrase	**Land**						

6 Rover

27.5. Voice Authentication Experiment 4 Part 2

75% Cutoff

Phrase 1 - 'Yellow' Trial 1

	Chati	Rob	Ross	Mike
Chati	Pass	Pass	Fail	Fail
Rob	Pass	Pass	Fail	Fail
Ross	Fail	Fail	Pass	Fail
Mike	Fail	Fail	Fail	Fail

Phrase 1 - 'Yellow' Trial 2

	Chati	Rob	Ross	Mike
Chati	Pass	Fail	Fail	Fail
Rob	Fail	Pass	Fail	Pass
Ross	Fail	Fail	Pass	Fail
Mike	Fail	Pass	Fail	Pass

Phrase 2 - 'Respective Name' Trial 1

	Chati	Rob	Ross	Mike
Chati	Pass	Pass	Pass	Fail
Rob	Pass	Pass	Pass	Fail
Ross	Pass	Pass	Pass	Fail
Mike	Fail	Fail	Fail	Fail

Phrase 2 - 'Respective Name' Trial 2

	Chati	Rob	Ross	Mike
Chati	Pass	Pass	Pass	Pass
Rob	Pass	Pass	Pass	Pass
Ross	Pass	Pass	Pass	Pass
Mike	Pass	Pass	Pass	Fail

Phrase 3 - 'Sri lanka' Trial 1

	Chati	Rob	Ross	Mike
Chati	Pass	Pass	Fail	Fail
Rob	Pass	Pass	Fail	Fail
Ross	Fail	Fail	Pass	Fail
Mike	Fail	Fail	Fail	Pass

Phrase 3 - 'Sri lanka' Trial 2

	Chati	Rob	Ross	Mike
Chati	Pass	Pass	Fail	Pass
Rob	Pass	Pass	Fail	Pass
Ross	Fail	Fail	Pass	Fail
Mike	Pass	Pass	Fail	Pass

Phrase 4 - 'City Of Birth' Trial 1

	Chati	Rob	Ross	Mike
Chati	Pass	Fail	Fail	Fail
Rob	Fail	Pass	Pass	Pass
Ross	Fail	Pass	Pass	Pass
Mike	Fail	Pass	Pass	Pass

Phrase 4 - 'City Of Birth' Trial 2

	Chati	Rob	Ross	Mike
Chati	Pass	Fail	Fail	Fail
Rob	Fail	Pass	Fail	Pass
Ross	Fail	Fail	Pass	Fail
Mike	Fail	Pass	Fail	Pass

Phrase 5 - 'Worcester, Mass' Trial 1

	Chati	Rob	Ross	Mike
Chati	Pass	Pass	Pass	Pass
Rob	Pass	Pass	Pass	Pass
Ross	Pass	Pass	Pass	Pass
Mike	Pass	Pass	Pass	Pass

Phrase 5 - 'Worcester, Mass' Trial 2

	Chati	Rob	Ross	Mike
Chati	Pass	Pass	Fail	Fail
Rob	Pass	Pass	Pass	Pass
Ross	Fail	Pass	Pass	Pass
Mike	Fail	Pass	Pass	Pass

Phrase 6 - 'Land Rover' Trial 1

	Chati	Rob	Ross	Mike
Chati	Fail	Fail	Fail	Fail
Rob	Fail	Fail	Fail	Fail
Ross	Fail	Fail	Pass	Fail
Mike	Fail	Pass	Fail	Pass

Phrase 6 - 'Land Rover' Trial 2

	Chati	Rob	Ross	Mike
Chati	Fail	Pass	Fail	Fail
Rob	Pass	Fail	Fail	Fail
Ross	Fail	Fail	Pass	Fail
Mike	Pass	Fail	Fail	Pass

27.6. Voice Authentication Experiment 4 Part 3

85% Cutoff

Phrase 1 - 'Yellow' Trial 1

	Chati	Rob	Ross	Mike
Chati	Fail	Pass	Fail	Fail
Rob	Pass	Pass	Fail	Fail
Ross	Fail	Fail	Pass	Fail
Mike	Fail	Fail	Fail	Pass

Phrase 1 - 'Yellow' Trial 2

	Chati	Rob	Ross	Mike
Chati	Pass	Fail	Fail	Fail
Rob	Fail	Fail	Fail	Fail
Ross	Fail	Fail	Pass	Fail
Mike	Fail	Pass	Fail	Pass

Phrase 2 - 'Respective Name' Trial 2

	Chati	Rob	Ross	Mike
Chati	Fail	Fail	Fail	Fail
Rob	Fail	Pass	Fail	Fail
Ross	Fail	Fail	Pass	Fail
Mike	Fail	Pass	Fail	Pass

Phrase 3 - 'Sri Lanka' Trial 1

	Chati	Rob	Ross	Mike
Chati	Pass	Pass	Fail	Fail
Rob	Pass	Pass	Fail	Fail
Ross	Fail	Fail	Pass	Fail
Mike	Fail	Fail	Fail	Pass

Phrase 3 - 'Sri Lanka' Trial 2

	Chati	Rob	Ross	Mike
Chati	Pass	Pass	Fail	Pass
Rob	Pass	Pass	Fail	Fail
Ross	Fail	Fail	Fail	Fail
Mike	Pass	Fail	Fail	Pass

Phrase 4 - 'City Of Birth' Trial 1

	Chati	Rob	Ross	Mike
Chati	Pass	Fail	Fail	Fail
Rob	Fail	Fail	Pass	Pass
Ross	Fail	Pass	Fail	Fail
Mike	Fail	Pass	Fail	Pass

Phrase 4 - 'City Of Birth' Trial 2

	Chati	Rob	Ross	Mike
Chati	Pass	Fail	Fail	Fail
Rob	Fail	Pass	Fail	Pass
Ross	Fail	Fail	Pass	Fail

Mike	Fail	Pass	Fail	Fail

Phrase 5 - 'Worcester, Mass' Trial 1

	Chati	Rob	Ross	Mike
Chati	Fail	Fail	Fail	Pass
Rob	Fail	Pass	Pass	Fail
Ross	Fail	Pass	Pass	Pass
Mike	Pass	Fail	Pass	Fail

Phrase 5 - 'Worcester, Mass' Trial 2

	Chati	Rob	Ross	Mike
Chati	Pass	Fail	Fail	Fail
Rob	Fail	Pass	Fail	Fail
Ross	Fail	Fail	Pass	Pass
Mike	Fail	Fail	Pass	Pass

Phrase 6 - 'Land Rover' Trial 1

	Chati	Rob	Ross	Mike
Chati	Pass	Fail	Fail	Fail
Rob	Fail	Fail	Fail	Fail
Ross	Fail	Fail	Pass	Fail
Mike	Fail	Fail	Fail	Pass

Phrase 6 - 'Land Rover' Trial 2

	Chati	Rob	Ross	Mike
Chati	Fail	Fail	Fail	Fail
Rob	Fail	Pass	Fail	Fail
Ross	Fail	Fail	Pass	Fail
Mike	Fail	Fail	Fail	Fail

27.7. Voice Authentication Experiment 5

MATLAB Signal Normalization Results

		Phrase 1 - 'Yellow'						**Phrase 2 - 'Respective Name'**			
	Chati	Rob	Jeff	Mom	Dad		Chati	Rob	Jeff	Mom	Dad
Chati	Pass	Fail	Fail	Fail	Fail	**Chati**	Pass	Fail	Fail	Fail	Fail
Rob	Fail	Pass	Fail	Fail	Fail	**Rob**	Fail	Pass	Fail	Fail	Fail
Jeff	Fail	Fail	Pass	Fail	Fail	**Jeff**	Fail	Fail	Pass	Fail	Fail
Mom	Fail	Fail	Fail	Pass	Fail	**Mom**	Fail	Fail	Fail	Pass	Fail
Dad	Fail	Fail	Fail	Fail	Pass	**Dad**	Fail	Fail	Fail	Fail	Pass

		Phrase 3 - 'Sri Lanka'						**Phrase 4 - 'City of Birth'**			
	Chati	Rob	Jeff	Mom	Dad		Chati	Rob	Jeff	Mom	Dad
Chati	Pass	Fail	Fail	Fail	Fail	**Chati**	Pass	Fail	Fail	Fail	Fail
Rob	Fail	Pass	Fail	Fail	Fail	**Rob**	Fail	Pass	Fail	Fail	Fail
Jeff	Fail	Fail	Fail	Fail	Fail	**Jeff**	Fail	Fail	Fail	Fail	Fail
Mom	Fail	Fail	Fail	Pass	Fail	**Mom**	Fail	Fail	Fail	Pass	Fail
Dad	Fail	Fail	Fail	Fail	Pass	**Dad**	Fail	Fail	Fail	Fail	Pass

		Phrase 5 - 'Worcester, Mass'						**Phrase 6 - 'Land Rover'**			
	Chati	Rob	Jeff	Mom	Dad		Chati	Rob	Jeff	Mom	Dad
Chati	Pass	Fail	Fail	Fail	Fail	**Chati**	Pass	Fail	Fail	Fail	Fail
Rob	Fail	Fail	Fail	Fail	Fail	**Rob**	Fail	Fail	Fail	Fail	Fail
Jeff	Fail	Fail	Pass	Fail	Fail	**Jeff**	Fail	Fail	Pass	Fail	Fail
Mom	Fail	Fail	Fail	Pass	Fail	**Mom**	Fail	Fail	Fail	Pass	Fail
Dad	Fail	Fail	Fail	Fail	Pass	**Dad**	Fail	Fail	Fail	Fail	Pass

28. Appendix G: Client-Server Code

Appendix H: RFID Progress Reports

28.1. June 7, 2006 – June 13, 2006

RFID Milestones over next 5 days

Hardware:

Talk to Parallax/Mouser:	1000	6/7/06	X
Customize and Order:	1030	6/7/06	X
Expected Arrival	-	6/14/06	

Software:

View Datasheets on Comm	1100	6/7/06	X
Find Previous Methods	1200	6/7/06	on datasheet
Determine SW Dev Tool	1400	6/7/06	X
Use sample code from Prev	1600	6/7/06	
Begin simple code	0900	6/8/06	
With single transmit	1000	6/8/06	
Check status of Order	1030	6/8/06	
Code with multiple transmit	1330	6/8/06	tentative

28.2. June 14, 2006 – June 20, 2006

Prior Seven Days:

- Researched and Ordered Parallax RFID Reader for arrival on 6/14/06
- Wrote sections on preliminary research and design of RFID
- Planned implementation to convert from single dataport to serial
- Arrival of RFID units to Worcester, 1 day early (6/13/06)
- Review of Parallax RFID datasheets

Next Seven Days:

- Discrete evaluation of received parts
- Review of Maxim 232 mount for serial conversion
- Code RFID to serial conversion
- Testing (into July if necessary)
    - RFID to reader
    - Reader to computer
    - Serial port to PHP protocol

Preliminary Schedule:

6/15/06

1700	Arrive Lab/Look over received parts
1730	Review Datasheets and effective prior applications
1900	Write-up plan of action for next day and report

6/16/06

900	Begin code
1200	Discrete building

1400	Review coding from day
1700	Write up progress of day into report
1900	Plan of action for next day

6/17/06

900	Into lab

28.3. June 21, 2006 – July 12, 2006

Prior Seven Days:

Assembled and tested reader alone
Received MAX232A IC with datasheets
Wrote sections on functional and form implementation
Found correct DE-9F port for connector
Drew up schematic for module

Next Three Days (7/12-7/14):

Complete circuit in functional implementation
Finish writing testing section
Test module through cascading stage tests

Preliminary Schedule:

6/23/06

900	Restart writing from prior day
1000	Meeting

6/27-7/11: Finish any critical writing

7/12/06

900	Complete circuit through MAX232A
1000	Test output to input of MAX232A
1030	Verify pin assignments of DE-9F
1100	Complete circuit through to computer
1130	Testing
1400	Write test results

28.4. July 13, 2006 – July 20, 2006

Recently Completed Tasks:
Completed Schematic

To be completed by next meeting:
Wiring Schematic
Communication over serial port
Testing
Output to file for use in interface

Upcoming Schedule

7/13/06
1300: Meeting
1500: Wiring Configuration
1600: Communication over serial port
1700: Test wiring configuration to output

7/14/06
1000: Design communication method
1100: Write section on communication
1300: Test communication from RFID card to computer
1500: Write section on testing

7/17/06
1000: Review overall section progress

29. Appendix I: Voice Authentication Progress Reports

29.1. June 2, 2006

Over the past week, sections on the development and testing of the voice authentication module have been written. During this week, sections will be written on system-server communication. The decisions about operating system and medium will be discussed. Upcoming, sections will be written on adaptations of the system from the selected CF RFID reader to the purchased USB RFID reader.

From the design standpoint, a physical interface using PHP or another graphical user interface will be designed. This will be used at the actual terminal that will prompt the user for their RFID card and voices.

29.2. June 5, 2006 – June 11, 2006

There were a lot of advancements that were made this week. The voice authentication algorithm went through a few changes. The changes to the program were made mainly to the peak_finder1 sub routine. Instead of taking just 10 frequencies and amplitudes, I decided to take 20 frequencies and amplitudes. By doing this I will be able to get more accurate results. All the frequencies are added together and compared to the frequencies of the original signal. The sums of both the old and new frequencies are divided by each other to get a certain percentage. The same concept was implemented to the amplitudes.

To get more consistent results I decided to check which distances the guards would be able to say their particular phrase. My results show that the guard should be no further than 10 inches away from the microphone. The guard should also make sure that he or she is speaking directly into the microphone. This is the only way to get consistent readings out of an individual.

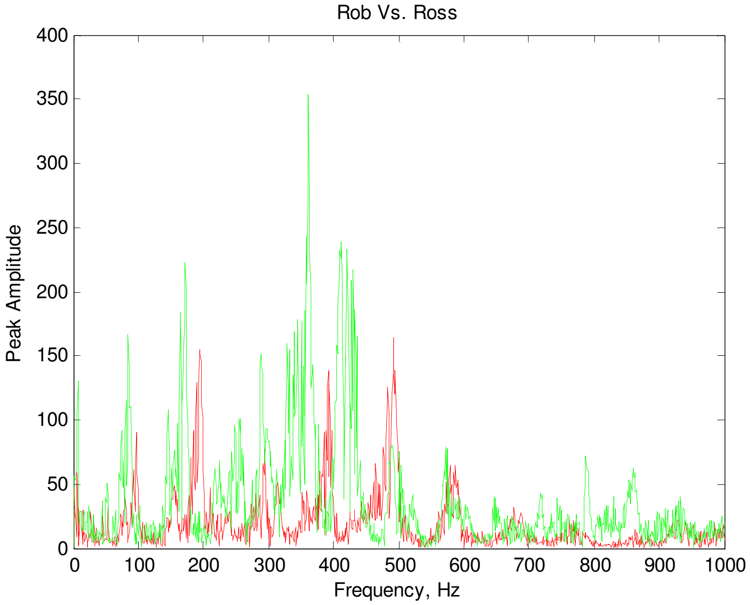
The frequencies emitted by a particular person seem to vary slightly from each recording, which exactly what we want. The problem that we are having is controlling the amplitudes. In some recordings, it is the amplitude alone that makes or breaks the voice authentication module. I am trying to find a way to control the amplitudes so that the readings become more consistent. Normalization of the signal is one way that I have achieved this. Once the signal is normalized the amplitude readings don't vary too greatly, but at times they can still cause problems.

For the rest of the week I plan to work on this problem, and also keep writing the final report. The final report will contain all the experiments, data, code and methodology used for the voice authentication system. Right now I consider my part to be a little behind schedule, but only by a few days. Writing the report while doing the project will save us time at the end. I also have attached and excel sheet of the experiments and the results that I have obtained from them.

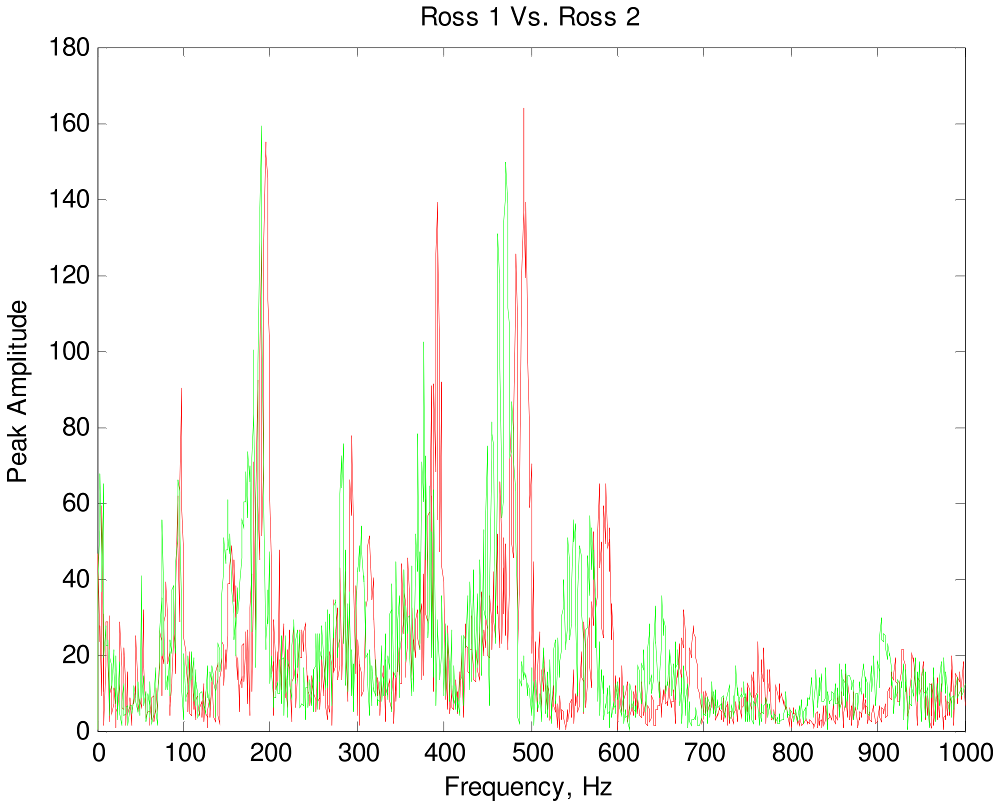
29.3. June 12, 2006 – June 18, 2006:

The frequencies work well when matching two of the same people, but it is very inconsistent when trying to match up two different people. I have been looking into another method of signal comparison using MATLAB. This problem will be worked on for most of the week.

When comparing the signals for two different people the results we get vary. The following is a plot of a speech signal from Ross compared to a speech from Rob.



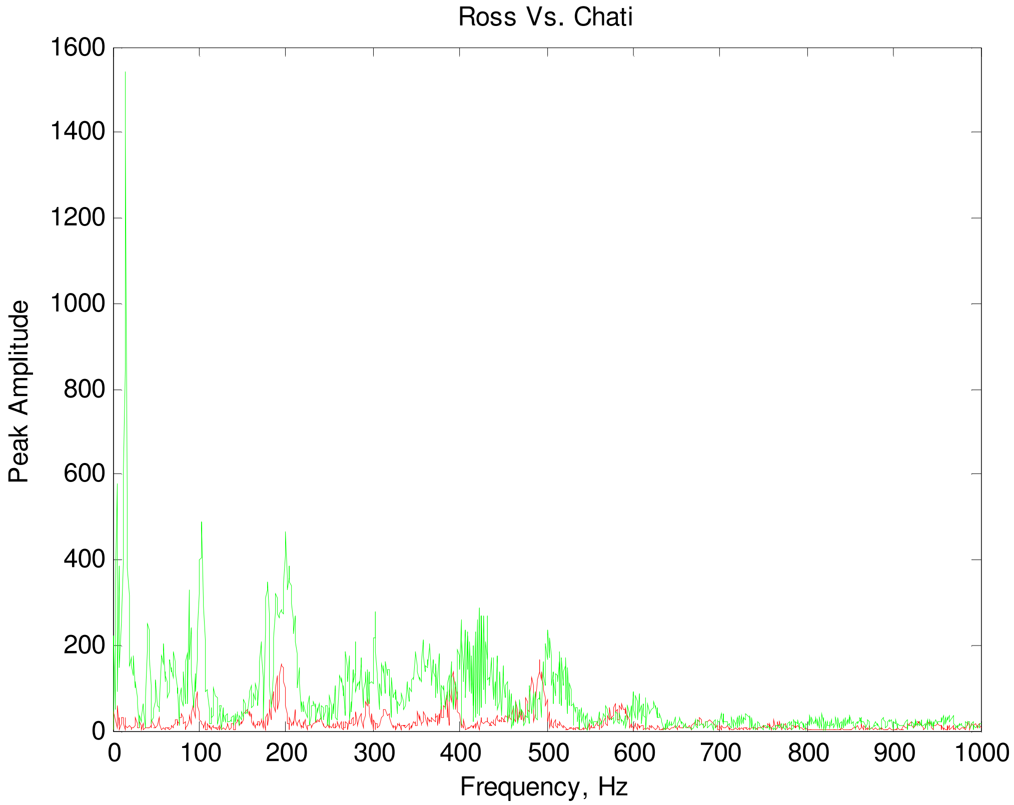
Ross is in red and Rob is in green. The following plot is when comparing Ross to himself:



As can be seen, these two signals are more similar than the Ross vs. Rob, but another method of comparison still needs to be found. We have thought about using correlation as another method of comparison and we currently trying to implement that into the MATLAB code.

By the end of the week more experiments will be performed, using female voice as well as male. If the system can differentiate between male and female, that will be a major success for now. More writing on the final paper will be done by the end of this week.

Other Comparisons:



37% Match Chati – green ; Ross - Red

29.4. June 19, 2006 – June 25, 2006

29.4.1. Experiment 3

Experiment number three was performed to check for any other weaknesses in the algorithm. All the subjects used in this experiment were male in their early twenties. There were quite a few weaknesses even though the second experiment had desirable results. Once big weakness that was present was that the frequencies alone were not enough to solely identity one person. The algorithm was taking one frequency for every 50 Hz of the speech signal. This was done because the function used to find the frequencies and peaks of the signal (find_peaks), was set to interpolate all of the frequencies found in a certain range of signal. So every 50 Hz of signal, the function would take about 25 frequencies and interpolate them to make them one frequency. By disabling the interpolation that the function was doing, the function was able to display all of the frequencies. This new method was applied to the previous methods of experimentation. The results produced a higher percentage of matching when the subjects were compared to themselves. All the results that were yielded format his experiment were in the 90 percentile range, with only three matches in the 80 percentile range. When comparing the signals visually through graphs and charts, most signals matched up with the "counterparts" and mismatched with the other speech signals.

Algorithm Changes III

The algorithm went through another set of changes once experiment three was completed. One of the major changes it underwent was the deletion of the interpolation function in the find_peaks subroutine. This allowed the subroutine to find more frequencies throughout the entire speech signal. When the frequencies were found and outputted they were all added to produce a sum of frequencies. This is what was done to both of the speech signals. This new method produced better results when a subject's voice was being compared to himself. Another subroutine has also been added to the algorithm; compare_plot. Compare_plot takes the fast Fourier transforms from the wav_filter subroutines and plots them against each other.

Experiment four was very similar to experiment three but covariance was added to the algorithm. By adding covariance to the signal comparison subroutine, the algorithm was partially able to start differentiating between two different subjects. Experiment four used the frequency comparison method used in the previous experiments, while also adding the signal covariance. The signal covariance subroutine took the variances of two different signals, and compared the variances to get another percent difference. The percent difference produced by the covariance is added to the percent match produced by the frequencies and then averaged. The algorithm then seems to get good results with subjects being compared to themselves or subjects being compared other subjects. There were two cutoffs that were implemented in this experiment, 75% and 85%. When the 75% cutoff was implemented, the algorithm passed two different speech signals inputted by two different subjects more often than when the 85% cutoff was implemented.

Phrase 3 - 'Sri lanka' Trial 1

	Chati	Rob	Ross	Mike
Chati	Pass	Pass	Fail	Fail
Rob	Pass	Pass	Fail	Fail
Ross	Fail	Fail	Pass	Fail
Mike	Fail	Fail	Fail	Pass

Table 16 - 75% and 85% Cutoff

As can be seen in Table 1, the algorithm still has a few glitches but for the most part it can differentiate between two voices. Though signal covariance might not be enough to fully implement the voice authentication system, it has shown a lot of promise and will be used along with the comparison of the frequencies.

While performing experiment four, other factors were also causing a few problems. By comparing only one or two words at a time, the system had a hard time differentiating between two subjects especially when they were saying the same phrase or two different phrases that were different but similar in sound. For example, the subjects were asked to input their cities of birth into the system, three of the subjects were born in cities that sounded very similar (Boston, Bolton, Brockton). When comparing the speech signals of these three cities, there was a lot of confusing present in the algorithm and the algorithm was unable to differentiate effectively; thus, passing most of the signals that sounded the same.

Phrase 4 - 'City Of Birth' Trial 1

	Chati	Rob	Ross	Mike
Chati	Pass	Fail	Fail	Fail
Rob	Fail	Pass	Pass	Pass
Ross	Fail	Pass	Pass	Pass
Mike	Fail	Pass	Pass	Pass

Table 17 - 75% Cutoff

As can be seen from Table 2, the algorithm passed all of the subjects whose cities of birth sounded very similar and failed the city of birth which obviously sounded different. The city of birth which failed with the other three cities was Rome. When comparing Rome to Boston, Bolton, and Brockton, the algorithm was able to recognize the difference of the word.

Algorithm Changes IV

A new subroutine was added to the algorithm after experiment four. This subroutines was called signal_cov, which stands for signal covariance. This subroutine takes the fast Fourier transform of the signals produced in wav_filter and wav_filter2, and takes the variances of each of the signals. The once the variances of each signal is found, it compares them and gets the percent difference of those two variances. This percent difference is then sent to the peak_compare0 subroutine which adds it on to the percent match of the frequencies and the takes the average of the two percentages. By using this method, the algorithm is starting to differentiate individual voices.

Milestones

Week 6/19-625
6/20 - Found a different way of comparing the signals along with the frequency comparison. Signal Covariance will also be used in the algorithm

6/22 –Might use signal correlation into algorithm to see if more accurate results can be produced.

Week 6/26 – 7/2

6/28 – Hoping to have the algorithm mostly functional, leaving room for a few additions or deletions

6/30 – Want to have a good portion of the voice authentication part of the module written for the final paper. So far 15 pages have been written on the topic.

So far, I have been on schedule, and seeing how work on the weekend goes, I might be ahead of schedule come Monday

29.5. June 26, 2006 – July 1, 2006

The voice authentication algorithm has finally started to work with consistent results. By comparing the inputted signal to two pre-recorded sound clips we can find an accurate representation of the person speaking into the security check-in station. When a guard speaks into the microphone the system will generate a percent match for both pre recorded signals. Once the percent matches are found, they will be averaged together. Once the percent matches are averaged together, the system will look at the following cutoffs:

- 79% - Below – Fail
- 80% - 84% - Pass to second trial
- 85% - Above – Definite Pass

After running about 250 trials or so there were only 4 trials that did not workout to our liking, proving that this system is not flawless but has a very good chance of authenticating the correct voice and failing unauthorized personnel. The table below are the typical Pass/Fail tests that were conducted, and as it can be seen all the fials and passes happen at the appropriate times

Phrase 1 - 'Yellow'

	Chati	Rob	Ross	Mike
Chati	Pass	Fail	Fail	Fail
Rob	Fail	Pass	Fail	Fail
Ross	Fail	Fail	Pass	Fail
Mike	Fail	Fail	Fail	Pass

To prove that this was not a one time occurrence, here are some more tables with the same results.

Phrase 3 - 'Sri Lanka'

	Chati	Rob	Ross	Mike
Chati	Pass	Fail	Fail	Fail
Rob	Fail	Pass	Fail	Fail
Ross	Fail	Fail	Pass	Fail
Mike	Fail	Fail	Fail	Fail

Milestones for this week:
6/26 – 6/27 Got the voice authentication portion functional
6/28 – Figure out a way to normalize in MATLAB and test it
7/2 – Work on final paper

Milestones Next Week:
7/3 – 7/5 Have a fully functional voice authentication program
7/5 – 7/12 Work on final paper

30. Appendix J: Client-Server Progress Reports