UNIVERSITY OF TARTU
FACULTY OF MATHEMATICS AND COMPUTER SCIENCE
Institute of Computer Science
Information Technology specialty

Uku Loskit

Adaptive Vocal Random Challenge Support for Biometric Authentication

Bachelor's thesis (6EAP)

Supervisor: Eero Vainikko, PhD
Co-supervisor: Jürmo Mehine, MSc

Author.. "...." May 2012
Supervisor... "...." May 2012
Co-supervisor.. "...." May 2012
Professor.. "...." May 2012

Tartu 2012

Table of Contents

Glossary

ASV – automated speaker verification

FAR – false acceptance rate

FRR – false rejection rate

EER – equal error rate

IVR – interactive voice response

PAD – playback attack detector

PIN – personal identification number

RA – replay attack

Introduction

In last 10 years[1] a great deal of interest has grown around using biometric solutions for identification and authentication. The usability of biometric solutions is on the rise with the ever-growing adoption of smart phones and cloud technologies which enable cost-effective solutions for example in the cases of face and speech recognition.

Biometry is a field that deals with identifying people based on their biological characteristics. These characteristics might include the person's voice, face, fingerprints, palm veins, but also the person's gait or writing style. Biometric authentication ascertains the person's identity by “who the person is”, as opposed to knowledge-based authentication which deals with identifying the person by “what the person has or knows”.

This thesis is written with the intent of being used in a proof of concept open source multi-modal biometric project for the Tartu University Biometry Group. The aim of the project is to create a biometric system that only relies on open source tools, but includes face recognition, speaker and voice recognition.

Ones of the motivations for using biometric authentication is that a correctly realized biometric solution might provide a more user-friendly and secure alternatives to knowledge-based solutions.

The main security problem for biometry are replay attacks – a class of attacks where the authentication process or authentication signal is replayed, thereby gaining access to the secured resource. The prevention of replay attacks is critical for assuring security for any biometric system.

This thesis focuses on a very specific way to protect a biometric system from replay attacks. A closer look will be taken at speaker-authentication systems where the user is authenticated based on the user's unique voice pattern. Naïve implementations of such a system would be vulnerable to replay attacks due to fact the user's speech could be easily recorded and replayed to

[1] For the latest developments in the field of biometric authentication see “Recent Applications in Biometrics” (2011) by Jucheng Yang and Norman Poh

the system thereby allowing access to unauthorized people, trivially breaking any sort of security that it would otherwise provide.

These aforementioned problems could be remedied by employing vocal random challenges. The idea behind vocal random challenges is that on every authentication attempt the user is prompted with a random challenge (or word or a sequence of words) that the user must then utter in order to gain access. Thusly the simpler forms of replay attacks can be prevented because the challenge will be unique for every authentication attempt and mere replays will not grant there attacker access.

One of the motivations for this thesis is that there has not been much academic publications about the implementation and security of vocal random challenges. In addition to this, there are many proprietary closed source applications, but no free open source ones available. Thus the goal was to a great a piece of software that could copy the functionality of that commercial products provide.

For the recognition of vocal random challenges, an open source PocksetSphinx[1][2] toolkit which was developed by Carnegie Mellon University is used. This piece of software will be compared to other free open source speech recognition solutions available. A brief overview of the available commercial closed source applications for speech recognition will also be given.

While biometric authentication in general can be applied to the same areas as knowledge-based authentication, we shall only consider the areas of interest where voice-based authentication is applicable due to cost-effectiveness and the nature of the business processes. Let us consider some of the areas where voice-based biometric authentication could be applied to see the motivations for employing voice-based biometric authentication. Voice Biometrics Group lists the following practical uses for voice-based authentication [3]:

- IVR[2]s and call centers – inbound calls could be handled and authenticated through the use of automated systems. These systems authenticate the user using either static text password or numeric pass phrases or free speech. For example, Kivox 4.0 by Agnitio provides this feature[4].
- Distance and online learning – educational institutions can verify unobtrusively that students that take courses and exams remotely, are who they claim to be.

[2] See the glossary for this and forthcoming abbreviations

- Password reset systems – banks, telecoms and other institutions can use automated systems that authenticate the users based on their voice in conjunction with a password phrase
- Multi-Factor web security – existing web-based solutions can supplement their existing knowledge-based solutions with voice-based biometric solutions.
- Parolee and offender monitoring – parolees and sex offenders can be monitored at their homes by random calls requiring them to utter a random word. A product called Shadowtrack provides this opportunity, for example. [5]
- Remote time and attendance – companies can use it to check in on their employees, so that they can see if they are working as required.
- Clinical trials and research – researchers in the fields of pharmacy and medicine are looking into implementing voice verification and identification systems to deter fraudulent report of results.
- Enterprise remote access – large corporations need to enable access for remote employees to the corporate the IT infrastructure.
- Forensic identification – forensic systems aim to identify who the speaker is, typically on a limited amount of sample data. This is particularly useful for military and intelligence communities, federal state and local governments. For example, S.P.I.D [6], software developed by Nuance provides this feature
- Smartphone security – smartphones can be used to make online payments or carry out other online transactions. VoicePay, a company based in Germany uses voice verification technologies by VoiceTrust [7]

To further exemplify the viability of voice-based biometric authentic, consider the following example where the cost-effectiveness of telephone-based password resets has been surveyed. The Gartner groups' research concluded that 80-90% of help-desk cost is composed of password resets for the users. A single call costs a company that uses help desk workers an estimated $30-$31. It should be fairly obvious that this enormous cost for the company could be greatly reduced by employing voice-based automated password resets instead to have them handled by human staff. The aforementioned Gartner survey claimed that a company that switched to the automated system using biometric authentication saved up to $600, 000 *per annum* [8]. This would mean that improving the cost-effectiveness through the use of voice-based authentication is definitely feasible and worth looking into.

In the first chapter of this thesis will give an overview of the current state of biometric authentication technology, and briefly summarize what the motivations, advantages and disadvantages of using biometric technology are.

The second chapter employs the concepts introduced in the first chapter for the voice-based biometric authentication. A demonstration of the main threat to voice-based biometric systems – replay attacks - is provided. The chapter concludes by weighing several options for dealing with this problem, and finally provides a solution that employs vocal random challenges.

Finally, in the third chapter, an overview of the greater biometric system in which the practical output of this thesis will be implemented will be given. For clarification a brief overview about how speech recognition works, is also given. This chapter also includes an assessment of the currently available open source speech recognition frameworks, and the rationale behind choosing Pocket Sphinx as the framework for the practical solution.

As a result of the thesis a working piece of software in the Python programming language will be produced using the aforementioned PocketSphinx voice recognition toolkit. More specifically two versions of the software will be produced:

1. A demonstration application with graphical user interface to demonstrate the possibilities of the authentication for hypothetical applications.
2. A command line utility that could be integrated into or invoked from any program to add speech recognition capabilities to it.

State of the Art

Publications

The existing literature on replay attacks on voice-based systems is quite sparse. Replay attacks are discussed in the literature, but most of them rely on building mathematical models for detecting impostures rather than employing random challenges. Despite this, it is useful to list these static pass phrase approaches discussed in the academia, because they might explain the rationale behind preferring them over random challenge-based methods.

Malik (2011) claims that while there has been a lot of research into countering attacks against synthesized voice, not a lot of research has been conducted on replay attacks. In the same paper Malik proposes a mathematical approach for modeling replay attacks. Malik claims that employing higher-order spectral analysis can be used to capture traces of nonlinearities in the cloned or replayed speech of the targeted speaker. Malik proposes a scale invariant moments based detection framework to detect cloned audio recording using replay attacks.[9]

Shang (2008) proposes a playback attack detector (PAD) for voice-based authentication systems to counter replay attacks. While this thesis concerns itself with non-static random passwords and pass phrases, then Shang discusses using predetermined, fixed passwords. Shang rationalizes this choice by referring to several weaknesses in the non-static random challenge scenarios. Shang's PAD approach relies on the random nature of human speech: if any two utterances of the pass phrase are deemed to be too similar to each other by the PAD system, the access to the system will not be granted.[10]

Genoud and Chollet (1999) demonstrate how an automatic speaker verification (ASV) system can be vulnerable to speech concatenation attacks. In a concatenation attack it is assumed that the attacker has managed to get a hold of the uttered pass phrase and some other sentences uttered by the person. After this, the attacker proceeds by splitting the recorded message into words that can then be concatenated to form any combination of the required random challenges. [11]

Other researchers have abandoned the idea of solely relying on voice or any single biometric characteristic altogether. Rather, they rely on the fusion of multiple biometric characteristics. Several researchers have approached the problem of biometric authentication with the novel approach of using face recognition in conjunction with voice recognition. They have employed the concept of *liveness*. The user is asked to blink or smile to prove in real-time that the authenticator is real alive person, instead of a prerecorded video presented by an impostor. [12]

Commercial solutions

Although there is not too much academic literature written on employing random pass phrases to secure voice-based biometrics as described in previous section, they are already being used by many of the commercial solutions. Some commercial solutions will be listed here, but this listing should by no means be considered conclusive.

RVA-Authenticate

This is a product by the Canadian company Perceive Solutions. It claims to authenticate the user based on whether the correct phrase was utter, and whether the phrase was uttered by the authorized person says. [13]

PERCEIVE claims that its software can be integrated into:

- IVR systems to provide automated and secure authentication using randomly generated pass phrases that meet the customer's specifications
- internet websites
- custom applications

ComBiom

ComBiom is product by a Swiss company Biometry AG. Their product uses a combination of face, lip movement, voice recognition, and also employs vocal random challenges. During enrollment the user is prompted to utter the digits from 0 to 9 [14]. ComBiom as a product is especially important for this thesis, because creating an application with the capabilities comparable to that of ComBiom was the main objective of the biometry project briefly discussed in the introduction of this thesis.

Chapter 1: Introduction to biometric authentication

1.1 The Concept of Biometric Authentication

1.1.1 Knowledge-based Authentication

Let us start by clarifying what exactly biometric authentication methods are. Currently most of the systems in practice employ authentication techniques that do not use biometric challenges, but are **knowledge-based**. Knowledge-based authentication refers to methods that require the person to prove to the system that they are who they present themselves as by using pre-shared information. This might appear in the form of a password, a PIN code, a secret question or the like.[15]

1.1.2 Biometric Characteristics

Biometric authentication methods conversely, rely on biological measurements for authentication. The user is authenticated not by “what the person has”, but rather “who he is”.[15]

It should be noted that not all biological measurements are suitable for authentication, however. In their article Jain *et al* (2004) point out four criteria for biometric characteristics. Any human physiological and/or behavioral characteristic can be used as biometric characteristic as long as it satisfies the following criteria[16]:

1. **universality** - each person should have the characteristic
2. **distinctiveness** - any two person should persons should be sufficiently different in terms of the characteristic
3. **permanence** - the characteristic should be sufficiently invariant
4. **collectibility** - the characteristic can be measured quantitatively

In addition an additional non-functional criterion is mentioned:

5. **user-friendliness or non-obtrusiveness** - the characteristic can be measured in a manner that is acceptable by the user. To further elaborate this criterion we can compare two modes of proposed biometric authentication: fingerprint scanning and DNA-collection. It is clear that

using a fingerprint scanner is far less obtrusive and faster than DNA-collection. DNA-collection is not cost-effective, cannot be easily automated, and furthermore, obtrusive modes of the collection of biological material in a very explicit manner can harm the public reception of biometric technology, as people become concerned for their privacy. [16]

1.2 How Biometric Authentication Works

Let's start out with defining the basic components and procedures of a biometric system. From these basic building blocks systems of any level of complexity can be created.

1.2.1 Components

1. *Sample* - A biometric measure presented by the user and captured by the data collection subsystem as an image or signal. [17]
2. *Feature* - A mathematical representation of the information extracted from the presented sample by feature extraction / selection. *[17]*
3. *Template* - A user's stored reference measure based on features extracted from enrollment samples. [17]

1.2.2 Enrollment and Recognition

A biometric system has of two main procedures - *enrollment* - the initial act in which an agent presents her biometric data to the system and enables access to a certain resource via this form of authentication, and *identification (recognition)* - the act in which an agent presents her biometric data to system to be checked an existing template to gain access to the aforementioned resource. Figure 1 depicts the general the process of enrollment and recognition for biometric authentication systems respectively. [16]

We can think of enrollment and identification as high level concepts that are presented to the end-user. This, however, says little about the inner workings of a biometric system. These processes will be now be further elaborated in the next section.

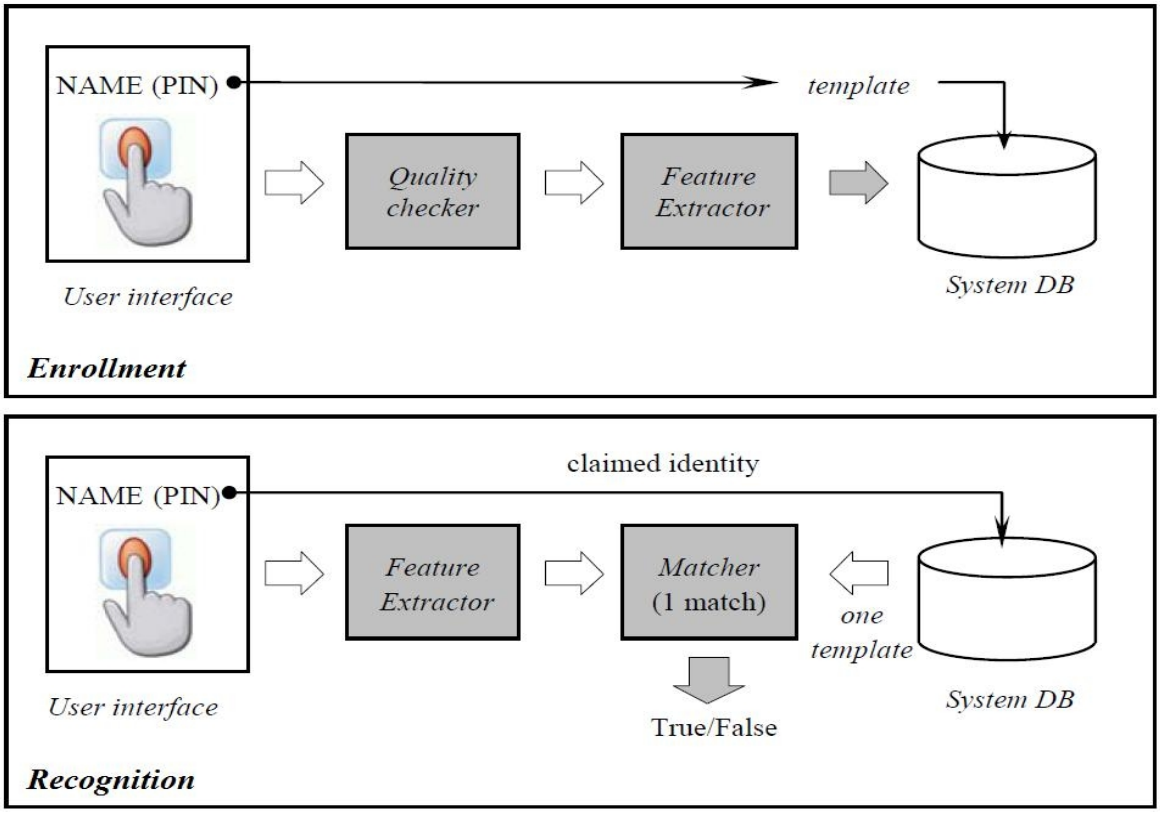


Figure 1: General schematic of biometric enrollment and recognition [16]

1.2.3 The Steps of Enrollment and Identification

Acquisition is the process through which biometric data is gathered from the user. This is one of critical parts of the system, because it usually determines the quality of the biometric system as a whole. If the acquired data is of poor quality, the whole system suffers. [18]

The first acquisition phase usually consists of two parts:

1. *quality assessment* - the quality of the acquired data is automatically measured, if the system employs a minimum quality threshold, a low quality sample maybe rejected, and a new sample maybe required of the user.
2. *segmentation* - the irrelevant data is separated from the relevant information.

The second phase is called *representation* - the data acquired in the first phase must be manipulated to be represented in a digital form. [18]

The third step is *feature extraction. Feature extraction* is the process by which key features of the sample are selected or enhanced. Typically, the process of feature extraction relies on a set of algorithms; the method varies depending on the type of biometric identification used. [18]

The fourth step is *matching* - in this step the extracted features are compared to the existing ones in a feature database. [18]

The last step is the *decision*. Based on the matching accuracy the system must make the decision of whether to accept or deny the user access. [18]

1.3 Advantages of biometric authentication

Knowledge- and key-based authentication are well-spread, but with the relevant security they provide, they have their fair share of problems as well. Reid [16] and Tsai *et al* [19] and offer a conclusive list of problems for knowledge-based authentication, which we will not further discuss here, but will instead focus on the advantages of biometric authentication.

Biometric authentication is arguably a lot more natural for human beings than knowledge-based authentication, as we use it in our every day lives all the time. In addition to this, biometric data is very difficult to forge.[20]. The signal of biometric characteristic can range in size from several hundred bytes to even megabytes. This is provides significantly more entropy than a usual password, and thus is unfeasible to brute force using current methods. Thus biometric solutions provide the speed of entering small insecure passwords (the validation time) with the security of having very long passwords. [21]

One of the advantages of biometric data is that it cannot be exchanged between people as in the case of knowledge-based solutions. A typical problem with password-based authentication schemes is that people readily exchange their passwords or write them down. Ironically the likelihood that the password will be written down increases with the increase in the password's entropy This problem could be avoided in the case of biometric solutions where this sort of exchange or external storage would be impossible.[18]

Another case could easily be made in favor of biometric solutions. Most password-choice strategies advise having both unique passwords and difficult to guess/brute-force for every separate resource (such as authenticating on a specific website). A modern user may thus need an infeasible number of unique passwords, very fast leading to password re-use or to the writing down the passwords as described above. Although this case maybe alleviated by the introduction of third party authentication services, we place an additional risk in trusting the third-party. In contrast, in a biometric system we do not need to take any extra measures or precautions, nor do we need to introduce a third party for the authentication process for such cases as described above because biometric data is difficult to forge, as described above. [18]

It has been suggested that due to the fact that password- or key-based authentication technologies and methods provide us with only a crisp binary output of either yes/no (was the user authenticated or not), could be seen as a negative aspect. A decision in a biometric system is always probabilistic, thus thresholds for match rates can be established to create systems that provide ways to establish user non-repudiation (that is that there is no possibility for the user to invalidate that he performed such and such actions). [22]

1.4 Problems with biometric authentication

Bruce Schneier (1994) discusses several problems with biometric authentication. Although biometric data is difficult to forge, it is easy to steal. An attacker may steal our biometric data, because we leave biologic material everywhere (our fingerprints are can be recovered everywhere, our face can be seen and voice heard in the public). Thus, according to Schneier, a method of biometric authentication is only secure as long as two conditions hold:

1. the data used for any given authentication attempt was generated at the time of authentication, and
2. that it can be matched with a master copy that is already present in the database. [20]

Schneier also points out that biometric data, once stolen, cannot be restored to its former secure state. In the cases of token- and knowledge-based authentication our certificates may be revoked or reinstated if the need be, but once biometric data has been compromised, it can never be securely reused again. For example, if one fears that someone has gotten a hold of one's password, one can always change it, but one cannot change his biometric characteristics such as voice or face (at least in a non-trivial manner). [20] Recently, however, several researches have advocated the use of cancelable biometrics, so this issue might be alleviated. [23]

Furthermore, biometric authentication methods do not conform with the principle that every object should have a unique access key. For knowledge- and token-based authentication methods password or key reuse is considered dangerous, biometric authentication cannot avoid key reuse due to its nature. [20]

We lose the ability to have different levels of security levels: it could prove a great security risk if a highly critical system used the exactly same authentication as one's home front door. In the case of knowledge- and token-based systems this problem can easily be alleviated by using different keys with different key lengths varying on the criticalness of the given resource. [20]

Recall section 1.1.2 where the concept of universality was introduced. It becomes apparent that for any given system there will always exist a subset of persons who will not be able to provide the biometric sample, undermining the overall usability and applicability of the biometric authentication solutions. [18]

In addition to the problems listed by Schneier, we have to take into consideration the probabilistic nature of biometrics. When biological data are introduced to the biometric sensor, and extracted in order to be compared to an existing template on the system, the extraction process is always lossy. Thus, we always end up with incomplete information that is further distorted by environmental noise. So the biometric system must make probabilistic judgment whether the person is who she claims to be. This introduces the possibility of both false positives (a great security risk) and false negatives (a great problem for the usability of biometrics). The next section will discuss precisely these probabilistic measurements that are essentially characteristic of biometric systems. [18]

1.5 Measuring the efficacy of biometric systems

Having introduced the general advantages and disadvantages of biometric authentication, it would seem reasonable to develop some sort of metrics that are specific to biometric authentication. Let us consider the problem of measuring the security, scalability and usability of a biometric systems. In order to have objective criteria for such characterizations, we must first define some mathematical measures.

1.5.1 FAR (False Acceptance Rate)

FAR is defined as the probability that a user making a false claim about his/her identity will be verified as that false identity [16]. FAR can be calculated as follows:

The probability that a fraudulent attempt is successful against a enrolled person n:

$$FAR(n)=\frac{Number\ of\ successful\ independent\ fraud\ attempts\ against\ a\ person\ n}{Number\ of\ all\ independent\ fraud\ attempts\ against\ a\ person\ n}$$ [24]

The overall FAR for N persons is defined as:

$$FAR=\sum_{n=1}^{N} FAR(n)$$ [24]

FAR as statistical measure actually measures the effectiveness of the underlying biometric algorithm. If the algorithm is ineffective and accepts too many fraudulent users it is indicated by a high percentage of FAR. [16]

1.5.2 FRR (False Rejection Rate)

The FRR is defined as the probability that a user making a true claim about his/her identity will be rejected as him/herself. [16]

The probability that a non-fraudulent person n is not successful on identification FRR(n):

$$FRR(n)=\frac{Number\ of\ rejected\ verification\ attempts\ for\ a\ qualified\ person\ n}{Number\ of\ all\ verification\ attempts\ for\ a\ qualified\ person\ n}$$ [24]

Similarly to to general FAR, the general FAR for N people can be calculated as follows:

$$FRR=\sum_{n=1}^{N} FRR(n)$$ [24]

FRR as a statistical measure indicates the robustness of the system. If the FRR is too high, in the best case scenario only the usability of the biometric system might slightly suffer, but on the other end of the spectrum, if the false rejections are too common, the system becomes unusable and inaccessible. [15]

Both FAR and FRR can be seen as measures of scalability. With the increased number of users the likelihood rises that the characteristics of any two persons are indistinguishable by the given biometric algorithm or undetectable by the biometric sensor.[15][24]

1.5.3 EER (Equal Error Rate)

Equal error rate (EER), often also referred to as cross-over error rate (CER) takes both of the two previously defined statistical measures of FAR and FRR into account. The interdependence of the two statistical measures is self-evident: for example, if we increase the sensitivity of the matching algorithm (require a tighter match), we may achieve a lower FAR, whereas the increased sensitivity might cause the ERR to rise, as more and more users are then falsely rejected. Conversely, when we “desensitize” the algorithm the opposite effect is produced: an increased FAR and a decreased EER. Due to this matter of fact, it makes sense to plot the ERR and FAR of a given system together. EER is thus a statistical measure that is indicative of the accuracy of the biometric system. [19]

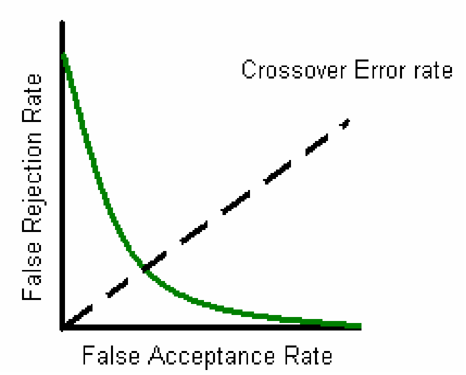


Figure 2: EER (CER) as a relation between FAR and FRR [25]

FAR, FRR and EER are not the only statistical measures, but they are arguably the more important ones. For the sake of brevity these other statistical measures will not be discussed in this thesis. [15][24]

Biometric	Cross-over accuracy
Retinal Scan	.000001%
Iris Scan	.000763%
Fingerprints	.2%
Hand geometry	.2%
Signature Dynamics	2%
Voice Dynamics	2%

Table 1: The cross-over rates of various biometrics by Liu and Silverman [25]

Chapter 2: Voice-based Biometrics

An brief overview of the current state of biometric technology, its advantages and disadvantages in comparison to knowledge-based authentication general was given in chapter 1. Let us now narrow our focus to voice-based authentication.

A **voice-based biometric system** is a biometric system that employs speaker verification or speech recognition or the conjunction of the two to authorize a user.

2.1 Evaluating Voice-based Biometric Authentication

Biometric characteristic	Universality	Distinctiveness	Permanence	Collectability	Performance	Acceptability
DNA	H	H	H	L	H	L
Ear	M	M	H	M	M	H
Face	H	L	M	H	L	H
Facial thermogram	H	H	L	H	M	H
Fingerprint	M	H	H	M	H	M
Gait	M	L	L	H	L	H
Hand geometry	M	M	M	H	M	M
Hand vein	M	M	M	M	M	M
Iris	H	H	H	M	H	L
Keystroke	L	L	L	M	L	M
Odor	H	H	H	L	L	M
Palmprint	M	H	H	M	H	M
Retina	H	H	M	L	H	L

Signature	L	L	L	H	L	H
Voice	M	L	L	M	L	H

Table 2: Comparison of various biometric technologies by Jain et al (2004) [16]

L – low, M -medium, H- high

Let us first evaluate the viability of voice-based biometrics based on some of the criteria introduced in the first chapter.

Reid (2004) grades voice-based biometric authentication on the scale of 0-10 (from bad to excellent). FAR for voice-based systems receives a rating of 6, and a FRR score of 6. These are quite low when compared to face recognition's score of 7.5 for both FAR and FRR. Reid argues that FAR and FRR score low (meaning that the FAR and FRR percentages are high) due to the noise in background, and points (similarly to Jain *et al.*) to the matter of fact that voice (its behavioral characteristic) may vary greatly due to the emotional state of the person. [15]

According to the table 1 in section 1.5.3 the EER is for one of the highest and is matched only by the EER of signature dynamics. This means that voice as a biometric is not very accurate when compared to other biometrics in general.

Recall the 5 criteria discussed in the grading is on the scale of 0 to 10 and is based on Reid (2004) From table 2 we can see that the strong points for voice recognition is that is universally available and easily collectible, while its not very distinctive, nor permanent. Although in the case of permanence Jain *et al* argue that the question is two-fold: while a person's voice is quite permanent (as a biological characteristic), voice also has a behavioral characteristic which might differ greatly due to emotions or illness. They add that despite voice-based authentication's shortcomings it's highly acceptable and non-obtrusive for the users, and irreplaceable for some situations like telephony-based applications. [15]

2.2 A Naïve Solution

Let us at first consider a very simplistic approach for using voice-based biometric authentication to demonstrate its weakness. A simplistic biometric system voiced-based authentication is seen in figure 3. This system uses a predefined PIN to authenticate its users. In step 1 the authorized user Bob can be seen uttering the PIN code. The system matches Bob's input to the enrolled version of him saying the very same password. However, without Bob's knowing his utterance has been recorded by a malicious user, Malroy. Now, in step 2, Malroy plays the very same utterance from a recording device to the system. Systems that rely on this kind of a naïve solution will accept the recording as genuine as long as the recording is of decent quality.[10] This is a well-known attack vector for biometric authentication which will be discussed in the next section.

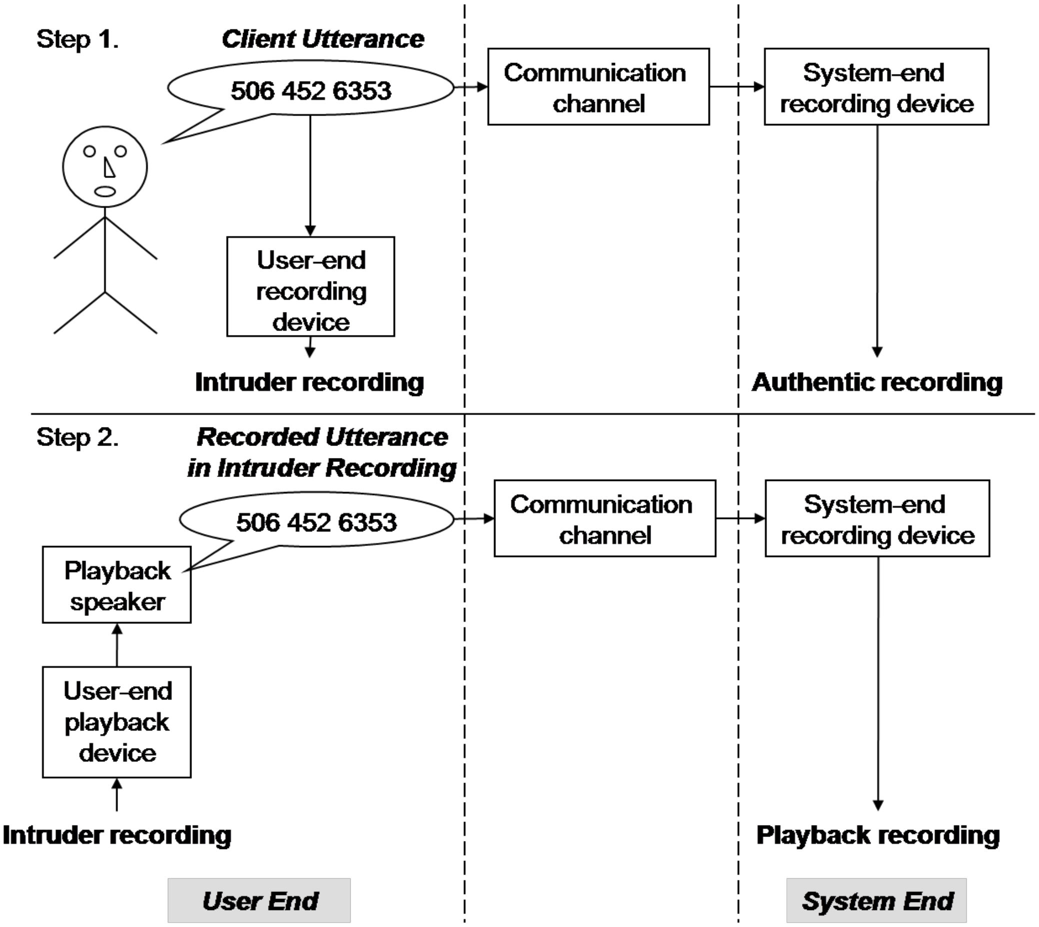


Figure 3: Execution of a playback (replay) attack introduced by Shang (2008) [10]

2.3 Replay attacks

Biometric systems are not completely invulnerable. A list of known attack vectors exists which target all of the major components of a biometric system. A conclusive listing and description of these known vulnerabilities is outside the scope of this thesis, but figure 4 illustrates which parts of a generic biometric system could possibly be attacked.

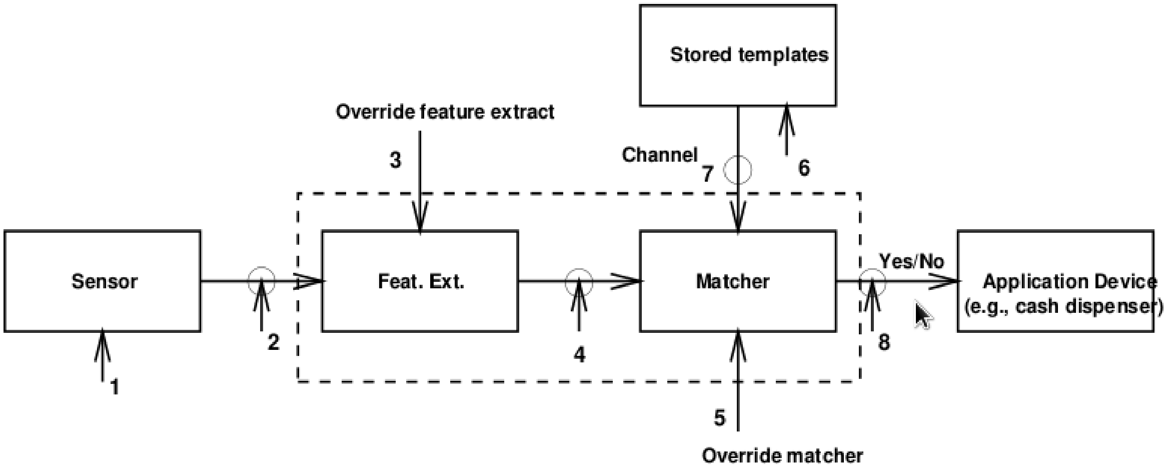


Figure 4: The possible attack points of a biometric system [21]

Recall the last section where the idea of a naïve implementation which was vulnerable to replaying a recorded utterance of the PIN was discussed. This form of an attack is known as a replay attack. In general, a **replay attack,** sometimes also referred to as a “playback attack”, is a form of network attack in which a valid data transmission is maliciously or fraudulently repeated or delayed. Thus, biometric authentication in a very general sense is also vulnerable to this attack vector [21].

In a typical biometric system replay attacks could occur between attack points point No. 1 either point No. 2, as depicted in figure 4. Point No. 2 stands for the communication between the sensor and the feature extraction device, whereas point No. 1 stands for the communication with the biometric sensor. The line of communication for point No. 2 is usually encrypted, thus providing relevant security between the sensor and the feature extraction device [21]. According to the definition given above both of these are forms of replay attacks, but let us differentiate between these two by means of two definitions.

Let us define a **on-the-wire replay attack** as tampering with the signal between the sensor and feature extraction device in attack point no. 2, in contrast to a **simple replay attack**, which relies on replaying a recorded signal to the sensor. This thesis will only concentrate on the latter as it requires a lot less technical expertise to carry out, and is thus privy to even non-technical attackers. Thus, in the next and forthcoming sections references to replay attacks refer only to simple replay attacks as defined above.

2.4 Considering Possible Solutions

Recall the literature discussed in the beginning of this thesis. Three general solutions were hinted at. These solutions can be explicated as follows:

1) similarity based solutions – these include PAD-like implementations like those posed by Shang (2008) which try to find the similarities or nonlinearities Malik (2011) in the signal to make sure it is not duplicated by replays.
2) liveness testing – a solution discussed by Toth (2005) where multi-modal (solutions that employ several biometrics in conjunction) systems that rely on challenging the users with facial expressions (the users are asked to smile or blink) to ascertain that a real *alive* person is there instead of a video feed.
3) vocal random challenge based solutions – the solution here is to prompt the user with a challenge of random word or a sequence of words. The randomness of the challenge would ensure that the replay attack could not occur.

This thesis maintains that the most simple and cost-effective solution would be to employ vocal random challenges. The rationale is the following: while signal comparison based solutions require either complicated mathematical frameworks or extensive training for attacker modeling, and the liveness based approaches require extra equipment which not be applicable for telephony-based cases, vocal random challenge based systems would be a lot easier and cheaper to implement. In addition, the development of these systems would require experts in the field, and as of yet no such open source frameworks exist. Due to the issues just mentioned, and due to the fact that relying solely on open source tools was one of the requirements for the university project, these other possible solutions will now be set aside.

2.5 Vocal Random Challenges

In the last section the rationale behind choosing vocal random challenges was discussed. Vocal random challenges can be of the following type:

1) single digits like, for example *ONE*
2) sequences of single digits, for example: *ONE-TWO-THREE*
3) numbers, for example *ONE THOUSAND AND ONE*
4) single words, for example *HORSE*
5) sequences of words (sentences), for example: *LIVE LONG AND PROSPER*

Even intuitively it would seem that accuracy of the systems which rely on a small amount of phrases would be greater, the reason for this will become apparent once an overview of the inner-workings speech recognition is given in chapter 3. On the other hand though, the systems that would rely on simpler methods, would be more vulnerable to concatenation attacks as described by Genoud and Chollet (1999), because the for example the digits from 1 to 9 could very easily be extracted from everyday speech and then concatenated.

2.6 Components of Vocal Random-challenge Based System

The system must known which words are being uttered and thus implement some form of a a **speech recognition expert.** But the system must also know that speaker is who he claims to be, so it must additionally implement a **speaker verification expert** as well. The result from both of these experts will be fused together in a decision module which then take the decision whether or not to authorize the person based on its configuration. The implementation of such a system will be discussed in the next chapter.

Chapter 3: The Practical Solution

3.1 Overview of the Biometric System

The concept of vocal random challenges was introduced in chapter 2. Figure 5 depicts the overall structure where the result of this practical thesis will be deployed. The system is composed of three main parts: the speech recognition expert, the speaker verification expert, and the decision module as seen in figure 4. This greater system is based on the commercial product that was introduced in the state of the art secton of this thesis - ComBiom.

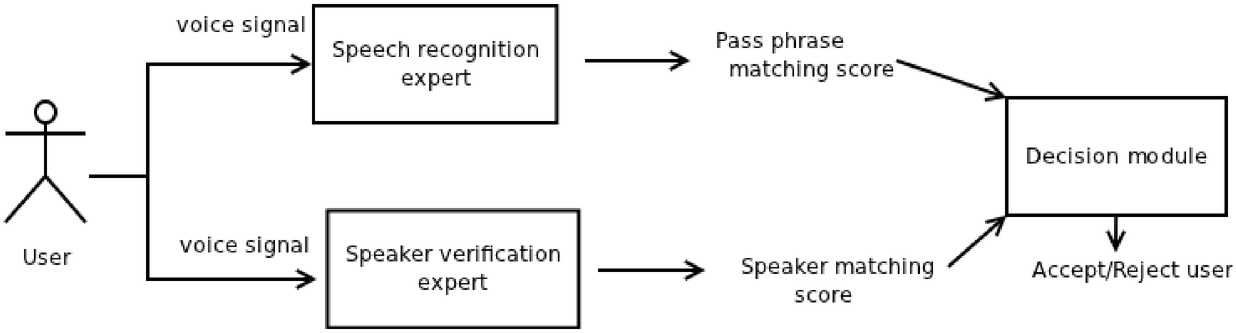


Figure 5: The structure of the complete biometric system

The **speech recognition expert** gives a scoring on how well the word was matched the challenge word. The Speaker verification gives a scoring on how certain it is that speaker is. The **decision module** makes the decision whether or not authorize the user on the predefined threshold value taking into consideration both of scores from two speech experts.

However, the scope of this thesis only limited to the speech recognition expert, so from hereon the speaker verification part will not be discussed further, the speaker verification expert will be considered to be a black box that the internals of which we are not concerned with. We just assume that it carries out its function and provides the decision module a scoring of how likely it is that the person is who he/she claims to be.

3.2 Tool Selection Process

A number of currently available voice recognition frameworks/technologies were considered for the practical output of this thesis, and as a result CMU Sphinx voice recognition toolkit was chosen. To understand the rationale behind the choice we must first establish what kind of criteria and characteristics were required of the proposed tool. The requirements were not arbitrary, but were imposed on me by the requirements of the aforementioned multi-modal biometric solution for Biometry Group of Tartu University. The list of requirements was the following:

1. Open source code - the framework must include its source code in case further improvements or development is necessary.
2. Free software - for our purposes the framework must be free of charge.
3. Permissive license - a permissive license is required if a commercial solution is ever created from the prototype. Due to this reason, all non-permissive and copy-alike licenses such as GPL would have to be discarded.
4. Optimized for mobile devices - the memory and CPU requirements for the chosen solution must be minimal to accommodate speech recognition on mobile devices.
5. An ideal solution would provide support for both Windows and Unix-like operating systems

Framework	**Free software**	**Permissive license**	**Programming language API/bindings**	**Operating systems**	**English Acoustic model available**	**Optimized for mobile devices**
CMU Sphinx	Yes	Yes (BSD)	Java, Python, C	Unix-like, Windows	Yes	Yes
Julius	Yes	Yes (BSD)	C, Windows SAPI	Unix-like, Windows	Yes	Yes

Simon	Yes	No (GPL)	N/A	Linux	No	No

Table 3 – a table of currently available open source voice recognition frameworks [1][26][27][28][29]

As can be seen from the information represent in table 3, only two out of the three proposed frameworks are suitable. Simon was discarded due to its unsuitable licensing, GUI only interface, and the non-existence of previously generated acoustic models for the English language. While Simon could prove to be an appropriate choice for other use cases, it does definitely fit with the 5 criteria listed above.

The final choice between Julius and CMU Sphinx was decided by the number of language bindings available, the size of the active developer community, and due to the fact that there was more documentation available for CMU Sphinx. Another deciding factor was that English acoustic models for Julius were not included with Julius itself, but were created by the VoxForge project as a joint community effort, whereas CMU Sphinx came with highly-trained acoustic models for English included. While the Julius speech decoder has a permissive licence, the VoxForge acoustic models cannot be used in any commercial solutions. [30]

CMU Sphinx has several different versions [2]. Again, due to the requirements imposed by the biometry project, a minimal version of the speech recognition framework - PocketSphinx - was chosen. PocketSphinx is written entirely in the C programming language with bindings for popular interpreted languages like Python[2]. This joins the ease of development with the best possible performance.

3.3 Brief Introduction to Speech Recognition

In order to show how the discussed tools for speech recognition work, an overview must given on how speech recognition works in general. This, however, requires that we first acquaint ourselves with several terms, definitions and concepts specific to this field. The overview of the theory behind speech recognition will be avoided, and will be represented in a very general manner due to its complexity and due to the fact that a thorough insight itself would amount to a bachelor's or master's thesis on its own.

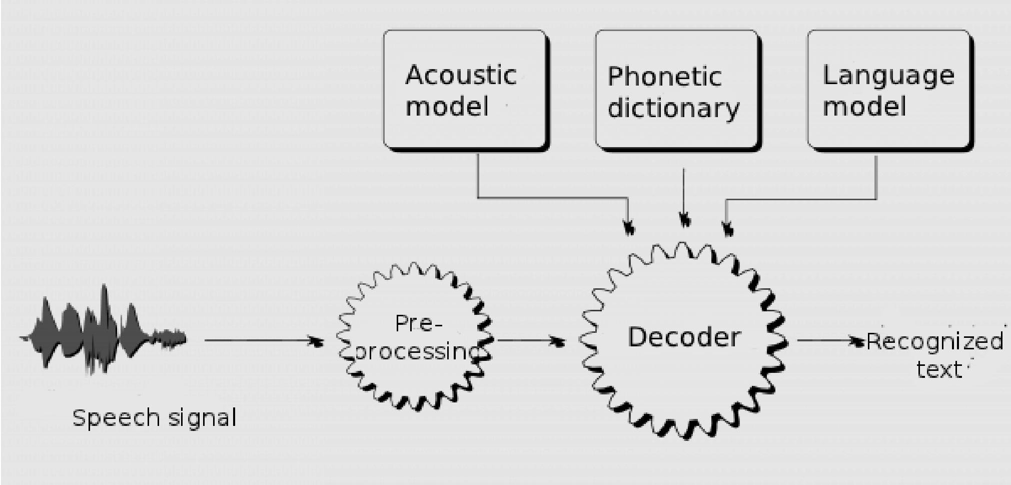


Figure 6 - A high level overview of the speech recognition process and its main constituents. Translated and modified image from Tallinn University of Technology materials [31].

First of all it should be noted that speech is a non-discrete phenomenon, meaning that is does not compose of discrete units that could be easily separated as spoken words, thus speech certainly cannot be regarded as "acoustic text", as it is often erroneously though possible. Due to this, the analog speech signal itself cannot directly be used for speech recognition. In order to use the signal for recognition, certain features must first be extracted from it. [31]

The first step of the recognition process is **pre-processing** (after the analog waveform as been digitalized), as depicted in Figure 1. During pre-processing speech is split into very small time units called **frames** (often as short as 10 milliseconds). For each such frame we can calculate a **feature vector**, a set of numbers representing the speech frame along with about 30 coefficients. The rationale behind using feature vectors is two-fold: to decrease the amount of information, and to accentuate those features that differ the most between different phones. The idea is that features should enable to differentiate between different phones, and at the same time disregard irrelevant aspects such as environmental and background noise, microphone idiosyncrasies, and the emotions of the speaker. The resulting output of the preprocessing step is a sequence of feature vectors, which represent the speech signal in much more compact manner than the initial digitalized waveform. For example, in the case of 16 kHz frequency and 16 bit depth digitalized speech signal with a duration of a second the size amounts to 32000 bytes per second, whereas its sequence of feature vectors may only amount to a mere 5200 bytes (in case of a 13-dimensional real number feature vector). [31]

After the pre-processing step has generated the sequence of feature vectors, the sequences are fed to the **speech decoder**. This brings us arguably the most important component of a speech recognition engine - the acoustic model. Acoustic models are used to model phones. The phonetic dictionary includes all of the known pronunciations of the applications. The idea behind acoustic models is that they determine how similar arbitrary feature vectors are to the speech signal, in other words, how probable is that the given speech signal is the speech signal we have described. Acoustic models rely on the use hidden Markov models (HMMs) [32]

Due to the fact that natural speech is too imprecise as people tend not to utter the words in their full form, leaving the end of words unsaid and thus multiple words overlapping, and when nothing about the language at hand is known we have no way of delimiting words in a sequence of phones. For example, Consider the phone sequence *thequickbrownfoxjumpsoverthelazydog*. This could result in different combinations of phones for words if we do not know which word the language contains. Thus, we need to have a representation of the language. The solution here is to employ **language models**. In very simplistic terms, language models can be considered to be a list of valid words for a given language. [31]

3.4 Practical Solution with PocketSphinx

3.4.1 The Components of PocketSphinx

In the last chapter the concepts of phonetic dictionaries, acoustic models. language models and speech decoders were introduced. These are also the very same constituents that make up PocketSphinx. By default two acoustic models are available: TIDIGITS[33] and WSJ1[34]. TIDIGITS is an acoustic model that is highly optimized for only digit recognition. Conversely the WSJ1 – Wall Street Journal – which was created to match a wide variety of words from natural speech and trained using dictations of hypothetical news reports by journalists [34]. The language model consists only of 11 words for TIDIGITS, but 6627 words for WSJ1[33][34].

3.4.2 Requirements for the Environment

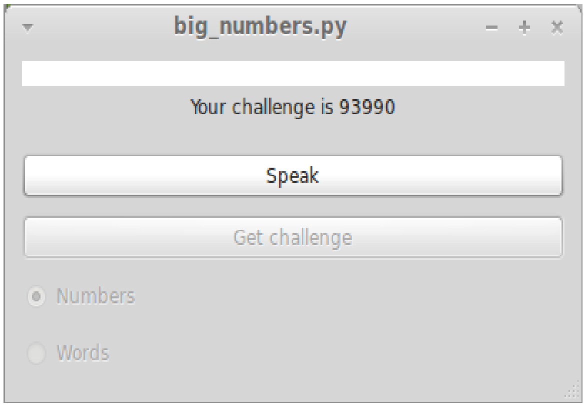
1. Unix-like operating system. Preferably a Debian-based distribution (Ubuntu), so that the packages for PocketSphinx are available on the repository and would not have to be manually compiled
2. Python v2.65 or greater installed
3. FFmpeg installed [35]
4. GStreamer bindings [36]

These dependencies should all be resolved by running the *installation.sh* file located in the root folder of this project.

3.4.3 Graphical User Interface

The application uses a mock door a to demonstrate the communication procedure with an outside device. The application uses the telnet protocol to send messages to and receive messages from the door. The door is just used as an analogy: the door could be replaced by any resource that requires authorization. according to. The lights on the door indicate whether the door is unlocked or locked.

The program can be started by running the shell script *start.sh* . This starts the GUI along with the mock door. The user is prompted with two regular buttons and a two radio buttons. The radio buttons stand for whether the challenges are words or numbers respectively. If the use clicks the *'Get challenge'* button, a vocal random challenge will be generated based on the previous position of the radio button (figure 8). In case “Words” was selected the challenges appear as pictures instead, as seen in figure 9. The user can then press *Speak* and utter the challenge. The program will automatically detect when the user has stopped talking and respond by evaluating the uttered phrase and comparing it to the random challenge. Once this process is completed, the program will acknowledge the user by a dialog whether the input was correct or not (figures 10, and 11 respectively), and open the door (figure 12) or do nothing (figure 7) .

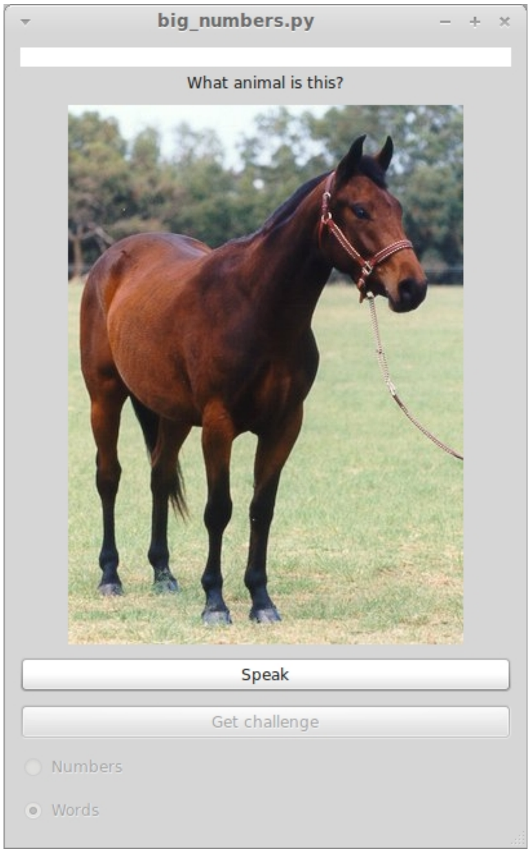


Your challenge is 93990

● Numbers

○ Words

Figure 8 – the user must utter 93990

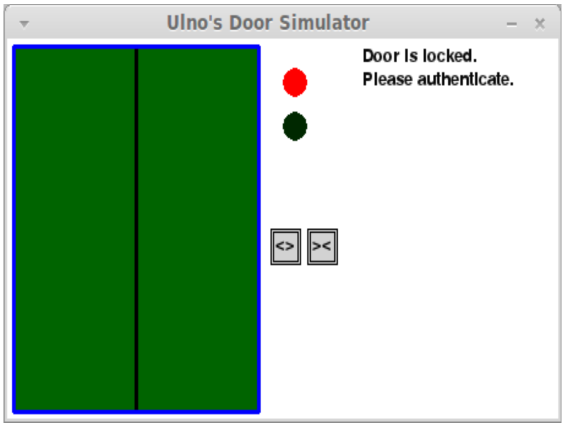


What animal is this?

○ Numbers

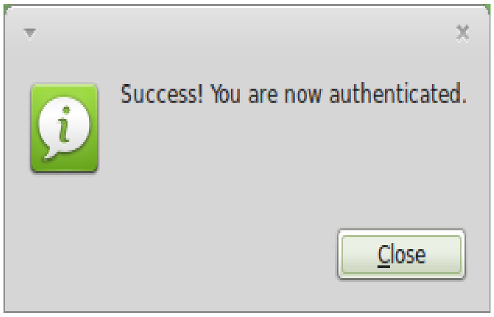
● Words

Figure 9: Example of a word-based challenge. The user must utter "horse" after pressing the speak button



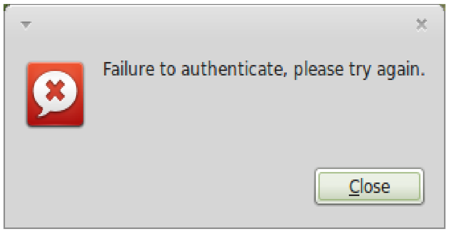
Door is locked.
Please authenticate.

Figure 7: A Closed mock door that illustrates a resource that needs authorization.



Success! You are now authenticated.

Figure 10: The user is displayed that the door is unlocked



Failure to authenticate, please try again.

Figure 11: The user is displayed an error message on failure

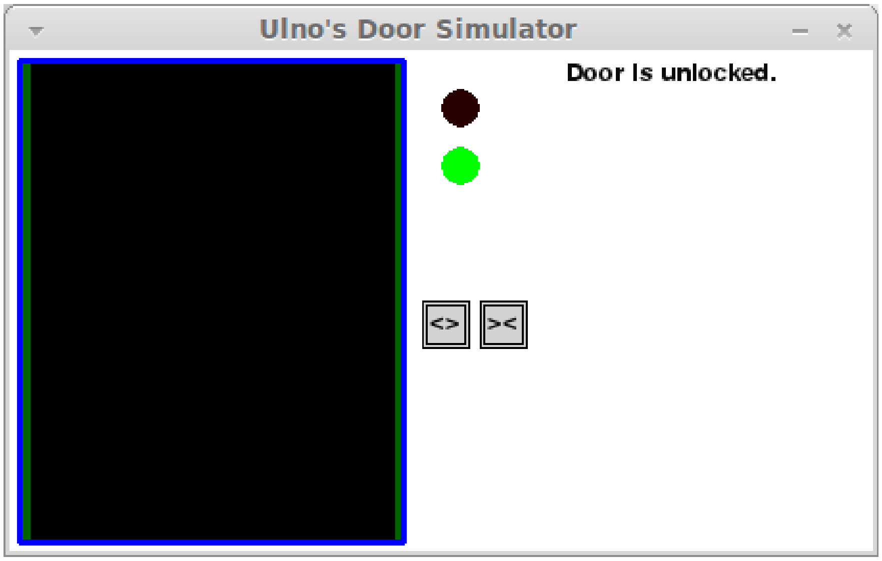


Figure 12: The door is opened on correctly guessing the word

3.4.4 Command Line Interface

While the GUI interface along with the door could be useful in some application, it would be more sensible to use it as a subprogram that is invoked from other programs. For this reason a command line version of the verification tool is included. The GUI version supported continuous speech by and the GStreamer plug-in took care of all the audio conversion. However, as most of the audio would be coming from different sources (for example, from a mobile device sent to a server) the audio must now be saved into wave form with the proper format. PocketSphinx uses 16 kHz frequency and a single audio channel. For this purpose, a conversion script (*convert.sh*) is included which uses FFMpeg to convert the audio into the proper format. *convert.sh* should used before running *check.sh*, if unsure about the audio format.

The validation program can be invoked using a shell script *check.sh.* The program will output the percentage of match to STDOUT (*standard output*).

The syntax for check.sh

./check.sh [filename.wav] [challenge to check against]

Example usage of *check.sh*

```
./check.sh onetwothree.wav 123
```

The syntax for convert.sh

./convert.sh [input-file] [output-file]

Example usage of *convert.sh*

./convert.sh onetwothree.mp3 onetwothree.wav

3.4.5 Configuration

Configuration is currently available for the demo application, for the command line application it must be changed from the source code.

The configuration is done by manually editing the configuration file *CONFIG* with a text-editor. Setting-variables are presented in uppercase letters, e.g "TIDIGITS_HMMDIR", and delimited from values by a ":" symbol:

VARIABLE: value

The following configuration settings are available:

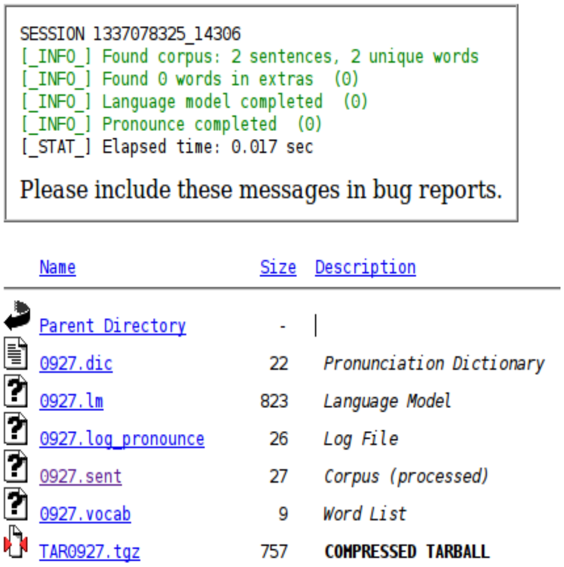
1. *TIDIGITS_HMMDIR* – specifies the directory where the acoustic model for the TIDIGITs is stored.
2. *WSJ_HMMDIR* – specifies the directory where the acoustic for the Wall Street Journal is located
3. *TIDIGITS_LANGUAGE_MODEL* – specifies the language model file for TIDIGITS
4. *TIDIGITS_DICTIONARY* – specifies the dictionary file for TIDIGITS
5. *WSJ_LANGUANGE_MODEL* – specifies the language model file for the Wall Street Journal model
6. *WSJ_DICTIONARY* – specifies the phonetic dictionary file the WSJ model

7. *NUMBER SEQUENCE_LENGTH* – specifies how long the random number are

Note that all the paths must be relative to the folder that *start.sh* is located in. Also note that the images for each word must be located in the *images* directory and the words must themselves be stored in the *vocab* file in the root folder.

3.4.4 Custom Language Model Generation

One can also define a custom dictionary of words which one might wish to recognize, the only limitation here being that the acoustic model has to be suitably trained. Luckily for the most common words you can use the WSJ1 (Wall Street Journal) which in great likelihood can be used to recognize the words.



```
SESSION 1337078325_14306
[_INFO_] Found corpus: 2 sentences, 2 unique words
[_INFO_] Found 0 words in extras  (0)
[_INFO_] Language model completed  (0)
[_INFO_] Pronounce completed  (0)
[_STAT_] Elapsed time: 0.017 sec
```

Please include these messages in bug reports.

Name	Size	Description
Parent Directory	-	
0927.dic	22	Pronunciation Dictionary
0927.lm	823	Language Model
0927.log_pronounce	26	Log File
0927.sent	27	Corpus (processed)
0927.vocab	9	Word List
TAR0927.tgz	757	COMPRESSED TARBALL

Figure 13: The list of resulting files from the online lmtool

To create one's own language model and phonetic dictionary, one must use the Sphinx online language modeling tool also known as lmtool [37]. The list of valid words (or sentences) is referred to as the **corpus** [37]. The corpus must be represented in file in which each word or sentence is delimited by a new line character. The site provides an upload form where one can upload this text-based corpus and it will generate the appropriate language model and phonetic dictionary. The site will then prompt the user with the files which can be downloaded either as a tarball or as single files, as demonstrated in figure 13. To use these custom dictionary and language model files, one must set them in the up in the CONFIG file as WSJ_DICTIONARY and WSJ_LANGUAGE_MODEL respectively. To add pictures, insert them in the images folder as *word_to_be_recognized.jpg*. The vocabulary file from the website (0927.vocab depicted in figure 13) should be replace the *vocab* file in the root directory.

Possible Future Work

Acoustic Model Accuracy Training

While the generic acoustic models are suitable for different English words and sequences of digits, they fall short when the words are very similar to each other. An attempt was made to accommodate an opportunity for the user to insert the sequences of digits as numbers instead. Thus, for the the challenge posed in section 3.4.3 in figure 5, the random challenge of 9713 could be uttered as “NINE-SEVEN-ONE-THREE” or “NINE THOUSAND SEVEN HUNDRED AND THIRTEEN”, both being equally valid. However, when attempting this, the Wall Street Journal acoustic model had to be used instead of TIDIGITS because TIDIGITS does not include the anything but the digits from 0 to 9. The result was that the Sphinx recognition could not tell apart between numbers such as seven**ty** and seven**teen**.

The author maintains that the security of the vocal random challenges for this system could be increased by varying whether the user has to present “9713” the sequence of digits or as a number. This could be improved my training the acoustic models

Offline Language Model Generation

Currently a big inconvenience for using the practical solution as a library is that the language models must be generated online using the Sphinx Knowledge Base Tool. A big improvement could be made to the system in terms of usability if the process of modifying the list of valid words could be achieved in a single step. This feature was not added to the practical solution due to time limitations.

Testing

For implementing the practical solution in a production environment it would be necessary to measure the actual FRR, FAR and EER of the solution. But testing at this point would not yield much information, as testing for the acoustic models has already been done by their acoustic model developers. To gather new valuable insight the testing would also have to be carried out in conjunction with the speaker verification expert, but this would time-consuming goal would have well been outside the scope of this thesis.

Conclusions

The goal for thesis was to create a working speech recognition software for a open source biometric authentication system. This goal has been fulfilled and the software is operational. PocketSphinx, an open source speech recognition tool developed at the Carnegie Mellon University, was used for this purpose. The practical outputs of the thesis are a demo application with graphical user interface which demonstrates the capabilities of PocketSphinx, and command line application that can be added to existing systems to provide a speech recognition capability.

While the current version of the software is fully functional, it has not been thoroughly tested, and generating language models is not very convenient for the user at this moment. Also the number recognition part is not yet implemented which could dramatically increase the overall security of the vocal random challenges. These three points should not be seen as deficiencies, but as goals for future work to be done on the software which were not achievable in the scope of this thesis.

Adaptiivne kõnepõhine juhuväljakutsete tugi biomeetrilisele autentimisele

Bakalaureusetöö (6 EAP)
Uku Loskit

Käesoleva bakalaureusetöö eesmärgiks oli arendada välja kõnetuvastusprogramm, mida saaks kasutada vokaalsete juhuväljakutse tarvis. Programmi eesmärgiks oli anda üks võimalik lahendus kõnepõhilise biomeetrilise autentimise kesksele turvaprobleemile – taasesitusrünnetele. Programm põhineb vabavaralisel PocketSphinxi kõnetuvastuse tööriistal ning on kirjutatud Pythoni programmeerimiskeeles.

Loodud rakendus koosneb kahest osast: kasutajaliidesega varustatud demonstratsiooniprogrammist ja käsurea utiilidist. Kasutajaliidesega rakendus sobib kõnetuvastusteegi võimete demonstreerimiseks, käsurea utiliiti saab aga kasutada mis tahes teisele programmile kõnetuvastusvõimekuse lisamiseks.

Kasutajaliidesega rakenduses saab kasutaja oma hääle abil programmiga vahetult suheldes avada näitlikustamiseks loodud demoprogrammi ust. Kasutaja peab ütlema õige numbrite jada või pildile vastava sõna inglise keeles, et programmi poolt autoriseeritud saada.

Mõlemat loodud rakendust saab seadistada luues oma keelemudeleid või muutes demorakenduse puhul numbriliste juhuväljakutsete pikkust.

Bibliography

[1] *CMU Sphinx homepage on SourceForge* http://sourceforge.net/projects/cmusphinx [Cited: May 15, 2012]

[2] *Versions of Decoders – CMUSphinx* http://cmusphinx.sourceforge.net/wiki/versions [Cited: May 15, 2012]

[3] *Voice Biometry Group - Practical Uses* http://www.voicebiogroup.com/uses.html [Cited: May 15, 2012]

[4] *Agnitio – Voice Biometrics (Kivox 4.0)* http://www.agnitio.es/producto.php?id_producto=1 [Cited: May 15, 2012]

[5] *House Arrest, Home Incarceration Alternative | ShadowTrack* http://www.shadowtrack.com/ [Cited: May 15, 2012]

[6] *Nuance – S.P.I.D* http://www.nuance.com/for-business/by-solution/customer-service-solutions/solutions-services/inbound-solutions/voice-authentication-biometrics/spid/index.htm [Cited: May 15, 2012]

[7] *Voiceplay homepage* http://www.voice-pay.com/index.php [Cited: May 15, 2012]

[8] R J. Witty, K. Brittain. *Automated Password Resets Can Cut IT Service Desk Costs*, Gartner, 13 December 2004

[9] H. Malik. *Securing Speaker Verification System Against Replay Attack*

[10] W. Shang. *A Playback Attack Detector for Speaker Verification Systems* ISCCSP, Malta, March 2008.

[11] D. Genoud., G. Chollet. *Deliberate Imposture: A Challenge for Automatic Speaker Verification Systems.* Eurospeech'99, Budapest, Hungary, pp 1971-1074. September 5-9 1999

[12] B. Toth, Biometric liveness detection" Information Security Bulletin, vol.

10, pp. 291-297, Oct. 2005.

[13] *About Perceive Solutions Inc.* http://perceivesolutions.com/rav_authenticate.php [Cited: May 15, 2012]

[14] *Biometry.com AG – ComBiom* http://www.biometry.com/combiom.html [Cited: May 15, 2012]

[15] P. Reid 2004 *Biometrics for Network Security* Prentice Hall Professional

[16] A. K. Jain, A. Ross, S. Prabhakar. *An Introduction to Biometric Recognition* IEEE Transactions on Circuits and Systems for Video Technology, vol. 14, no. 1, January 2004

[17] *ECE1517 - Biometric Signals and Systems* lecture slides by Kostas Plataniotis (2009) http://www.ipsi.utoronto.ca/docs/ECE1517-Intro-Lecture.pdf [Cited: May 15, 2012]

[18] V. Matyàs and Z. Rìha. *Biometric authentication - Security and usability* in Proc. 6th IFIP TC6/TC11 Conf. Commun. Multimedia Security pp. 227–239. 2002

[19] C. Tsai, C. Lee, and M. Hwang. *Password authentication Schemes: Current Status and Key Issues.* International Journal of Network Security*;*3(2)*:* pp.101–15. 2006

[20] B. Schneier. *The uses and abuses of biometrics*, Comm. ACM, vol. 42, no. 8, pp. 136, Aug. 1999.

[21] N. K. Ratha, J. H. Connell, and R M. Bolle. *An Analysis of Minutiae Matching Strength.*

Lecture Notes in Computer Science*, v*ol. 2091*, pp.* 223-228. 2001

[22] Bolle, Ruud. *Guide to Biometrics. Springer, 2004.*

[23] *Cancelable Biometrics* by Jin *et al* *http://www.scholarpedia.org/article/Cancelable_biometrics [Cited: May 15, 2012]*

[24] *Biometrics FAQ* by Dr. Manfred Bromba http://www.bromba.com/faq/biofaqe.htm [Cited: May 15, 2012]

[25] *An Exploration of Voice Biometrics* (2004) by Lisa Myers http://www.sans.org/reading_room/whitepapers/authentication/exploration-voice-biometrics_1436 [Cited May 15, 2012]

[26] *Sphinx 4 license*

http://cmusphinx.sourceforge.net/sphinx4/license.terms [Cited: May 15, 2012]

[27] *PocketSphinx license* https://cmusphinx.svn.sourceforge.net/svnroot/cmusphinx/branches/pocketsphinx-0.6/pocketsphinx/COPYING [Cited: May 15, 2012]

[28] *Terms and conditions of license of Julius* http://julius.sourceforge.jp/LICENSE.txt [Cited: May 15, 2012]

[28] *Simon on SourceForge* http://sourceforge.net/projects/speech2text/ [Cited: May 15, 2012]

[29] *Simon documentation* http://simon.gibolles.com/doc/0.3/simon/en/index.pdf [Cited: May 15, 2012]

[30] *Voxforge About* http://www.voxforge.org/home/about [Cited: May 15, 2012]

[31] *Kõnetuvastus [Foneetika ja kõnetehnoloogia laboratoorium]* http://www.phon.ioc.ee/dokuwiki/doku.php?id=konetuvastus.et [Cited: May 15, 2012]

[32] *Hidden Markov model* http://en.wikipedia.org/wiki/Hidden_Markov_model [Cited: May 15, 2012]

[33] *TIDIGITS < Sphinx 4 < TWiki* http://www.speech.cs.cmu.edu/sphinx/twiki/bin/view/Sphinx4/TIDIGITS [Cited: May 15, 2012]

[34] *Wallstreet Journal < Sphinx 4 < TWiki* http://www.speech.cs.cmu.edu/sphinx/twiki/bin/view/Sphinx4/WallStreetJournal [Cited: May 15, 2012]

[35] *FFMpeg* http://www.fmpeg.com [Cited: May 15, 2012]

[36] *Using PocketSphinx with GStreamer and Python* http://cmusphinx.sourceforge.net/wiki/gstreamer [Cited: May 15, 2012]

[37] *Sphinx Knowledge Base Tool Version 3* by Alex Rudnicky http://www.speech.cs.cmu.edu/tools/lmtool-new.html [Cited May 15, 2012]

[38] *Training Acoustic Model for CMU Sphinx* http://cmusphinx.sourceforge.net/wiki/tutorialam [Cited May 15, 2012]

Appendix

The source code of the practical solution, the required language and acoustic models, and phonetic dictionaries that were produced as a result of this thesis are included on a DVD that is attached to the back cover of this thesis.