Security in Voice Authentication

by

Chenguang Yang

A Dissertation

Submitted to the Faculty

of the

WORCESTER POLYTECHNIC INSTITUTE

In partial fulfillment of the requirements for the

Degree of Doctor of Philosophy

in

Electrical and Computer Engineering

by

March 2014

APPROVED:

Professor Berk Sunar Dissertation Advisor ECE Department	Professor Thomas Eisenbarth Dissertation Committee ECE Department
Professor Krishna Kumar Venkatasubramanian Dissertation Committee Computer Science Department	Professor Yehia Massoud Head of Department ECE Department

Abstract

We evaluate the security of human voice password databases from an information theoretical point of view. More specifically, we provide a theoretical estimation on the amount of entropy in human voice when processed using the conventional GMM-UBM technologies and the MFCCs as the acoustic features. The theoretical estimation gives rise to a methodology for analyzing the security level in a corpus of human voice. That is, given a database containing speech signals, we provide a method for estimating the relative entropy (Kullback-Leibler divergence) of the database thereby establishing the security level of the speaker verification system. To demonstrate this, we analyze the YOHO database, a corpus of voice samples collected from 138 speakers and show that the amount of entropy extracted is less than 14-bits. We also present a practical attack that succeeds in impersonating the voice of any speaker within the corpus with a 98% success probability with as little as 9 trials. The attack will still succeed with a rate of 62.50% if 4 attempts are permitted. Further, based on the same attack rationale, we mount an attack on the ALIZE speaker verification system. We show through experimentation that the attacker can impersonate any user in the database of 69 people with about 25% success rate with only 5 trials. The success rate can achieve more than 50% by increasing the allowed authentication attempts to 20. Finally, when the practical attack is cast in terms of an entropy metric, we find that the theoretical entropy estimate almost perfectly predicts the success rate of the practical attack, giving further credence to the theoretical model and the associated entropy estimation technique.

Acknowledgements

I would like to express my gratitude to my advisor Professor Berk Sunar who has accepted me as his Ph.D student, who has been guiding me to the research area with his knownledge and patience. I also need to sncerely thank to Dr. Ghaith Hammouri who acts as my elder brother in the research lab who help me a lot in the research. I would like to thank to the committee member of my research: Professor Eisenbarth Thomas, Professor Krishna Kumar Venkatasubramanian who give me valuable assistance on my dissertation.

I also would like to give special thank to my friends in the lab: Michael Moukarzel, Yin Hu, Yarkn Doroz, Kahraman Daglar Akdemir, Deniz Karakoyunlu, Xin Ye and Wei Dai who has accompanied with me for these years and who make my life colorful.

I would like to reserve the my gratitude in the last to my family. I would like to show my gratitude to my gradmother Deying Jiang who brougt me up. I would like to show my gratitude to my father Yulu Yang and mother Xinmin Li who acted a my role model and has been always backing me up. I would like to show my gratitude to my aunt Luping Yang who gives me love like parents. I would like to thank all my family member. In the last, this dissertation is for my beloved girl friend Shiwei Wang.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

In the last decade, we have witnessed large scale adoption of biometric technologies, e.g. fingerprint scanners on laptops, cameras with built-in face recognition capabilities at airport terminals and stadiums, and voice based authentication technologies for account access on smartphones. Among biometric authentication technologies, voice based authentication is playing a pivotal role due to the exponential growth in the smartphone user base [1] and due to the unparalleled convenience it offers. Indeed, human voice can be easily captured over large distances simply over a standard phone line without requiring any special reader device. Furthermore, compared to other biometric schemes voice authentication offers the user a greater degree of freedom during signal acquisition.

Voice verification comes in two flavors: text dependent and text independent. Text independent voice verification, i.e. speaker verification, is not concerned with the text that is spoken. In contrast, in text dependent systems, the verification requires a match on the spoken text as well as a match on the user. With rapid developments in mobile computing and voice recognition technologies, it is convenient to use voice verification in the service of biometric authentication. Typically, in

commercial speaker verification systems, speech recognition is applied before speaker verification to prevent playback attacks. The user is asked to recite a randomly generated pass-phrase, and only if what the user says matches the pass-phrase, the system proceeds to the text-independent voice verification step.

Given the usability and ease of deployment, a number of companies are now offering voice based authentication services: PerSay's VocalPassword and FreeSpeech, Agnitio's Kivox and VoiceVault's VoiceSign, VoiceAuth products, etc. Unfortunately, the precise details of the extraction techniques used are not made public. We can only speculate on connections to academic work developed in the last decade. During the late 1990s to early 2000s, researchers in [2, 3] introduced the Gaussian Mixture Model (GMM) and the adapted GMM (GMM-UBM) to the speaker verification task. Later in [4, 5], the authors propose a new classifier by combining a support vector machine with a Gaussian Mixture Model verifier. In the meantime, researchers in [6, 7, 8] developed Joint Factor Analysis (JFA), a modification of GMM in order to tackle channel variability. Today, the state-of-the-art I-Vector method [9] is a derivative of JFA.

With all this deployment of voice authentication technologies, it becomes crucial to evaluate voice authentication technologies from a security point of view. Specifically, we are interested in text independent voice authentication. We want to explore whether or not human voice allows us to build a voice based authentication scheme whose security matches that of a cryptographic authentication scheme.

Studies in [10, 11, 12, 13, 14] indicate in general that the security level of biometric features themselves may be low. A number of studies in [15, 16, 17, 18] show that the speaker verification systems are vulnerable to synthesized speech attacks. All these attacks assume the attacker has acquired speech samples from target victims. The attacker builds synthesized speech through some transformation method based

on the parameters trained from the speech sample of the victim. To counter such attacks, authors of [19, 20] developed an anti-spoofing system targeting the synthesized speech attack. However, researchers seldom examine the speaker verification system from the information security point of view. In this work we take an initial step in that direction.

1.1 Motivation

To examine the security level of text-independent speaker verification systems, the performance of the current systems need to be reviewed. Two basic parameters are introduced in terms of evaluating the performance of a speaker verification system: false negative rate and false positive rate. These two rates represent the probability of a valid user being rejected by the verification system and probability of a fake user being accepted by the verification system, respectively. For a specific system one can always decrease one of these two rates by increasing the other one. The overall performance can be evaluated by a single parameter called equal error rate (EER). The EER is reached when the false negative rate equals false positive rate. A lower EER, in general, is the attribute of a better speaker verification system. Based on the summary of the performance of current text-independent speaker verification systems, EER is ranging from 1% to 10% [6, 7, 3]. The 1% false positive rate means that out of every 100 attempts to carry out voice authentication by random people one person will succeed. In other words, a person's account will have a 1% probability of being compromised by a random attacker. Since 1% implies $-\log_2(\frac{1}{100}) = 6.6 \approx 7$ bits, this corresponds to the security offered by an encryption algorithm with a key size of about 7 bits. One way to increase the security level would be to decrease the false positive rate. However, decreasing the false positive

rate would correspond to a higher false negative rate. Even if we tolerate the very high noise level that comes with decreasing the false positive rate by a factor of 100 the increase in security would be effectively doubling the key length. That is, going from around 7 bits to about 14 bits. This logic can be seen as a strong indicator of the limited amount of entropy that can be extracted from human voice.

The importance of establishing an entropy estimation is the real measure of identification capacity. To elaborate, in any voice identification system, the person's voice features will eventually be converted into a model or a key or any other mathematical representation. The form of which must be known publicly according to Kerckhoffs' basic security principles. This representation of the speaker's voice features will only be as unpredictable to an attacker as allowed by the entropy of the voice features. Therefore, if the entropy is low an attacker will simply test every single instant of the voice representation (which will be small due to the low entropy) until the attacker finally succeeds in impersonating that speaker. Security cannot be guaranteed by a complicated extraction process, but instead has to be rooted in entropy. Establishing the entropy content of a human voice password database in a way that is meaningful for security applications is the goal of this dissertation. Thus far – to the best of our knowledge – no rigorous study of the extractable entropy from human voice has appeared in the literature.

1.2 Dissertation Outline

In this dissertation, we show from both theoretical and practical views that the entropy of the Voice Password Databases is limited. We estimate the security level of text-independent speaker verification systems presenting a full fledged information theoretical analysis. The remainder of this dissertation is organized as follows.

In Chapter 2, we overview the related background knowledge of text-independent voice authentication/verification technologies. We start by reviewing how a typical voice authentication system works. Next, we explain feature extraction and feature modeling techniques. We focus on the fundamental text-independent voice verification techniques, i.e. we review Mel-Frequency Cepstral Coefficients (MFCC) and Gaussian Mixture Model (GMM).

In Chapter 3 we explain how we estimate the entropy of a human voice password database. We start by introducing the concept of entropy in security and enumerating different entropy metrics. Next, we explore the possibility of applying the differential entropy metric to a human voice password database and point out the deficiency of using this metric. Eventually, we propose and show by experiments that relative entropy is an appropriate measurement of uncertainty of the voice password database. We finally relate the theoretical relative entropy estimation to the attack introduced in detail in Chapter 4.

In Chapter 4, we propose an attack on a simulated voice authentication system. We begin with describing the MFCC-GMM based voice authentication system that we are attacking. We then explain the attack rationale and the detail of the attack. Finally, we show the experimental results. The successful attack in this chapter is a support of our theoretical entropy estimation in Chapter 3.

In Chapter 5 we take a further step to attack a third-party open-source text-independent voice authentication system. We first introduce the target system ALIZE. Then we explain the attack rationale. After reviewing why the attack method in Chapter 4 cannot apply directly to the ALIZE system, we propose an effective method of building mock speech signals. Next, we apply the proposed attack method to the YOHO database and show the experiments. Finally, we relate this attack to the previous attack in Chapter 4.

In Chapter 6 we relate our work to the state-of-the-art speaker verification systems. We explain that although we demonstrate our work using the classical GMM-UBM speaker verification system, the limited entropy which enables our attack on GMM-UBM is an intrinsic property of the speech signal still present even if JFA or an I-Vector based system is employed. Meanwhile, we propose possible utilization of the voice passwords.

Finally, in Chapter 7 we conclude the dissertation.

Chapter 2

Background

In this chapter, we briefly review existing voice authentication technologies. In Section 2.2 we explain short term cepstral analysis and two representative cepstral features: MFCC and LPCC. In Section 2.3, we introduce Gaussian Mixture Model, the feature modeling methods targeting text-independent voice authentication tasks.

2.1 Introduction

Speaker verification systems work in two phases: enrollment and verification [21]. During enrollment, a speaker is asked to contribute speech samples whose features are then extracted as shown in Figure 2.1. The speech features are then used to develop the users' speech models. The speech model is stored for future comparison. At a later time, when verification is required, see Figure 2.2, fresh samples are collected from the user. After similar extraction phases, the resulting extracted features are compared against the model stored during enrollment.

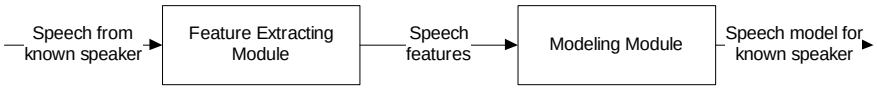

Figure 2.1: Speaker enrollment.

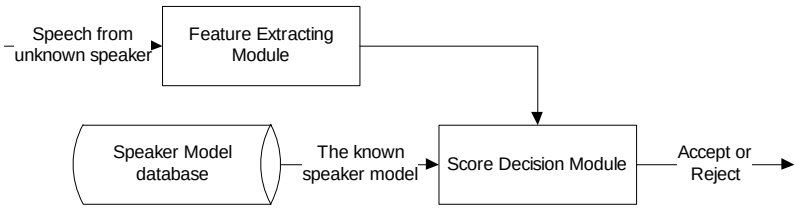

Figure 2.2: Speaker verification.

Feature extraction, also known as speech parameterization, is dominated by the cepstral family. Mel-frequency cepstral coefficients (MFCC) and Linear predictive coding coefficients (LPCC) are the representative technologies. Feature modeling methods can be classified as generative approaches and discriminative approaches based on the training mechanism. The generative approaches capture within class features, including Gaussian Mixture Models (GMM), Hidden Markov Models (HMM), Vector Quantization (VQ), and the well known Joint Factor Analysis (JFA). Note that the HMM take into the consideration the temporal sequence of

the feature therefore widely used in text-dependent speaker verification tasks. Other techniques do not model temporal information, mainly used for text-independent speaker verification tasks. The discriminative approaches capture the boundary between two classes. The representation of discriminative approaches are artificial neural networks (ANNs) and Support Vector Machines (SVM). Table 2.1 summarizes the popular feature extraction and feature modeling methods [22].

Table 2.1: Summary of tasks in speaker verification.

<table>
<tr><td>Feature Extraction Methods</td><td colspan="3">Feature Modeling Methods</td></tr>
<tr><td rowspan="3">MFCC, LPCC</td><td colspan="2">Generative approaches</td><td>Discriminative approaches</td></tr>
<tr><td>Text-independent</td><td>Text-dependent</td><td rowspan="2">SVM, ANN</td></tr>
<tr><td>GMM, VQ, JFA</td><td>HMM</td></tr>
</table>

2.2 Feature Extraction

The most popular feature extraction technique used in voice verification systems is based on *short term cepstral analysis* including MFCC, LPCC, etc [23, 24, 25]. We now briefly review two basic techniques that will be essential to our entropy estimation and our attack.

2.2.1 Cepstral Analysis

Cepstral analysis [26] is widely used technique today for speech and speaker feature extraction tasks. A human voice signal $s(n)$ is typically modeled as a convolution $s(n) = e(n) * \theta(n)$ where $\theta(n)$ represents the response of the vocal system and the $e(n)$ represents the excitation. In both speech recognition and speaker recognition applications the goal is to extract unique features representing the speaker or the speech. This information is precisely the kind of information that is captured by

the response of the vocal system $\theta(n)$. Consequently, the initial task of any speech or speaker recognition application becomes the extraction of $\theta(n)$. To achieve this goal, cepstral analysis may be employed. According to [27], the cepstrum of a signal $s(n)$ is defined as

$$c_s(n) = \mathrm{DFT}^{-1}\{\log\{|\mathrm{DFT}(s(n))|\}\} = \frac{1}{2\pi}\int_{-\pi}^{\pi} C_s(w)e^{jwn}dw \ . \tag{2.1}$$

Here DFT represents the Discrete Fourier Transform, $S(\omega)$ denotes the DFT coefficients of $s(n)$, and the cepstrum is obtained by $c_s(n) = \mathrm{DFT}^{-1}\{\log(|S(\omega)|)\}$. Based on equation 2.1, the steps in computing the cepstrum can be explained as follows:

1. Compute the DFT of the speech signal $s(n)$: by computing the DFT, the convolution of the vocal system $\theta(n)$ and the excitation $e(n)$ turns into a multiplication, i.e. $S(\omega) = E(\omega) \cdot \Theta(\omega)$.

2. Compute the logarithm of the amplitude of $S(\omega)$: by computing the logarithm of $S(\omega)$, the multiplication $E(\omega) \cdot \Theta(\omega)$ turns into addition $C_s(\omega) = \log(|S(\omega)|) = \log(|E(\omega)|) + \log(|\Theta(\omega)|) = C_e(\omega) + C_\theta(\omega)$.

3. Compute the inverse DFT [1] of the above: this step will effectively take the signal into a different frequency domain (called quefrency) where the signal $c_e(n)$ representing $C_e(\omega)$ in quefrency domain resides at the higher quefrency part while $c_\theta(n)$ (the quefrency representation $of C_e(\omega)$) will reside at the lower quefrency part. This allows us to extract the response of the vocal system.

The steps above highlight a general framework for extracting voice features. In order to produce more effective results different scaling techniques can be applied to the

[1] Note that signal $C_s(\omega)$ is even, so the DFT is equivalent to the Discrete Cosine Transform (DCT) which is adopted in practical implementations.

process. In particular, we will be interested in using the cepstrum along with the Mel-scale filter banks and short term processing. In the next two sections we briefly explain these techniques.

2.2.2 Short-term Processing

Commonly a speech signal $s(n)$ is not processed as a whole. Instead, signals are initially divided into a number of overlapping smaller segments each of which is processed individually. This partitioning of the signal is motivated by the dynamic variation of the signal which is better captured through the smaller segments.

There are two important concepts in short term processing: frame rate (step) and window duration. Typically both are expressed in milliseconds. Window duration is the length of one short term piece. Where as frame rate is the time duration between the beginning of two windows [28]. Research shows that the quickest movement of vocal articulators are in the order of 10 ms [27, 29] and therefore is used as the frame rate. The window duration is commonly 2–3 times the frame rate causing an overlap of adjacent short term pieces.

2.2.3 Mel-scale & Filter Banks

As mentioned earlier the cepstrum spectrum is typically used along with Mel-scale filter banks which yields the Mel-frequency cepstral coefficients (MFCC) [30]. The central idea of the Mel (melody) frequency scale is to apply a linear map on the signal's frequency components below 1000 Hz and a logarithmic map for frequency components above 1 kHz. As such, the Mel-scale can be represented using the

following equation [31].

$$\text{Mel}(f) = 2595 \cdot \log_{10}\left(1 + \frac{f}{700}\right) \ ,$$

where f represents the normal frequency components in the range 0 to 4 kHz — the range of the traditional telephone bandwidth.

It is important to note that the Mel-frequency components are influenced by the amplitudes of nearby components. This effect forms a *critical-band*. The critical-band is constant for about 100 Mels on Mel-scale. Mel filter banks are designed with the critical-band in mind. The original Mel filter banks are 20 overlapped filter banks with 10 evenly distributed banks below 1 kHz in Hertz scale and 10 banks with a log distribution covering the range of $1 - 4$ kHz. To compute the Mel frequency components, $C_s(w)$ will go through the group of Mel filter banks yielding $C_s^{mel}(w)$, the Mel-frequency version of $C_s(w)$, which will be used in the later steps of cepstral analysis.

2.2.4 MFCC and LPCC

Mel-frequency cepstral coefficients (MFCC) [30, 32] have been shown to outperform any other Short Term Cepstrum feature extraction technique in speech recognition and later on widely used in speaker verification tasks. Similar to other cepstral features, MFCC is obtained from a speech signal through a combination of transforms [21, 33, 25]. Particularly, MFCC can be carried out with the following steps.

1. Break the input into a number of time frames to be processed independently. Each frame is typically $20 - 30$ ms.

2. Using a Fast Fourier Transform (FFT) compute the frequency components of each of the time frames and take the amplitude.

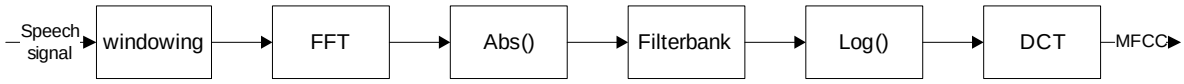

Figure 2.3: Transformation from signal to MFCC.

3. Use a number of triangular band-pass filters in order to project the frequency components of each frame into the Mel-scale.

4. Take the logarithm.

5. Apply a discrete cosine transform (DCT) on the output of the filters in order to compute the MFCC for each frame.

The output of the above steps will be a matrix C where the entry c_{ij} represents the i^{th} Mel-frequency cepstral coefficient for the j^{th} time frame of the input sound as shown in Figure 2.3. To remove the channel filter bias and intra-speaker variability, compensation methods can be applied [21]. Note that MFCC processing is invertible by inverting each step in the MFCC computation steps. However, because some of the MFCC processing steps are non-linear, the inversion will be a lossy process. The inversion details can be found at [34].

The other often used voice feature is Linear Predictive Coding coefficients (LPCC) [35]. The LPCC is also based on short term processing and cepstral analysis (see Section 2.2.1 and 2.2.2 for detail). The first two and last steps of LPCC transformation are exactly the same as MFCC. The rationale behind LPCC is that the voice signal can be viewed as a group of auto regressive (AR) filters. The LPCC estimates the parameter of the AR filter for each windowed speech signal. For each windowed signal, the parameter of the AR filter is estimated by previous parameters. That is the reason why it is called linear predictive coding.

From mathematical point of view, MFCC and LPCC are very similar to each other. They can both be viewed as high dimensional continuous random variables. We name them as short-term processed cepstral feature (STC) because of their similarity in math form. Whenever the STCs are extracted, they are not directly used as templates for verification task. Instead, they are commonly further compressed as certain mathematical model. In the next section we will discuss how the voice feature is modeled.

2.3 Feature Modeling

In voice verification, the extracted feature is not directly used as the voice template. Instead, a more compressed probabilistic representation will be generated based on the voice feature. This process is called feature modeling. We now briefly review Gaussian mixture model. This approach founds the basis of text-independent speaker verification algorithms.

2.3.1 The Gaussian Mixture Model

The Gaussian Mixture Model (GMM) [2] is one of the most widely used voice models in text-independent speaker verification. Based on the GMM, many other methods were derived. One of the most popular methods is Joint Factor Analysis (JFA) [6, 7, 8]. The JFA model tackles the inter-speaker variability problem and therefore improves the performance of the speaker verification process. The GMM is a uni-state HMM [36] regardless of temporal information. The GMM is based on the fact that any probability distribution can be expressed as a collection of weighted Gaussian distributions with different means and variances [37]. Each Gaussian may reflect one aspect of features of the human voice. What is interesting about the

GMM is that the model is trained using unsupervised computer clustering which means that the individual Gaussian distributions are unlabeled. Therefore, we may not know which Gaussian distribution captures which features of the human voice.

The GMM is a collection of weighted Gaussian distributions λ which reflects the real distribution of mass[2]. A GMM is denoted by $\lambda = \{p_i, \mu_i, \Sigma_i\}\ i = 1, 2, ...N$ where p_i gives the weight of i^{th} component. Therefore, $\sum p_i = 1$. The mean and variance of the i^{th} component are represented by μ_i and Σ_i, respectively. N represents the number of Gaussian components. The Gaussian Mixture Density is defined as

$$p(X|\lambda) = \sum_{i=1}^{M} p_i b_i(X) \ . \tag{2.2}$$

Where X is a random vector, $b_i(X)$ is probability density function of i^{th} component explicitly given as

$$b_i(X) = \frac{1}{\sqrt{2\pi|\Sigma_i|}} e^{-\frac{1}{2}(X-\mu_i)'\Sigma_i^{-1}(X-\mu_i)} \ .$$

Given K observations of the random vector X, the probability of X following the GMM λ can be expressed as

$$q(X|\lambda) = \left(\prod_{k=1}^{K} p(X_k|\lambda) \right)^{\frac{1}{K}} \tag{2.3}$$

where X_k is the k^{th} observation of X. For a known speaker j, the GMM model λ_j is computed such as to maximize the overall probability $q(X_j|\lambda_j)$. Therefore, the GMM λ_j provides a voice template. In GMM based biometric verification system, a two phase scenario is applied. In the enrollment phase, a feature X_j extracted from a person j, is used to generate a template GMM λ_j. In the verification phase,

[2]In the voice verification case this corresponds to the cepstral features.

a decision function

$$q(X'|\lambda_j) \begin{cases} \geq T \text{ accept} \\ < T \text{ reject} \end{cases} \tag{2.4}$$

is computed. Where T is the pre-defined constant threshold and X' is a fresh feature extracted from an unknown person who claims to be j. If the likelihood $q(X'|\lambda_j)$ is greater than the thresh hold, the unknown person passes the verification as j otherwise the authorization fails.

Finally, we note that a more popular version of the GMM, namely the Adapted Gaussian Mixture Model, is in use today [3]. In the Adapted GMM, a universal background model λ_b is generated by training with samples collected from all speakers. Afterwards, each speaker is modeled by adapting the background model. In the verification phase, the Gaussian mixture density $p(x)$ in Equation 2.2 is substituted by $p'(x) = \frac{p(x|\lambda_j)}{p(x|\lambda_b)}$. The details of the adapted GMM modeling algorithm can be found in [3]. The main advantage of the adapted GMM is that the training phase for a speaker is much faster while at the same time it gives a more accurate verification performance. In this dissertation we will base our analysis on the more popular adapted GMM.

Chapter 3

Entropy Estimation

In this chapter, we estimate the entropy of a human voice password database assuming a voice authentication technology that uses MFCC features and a GMM. In Section 3.2 and 3.3 we first explain the relationship between entropy and security and introduce some basic concepts of entropy. In Section 3.4, we explore the estimation of differential entropy on STC and point out the problem of using this entropy metric. In Section 3.5 we propose an entropy estimation on STC using the relative entropy and show the relative entropy is an good measurement of entropy bits of the voice features. In Section 3.6, we show experimental results. Finally in Section 3.7, we connect the entropy estimation to the attack in Chapter 4.

3.1 Introduction

Entropy is a measurement of uncertainty of a random variable. The importance of establishing an entropy estimation is the real measure of identification capacity. Security cannot be guaranteed by a complicated extraction process, but instead has to be rooted in entropy. Establishing the entropy content of a human voice password database is crucial in estimating the achievable secure level of a voice authentication system.

In this chapter we develop an estimation on the relative entropy between users' voices from human voice password database using the Short Term Cepstrum features. By understanding the relative entropy between humans' voices using a certain verification technique we are essentially achieving two goals:

1. We place an approximation on the number of people who can be uniquely identified using the studied verification technique.

2. We place an approximation on the computational effort required to break the identification technique. That is, we capture the amount of effort needed to mimic someone else's voice without previous knowledge of that person's voice characteristics.

Under this view we can start evaluating various voice identification software and technologies in a way that takes into account the most important of all factors, i.e. security. Further, in this chapter we derive a technique and then use it to estimate the entropy extracted from the YOHO voice verification corpus [40].

3.2 Entropy and Security

The entropy [41] of a discrete random variable X is defined as

$$H(X) = E\left[\frac{1}{\log P(X)}\right] = \sum_{X \in \chi} P(X) \log P(X) \tag{3.1}$$

where $P(X)$ is the probability density function of the r.v. X. It is a measurement of information bits, i.e. a quantity measurement of the minimum number of yes/no questions one needs to ask to determine the value of X. The entropy of the English alphabet, for example, is 4.7 bits assuming the frequency of appearance of each letter is equal (see Example 1). This means given a letter, at least 4.7 binary questions need to be asked in order to successfully guess the letter. In other words, one needs to make a guess among all $2^{4.7} = 26$ possibilities to obtain the right letter. In reality, however, the frequency of letters is not uniform as it is a commonsense that the letter "e" has the highest frequency while the "z" has the lowest frequency in the daily usage of the English language. If the frequency of letter usage has been taken into consideration, the entropy of the English alphabet is only 4.14 bits [42]. That is to say, one needs less effort to successfully guess a letter randomly picked from an English article than to guess a letter randomly picked from the alphabet set.

Example 1. *Let $\alpha = \{a, b, c, \ldots, z\}$ be the alphabet of English. Assume the frequency of each letter is equal, then the entropy of α is*

$$H(\alpha) = \sum_{i=1}^{26} \frac{1}{26} \log \frac{1}{26} = 4.7 \ bits$$

The concept of entropy highly relates to the security level of cryptographic authentication systems. The entropy of a cryptographic key indicates the expected

number of trials that an attacker needs in order to successfully guess the key. To make the concept clear, we review Example 1. Assume a key generation system produces a key with a single letter picked from the English alphabet, then an attacker may obtain that key with $2^{4.7-1}$ trials on average using brute-force attack. The probability that the attacker successfully guesses the key with one attempt is $p_r = \frac{1}{2^{4.7}} = \frac{1}{26}$. Now assume the attacker is memory-less, i.e. he may re-pick the same key in his attempts, with N attempts the attacker has $P = 1 - (1 - p_r)^N$ success rate in guessing the key. The attacker will have more than 50% success rate to get the correct key with 18 trials. If the authentication system picks the letter randomly from an English book, i.e. the frequency of letter usage is taken into consideration, the attacker is expected to obtain the key with only $2^{4.14-1}$ trials. The connection between entropy and security is also valid in voice authentication systems. Remember that the voice feature can be viewed as the high dimensional random variable (see Section 2.2.4 for detail). The entropy of the voice feature predicts the effort that an attacker needs to reproduce a victim's voice feature.

3.3 Shannon Entropy, Differential Entropy and Relative Entropy

The entropy in the form of Equation 3.1 is also known as Shannon entropy which measures the quantity of expected information bits of a random variable. The limitation of Shannon entropy is that it only applies to the discrete random variable.

To extend the Shannon entropy to continuous form, the differential entropy (continuous entropy) is introduced. The differential entropy $h(X)$ of a continuous

random variable X is defined as

$$h(X) = -\int_S f(x) \log f(x) dx \tag{3.2}$$

where $f(x)$ and S are the probability density function and the support set of the r.v. X, respectively [41]. However, the differential entropy is quite different from the Shannon entropy in terms of its properties. For instance, the differential entropy can be negative while the Shannon entropy can not. A more important difference is the differential entropy is not a quantity measurement of expected information bits as Shannon entropy is. Instead, the differential entropy reflects the compactness of a random variable in distribution, i.e. a small value of the differential entropy means the random variable is confined to a small area of distribution, while a large value of the dfferential entropy indicates the random variable are more spread in the distribution. The differential entropy relates to Shannon entropy through the quantization of a random variable [41]. Suppose the range of a continuous r.v. X is divided into n bins. Each bin has a length of $\Delta = 2^{-n}$. The quantized random variable X^{Δ} is defined as

$$X^{\Delta} = x_i \qquad when \Delta i \leq X < \Delta(i+1) \tag{3.3}$$

where x_i is the expected value of i^{th} bin of the r.v. X. The entropy of the discrete r.v. X^{Δ} which is the quantized version of the r.v. X should be

$$H(X^{\Delta}) = -\sum \Delta f(x_i) \log f(x_i) - \log \Delta \tag{3.4}$$

As the $\Delta \to 0$,

$$H(X^{\Delta}) + \log \Delta \;\; \to \;\; h(X) \tag{3.5}$$

$$H(X^{\Delta}) - n \;\; \to \;\; h(X) \tag{3.6}$$

The other form of entropy metric is relative entropy a.k.a. Kullback-Leibler divergence. Relative entropy between two probability mass functions $p(x)$ and $q(x)$, is defined as

$$D(p||q) = E_p \log \frac{p(x)}{q(x)} \tag{3.7}$$

$$= \sum_{x \in \chi} p(x) \log \frac{p(x)}{q(x)} \tag{3.8}$$

Similarly, the relative entropy in continuous case is in the form of

$$D(p||q) = \int p(x) \log \frac{p(x)}{q(x)} \tag{3.9}$$

where the $p(x)$ and $q(x)$ are now probability density functions [41]. The relative entropy is a counting measure similar to Shannon entropy, therefore it measures the information in bits. On the other hand, unlike Shannon entropy/differential entropy, the properties of relative entropy are the same in both discrete and continuous cases. The relative entropy is conceptually known as a measurement of the "distance" between two distributions of random variables. Note that the relative entropy is not a symmetric metric, i.e. $D(p||q)$ is not commonly equal to $D(q||p)$. According to [41], the relative entropy measures the inefficiency using distribution q to describe the real distribution p.

3.4 Exploration of Differential Entropy on STC

Since the voice feature STC is continuous random variable, we first explore the possibility of applying differential entropy to the voice feature.

Proposition 1. *Let* $\mathrm{STC}(S) \in \mathbb{R}^{Q \times L}$ *be a matrix of* L Q*-dimensional STC vectors of a speech signal* S *representing the concatenation of all speech signals in a database* $\mathcal{D}$*, and let* σ_j *be the standard deviation of the* j^{th} *row of* $\mathrm{STC}(S)$*, then*

$$h\left(\mathrm{STC}(S)\right) \leq \frac{1}{2}\sum_{j=1}^{Q}\log_2(2\pi e\sigma_j^2) \quad , \tag{3.10}$$

where $h\left(\mathrm{STC}(S)\right)$ *is the differential entropy of* $\mathrm{STC}(S)$*.*

Proof. Let the C_j be the j^{th} row of $\mathrm{STC}(S)$, then

$$h\left(\mathrm{STC}(S)\right) = h(C_1, C_2, \ldots C_Q) \leq \sum_{j=1}^{Q} h(C_j) \quad , \tag{3.11}$$

where the equality holds if and only if the $C_1, C_2, \ldots C_Q$ are independent. According to [41] the differential entropy of any random vector X with $E(X) = \mu$ and $Var(X) = \sigma^2$ is upper bounded by the entropy of the Gaussian distribution $N(\mu, \sigma^2)$. As such, we can upper bound $h\left(\mathrm{STC}(S)\right)$. First we write

$$h(C_j) \leq h(N(\mu_j, \sigma_j^2)) \quad , \tag{3.12}$$

where μ_j and σ_j^2 represent the mean and variance of C_j. We now combine Equations 3.11 and 3.12 to upper bound $h\left(\mathrm{STC}(S)\right)$ as

$$h\left(\mathrm{STC}(S)\right) \leq \sum_{j=1}^{Q} h(N(\mu_j, \sigma_j^2)) = \frac{1}{2}\sum_{j=1}^{Q}\log_2(2\pi e\sigma_j^2) \quad . \tag{3.13}$$

□

Equation 3.13 gives a loose upper bound on the differential entropy of the STC features. We next discuss the possibility of deducing more rigorous bounds. To do this, we analyze the source of entropy inside STC features. There are 3 main sources of entropy which contribute to the total entropy found in STC(S); the text-independent voice entropy $h(V)$, the entropy in the spoken words $h(W)$ and noise. The entropy added by the noise is useless from a security point of view. However, for a general bound we assume the noise is a part of $h(V)$. Note that this assumption can only increase $h(V)$ and therefore will not effect the validity of our bound. Based on this we can write

$$h(\text{STC}(S)) = h(V, W) \quad .$$

Our goal is to upper bound $h(V)$ which according to the joint entropy definition can be written as

$$h(V) = h(V, W) - h(W|V) \quad . \tag{3.14}$$

$h(V)$ is the text-independent entropy that is solely contributed by the speaker's voice, $h(W|V)$ is the entropy of the said words given a specific speaker and $h(V, W)$ is the total entropy in the speech signal. The upper bound on $h(V)$ can be computed using $h_u(V, W)$ the upper bound of $h(V, W)$ and $h_l(W|V)$ the lower bound of $h(W|V)$. That is,

$$h(V) \leq h_u(V, W) - h_l(W|V) \quad . \tag{3.15}$$

Upper-bounding $h(V, W)$ is equivalent to upper-bounding $h(\text{STC}(S))$. According to Equation 3.13, we can upper bound the entropy in $h(V, W)$

$$h(V, W) \leq \frac{1}{2} \sum_{j=1}^{Q} \log_2(2\pi e \sigma_j^2) = h_u(V, W) \quad , \tag{3.16}$$

Next, we compute a lower bound on $h(W|V)$. By definition (see [41]),

$$h(W|V) = \sum_{i=1}^{M} p(V_i) h(W|V_i) \quad .$$

Since we are restricting ourselves to voice signals after being processed by STC, $h(W|V_i)$ will be $h(\mathrm{STC}_i)$. That is, the entropy in the speech of speaker i. $p(V_i)$ is the probability that speaker i is selected from the database of speakers. With M speakers in the database all equally likely to be chosen we have $p(V_i) = 1/M$. So we can simplify the equation of $h(W|V)$ as

$$h(W|V) = \frac{1}{M} \sum_{i=1}^{M} h(W|V_i) = \frac{1}{M} \sum_{i=1}^{M} h(\mathrm{STC}_i) \ . \tag{3.17}$$

To compute a lower bound on $h(W|V)$ we resort to GMM. The Gaussian mixture model is a probabilistic technique for modeling an arbitrary random vector. As the number of components in the GMM λ increases it produces a more accurate approximation of the distribution $\mathcal{X}$ from which an arbitrary random vector X is drawn. In fact, given a sufficient number of components a GMM can approximate any smooth function to arbitrary accuracy [43]. Therefore, with sufficiently many components in λ one can approximate the entropy of $\mathcal{X}$ by using the entropy of λ, that is $h(\mathcal{X}) \approx h(\lambda)$. We now borrow the following proposition from [37].

Proposition 2. ***([37])*** *Given a set of random vectors $X_i \in \mathbb{R}^Q$ from some distribution $\mathcal{X}$ and modeled by a GMM $\lambda = \{w_i, \mu_i, \Sigma_i\}$ where $i = 1, 2, ...N$, a lower bound on the entropy of λ can be written as*

$$h_l(\lambda) = -\sum_{i=1}^{N} w_i \cdot \log_2 \left(\sum_{j=1}^{N} w_j \cdot z_{i,j} \right) \quad , \tag{3.18}$$

where

$$z_{i,j} = N(\mu_i; \mu_j, \Sigma_i + \Sigma_j) = \frac{1}{\sqrt{2\pi|\Sigma_i + \Sigma_j|}} e^{-\frac{1}{2}(\mu_i - \mu_j)'(\Sigma_i + \Sigma_j)^{-1}(\mu_i - \mu_j)} \quad .$$

According to [43], as N goes to infinity, λ tends to describe $\mathcal{X}$ which is the true distribution of X. With that we state the following corollary to Proposition 2.

Corollary 1.

$$h(\mathcal{X}) = \lim_{N\to\infty} h(\lambda) \geq \lim_{N\to\infty} h_l(\lambda) \quad , \tag{3.19}$$

where $h(\cdot)$ is the differential entropy.

Now let λ_i be a GMM for STC_i then according to Corollary 1 we have

$$h(\text{STC}_i) = \lim_{N\to\infty} h(\lambda_i)$$

where N denotes the number of Gaussian components. Using Proposition 2 and Equation 3.17 we have

$$h(W|V) \geq \quad \tfrac{1}{M} \textstyle\sum_{i=1}^{M} \lim_{N\to\infty} h_l(\lambda_i) = h_l(W|V) \quad . \tag{3.20}$$

Finally, using Equations 3.15, 3.16 and 3.20 we get

$$h(V) \leq \frac{1}{2} \sum_{j=1}^{Q} \log_2(2\pi e \sigma_j^2) - \frac{1}{M} \sum_{i=1}^{M} \lim_{N\to\infty} h_l(\lambda_i) \tag{3.21}$$

Equation 3.21 gives the upperbound of differential entropy. As we discussed in Section 3.3, the differential entropy is not directly a quantity measurement of information in bits. To convert the differential entropy to Shannon entropy, we have to apply Equation 3.6. Therefore, we apply Equation 3.6 to Equation 3.16, 3.20 and

3.21 to get

$$H(V,W) \leq \frac{1}{2}\sum_{j=1}^{Q}\log_2(2\pi e\sigma_j^2) + c_1 n \tag{3.22}$$

$$H(W|V) \geq \frac{1}{M}\sum_{i=1}^{M}\lim_{N\to\infty} h_l(\lambda_i) + c_2 n \tag{3.23}$$

$$H(V) \leq \frac{1}{2}\sum_{j=1}^{Q}\log_2(2\pi e\sigma_j^2) - \frac{1}{M}\sum_{i=1}^{M}\lim_{N\to\infty} h_l(\lambda_i) + (c_1 - c_2)n \tag{3.24}$$

where n is quantization bit while c_1 and c_2 are two constant.

In this section we developed an upperbound of differential entropy of voice features. However, we have no idea of the relation of c_1 and c_2. Unless c_1 equals c_2 the bound on $H(V)$ will not be constant. For instance, if $c_1 > c_2$ then $H(V)$ is in a form of a constant plus n bit. By increasing n, we can increase the extracted entropy. With the presented analysis and following brief discussion we can see the difficulty in estimating information bits in human voice through differential entropy. This motivated us to consider another entropy metric, i.e. relative entropy, which will be discussed in the following section.

3.5 Relative Entropy on STC

Our goal in this section is to introduce relative entropy as an measurement of uncertainty of the STC voice features. Several estimation methods and bounds are discussed in Section 3.5.2. Relative entropy is useful for measuring the security obtained from using human voice identification on a given database of speech samples.

3.5.1 Relative entropy estimation of voice features

Since voice features are continuous[1], the classic entropy (the Shannon entropy) [41] which measures the uncertainty of a discrete random variable does not apply directly. Differential entropy once appeared to be the more appropriate metric. Based on the discussion in the previous section, it is not. The reason is differential entropy does not inherit many properties of entropy in the discrete case and therefore does not reflect the accurate number of bits that is necessary to represent an individual's voice. More importantly, the security of voice passwords is not rooted in the absolute entropy of voice itself. Instead, it is rooted in the "distance" between the different people's voice signals.

To overcome these obstacles, we make use of the notion of *relative entropy*. Relative entropy is defined by Equation 3.7 and 3.9 Unlike entropy which has two different forms in case of discrete and continuous, relative entropy maintains the same form from discrete to continuous case. In [44], relative entropy is used to represent a measurement of biometric information i.e. how close two biometric features are. Now, we use relative entropy to measure entropy residing in voice. Assume $V_i(X)$ and $V_j(X)$ refers to GMM of voice from person i and j, the relative entropy

$$D_{KL}(V_i||V_j) = E_{V_i(x)}\left[\log(\frac{V_i(x)}{V_j(x)})\right] \tag{3.25}$$

measures how many additional bits one would need to express V_i given an expression for V_j. The measurement of relative entropy is essential to estimate the security level of voice authentication systems. It estimates the effort that an attacker need to successfully impersonate an arbitrary victims i.e. the relative entropy indicates how many attempts the attacker needs to make in order to modify his own voice

[1]Note that state-of-the-art text-independent speaker authentication uses a continuous probability density function GMM to model voice

feature to be the victim's voice feature.

3.5.2 Estimation of Relative Entropy for two GMMs

There is no closed form expression that allows us to compute the relative entropy for two given GMMs. However, several estimation methods can be applied. As mentioned in [45], assume two GMMs $\lambda(x)$ and $\lambda'(x)$, an accurate estimation of $D_{KL}(\lambda||\lambda')$ can be obtained by Monte Carlo Sampling:

1. Draw from λ i.i.d. samples $\{x_i\}_{i=1}^n$.
2. Compute $D_{MC}(\lambda||\lambda') = \frac{1}{n}\sum_{i=1}^n \log\left(\frac{\lambda(x_i)}{\lambda'(x_i)}\right)$.

Note that as n goes to infinity, D_{MC} converges to $D_{KL}(\lambda||\lambda')$. The weakness of Monte Carlo Sampling is the high complexity of computation. In practice, it is more convenient to use a bound over an estimation procedure. Here we borrow the following proposition from [45].

Proposition 3 (Relative Entropy Upper Bound [45])**.** *An upper bound on relative entropy between two GMMs $\lambda(x)$ and $\lambda'(x)$ is given as*

$$D_{KL}(\lambda||\lambda') \leq \sum_{i,j} p_i p'_j D_{KL}(N(\mu_i,\Sigma_i)||N(\mu'_i,\Sigma'_i))) = D_{u1}(\lambda||\lambda') \ . \quad (3.26)$$

Alternatively, if two GMMs have same number of components n, the upper bounded can also be expressed as

$$\begin{aligned} D_{KL}(\lambda||\lambda') &\leq D_{KL}(p||p') + \sum_i^n p_i D_{KL}(N(\mu_i,\Sigma_i)||N(\mu'_i,\Sigma'_i)) \\ &= \sum_i^n p_i(\log(\frac{p_i}{p'_i}) + D_{KL}(N(\mu_i,\Sigma_i)||N(\mu'_i,\Sigma'_i))) \\ &= D_{u2}(\lambda||\lambda') \end{aligned} \quad (3.27)$$

where $\lambda = \{p_i, \mu_i, \Sigma_i\}_{i=1}^{n}$ *and* $\lambda' = \{p'_i, \mu'_i, \Sigma'_i\}_{i=1}^{n}$.

We next demonstrate the application of these estimates on a real voice database.

3.6 Experiments & Results

In this section we will utilize the relative entropy estimation explained in Section 3.5 to estimate the entropy available to a speaker verification system. Our focus will be on speaker verification systems that use the MFCC extraction technique along with GMM modeling. The voice samples that we use for our experiments are collected from the YOHO database which contains 138 speakers, with each speaker reciting a random combination of three two digit numbers. We will start this section by explaining our choice of parameters for the speaker verification system that we analyze.

3.6.1 Parameter Selection

In the system that we analyze, the voice signal is first broken into a number of overlapped 10 ms frames which gives the highest time resolution [27, 29]. Each frame then goes through a hamming window of length 32 ms. Each frame will then produce a 26 dimensional MFCC vector. The first 13 features except 0^{th} dimension are kept as the MFCC features and the rest are discarded. Although the optimal number of filter banks is an open question, typically, the number of filters ranging from 20 to 32 is suggested for frequencies ranging from 0 to 4 kHz [25, 46]. The dimension of the MFCC coefficients should be smaller than the number of filters i.e. the high dimensions are discarded [21]. According to [47], 12 coefficient is sufficient for speaker recognition. The 12-coefficients MFCC setup is used in [48, 49]. Also note that the first and second derivatives of the MFCC features are sometimes

concatenated as the last dimension to the features in order to increase the entropy. However, experiments [47] have shown that dynamic (derivative) features contribute far less to the speaker verification performance than normal features. Furthermore, research in [50] have shown that adding dynamic features contributes very little to the overall identification performance. Therefore, our speaker verification module includes only standard MFCC features. Now, based on the 13 dimensional MFCC features, a 256 component adapted GMM is trained as outlined in [3] which can be summarized as follows

1. A 256 component universal background model λ_b is trained

2. For each person i, the GMM λ_i is trained by adapting the mean vector of λ_b.

Conceptually, the precision of the GMM can be improved by using more components. However, increasing the number of components beyond a certain point increases the computational cost without yielding any significant benefit in the precision. Therefore, we fix the number of Gaussian components to 256 which gives a very good approximation of the original signal [3]. Since the YOHO dataset is only constrained to voice samples reading digits, 256 GMM should be sufficient to accurately estimate both the background model and the speaker model [21].

Note that we only adapt the mean of the background model which is what is suggested in [3]. Also note that the covariance matrices in the background model can be either full or diagonal. Although the full matrices will do a little better than the diagonal matrices in terms of recognition they require more computation time. Based on our database, the verification performance yield an EER of 1.12% and 1.42% using the full and the diagonal covariance matrices, respectively.

3.6.2 Entropy Upper Bound

We now proceed to estimate the relative entropy bound using Equation 3.27. We introduce two quantities: $D_{u2}(\lambda_i||\lambda_b)$ which reflects the expected upper bound on entropy in bits found in λ_i, given λ_b. The second quantity is $D_{u2}(\lambda_i||\lambda_j)$ which reflects the expected upper bound on entropy in bits found in λ_i, given λ_j. Recall that λ_i and λ_b represent the GMM of the i^{th} person and the GMM of the background, respectively. In order to get a good understanding of these numbers we define the averages over all people in the database. We define D_b as the average of $D_{u2}(\lambda_i||\lambda_b)$ over all speakers which can be computed as

$$D_b = \sum_{i=1}^{N} p(\lambda_i) D_{u2}(\lambda_i||\lambda_b) \ .$$

We also define $\bar{D}$ which is the average of $D_{u2}(\lambda_i||\lambda_j)$ over all (i, j) combinations which can be computed as

$$\bar{D} = \sum_{j=1}^{N} p(\lambda_j) \sum_{i=1}^{N} p(\lambda_i) D_{u2}(\lambda_i||\lambda_j) \ .$$

The value D_b is the average upper bound on relative entropy between the people and the background which reflects the amount of information the background reveals about the people. On the other hand, $\bar{D}$ is the average upper bound on relative entropy between any two people in the database which reflects the amount of information that the model of one person yields about another person's model. Table 3.1 shows the results for D_b and $\bar{D}$ when using a diagonal and a full covariance matrix in the model. Figure 3.1 shows the value of $D_{u2}(\lambda_i||\lambda_b)$ for various values of i as well as $D_i = \sum_{j=1}^{N} p(\lambda_j) D_{u2}(\lambda_j||\lambda_i)$.

The results in Table 3.1 and Figure 3.1 suggest that there is no more than 14

	Full Variance	Diagonal Variance
D_b	7.23	8.07
D	12.54	13.95

Table 3.1: Upper bound of Relative Entropy in the YOHO voice database.

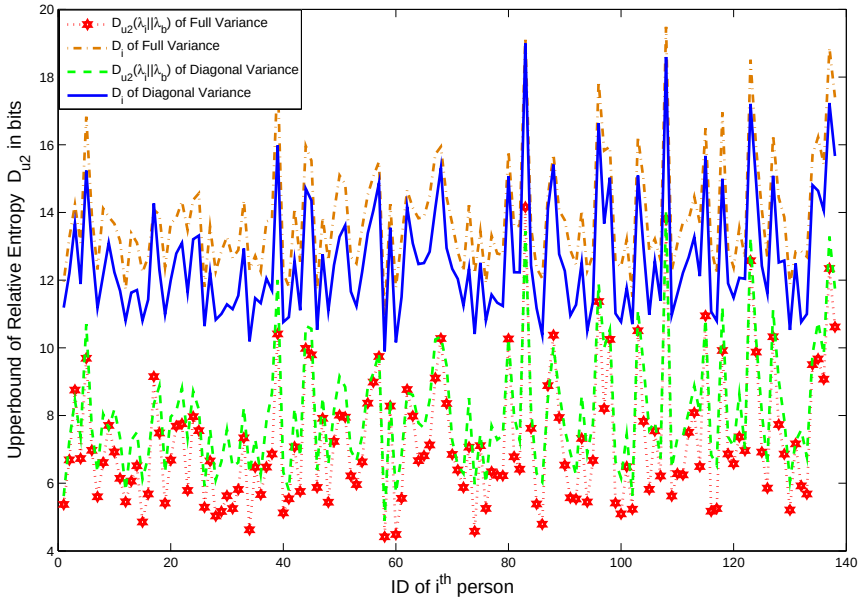

Figure 3.1: Averaged relative entropy bounds D_{u2} for each person in the YOHO database.

bits of entropy that separates one person's GMM from another. This suggests that an attacker with access to his GMM in a database can impersonate another person in the database with less than 2^{14} authentication attempts on average. Note here that a speaker typically will not have access to his GMM in a database. Therefore such a bound might not immediately have any practical implications. However, the low amount of entropy does suggest that an attack might exist to compromise the use of human voice for authentication. Next we explore this possibility.

For an attacker to mount an attack, the first question that should be asked is "How much information will his voice model without background adaptation (denoted $\widehat{\lambda}$) yield about anyone else's voice model?" This question is relevant because without background adaption the attacker can attempt to manipulate his voice directly in order to attack a database of speakers.

To understand this relation, we let the attacker model his voice with a single Gaussian $N(\widehat{\mu}, \widehat{\sigma})$, where $\widehat{\mu}$ and $\widehat{\sigma}$ are the mean and variance of the attacker's own voice features. Now with this model we can estimate the $D_{u1}(\lambda_i||N(\widehat{\mu}, \widehat{\sigma}))$ by Equation 3.26. The result is shown in Figure 3.2.

This result indicates that the distance between the voice model built by the attacker and the voice models in the system is no more than 11 bits. In the next section, we will show that $\widehat{\lambda}$ can be even made more closer to λ_i by increasing the number of components that is used to generate $\widehat{\lambda}$.

Figure 3.2 suggests that an attacker can manipulate his voice in order to impersonate someone else in a database in less than 2^{11} tries on average. Of course these results hint the existence of an attack but do not expose an attack algorithm. Also note that so far we have been looking at the upper bound. In the next section we will re-visit the relative entropy estimates before we relate the quantities to an actual attack.

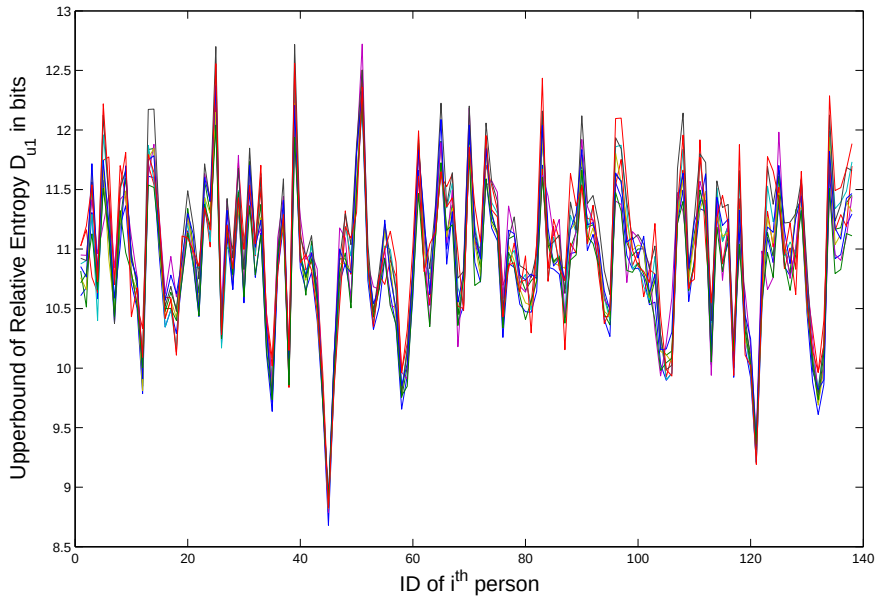

Figure 3.2: $D_{u1}(\lambda_i||N(\widehat{\mu}_j, \widehat{\sigma}_j))$ for 10 selected js. Here $j = 1, 2, \ldots, 10$. Each color represents a j value.

3.6.3 Empirical Results on Monte Carlo Sampling Method

From the estimation above, we know the upper bound of relative entropy in voice. Now, we use Monte Carlo Sampling to derive a more accurate estimate. Assume $\text{MFCC}_i = \{cc_{i,j}\}_{j=1}^{n}$ is the collection of MFCC features for the i^{th} person in a voice database, where $cc_{i,j}$ represents the features of the j^{th} time frame. Then

$$D_{MC}(\lambda_i||\widehat{\lambda}) = \frac{1}{n}\sum_{j=1}^{n}\log\left(\frac{\lambda_i(cc_{i,j})}{\widehat{\lambda}(cc_{i,j})}\right) \quad .$$

Recall that $\widehat{\lambda}$ is the voice model built by the attacker i.e. the GMM without background adaptation. Also note that this is a generous estimation since Monte Carlo Sampling assumes i.i.d. samples. For the YOHO database, n is around $30,000$ which is large enough to yield a decent estimation [45]. To simplify the computation, we pick a single random speaker from the 138 that are in the YOHO database to act as

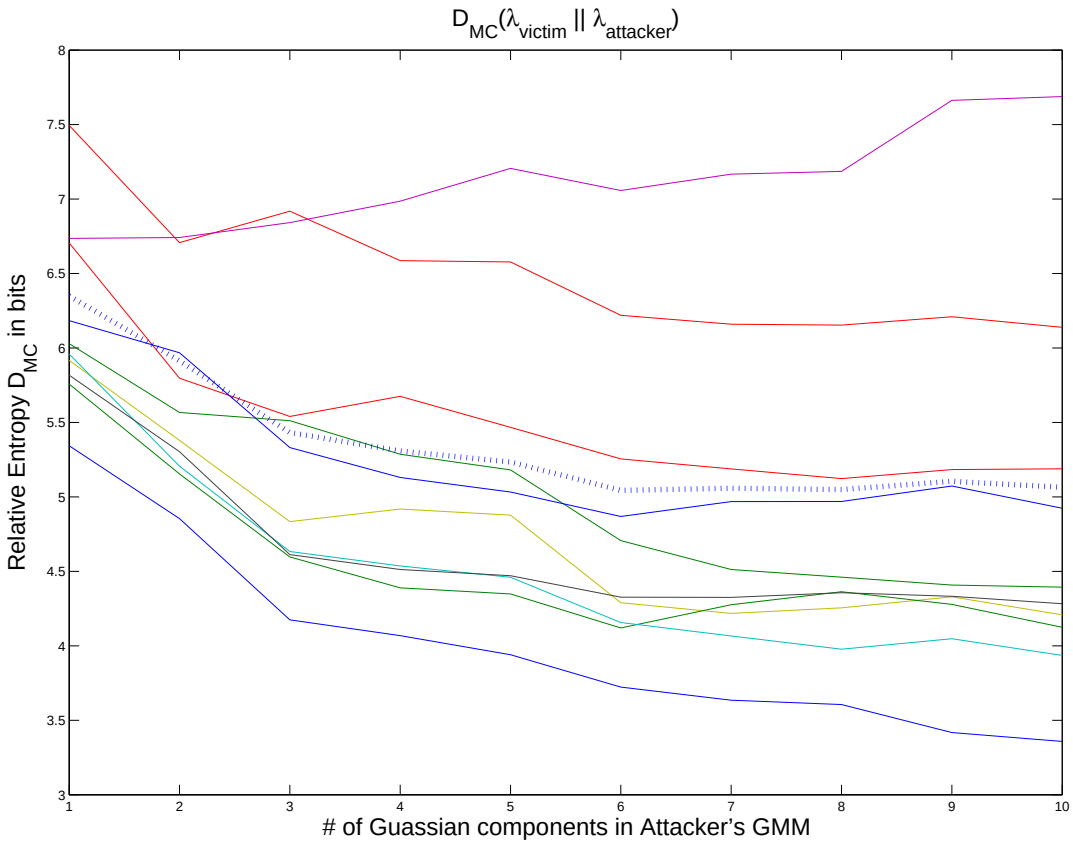

Figure 3.3: Monte Carlo Estimation of relative entropy. The dot line shows the average of 138 victims versus one designated attacker while the solid lines show 10 randomly picked victims versus the attacker.

the attacker. The attacker varies the number of components that he uses to build GMM from 1 to 10. The experiment results can be found in Figure 3.3. We can see that the relative entropy is about 5 to 6 bits on average. Meanwhile as the number of components increases, the relative entropy decreases a little.

3.7 From Entropy Estimates to a Practical Attack

Relative Entropy is a measure of the amount of information that one distribution reveals about another. Low relative entropy is indicative of security weakness in

an authentication system. Although the multi-dimensional MFCC vectors appear to have high degree of freedom, the results we have shown so far indicate that it should be easy for an attacker to impersonate another user's voice in the same database. The results shared in the previous section suggests the existence of an attack that would allow an attacker to run an impersonation attack with an average of about 32 to 64 authentication attempts. Indeed, the authors of this dissertation have successfully demonstrated such attack in [51]. Here we review the attack and expose the connections between the attack in [51] and the results of the previous section. More detail of the attack will be explained in Chapter 4 and Chapter 5.

The attack targets a voice identification system that uses adapted GMM to model basic MFCC features of speakers. Moreover, the attack assumes processing parameters identical to what is outlined in this dissertation.

A successful attack signal needs to pass the speaker verification *and* the speech recognition processes. To produce such a signal we create two attack signals each of which targets one of these two modules. These two attack signals will be merged later, in order to produce the final attack signal which we will refer to as the *hybrid signal*.

The first attack signal will target the speech recognition module. Creating this signal will amount to speech synthesis and therefore will be straightforward. The attacker may simply use his/her own voice to speak the challenge words provided for verification. As we discussed earlier, these signals are used to ensure the freshness of the audio signal that is fed to the system. We refer to this signal as S_1.

The second attack signal requires constructing a speech signal with impulse functions centered at the average of each MFCC feature. More specifically, the attacker analyzes a large amount of his voice signals and then transforms his voice into a number of MFCC features. The attacker can then model his features using

% of passing people

mock signal ratio (q)	# of Gaussian components in GMM (n)									
	1	2	3	4	5	6	7	8	9	10
1	0.00	0.00	0.00	0.00	4.41	6.62	6.62	6.62	5.88	5.88
2	0.00	0.74	0.74	9.56	25.74	29.41	32.35	39.71	52.94	55.15
3	0.00	1.47	11.76	26.47	40.44	45.59	52.21	59.56	55.15	55.15
4	0.00	4.41	19.12	41.18	47.79	60.29	63.97	72.79	74.26	77.94
5	0.00	8.09	29.41	55.15	56.62	70.59	72.06	80.88	88.24	90.44
6	0.00	5.88	34.56	58.82	64.71	75.74	77.94	90.44	93.38	94.12
7	2.21	11.03	45.59	62.50	71.32	77.94	82.35	91.91	94.12	95.59
8	1.47	12.50	43.38	62.50	70.59	80.15	86.76	92.65	98.53	97.79

Table 3.2: Success rate of attack with various parameters.

a few component GMM (in this attack we use 9-component GMM). The attacker's GMM will contain a number of means (in our case 9). Now the attacker will create a sound signal which corresponds to an impulse function centered at one of the GMM means by inverting the GMM model for MFCC features [34]. The impulse function will correspond to a voice signal where every time frame gives rise to the same exact MFCC value (the value of the chosen mean). This means that the attacker will have several candidates for the attack signal one corresponding to every mean in the GMM. We refer to these signals as S_2^i where $i \in [1, \ldots, n]$ and n represents the number of components in the GMM model.

In the last step of the attack we merge the two attack signals to create the final hybrid signal. There are a number of strategies to merge the signals. Our results show that the most successful method is a simple concatenation. This means that the hybrid signal will consist of the first attack signal followed immediately with the second attack signal. There is a degree of freedom here, i.e. the duration of the two signals relative to each other. In the next section we will show that the best results where achieved when the second attack signal was several times the size of the first attack signal. We refer to the hybrid signal as $H_i = [S_1|S_2^i]$. Note that the first attack signal needs to be computed in real time due to the challenge. However, the

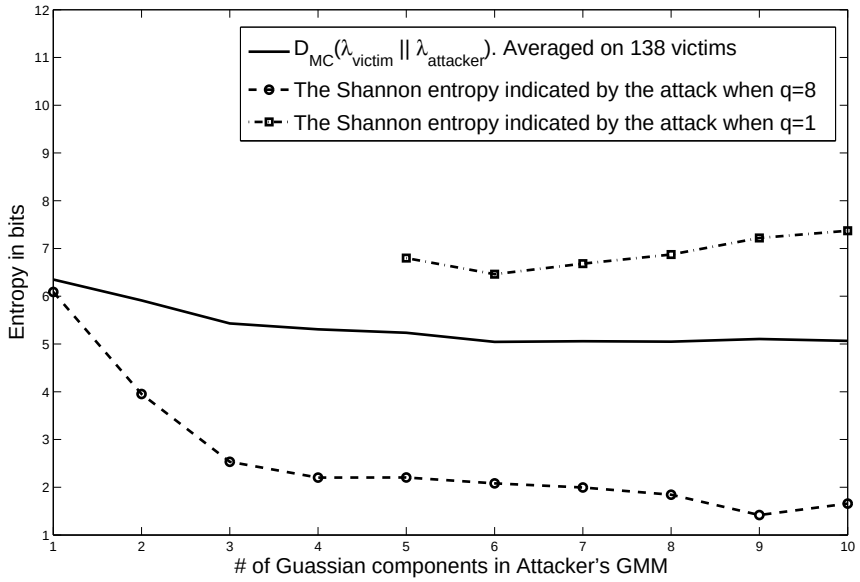

Figure 3.4: Relation between Relative entropy and attack. Note that when $q = 1$, the Shannon entropy goes to infinity in case of $n = 1, 2, 3, 4$.

second attack signal can be precomputed.

Table 3.2 shows the success rate of the attack with different parameter. Here the q parameter represents the ratio of the length of the mock signal S_2^i to the speech signal S_1. The variable n denotes the number of components in the attacker's GMM. The larger the q parameter is, the more dominant the mock signal part is. From an authentication protocol perspective, the parameter n determines the max number of trials the attacker is allowed to make before triggering an authentication failure. Note that an attacker can succeed with a very high probability ($> 98\%$) in as little as 9 trials.

To relate success rates of this attack to the relative entropy, we convert the success rates in Table 3.2 into bits measured using Shannon entropy. To do this we model each attack as a series of n independent repeated Bernoulli experiments. As shown in Table 3.2, n represents the number of GMM components, while p

represents the success rate by up to n trials. The outcome of each trial therefore can be captured by a Bernoulli probability distribution $(r, 1-r)$ where

$$r = \left(1 - (1-p)^{\frac{1}{n}}\right),$$

resulting in a Shannon entropy of $-\log r$. Figure 3.4 shows a simple comparison between relative entropy estimation D_{MC} and Shannon entropy estimates applied to the Bernoulli distributions. We focus on $q = 1$ (least successful attack) and $q = 8$ (most successful attack) where the mock signal is least and most dominant respectively. It should be clear that the entropy estimates obtained from a real attack correlates with the entropy estimate derived in the previous section as illustrated by Figure 3.4.

3.8 Conclusions

In this chapter, we took a first step towards evaluating voice password databases from a security perspective. We defined an authentication model that captures what is nowadays used in many voice password products. Furthermore, we developed a theoretical framework based on the notion of relative entropy in order to estimate the entropy of a voice database. Using the derived entropy estimates we were able to estimate the security level offered by a voice authentication system relying on a voice database. Our experimental results are strictly based on voice authentication systems that use short term cepstrum for feature extraction. However, it remains as an open question whether it might be possible to extend similar results to other types of feature extraction techniques. Finally, in order to verify our security estimates, we carried out a number of experiments using the YOHO voice database. We first estimated that for the 138 speakers in the database only 14 bits of entropy could

be extracted. As a confirmation to this result, we outlined a practical attack that succeeded in impersonating any of the 138 people in the database with as little as 9 tries and a success probability of 0.98.

Chapter 4

Attack on Simulated Voice Authentication System

In this chapter, we develop an attack on a simulated voice authentication system. We will start by describing the type of system that we are attacking in Section 4.2 and then explain the rationale behind the attack in Section 4.3. In Section 4.4 we describe the attack in detail and outline its limitations. Finally, we show the experimental result of the attack in Section 4.5.

4.1 Introduction

In this chapter we demonstrate an attack on basic voice authentication technologies. Specifically, we show how one member of a voice database can manipulate his voice in order to gain access to resources by impersonating another member in the same database. The attack targets a voice authentication system built around parallel and independent speech recognition and speaker verification modules and assumes that adapted Gaussian Mixture Model (GMM) is used to model basic Mel-frequency cepstral coefficients (MFCC) features of speakers. We experimentally verify our attack using the YOHO database. The experiments conclude that in a database of 138 users, an attacker can impersonate anyone in the database with a 98% success probability after at most nine authorization attempts. The attack still succeeds, albeit at lower success rates, if fewer attempts are permitted. The attack is very simple to carryout and opens the door for many varients which can prove quite effective in targeting voice authentication technologies. The attack also highlights the limited amount of entropy that can be extracted from the human voice when using MFCC features.

4.2 Voice Authentication Assumptions

In the following, we list the assumptions we make on the targeted voice authentication system.

Assumption 1: Parallel Processing In the previous sections we explained that typical voice authentication systems will randomly chose a number or words and prompt the user to utter the chosen words in order to prevent replay attacks. Once the system captures the voice, it will proceed by running two parallel tasks:

1. A *speech* recognition process to insure that the speech signal corresponds to the randomly chosen text confirming freshness.

2. A *speaker* verification process to ensure the identity of the speaker.

In our attack we will assume that these two processes, speech and speaker verification, are applied in parallel. That is to say that the system will process the speech signal through a speech recognition module and a speaker verification module independently and simultaneously and will only authorize the speaker if both modules return a positive result.

Assumption 2: Basic MFCC and GMM As discussed in Section 2, the basic idea of speaker verification relies on extracting MFCC features and modeling them using a GMM. Many variants of the standard MFCC and GMM model are utilized today. For a general result we assume that the attacked system will have a speaker verification module which utilizes a standard MFCC feature extraction step followed by a standard GMM modeling step.

4.3 Attack Rationale

The strategy we follow in our attack is to synthesize a rogue speech signal that will satisfy the speaker verification module without degrading the performance of the speech recognition process too much. Since it is the center piece of our attack, we briefly review (in informal terms) the speaker verification process. In the enrollment step, a person's voice is modeled as a probability distribution over the MFCC features. The features are extracted from captured voice samples. In the speaker verification step newly captured voice samples are processed, and the resulting features are placed into the model yielding an aggregate metric that captures the

likelihood of the features extracted from the new sample coming from the same person. With more voice samples, the model becomes more accurate, in turn improving the accuracy of the likelihood predictions.

In order to capture this probability distribution a GMM model is built. Before elaborating on the attack rationale we make two observations:

1. As explained earlier a GMM model contains a number of Gaussian distributions which are trained by varying its mean, variance and weight. According to [3] the best results are achieved when GMMs are assigned fixed variances and weights and are trained by only moving around the means of the Gaussian. Essentially, the means of the Gaussians in a GMM model will capture the peaks of the modeled feature distribution.

2. In general, GMMs behave in a manner similar to any other basis system where adding more GMM components will result in a more accurate model of the distribution. This suggests that maximizing the number of components in the GMM will yield significantly better results. This hypothesis was investigated in [3] where the authors found that the equal false positive and false negative rates saw very little improvement beyond 256 components. Another important results of [3] is that increasing the number of components from 16 to 2048 improved the equal false positive and false negative rate from 20% to 10%. This means that 80% of the speakers were properly identified using a mere 16 component GMM. In essence, the general shape of the probability distribution of MFCC features will be captured with a small number of GMM components.

These observation lead us to the following hypothesis:

Given a probability distribution of an MFCC feature modeled through a GMM with a small number of components, the means of the GMM reflect the most likely

values of the MFCC feature.

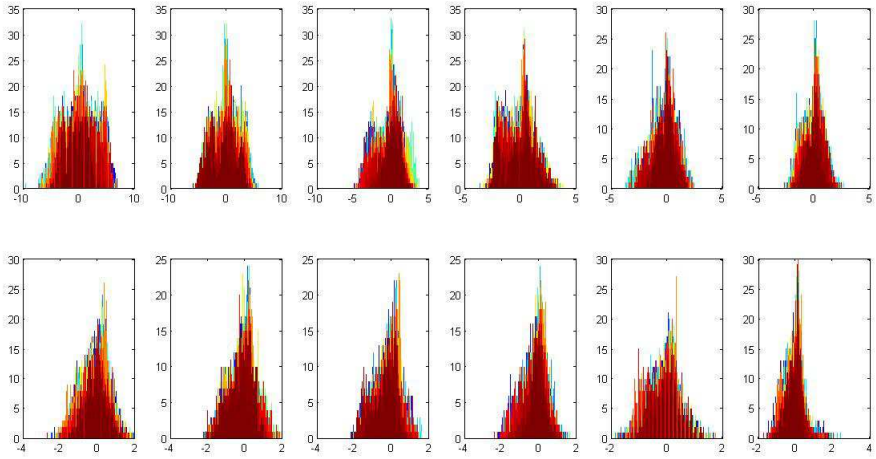

Figure 4.1: 12 MFCC features each modeled using 256 GMM components with each color representing one of 138 people.

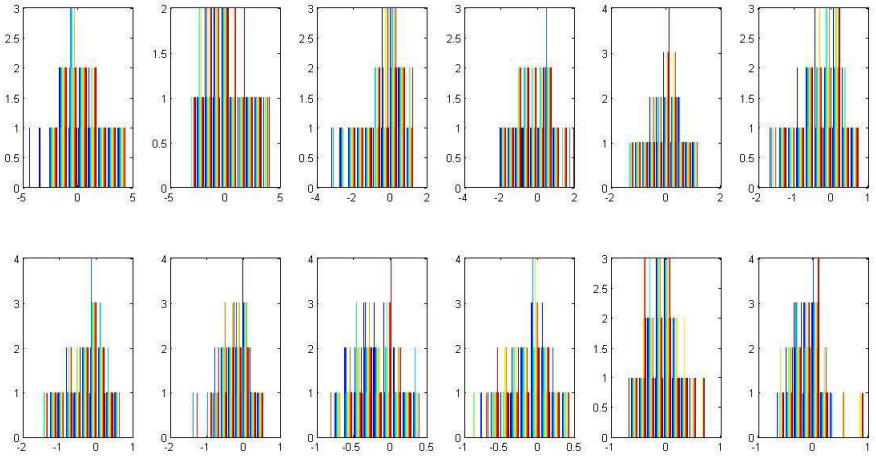

Figure 4.2: 12 MFCC features each modeled using 4 GMM components with each color representing one of 138 people.

With a lower number of components in the GMM model the training algorithm has little room to work in. Therefore, it becomes likely that the means of the

Gaussians will capture the likely values of the MFCC features. Figures 4.1 and 4.2 show the distribution for 12 MFCC components of 138 different people. Figure 4.1 uses 256 component GMM and Figure 4.2 uses 4 component GMM. It should be clear that the general shape and peaks of the distributions are preserved even when using as little as 4 components in the GMM. Simply using the means of the 4 component GMM gives a pretty accurate reflection of the peaks in the more accurate 256 component GMM.

Another observation concerning Figures 4.1 and 4.2 is the range of the features. The MFCC features do not span a large range of values which means that there will be many overlaps between the people's voice features even in a group of 138 people. Different people will have different feature distributions but with a significant overlap with other people. This is indicative of the limited amount of entropy that can be extracted from the MFCC features. This should make it clear that the means in one person's GMM will with a good probability fall into another person's MFCC distribution. This is exactly the point of weakness that our attack targets.

In general the goal of a MFCC based speaker verification system is to to test whether a given set of MFCC features belong to a specific person or not. When considering full distributions of a MFCC components belonging to two different people an overlap will occur but that will not be sufficient to create a misidentification. The nature of the Gaussian's in the GMM spread the probability on the range so that although an overlap exists it will preserve the uniqueness of every person. However, due to the observations we made earlier the means in a GMM capturing one person's features will with good probability be close to the means of a different GMM capturing the features of another person. This is where the system can be manipulated.

If a person's feature distribution is replaced with an impulse function represent-

ing one of the means in their feature GMM, then we can expect the system to pass that person as someone else with a good probability. Since the means of the GMM are close to other people's feature distribution peaks, and since we are using an impulse function to place all the concentration of the distribution on these means we will likely be a able to stimulate a misidentification. In the next section we explain this attack in more detail.

4.4 The Attack

A successful attack signal needs to pass the speaker verification *and* the speech recognition processes. To produce such a signal we create two attack signals each of which targets one of these two modules. These two attack signals will be merged later on in order to produce the final attack signal which we will refer to as the *hybrid signal*.

The first attack signal will target the speech recognition module. Creating this signal will amount to speech synthesis and therefore will be straightforward. The attacker may simply use his/her own voice to speak the challenge words provided for verification. As we discussed earlier, these signals are used to ensure the freshness of the audio signal that is fed to the system. We refer to this signal as S_1.

The second attack signal requires the creation of the MFCC impulse functions that we discussed in the previous section. More specifically, the attacker analyzes a large amount of his voice signals and then transforms them into a number of MFCC features. The attacker can then model his features using a few component GMM (in our attack we use 9-component GMM). The attacker's GMM will contain a number of means (in our case 9). Now the attacker will create a sound signal which corresponds to an impulse function centered at one of the GMM means by inverting

the GMM model for MFCC features [34]. The impulse function will correspond to a voice signal where every time frame gives rise to the same exact MFCC value (the value of the chosen mean). This means that the attacker will have several candidates for the second attack signal one corresponding to every mean in the GMM. We refer to these signals as s_2^i where $i \in [1, \ldots, n]$ where n represents the number of components in the GMM model.

In the last step of the attack we merge the two attack signals to create the final hybrid signal. There are a number of ways to merge these two signals. Our results show that the most successful method is a simple concatenation. This means that the hybrid signal will consist of the first attack signal followed immediately with the second attack signal. There is a degree of freedom here, i.e. the duration of the two signals relative to each other. In the next section we will show that the best results where achieved when the second attack signal was several times the size of the first attack signal. We refer to the hybrid signal as $H_i = [S_1|S_2^i]$.

Note that the first attack signal needs to be computed in real time due to the challenge. However, the second attack signal can be precomputed. So when attacking a live system the attacker proceeds as outlined in Table 4.1.

4.5 Experimental Results

Our experiments utilize the YOHO database which contains voice samples collected from 138 different speakers with a sampling frequency of 8 kHz [40]. Each speaker's voice is recorded reciting a random combination of three two digit numbers. For each speaker, YOHO has 4 enrollment sessions and 10 test sessions. Each enrollment session contains 24 phrases (which are roughly equivalent to 3 of minute speech) while each test session contains 4 phases (which are roughly equivalent to 20 second

Table 4.1: Steps of the proposed voice password impersonation attack.

Impersonation Attack:
1. The system will ask the attacker to say a certain word.
2. The attacker creates the S_1 signal that corresponds to him saying the given word.
3. Let $i = 1$:
4. The attacker feeds the authorization system the signal H_i. If the system accepts the voice signal then the attack has succeeded. If $i = n$ then the attack has failed.
5. Otherwise, $i = i + 1$.
6. Go back to Step 4.

speech).

We start by explaining our setup for the voice authentication system that we will be attacking.

4.5.1 Voice Authentication Setup

As explained in the previous section, our voice authentication system is composed of two parallel sub-modules, i.e. speech recognition and speaker verification. Speech recognition will be concerned with the actual speech spoken by the user. For this module we decided to use a standard library for speech recognition. This is why we used the Windows .NET Framework [52]. We treated the speech recognition module as a black box that takes in a voice signal and returns the written form of the speech input.

In the speaker verification, the voice signal is first broken into a number of overlapped 10 ms frames. Each frame goes through a hamming window length 32 ms. Then for each frame a 26 dimensional MFCC is calculated. The first 13 feature except Zero'th dimension are kept as the MFCC features. Note that first

and second derivatives of MFCC are sometimes concatenated to the last dimension of MFCC feature in order to increase entropy. However, experiments in [47] have shown that dynamic (derivative) features contribute far less to the speaker verification performance than normal features do. Furthermore, research in [50] have shown that adding derivatives of MFCC contributes very little to the overall identification performance. Therefore, our speaker verification module includes only standard MFCC features. Next, based on the 13 dimensional MFCC features, a 256 components adapted GMM is trained following [3]:

1. A 256 component universal background model λ_b is trained
2. Each person's GMM λ_i is trained by adapting only the mean vector of λ_b where i refers to the i^th person.

The verification process proceeds as follows. Given a sound signal from a person x, the MFCC components are extracted and passed through a decision function D, where

$$D_j(\text{MFCC}_x) = \log\left(\frac{p(\text{MFCC}_x|\lambda_j)}{p(\text{MFCC}_x|\lambda_b)}\right) \ . \tag{4.1}$$

Given a threshold T, if $D_j(\text{MFCC}_x) > T$ the voice originating from x is passed as the person j, otherwise the authorization fails. We set the threshold $T = 0.1$ such that it yields a false positive rate of 0.48% and false negative rate of 3.1% [1]

Note that the voice signal will pass the voice authentication if and only if it passes both the speech and the speaker verification modules.

[1]For the equal error rate (where the false positive equals the false negative) our data happens to be at 1.21%.

4.5.2 Hybrid Signal Setup

Now we introduce our hybrid signal setup. Remember that the hybrid signal refers to the signal that is used to mimic the voice of any target person. In our setup we randomly chose one ID out of the 138 speakers that are in our data set, and denote this person as x. Clearly, an attacker has access to his own voice. Therefore he will always have the ability to build $\text{MFCC}_x(W)$ representing any pass-phrase W. Since the attacker does not know the background model [2], his own GMM is simply trained by the K-mean method without background adaptation. Let us denote the mean vector of his own GMM as $m_x(i)$ where $i = 1 \ldots n$ is the index of the mean vector of the GMM.

To build up hybrid signal the attacker takes the following three steps:

1. Pick up one of $m_x(i)$,
2. Append a block of repetition of $m_x(i)$ to the last frame of MFCC_x,
3. Invert the MFCC signal to synthesize the corresponding voice signal H_i.

The hybrid signal will compose of a noisy pass-phrase recited by the attacker followed by a block of mock signal built up from $m_x(i)$. Note that the mock signal will appear as noise to the naked eye. The first part is used to pass the speech verification process while the second part is used to pass the speaker verification step.

4.5.3 Empirical Results

Two parameter values are decided in building the hybrid signals: the block length of the repetition of $m_x(i)$ denoted as q and the number of components in the attacker's GMM denoted as n. The q parameter represents the ratio of the mock signal S_2^i

[2]The attacker can build a background model from a separate dataset that he constructs. Here we just assume that he does not know the background.

Table 4.2: Success rate of Attacking with different parameters. Assume speech signal $S1$ is with a ratio of 1.

% of passing people

		# of GMM (n)									
		1	2	3	4	5	6	7	8	9	10
mock signal ratio (q)	1	0.00	0.00	0.00	0.00	4.41	6.62	6.62	6.62	5.88	5.88
	2	0.00	0.74	0.74	9.56	25.74	29.41	32.35	39.71	52.94	55.15
	3	0.00	1.47	11.76	26.47	40.44	45.59	52.21	59.56	55.15	55.15
	4	0.00	4.41	19.12	41.18	47.79	60.29	63.97	72.79	74.26	77.94
	5	0.00	8.09	29.41	55.15	56.62	70.59	72.06	80.88	88.24	90.44
	6	0.00	5.88	34.56	58.82	64.71	75.74	77.94	90.44	93.38	94.12
	7	2.21	11.03	45.59	62.50	71.32	77.94	82.35	91.91	94.12	95.59
	8	1.47	12.50	43.38	62.50	70.59	80.15	86.76	92.65	98.53	97.79

to the speech signal S_1. The larger the q parameter is, the more dominant the mock signal part is. From an authentication protocol perspective, the parameter n determines the max number of trials the attacker is allowed to make before triggering an authentication failure.

In our experiments, we applied different ratios of the mock signals. We varied q from 1 to 8. Meanwhile we varied the n parameter from 1 to 10. There were a total of 137 victims that the attacker can try and impersonate. For this, given a fixed q, for each victim the attacker tries n times, each time with a different hybrid signal H_i. Table 4.2 summarizes the results of the attack. For select q values these results are also plotted in Figure 4.3.

The results clearly demonstrate that the attacker can certainly impersonate other people in the database with a high success rate if the attack parameters are chosen carefully. At first glance it is clear that with 9 GMMs an attacker can almost certainly impersonate anyone in the database (98.5% success rate). The problem of course is that a real system might not allow as many as $n = 9$ trials. Even under such a restriction the 4 GMM scenario can produce pretty impressive results at 62% success rate. These results strongly demonstrate a sever limitation in the intrinsic

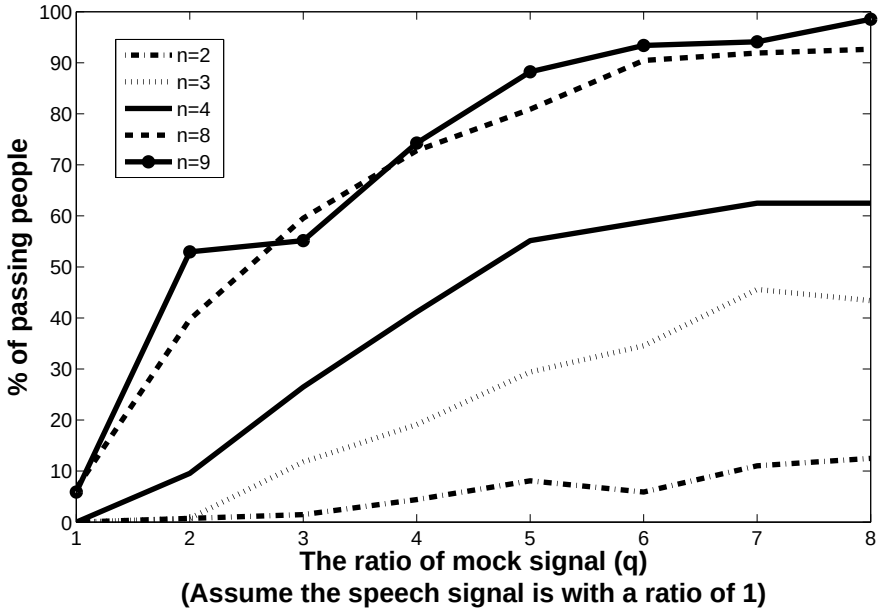

Figure 4.3: Attack success rate for select q values.

security of voice password authentication systems.

4.5.4 Limitations and Possible Improvements

In the previous sections we have outlined an attack targeting a sanitized voice password authentication system, and shared some experimental results showing the efficacy of the proposed attack. Before we draw the conclusions we would like to point out a number of limitations of the attack and briefly discuss possible improvements:

1. Our system carries out the speech recognition and speaker verification steps in parallel (Assumption 1). If the speech recognition module is applied first to the signal it might impose certain filters on speech signal thus eliminating the second part of the speech signal. Such a procedure would prevent our attack. This is in part due to the straightforward concatenation between the speech signal and the added MFCC signals. More involved steps of signal mixing can

be explored in order to strengthen our attack.

2. These results apply to a particular voice authentication system that uses standard MFCC features followed by GMM modeling (Assumption 2). While this particular setting is commonly used in practice, the specifics vary from one implementation to another. Specifically, we do not include derivative features into our assumed system. Hence the success rate when applied to an actual product will vary as well. Further work is required to assess the vulnerability, and the precise success rate for actual products in the market.

4.6 Conclusions

In this chapter we demonstrated an attack on basic voice authentication systems. We demonstrated how one member of a voice database can manipulate his voice in order to attack the other voice password accounts in the system. We demonstrated our attack using the YOHO database which contains 138 people, and we showed how an attacker can impersonate anyone in the database with a 62% success probability after at most four authorization attempts. The attack reaches a 98% success probability if up to nine authorization attempts are permitted. Our approach presents the first steps towards attacking real-world voice authentication systems.

Chapter 5

Attack on a Third-party Voice Authentication System

In this chapter, we demonstrate an attack on a third-party open-source text-independent voice authentication system. We first introduce our target system: ALIZE voice authentication system in Section 5.2. In Section 5.3, 5.4 and 5.5, we discuss the attack rationale and propose how the mock speech signal can be built. This is followed by Section 5.6 where we show experimental results of the application of our attack to the YOHO database.

5.1 Introduction

In this chapter, we mount an effective attack on a third-party open-source text-independent voice authentication system. Specifically, we show how an attacker can simply use a signal generated at fixed frequency to pass voice authentication and gain access to other user accounts. We demonstrate this attack on the ALIZE voice authentication system using the YOHO voice database. We show through experimentation that the attacker can impersonate any users in the database of 69 people with about 25% success rate with only 5 trials. The success rate can achieve more than 50% by increasing the allowed authentication attempts to 20. These success rates are significantly higher than the rates achievable by exploiting the false positive rate of the voice authentication system using random trials. Note that our attack in this chapter together with the attack in the previous chapter do not assume the attacker has any knowledge of the victim's voice. That is the main difference between our attack and synthesized speech attack mentioned in [15, 16, 17]. The experiments on YOHO database shows that the success rate of impersonating users exploiting the attack methods matches the prediction given by the theoretical entropy estimation. The theoretical entropy estimation together with the fact that our attack takes effect indicate the entropy of human voice database is limited. The limited entropy explains from information point of view, why speaker verification system is vulnerable to spoofing attack.

5.2 Open Source Software

ALIZE is an open-source voice authentication system implementing the MFCC and GMM based text-independent speaker verification algorithm [3]. It was originally invented from University of Avignon [53, 54]. Later on as mentioned in [55] and [56]

ALIZE became a baseline reference system under BioSecure Speaker Verification Benchmarking Framework. Note that BioSecure Network of Excellence was founded in 2004. More information can be found in [57] and their website [58]. The other baseline reference system adopted by BioSecure is BECARS [56, 57]. The ALIZE system can be downloaded from [59]. In this chapter we mount attack on ALIZE system.

5.3 Attack Rationale

Our attack is based on the indication that human voice has low entropy (see Chapter 3 for theoretical analysis). The speech feature of human voice concentrates in a relatively small high dimensional space because of the low entropy. Therefore it becomes possible to build a mock speech whose speech feature is similar to the feature of other users. The strategy of our attack is to synthesize a mock speech signal that satisfies the speaker model generated from the speaker verification module.

To make our point, we briefly review the speaker verification process. In the enrollment phase, a user's voice is modeled as GMM over MFCC features. In the verification phase, a newly collected voice sample is processed. The resulting MFCC feature is placed into GMM, yielding a likelihood metric measurement. If that measurement is greater than a given threshold, the new voice sample passes the authentication, otherwise it fails. Note that in both phases the MFCC (and its derivatives) is used to represent the feature of voice. Thus, later on in the context, the word "speech feature" and the MFCC are interchangeable to each other and both refer to the high dimensional vector containing MFCC (and its derivatives). To simplify the concept, we can view the MFCC as orderless high dimensional random variable (r.v.). We denote such a r.v. as C. The Gaussian Mixture Model captures

the probability density function (pdf) of the r.v. C . We denote the Gaussian Mixture Density function in Equation 2.2 as $G(C)$, specifically

$$G_i(C) = p(C|\lambda_i) \ . \tag{5.1}$$

Here Gi(C) denotes the Gaussian mixture density function defined with parmeter λ_i.

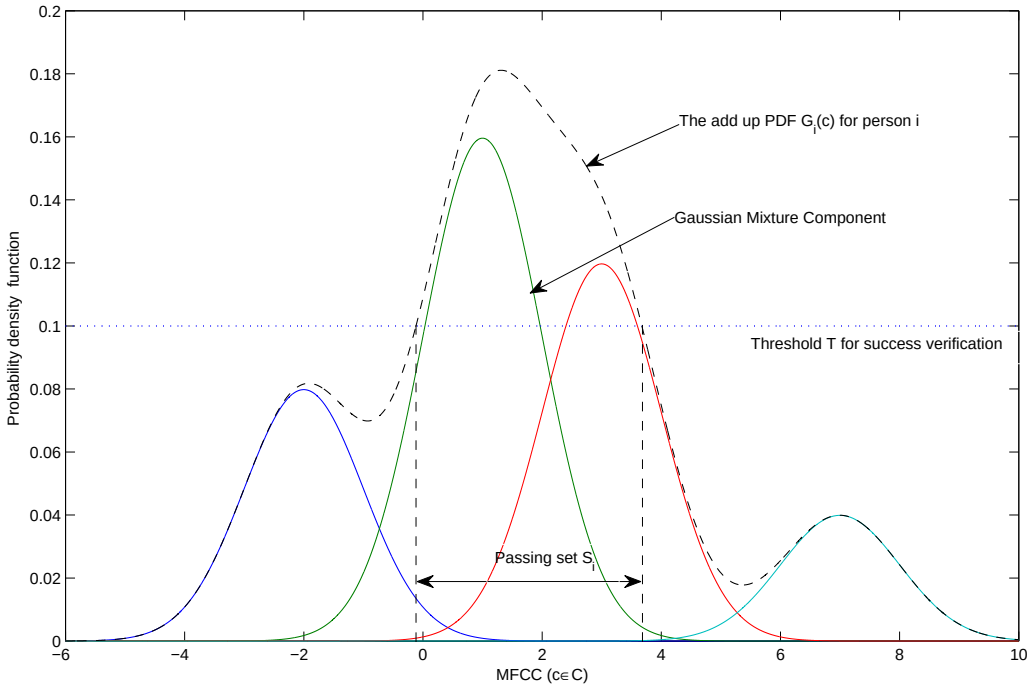

Figure 5.1: Demonstration of point-wise verification process of GMM based voice authentication.

Now we discuss the rationale behind the attack. At first glance, the attacker has to retrieve the whole distribution of a voice feature in order to impersonate other users. However, we claim that this will not be necessary. We now explain this by examining the verification process. Assume in the verification phase, the system is judging whether or not a fresh MFCC is extracted from the user i who

has previously enrolled GMM template $G_i(C)$. The following two observations can be made:

1. Based on Equation 2.3 Equation 2.4 the overall likelihood score measures the geometric mean of point-wise "distance" from each of the MFCC points to the template GMM $G_i(C)$. A successful verification can be expected for sure in case that each point-wise measurement yields a likelihood greater than T (the given threshold in Equation 2.4).

2. For the point-wise measurement, i.e. consider only a single point of MFCC in each measurement, the possible value of MFCC to pass the verification is not determined by the entire distribution $G_i(C)$, but a range S_i of C, where the corresponding probability $G_i(C)$ is greater than the threshold T. In other words, any value within range S_i would lead to a successful point-wise verification (see demonstration in Figure 5.1). Furthermore, based on Observation 1, any collection from S_i can be used as voice feature to pass the verification as the user i.

The above observations indicate that the attacker could impersonate the user i by picking up mock speech feature from any value in the range of S_i. We formalize the attack rationale to the following proposition:

Proposition 4. *In GMM based speaker verification [3], given a threshold T and a speaker model GMM $G_i(C)$ capturing the distribution of i^{th} user's voice feature, we define the passing set of the user i as $S_i = \{c : G_i(c) > T\}$. If the attacker can find any subset $c_a \subset S_i$, the attacker can impersonate the user i using the voice feature built from c_a.*

Proof. Assume in the feature domain, c_a is known. Then an n-frame mock voice feature M can be built up. Here $M = \{c_1, c_2, \ldots, c_n\}$ where $c_i \in c_a$ for all i and n

is an arbitrary positive integer number. Since $c_a \subset S_i = \{c : G_i(c) > T\}$ we have $G_i(c_k) > T$ for all k. Furthermore, from Equation 2.3 and Equation 5.1 we derive that $q(M|\lambda_i) = (\prod_{k=1}^{n} G_i(c_k))^{\frac{1}{n}} > T$. According to Equation 2.4, the mock voice feature M will pass the authentication as the user i. □

Now we demonstrate a visualized explanation for Proposition 4 with Figure 5.1. On this one dimensional example, the model of voice feature (the MFCC) is the probability density function (the black dash curve) built up by 4 equal weight Gaussians (the colorful solid curves). Assuming the threshold is T, in case that the voice feature contains only one point and the feature point falls within the range of S_i, such voice feature will definitely lead to a successful verification. In case that the voice feature contains more than one points and all the feature points fall within the range of S_i, such voice feature will also for sure pass the verification.

Proposition 4 simplifies the attack from acquiring an entire distribution $G_i(C)$ of a voice feature to searching for a few points c_a. Note that this proposition applies to both GMM and GMM-UBM systems. In case of GMM-UBM system, the Equation 5.1 is substitute by

$$G_i(C) = \frac{p(C|\lambda_i)}{p(C|\lambda_b)}$$

where λ_b is the universal background model. Again, due to the low entropy of human voice, c_a should not be hard to find. To synthesize a mock speech signal, we can go from two directions: either find c_a first and invert it back to time domain, due to the invertible property of MFCC (see Section 2.2.4), or build a time domain signal that yields c_a. In the following sections we will discuss how to generate attack signals in detail.

5.4 Review of an Earlier Attack

In this section, we briefly review the attack introduced in [51] which defines a voice authentication systems following the GMM based speaker verification technology specified in [3]. The attack works by building a mock MFCC and then inverting it back to time domain. The attack assumes that the attacker has access to his/her own speech samples and therefore can produce the GMM from his/her own speech feature. Next, a mock MFCC is built by repeating one of the means extracted from the attacker's GMM. Finally, the mock speech signal is generated by inverting the mock MFCC signal back to time domain. Note that each of the means of the GMM is considered as a candidate point of c_a. Thus with an N component GMM, the attacker would have up to N candidate mock speech signals, i.e. the attacker may try the authenticate up to N times. The experiments in [51] showed that the attacker can impersonate more than half of the voice in the target database with $N = 4$ and can impersonate almost everyone with $N = 9$.

A possible reason why the mean of the GMM can be good candidates of c_a is because of the low entropy of voice. The same reference, i.e. [51], provides the following explanation: Given a probability distribution of an MFCC feature modeled through a GMM with a small number of components, the means of the GMM reflect the most likely values of the MFCC feature. The claim is based on two observations in [51]:

1. MFCC features of different people are similar to each other and therefore highly overlapped.

2. Even with only a few components the GMM captures the general shape of a MFCC distribution.

However, this attack does not apply to ALIZE authentication system, due to an

additional normalization step in MFCC processing.

5.5 Attack Method Targeting ALIZE: From Time Domain Signal to MFCC

In this section we propose a successful attack method targeting the ALIZE authentication system. We start from examining what prevents the previous method in Section 5.4 from working on the ALIZE. In the ALIZE the MFCC processing step is implemented exactly in the same way as we described in Section 2.2.4 except that there are two more additional steps: energy detection and normalization. The energy detection is used for silence removal. The bottom $N\%$ of low energy frames is considered as background noise and therefore removed. The normalization deals with the environmental mismatch [1]. Given the MFCC feature $c \in C$, the normalization step is processed by

$$c_n = \frac{c - E(c)}{Var(c)} \ . \tag{5.2}$$

Where the $E(c)$ and $Var(c)$ are mean and variance of the original MFCC. The normalized MFCC will have zero means and one variance for each dimension.

These two new steps compared with the system in [51] change the value of desired mock MFCC and therefore cause the failure of applying attack method in Section 2.2.4 to ALIZE. Especially, the normalization step treats the constant-like mock MFCC [2] as DC offset and reset the mock MFCC back to zero. The normalization step makes it harder to convert MFCC back to time signal.

Instead of finding a de-normalization step, we present an alternative way of

[1] Note that normalization may not be the necessary step in case of clean environment databases such as YOHO database the one we used in the experiment.

[2] Remember the mock MFCC is a repetition of a means of a GMM in [51]

building the attack signal. That is we explore time domain signals which have decent probability yielding a mock MFCC $\in c_a$. Remember in [51], the mock MFCC was generated by the repetition of mean of Gaussians. The reason the authors did this was because they wanted a MFCC feature with repetition of a simple pattern which contains as few different values as possible so as to increase the chance of hitting the set S_i (see Proposition 4 for detail). Similarly, we want our time domain mock speech signal maps to a simple MFCC pattern. Based on this criteria, sine wave may be a good candidate. Since it maps to an impulse function in frequency domain the sine wave leads to a simple pattern in MFCC domain. To deal with the energy detection we apply Gaussian modulation to the pure sine wave. Finally, our mock speech signal appears to be a Gaussian modulated sine wave with a form

$$y(t) = \cos(2\pi f_c t) e^{-\frac{t^2}{\tau^2}} \quad . \tag{5.3}$$

Here f_c corresponds to the frequency of sine wave, and τ determines the width of the pulse in time domain. The possible frequencies for f_c in the pulse ranges from 0 to $\frac{f_s}{2}$ where the f_s is the sample frequency. For each mock speech, the frequency is fixed. This mock speech signal overcomes the normalization problem in ALIZE software. In the next session we will demonstrate the performance of the attack.

5.6 Experimental Results

We now demonstrate our experimental results. We start by discussing target system selection and the voice database selection. Then, we describe the parameter setup for both the target system and the mock speech signal. Next, we show the attack performance. Finally, we conclude this section with discussions.

5.6.1 Target System Selection – Why ALIZE

An ideal target system should be a well known commercial voice authentication system. However, the detail implementation of such commercial system can not be found easily. Instead, we found ALIZE. The reasons why we choose ALIZE are as following: first, it is an open-source software. We can know exactly how each step works inside the system. Second, it is quality guaranteed and thoroughly tested. Related to which, many research papers have been published. Third, it is well documented. Besides research publications, AIZE has user guide which can be downloaded along with the software. More detail of ALIZE can be found in Section 5.2.

5.6.2 Database Selection

Our experiments utilize the YOHO database which contains voice samples collected from 138 different speakers with a sampling frequency of 8 kHz [40]. Each speaker's voice is recorded reciting a random combination of three two digit numbers. For each speaker, YOHO has 4 enrollment sessions and 10 test sessions. Each enrollment session contains 24 phrases (which are roughly equivalent to 3 of minute speech) while each test session contains 4 phrases (which are roughly equivalent to 20 second speech).

Note that in the provided document in [59], ALIZE is evaluated under database BANCA and NIST 2005. We did not use these two databases because both of these two databases are unconstraint speeches yielding high Equal Error Rates (EER) around 10% which in turn gives high false positive rates. Such an high equal error rate itself gives too much advantages to the attacker. Thus, instead, we use another NIST database YOHO which constrains the speech content to digital numbers. Such

that it is more robust and secure. We keep the parameter selection in most case in the same way as the selections targeting NIST 2005 in [59], unless there were necessary changes.

5.6.3 ALIZE Setup

We now describe the parameter setup for ALIZE system with YOHO database. In ALIZE, the voice signal is first broken into a number of overlapped 10 ms frames. Each frame goes through a hamming window length 20 ms. Then for each frame a 24 dimensional MFCC is calculated. The first 16 feature except Zero'th dimension as long as their first derivatives are kept as the MFCC features. Note that the signal energy and derivatives of the energy are also appended to the feature. The final voice feature contains totally 34 dimensions (16 MFCC, 16 delta MFCC, 1 energy and 1 delta energy). The energy detection step follows this step. Based the energy dimension, the frames are classified to two clusters. The cluster corresponds to the high energy is kept as speech frames while the cluster corresponds to the low energy is discarded. Note that after this step the energy dimension is discarded, i.e. the voice feature contains only 33 dimensions in the training and the verification session. Next, the speech frames is normalized by Equation 5.2. After that, a 512 components adapted GMM is trained following [3]:

1. A 512 component universal background model λ_b is trained
2. Each user's GMM λ_i is trained by adapting only the mean vector of λ_b where i refers to the i^th user.

The verification process proceeds as follows. Given a sound signal from a user x, the MFCC components are extracted and passed through a decision function D, where

$$D_j(\text{MFCC}_x) = \log\left(\frac{p(\text{MFCC}_x|\lambda_j)}{p(\text{MFCC}_x|\lambda_b)}\right) \ . \tag{5.4}$$

Given a threshold T, if $D_j(\text{MFCC}_x) > T$ the voice originating from x is passed as the user j, otherwise the authorization fails.

We set the threshold $T = 0.5$ such that it yields a false positive rate of 0.44% and false negative rate of 1.56%. [3]

5.6.4 Mock Signal Setup

The mock speech signal is built up from fixed frequency Gaussian-modulated sinusoidal pulse which is given by

$$y(t) = \cos(2\pi f_c t) e^{-\frac{c_0 t^2}{(b_w f_c)^2}} \ . \tag{5.5}$$

Here, $c_0 > 0$ is a constant, f_c is the center frequency and b_w is the fractional bandwidth. The center frequency f_c varies from 1 to 4000 Hz. We select the fractional bandwidth as $b_w = \frac{200}{f_c}$ such that all pulses will have same time length and one pulse takes effect within 20 ms which equals to the length of windowed frame in the ALIZE setup. We repeatedly generate the pulse every 100 ms, such that pulses do not interfere with each other during the windowing process. Figure 5.2 and Figure 5.3 show a typical waveform of mock signal in time domain.

[3] For the equal error rate (where the false positive equals the false negative) our data happens to be at 0.94%.

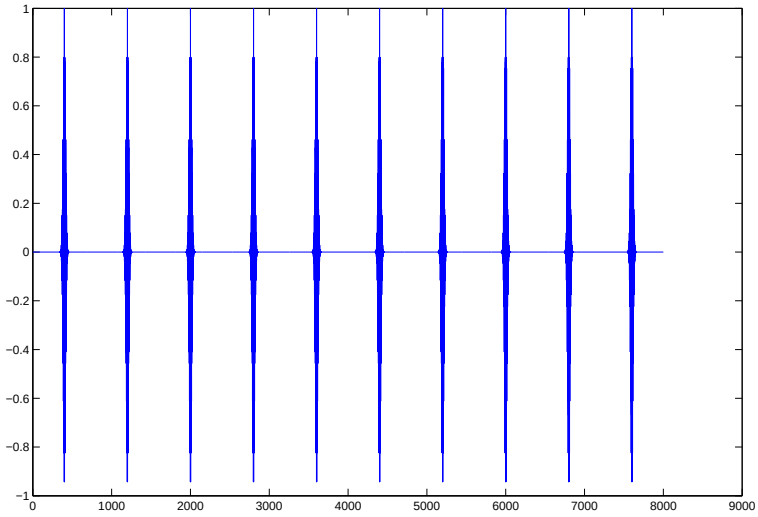

Figure 5.2: Example of mock speech.

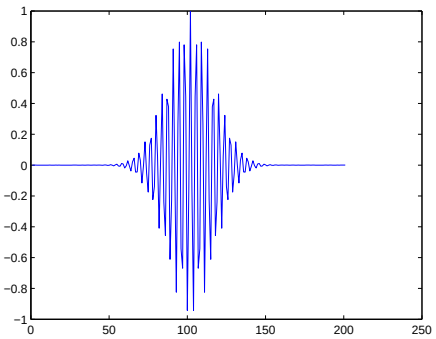

Figure 5.3: Example of mock speech (zooming in one pulse).

5.6.5 Empirical Results

Due to the existence of the false positive rate of the system, one may pass the authentication as someone else with small probability. With huge amount of trials of different voice samples collected from random users, the attacker will eventually successfully impersonate a designated user and get access to his account. The success

rate of this naive brute-force attack is given by

$$S_r = 1 - (1 - f_n)^N \tag{5.6}$$

depending on false positive rate f_n and number of trials N. We name such an naive attack as *random test*. Note that in our system setup the f_n is 0.44%. We will show in the following experiments our attack scheme performs overwhelmingly better than the random test.

We now test the success impersonation rate of all mock speech signals with the center frequencies ranging from 1 to 4000 Hz. As shown in Figure 5.4, in most cases of all 4000 frequencies, our mock signals achieve higher success rate than random test. Figure 5.5 shows the number of times that each speaker in the database is impersonated by the mock speech signals. We can also see from Figure 5.5 that in case of 97 speaker IDs out of 138, our mock signals pass the verification more times than random test. Based on Figure 5.5, 89.9% of the users in the database can be impersonated at least once after trials of all 4000 frequencies. Meanwhile, on average the mock signal succeeded 110 times out of 4000 trials which is equivalent to 2.76% success rate with one trial. The 2.76% success rate outperforms the rate of random test which equals 0.44% with one trial. The experiment so far indicates that our mock speech signals built from 4000 various Hz perform in overall better than random test. However with 4000 trials, the random test itself can almost get access to any designated account. The coming up question is if we reduce number of trials, i.e. building less mock signals, can our attack scheme still achieve a better success rate than random test does?

To answer the question above, we next show that with a few number of trials our attack will still successfully get access to a decent number of accounts. To do

this we need obtain some *pre-knowledge* that which frequencies are statistically more likely to pass the verification. Such a job will be easy, if Figure 5.4 is given to the attacker in advance. For instance, we can see from Figure 5.4 that some frequencies around 2300 Hz have more than 10% of chance to mimic the users' voice in the database. Meanwhile, frequencies around 500 Hz, 1150 Hz and 3400 Hz also have decent probability to impersonate users in the database. Knowing this, we would start to try our luck with the mock signal built from 2300 Hz. Then we would try 500 Hz, 1150 Hz and 3400 Hz consequentially, so on and so forth. Although in the reality, the attacker is not allowed to get access to the voice database of the target system, he can collect his own database and obtaining the "pre-knowledge" from his own database.

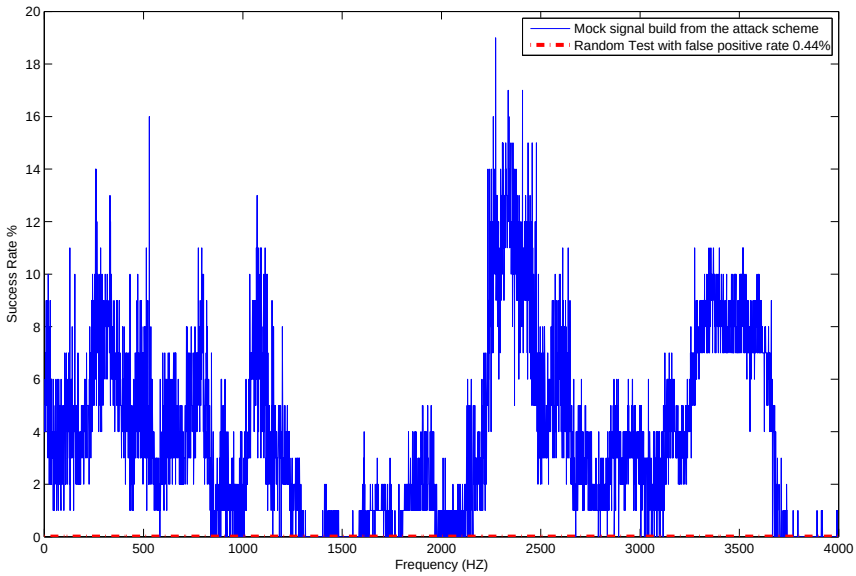

Figure 5.4: Success rate of impersonating users using fixed frequencies ranging from 1 to 4000 Hz on YOHO database with 138 users.

In the next experiment, we assume the attacker has his own voice database[4].

[4]Note that since human voice is relatively easy to collect, it's reasonable to make such an assumption

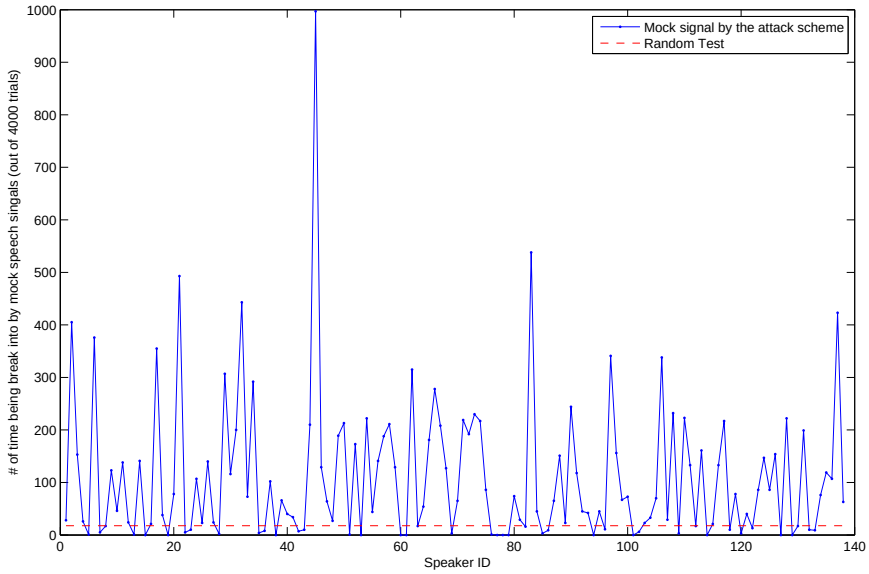

Figure 5.5: Number of times that each speaker is impersonated by mock speech signals testing on YOHO database with 138 people.

The attacker then can better select candidate signal out of 4000 frequencies with the help of his own database. We name this attack as *selected frequency attack*. To simulate this, we divide the YOHO database into two equal size subsets DB_1 and DB_2, each containing 69 speakers. We assume DB_1 belong to the attackers and used for training the "pre-knowledge" while DB_2 acts as database of the target system. To get the "pre-knowledge", we build up a matrix M_p based on the database DB_1. Here in M_p, the row indexes the speaker ID in DB_1; the column indexes the frequencies of the mock speech signals (i.e. the ID of mock speech signal). In the attacker's training process, we set $M_p(i, j) = 1$ whenever the mock speech built from j^{th} frequency successfully impersonate the i^{th} user in DB_1. Note that the matrix M_p reflect statistically which frequencies are more likely to mimic the users' voice in the database. We then mount the attack to the ALIZE system which uses database DB_2. Having the "pre-knowledge" matrix M_p, we select candidate signals

Impersonation Attack using selected frequencies:

1. Select $\hat{j}^{th}$ mock speech signal such that $\hat{j} = \arg(\max_j(\sum_i M_p(i,j)))$.
2. Find all $\hat{i}$ such that $M_p(\hat{i},\hat{j}) = 1$. Then set $M_p(\hat{i},j) = 0$ for all j, set $M_p(i,\hat{j}) = 0$ for all i.
3. Go back to step 1 unless $M_p(i,j) = 0$ for all i and j.

Figure 5.6: Steps of selecting mock speech signal based on M_p.

using the algorithm as shown in Figure 5.6. We then repeat the same experiment by exchanging the roles of DB_1 and DB_2, i.e. DB_2 acts as the attacker's database while DB_1 becomes ALIZE's database in the second experiment.

Figure 5.7 gives the success rate of the attack versus number of trials the attacker attempts. As we can see in Figure 5.7, using our attack method, around 25% of the accounts are compromised within 5 trials compared to 2.18% success rate through random test with same amount of trials. If the number of allowed trials is rised up to 20, more than 50% of the accounts are compromised while the random test can only impersonate 8.44% of users with 20 trials.

5.6.6 Discussions

Now we relate the result in this chapter to the entropy estimation in Chapter 3 and the simulated attack in Chapter 4. We see certain performance degradation from Table 4.2 to Figure 5.7, explained as follows:

1. ALIZE uses 34 dimensions of voice features (including delta MFCC), while assumed authentication in Chapter 4 uses 12 dimensions of voice feature. The increasing of dimensions adds up entropy. However, as explained in [51], delta MFCC will not add too much entropy since they are highly dependent of MFCC. That explains why we need more trials to mount a successful attack

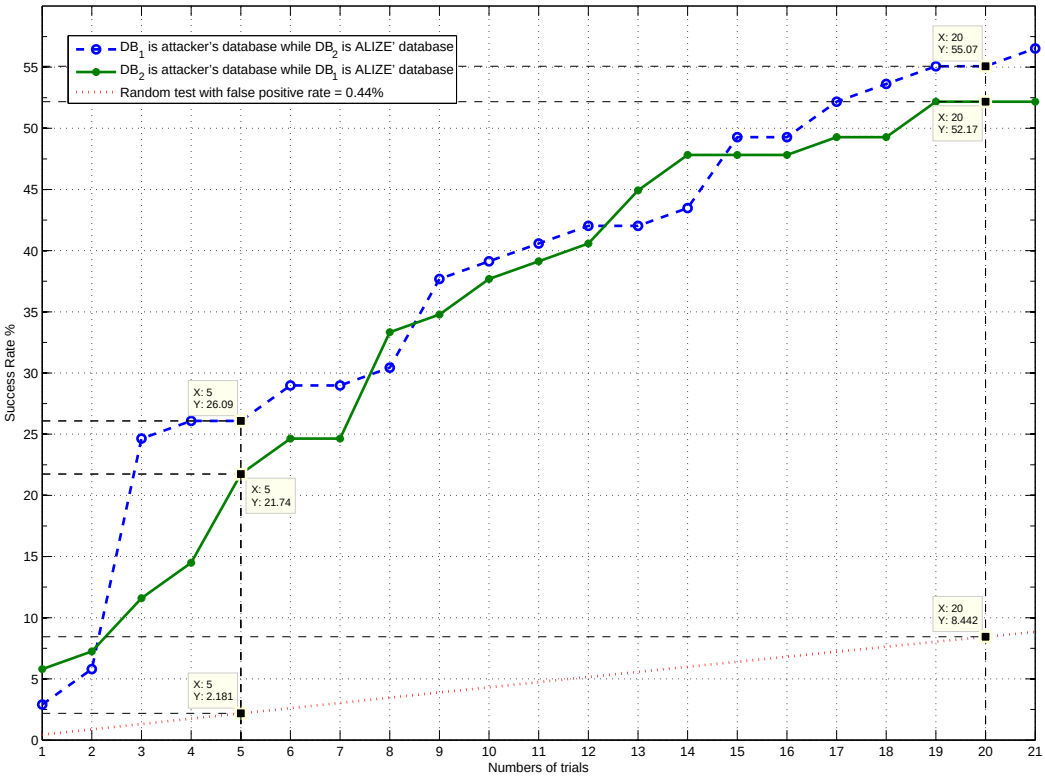

Figure 5.7: Selected frequency attack assuming the attacker makes usage of his own voice database.

to ALIZE while the number of trials does not increase tremendously compared with [51].

2. ALIZE applying normalization to voice feature in a way that makes it hard to build mock signal from MFCC domain (see details in Section 5.5). Alternatively, we propose a possible attack that builds signals from time domain using various fixed frequencies.

3. The proposed attack in this chapter may not be the optimized one. Different from the means of GMM which are good representation of c_a in MFCC domain due to the similarity of GMMs, there is no direct connection from the frequency

signal to c_a. Each specific frequency signal in time domain forms a unique pattern in MFCC domain. We search blindly among frequencies to find a pattern that coincidentally matches to some points in c_a.

The fact that we can still find such a match with only a few trials is an empirical confirmation of the analysis in Chapter 3 that the entropy of human voice is low. As shown in Figure 5.7 the attack signal can impersonate about 25% of users with only 5 trials. As the number of allowed trials is increased to 20, the impersonation rate can achieve more than 50% success. The attack is significantly more effective than random test which yields success rates of 2.18% and 8.44% with 5 and 20 trials, respectively.

5.7 Conclusions

In this chapter, we demonstrated an attack on the ALIZE voice authentication systems. We showed how an attacker can use fixed frequency stimulus to build mock speech signals which can pass the voice authentication with a limited number of trials. We demonstrated the attack on ALIZE voice authentication system using the YOHO database. The attack achieves more than 50% success rate of impersonating any users in the database after at most 20 authentication attempts. The attack can still succeed with about 25% success rate if only 5 attempts are allowed. The comparison of success rate between the attack and the random test shows the effectiveness of our attack.

Chapter 6

Further Discussion

6.1 How Does Our Work Relate to State-Of-the-Art Systems?

Since the introduction of GMM-UBM more than a decade ago, many new techniques have appeared in the literature gaining popularity over GMM-UBM. Joint Factor Analysis (JFA) [60, 6, 7] and I-Vector [9] are representatives of state-of-the-art text-independent speaker verification systems. However, these systems are still tightly related to GMM-UBM.

Joint Factor Analysis (JFA) [8] is an extension of GMM. The JFA models inter-session variability which is mainly contributed by channel variability and intra-speaker variability. Whereas GMM-UBM only models the intra-speaker variability. Suppose F represents the number of dimensions of voice features, e.g. MFCC and N stands for the number of Gaussian components in a universal background model. The JFA model defines a super-vector M with dimension of $F \times N$ by concatenating the F dimensional mean vectors in the GMM of a given speech. The super-vector M is the composition of the UBM mean vector $\mathbf{m}$, speaker factors $\mathbf{v}y + \mathbf{d}z$ and the

channel factor $\mathbf{u}x$

$$M = \mathbf{m} + \mathbf{v}y + \mathbf{d}z + \mathbf{u}x \ . \tag{6.1}$$

JFA can be viewed a less noisy version of GMM-UBM. If we set $\mathbf{u} = 0$, the super-vector M is very close to the adapted GMM of a speaker. From the entropy point of view, since GMM-UBM is more noisy (i.e. the adapted GMM counts the channel variability into speaker variability), the entropy estimation result on GMM-UBM should be higher than JFA, therefore, the estimation on GMM-UBM will yield an entropy upper-bound. Furthermore, the scoring process of JFA is similar to GMM-UBM. The likelihood scoring method used by GMM-UBM can be directly applied to JFA [61, 62]. In this sense, the attack rationale in our dissertation also applies to the JFA based system. Moreover, it may not be necessary to use JFA in our experiment setup. The overall performance of JFA is better than GMM-UBM because JFA compensates channel variability. However, in case that all speakers' voice samples are recorded by the same device in a clean environment, as in the case of the YOHO database that we are using, JFA will have no significant advantage over GMM-UBM as the term u is close to 0.

The I-Vector [39] is based on JFA. Similar to JFA the I-Vector represents the super-vector M as

$$M = \mathbf{m} + \mathbf{T}w \ , \tag{6.2}$$

where $\mathbf{T}$ is the total variability matrix with low rank and w is a random vector, i.e. the I-Vector characterizing the inter-speaker variability. The angle between pre-enrolled I-Vector and target I-vector represents the similarity score. Since there are no channel related terms, extra channel compensation steps are needed before scoring. Popular compensation methods include WCCN, LDA plus WCCN [9].

The I-Vector extracts more compressed information from GMM. A simple mod-

ification on the attack rationale in Section 5.3 can make it fit to the case of the I-Vector. Instead of using Proposition 4, the attacker needs to find a signal representation leading to a I-Vector who has an angle within the range of $\theta - \delta$ to $\theta + \delta$ where θ is the angle of the pre-enrolled I-Vector belonging to a victim. The δ is the threshold to determine whether or not the verification will succeed. If the entropy of voice is low, such a signal representation will be easy to find.

From GMM to JFA to I-Vector, the voice characters become more precise and more compact. However, our work shows from both theoretical and practical views that the entropy of the Voice Password Databases is limited. The limited entropy which enables our attack on GMM-UBM is an intrinsic property of the speech signal still present even if JFA or an I-Vector based system is employed. The experiments in [39] complement our attack. As seen in Table 2 of [39] the 2 second short utterance in testing yields an incredibly high equal error rate no matter how long the training sessions are. This experiment can also be viewed as an "attack" to the speaker verification system. The attack signal is the 2 second short utterance. This results in [39] shows JFA and I-Vector based systems are also vulnerable to an attack similar to the one presented in this dissertation due to the limited entropy of human voice.

6.2 Utilization of Voice Passwords

The experiment of entropy estimation in Chapter 3 and the attacks in Chapter 4 and 5 are indications that the entropy of the text-independent human voice is limited. One needs to be very careful when they want to use the human voice for the authentication task.

The experiments in Chapter 3 show that the entropy of voice feature extracted from the YOHO database containing 138 users is less than 14 bits. The 14 bits is the

amount of entropy that a 4-digits number password can provide. Voice passwords can be applied to the place where a weak password is acceptable. For example, the voice passwords can be a substitution of 4 digits lock code on Iphones.

The strength of the text-independent voice password can be reinforced by taking into consideration the content of the speech i.e. using text-dependent voice authentication. In such case, in the enrollment phase, the user has to remember a passphrase and record his passphrase along with his voice. In the verification phase, the user needs to recite the same passphrase. If and only if the voice and the passphrase both match, the user can pass the authentication. We now estimate the entropy of such a text-dependent voice password. Assume the entropy provided by the text-independent voice is n_v bits and the length of the passphrase is c. Each character of the passphrase composes of letter a to z and digit 0 to 9. As a result the entropy that the password can provide is $c \log_2 36$. The total entropy of such a text-dependent password is $n_v + c \log_2 36$ bits. Assuming the $n_v = 14$ and $c = 4$, the entropy of such voice password is about 34 bits. A higher entropy bits can be reached by using longer passphrase. Similarly, instead of using passphrase, we may combine other biometric sources with voice. The security level of such a fusion system will be strengthened. The total entropy of such a fusion system is $\sum_{i=1}^{N} n_i$ where the N is the number of biometric sources and n_i is the entropy of the i^{th} source.

Chapter 7

Conclusion

In this dissertation, we estimated the security level of voice password databases. We reviewed an authentication model i.e. the MFCC and GMM based authentication model that captures what is nowadays used in many voice password products. Next, we developed a theoretical framework based on the notion of relative entropy in order to estimate the entropy of a voice database. Using the relative entropy estimates we were able to estimate the security level offered by a voice authentication system relying on a voice database. In order to verify our security estimates, we carried out a number of experiments using the YOHO voice database. We showed with experiments that the number of entropy which can be extracted from the YOHO database which contains 138 speakers is no more than 14 bits. To confirm our theoretical entropy estimation, we propose an attack on the MFCC and GMM based basic voice authentication systems. We demonstrated our attack using the YOHO database and showed how an attacker can impersonate anyone in the database with a 62% success probability after at most four authorization attempts. The attack reaches a 98% success probability if up to nine authorization attempts are permitted. Furthermore, we demonstrated an attack on a third-party system, the ALIZE

voice authentication systems. We showed how an attacker may use fixed frequency stimulus to build mock speech signals which passes the voice authentication with a limited number of attempts. Again, we demonstrated the attack using the YOHO database and showed the attack can achieve more than 50% success rate of impersonating any users in the database after at most 20 authentication trials. The attack can still maintain a success rate of about 25% if only 5 attempts are allowed. The comparison of success rate between the attack and the random test shows the effectiveness of our attack. With these, we take an initial step towards evaluating voice password databases from a security perspective.

Bibliography

[1] Dan Miller and Derek Top. Voice biometrics 2010: A transformative year for voice-based authentication, May 2010.

[2] D.A. Reynolds and R.C. Rose. Robust text-independent speaker identification using Gaussian mixture speaker models. *IEEE Transactions on speech and audio processing*, 3(1):72–83, 1995.

[3] D.A. Reynolds, T.F. Quatieri, and R.B. Dunn. Speaker verification using adapted Gaussian mixture models. *Digital signal processing*, 10(1-3):19–41, 2000.

[4] N. Krause and R. Gazit. SVM-based Speaker Classification in the GMM Models Space. In *Speaker and Language Recognition Workshop, 2006. IEEE Odyssey 2006: The*, pages 1–5. IEEE, 2006.

[5] WM Campbell, DE Sturim, and DA Reynolds. Support vector machines using GMM supervectors for speaker verification. *IEEE Signal Processing Letters*, 13(5):308–311, 2006.

[6] P. Kenny, G. Boulianne, P. Ouellet, and P. Dumouchel. Joint factor analysis versus eigenchannels in speaker recognition. *Audio, Speech, and Language Processing, IEEE Transactions on*, 15(4):1435–1447, 2007.

[7] P. Kenny, P. Ouellet, N. Dehak, V. Gupta, and P. Dumouchel. A study of interspeaker variability in speaker verification. *Audio, Speech, and Language Processing, IEEE Transactions on*, 16(5):980–988, 2008.

[8] P. Kenny. Joint factor analysis of speaker and session variability: Theory and algorithms. *CRIM, Montreal,(Report) CRIM-06/08-13*, 2005.

[9] Najim Dehak, Reda Dehak, James Glass, Douglas Reynolds, and Patrick Kenny. Cosine similarity scoring without score normalization techniques. In *Proc. Odyssey Speaker and Language Recognition Workshop*, pages 71–75, 2010.

[10] U. Uludag, S. Pankanti, S. Prabhakar, and A.K. Jain. Biometric cryptosystems: issues and challenges. *Proceedings of the IEEE*, 92(6):948–960, 2004.

[11] L. Ballard, F. Monrose, and D. Lopresti. Biometric authentication revisited: Understanding the impact of wolves in sheepos clothing. In *15th Annual USENIX Security Symposium*, pages 29–41, 2006.

[12] L. Ballard, S. Kamara, and M.K. Reiter. The practical subtleties of biometric key generation. In *Proceedings of the 17th conference on Security symposium*, pages 61–74. USENIX Association, 2008.

[13] H. Beigi. Effects of time lapse on speaker recognition results. In *Proceedings of the 16th international conference on Digital Signal Processing*, pages 1260–1265. Institute of Electrical and Electronics Engineers Inc., The, 2009.

[14] K.P.H. Sullivan and J. Pelecanos. Revisiting Carl Bildt's Impostor: Would a Speaker Verification System Foil Him? In *Proceedings of the Third International Conference on Audio-and Video-Based Biometric Person Authentication*, page 149. Springer-Verlag, 2001.

[15] Driss Matrouf, J-F Bonastre, and Corinne Fredouille. Effect of speech transformation on impostor acceptance. In *Acoustics, Speech and Signal Processing, 2006. ICASSP 2006 Proceedings. 2006 IEEE International Conference on*. IEEE, 2006.

[16] Jean-François Bonastre, Driss Matrouf, and Corinne Fredouille. Artificial impostor voice transformation effects on false acceptance rates. In *Eighth Annual Conference of the International Speech Communication Association*, 2007.

[17] Tomi Kinnunen, Zhi-Zheng Wu, Kong Aik Lee, Filip Sedlak, Eng Siong Chng, and Haizhou Li. Vulnerability of speaker verification systems against voice conversion spoofing attacks: The case of telephone speech. In *Acoustics, Speech and Signal Processing (ICASSP), 2012 IEEE International Conference on*, pages 4401–4404. IEEE, 2012.

[18] Q. Jin, A.R. Toth, A.W. Black, and T. Schultz. Is voice transformation a threat to speaker identification? In *Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP'08)*, pages 4845–4848. Citeseer, 2008.

[19] Zhizheng Wu, Eng Siong Chng, and Haizhou Li. Detecting converted speech and natural speech for anti-spoofing attack in speaker recognition. In *INTERSPEECH 2012, 13th Annual Conference of the International Speech Communication Association*, 2012.

[20] Federico Alegre, Ravichander Vipperla, Nicholas Evans, et al. Spoofing countermeasures for the protection of automatic speaker recognition systems against attacks with artificial signals. In *INTERSPEECH 2012, 13th Annual Conference of the International Speech Communication Association*, 2012.

[21] F. Bimbot, J.F. Bonastre, C. Fredouille, G. Gravier, I. Magrin-Chagnolleau, S. Meignier, T. Merlin, J. Ortega-García, D. Petrovska-Delacrétaz, and D.A. Reynolds. A tutorial on text-independent speaker verification. *EURASIP Journal on Applied Signal Processing*, 4:430–451, 2004.

[22] Tomi Kinnunen and Haizhou Li. An overview of text-independent speaker recognition: From features to supervectors. *Speech communication*, 52(1):12–40, 2010.

[23] S. Furui. 40 Years of Progress in Automatic Speaker Recognition. In *Proceedings of the Third International Conference on Advances in Biometrics*, pages 1050–1059. Springer-Verlag, 2009.

[24] T. Ganchev, N. Fakotakis, and G. Kokkinakis. Comparative evaluation of various MFCC implementations on the speaker verification task. In *Proceedings of the SPECOM*, volume 1, pages 191–194. Citeseer, 2005.

[25] T. Ganchev, N. Fakotakis, and G. Kokkinakis. Comparative evaluation of various MFCC implementations on the speaker verification task. In *Proceedings of the SPECOM*, volume 1, pages 191–194. Citeseer, 2005.

[26] Sadaoki Furui. Cepstral analysis technique for automatic speaker verification. *Acoustics, Speech and Signal Processing, IEEE Transactions on*, 29(2):254–272, 1981.

[27] J.R. Deller, J.G. Proakis, and J.H.L. Hansen. *Discrete-time processing of speech signals*. Macmillan New York, 1993.

[28] J.W. Picone et al. Signal modeling techniques in speech recognition. *Proceedings of the IEEE*, 81(9):1215–1247, 1993.

[29] V. Mantha, R. Duncan, Y. Wu, J. Zhao, A. Ganapathiraju, and J. Picone. Implementation and analysis of speech recognition front-ends. *IEEE Southeastcon'99. Proceedings*, pages 32–35, 1999.

[30] S. Davis and P. Mermelstein. Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences. *Acoustics, Speech and Signal Processing, IEEE Transactions on*, 28(4):357–366, 1980.

[31] J.P. Campbell et al. Speaker recognition: A tutorial. *Proceedings of the IEEE*, 85(9):1437–1462, 1997.

[32] T. Kamm, H. Hermansky, and A.G. Andreou. Learning the Mel-scale and optimal VTN mapping, 1997.

[33] R. Vergin, D. O'Shaughnessy, and V. Gupta. Compensated mel frequency cepstrum coefficients. In *ICASSP '96: Proceedings of the Acoustics, Speech, and Signal Processing, 1996. on Conference Proceedings., 1996 IEEE International Conference*, pages 323–326, Washington, DC, USA, 1996. IEEE Computer Society.

[34] Daniel P. W. Ellis. PLP and RASTA (and MFCC, and inversion) in Matlab, 2005. http://www.ee.columbia.edu/ dpwe/resources/matlab/rastamat.

[35] Douglas A Reynolds. Experimental evaluation of features for robust speaker identification. *Speech and Audio Processing, IEEE Transactions on*, 2(4):639–643, 1994.

[36] BH Juang and LR Rabiner. Hidden Markov Models for Speech Recognition. *Technometrics*, 33(3):251–272, 1991.

[37] Huber Marco F, Bailey Tim, Durrant-Whyte Hugh, and Hanebeck Uwe D. On entropy approximation for Gaussian mixture random vectors. In *IEEE International Conference on In Multisensor Fusion and Integration for Intelligent Systems*, 2008.

[38] Allen Gersho and Robert M Gray. *Vector quantization and signal compression*. Springer, 1992.

[39] Ahilan Kanagasundaram, Robbie Vogt, David B Dean, Sridha Sridharan, and Michael W Mason. i-vector based speaker recognition on short utterances. In *Proceedings of the 12th Annual Conference of the International Speech Communication Association*, pages 2341–2344. International Speech Communication Association (ISCA), 2011.

[40] A. Higgins, J. Porter, and L. Bahler. YOHO speaker authentication final report. *ITT Defense Communications Division*, 1989.

[41] T.M. Cover and J.A. Thomas. *Elements of information theory*. wiley, 2006.

[42] Claude E Shannon. Prediction and entropy of printed english. *Bell system technical journal*, 30(1):50–64, 1951.

[43] V. Mazya and G. Schmidt. On approximate approximations using gaussian kernels. *IMA J. Numer. Anal.*, 16, 1996.

[44] A. Adler, R. Youmaran, and S. Loyka. Towards a measure of biometric information. In *Electrical and Computer Engineering, 2006. CCECE'06. Canadian Conference on*, pages 210–213. IEEE, 2006.

[45] J.R. Hershey and P.A. Olsen. Approximating the kullback leibler divergence between gaussian mixture models. In *Acoustics, Speech and Signal Processing, 2007. ICASSP 2007. IEEE International Conference on*, volume 4, pages IV–317. Ieee, 2007.

[46] F. Zheng, G. Zhang, and Z. Song. Comparison of different implementations of MFCC. *Journal of Computer Science and Technology*, 16(6):582–589, 2001.

[47] T. Kinnunen. Spectral features for automatic text-independent speaker recognition. *Licentiatesthesis, Department of computer science, University of Joensuu*, 2003.

[48] Fabian Monrose, Michael K Reiter, Qi Li, and Susanne Wetzel. Cryptographic key generation from voice. In *Security and Privacy, 2001. S&P 2001. Proceedings. 2001 IEEE Symposium on*, pages 202–213. IEEE, 2001.

[49] F. Monrose, M.K. Reiter, Q. Li, and S. Wetzel. Using voice to generate cryptographic keys. In *2001: A Speaker Odyssey-The Speaker Recognition Workshop*. Citeseer, 2001.

[50] F.K. Soong and A.E. Rosenberg. On the use of instantaneous and transitional spectral information in speaker recognition. *Acoustics, Speech and Signal Processing, IEEE Transactions on*, 36(6):871–879, 1988.

[51] C. Yang, G. Hammouri, and B. Sunar. Voice passwords revisited. In *International Conference on Security and Cryptography (SECRYPT)*, Rome, Italy, July 2012.

[52] Microsoft Corporation. System.speech programming guide for .net framework 4.0, October 2011. Microsoft Developer Network (MSDN).

[53] J.F. Bonastre, F. Wils, and S. Meignier. Alize, a free toolkit for speaker recognition. In *Acoustics, Speech, and Signal Processing, 2005. Proceedings.(ICASSP'05). IEEE International Conference on*, volume 1, pages 737–740. IEEE, 2005.

[54] B.G.B. Fauve, D. Matrouf, N. Scheffer, J.F. Bonastre, and J.S.D. Mason. State-of-the-art performance in text-independent speaker verification through open-source software. *Audio, Speech, and Language Processing, IEEE Transactions on*, 15(7):1960–1968, 2007.

[55] D. Petrovska-Delacrétaz, A. El Hannani, and G. Chollet. Text-independent speaker verification: state of the art and challenges. *Progress in nonlinear speech processing*, pages 135–169, 2007.

[56] A.E. Hannani, D. Petrovska-Delacrétaz, B. Fauve, A. Mayoue, J. Mason, J.F. Bonastre, and G. Chollet. Text-independent speaker verification. *Guide to Biometric Reference Systems and Performance Evaluation*, pages 167–211, 2009.

[57] G. Chollet, G. Aversano, B. Dorizzi, and D. Petrovska-Delacretaz. The 1st biosecure residential workshop. In *Proc. of ISPA*, pages 251–256, 2005.

[58] Biosecure. http://www.biosecure.info/.

[59] The link to alize. http://svnext.it-sudparis.eu/svnview2-eph/ref_syst//.

[60] P. Kenny, G. Boulianne, and P. Dumouchel. Eigenvoice modeling with sparse training data. *Speech and Audio Processing, IEEE Transactions on*, 13(3):345–354, 2005.

[61] Ondrej Glembek, Lukas Burget, Najim Dehak, Niko Brummer, and Patrick Kenny. Comparison of scoring methods used in speaker recognition with joint factor analysis. In *Acoustics, Speech and Signal Processing, 2009. ICASSP 2009. IEEE International Conference on*, pages 4057–4060. IEEE, 2009.

[62] Claudio Vair, Daniele Colibro, Fabio Castaldo, Emanuele Dalmasso, and Pietro Laface. Channel factors compensation in model and feature domain for speaker recognition. In *Speaker and Language Recognition Workshop, 2006. IEEE Odyssey 2006: The*, pages 1–6. IEEE, 2006.