Voice Passwords Revisited

Chenguang Yang[1], Ghaith Hammouri[2] and Berk Sunar[1]
[1]*CRIS Lab, Worcester Polytechnic Institute, 100 Institute Road, Worcester, MA 01609-2280 U.S.A.*
[2]*Claveo Software, 810 E Montecito Street, Santa Barbara, CA 93101 U.S.A.*

Keywords: Voice, Entropy, Mel Frequency Cepstral Coefficients, Gaussian Mixture Model.

Abstract: 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 build 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 quite practical and highlights the limited amount of entropy that can be extracted from the human voice when using MFCC features.

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 (Miller and Top, 2010) 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 connections to academic work developed in the last decade. In 2001, (Monrose et al., 2001a; Monrose et al., 2001b) were the first group to extract cryptographic keys from human voice. In (Krause and Gazit, 2006), the authors propose a new classifier by combining a supported vector machine with a Gaussian Mixture Model (GMM) verifier. Other GMM based schemes may be found in (Heck and Mirghafori, 2000; Mirghafori and Heck, 2002; Teunen et al., 2000) use cepstrum based features as the front-end processing feature. Agnitio's Kivox hosts a speaker verification system based on MFCC-GMM modeling technique (Brummer and Strasheim, 2009). Further detailes can be found in (Kenny et al., 2005; Kenny et al., 2007; Kenny et al., 2008). Note that, MFCC and GMM are the most popular extraction and modeling techniques for text-independent speech

Yang C., Hammouri G. and Sunar B..
Voice Passwords Revisited.
DOI: 10.5220/0004060201630171
In *Proceedings of the International Conference on Security and Cryptography* (SECRYPT-2012), pages 163-171
ISBN: 978-989-8565-24-2

recognition and are used as building blocks in numerous speaker verification sytems, e.g., see (Heck and Mirghafori, 2000; Mirghafori and Heck, 2002; Teunen et al., 2000; Brummer and Strasheim, 2009). With all this deployment of voice authentication technologies, it becomes crucial to evaluate voice authentication technologies from a security point of view. In this work we take a step in this direction.

Our Contribution. In this paper we demonstrate an effective attack on basic voice authentication technologies. In particular, we show how one member of a voice database can manipulate his voice in order to attack a voice authentication system which uses speech recognition in parallel with speaker verification. We assume that the speaker verification uses adapted GMM to model basic MFCC features of speakers. We demonstrate our attack using the YOHO database which contains 138 different people, and we show how an attacker can impersonate anyone in the database with a 98% success probability after at most nine authorization attempts. 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.

The remainder of this paper is organized as follows. In the next section we introduce the basic background and relevant terminology. In Section 3 we explain our attack and highlight why the attack works. This is followed by Section 4 where we show detailed results of the application of our attack to the YOHO database. Finally we present the conclusions in Section 5.

2 BACKGROUND

Speaker verification systems work in two phases: enrollment and verification (Bimbot et al., 2004). During enrollment, a speaker is asked to contribute speech samples whose features are then extracted as shown in Figure 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, fresh samples are collected from the user. After similar extraction phases, the resulting extracted features are compared against the model stored during enrollment. The most popular feature extraction technique used in voice verification systems is based on *short term cepstral analysis* which includes mel-frequency cepstral coefficients (MFCC), linear predictive cepstral coefficients (LPCC), etc. We now briefly review two basic techniques that will be essential to our attack.

Figure 1: Speaker enrollment.

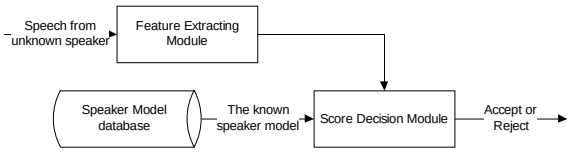

Figure 2: Speaker verification.

2.1 Extracting Mel Frequency Cepstrum Coefficients

Mel-frequency cepstral coefficients (MFCC) is the short term cepstral representation of speech signal in the mel scale. Short term cepstral features capture the information of vocal tract in order to reflect the uniquness of a speaker's voice. The mel scale is used to appoximate the response of the human auditory system. MFCC was first introduced to speech recognition in (Davis and Mermelstein, 1980) and later on was used in speaker verification. MFCC have been shown to outperform any other Short Term Cepstrum feature extraction technique in speech recognition (Davis and Mermelstein, 1980). MFCC also provides the most robust features for text-independent speaker recognition (Reynolds et al., 2000; Reynolds and Rose, 1995). In the following discussion, we will focus on MFCC alone. Introduction to other features such as Linear Prediction Cepstral Coefficients can be found in (Bimbot et al., 2004). Similar to other cepstral features, MFCC is obtained from a speech signal through a combination of transforms (Bimbot et al., 2004; Vergin et al., 1996; Ganchev et al., 2005). 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.
3. Use a number of triangular band-pass filters in order to project the frequency components of each frame into the Mel-scale.
4. Compute 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 3. To remove the channel filter bias and intra-speaker variability, compensation methods can be applied (Bimbot et al., 2004).

According to (Reynolds and Rose, 1995), cepstral mean subtraction which is called spectral shape compensation gives the best identification performance. 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 (Ellis, 2005).

2.2 The Gaussian Mixture Model

The Gaussian Mixture Model (GMM) is one of the most widely used voice models in text-independent speaker recognition (Reynolds and Rose, 1995). 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 (Marco F et al., 2008). 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[1]. 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

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

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

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

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

$$p(X|\lambda) = \prod_{k=1}^{K} p(X_k|\lambda) \tag{3}$$

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 $p(X_j|\lambda_j)$. 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, a decision function

$$D_j(X') = p(X'|\lambda_j) \tag{4}$$

is computed where X' is a fresh feature extracted from an unknown person who claims to be j. Given a predefined constant threshold T, a decision will be made based on the condition $D_j(X') > T$ holding. If it does, 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 (Reynolds et al., 2000). 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, instead of having Equation 4 adapted GMM uses a decision function

$$D_j(X') = \frac{p(X'|\lambda_j)}{p(X'|\lambda_b)} . \tag{5}$$

The details of the adapted GMM modeling algorithm can be found in (Reynolds et al., 2000). 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 paper we will base our analysis on the more popular adapted GMM.

3 OUR CONTRIBUTION

In this section we explain the proposed attack. We will start by describing the type of system that we are attacking and explain the rationale behind the attack. Finally, we describe the attack in detail and outline its limitations.

3.1 Voice Authentication Assumptions

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

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

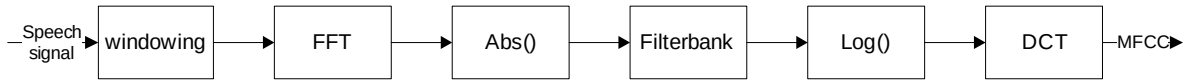

Figure 3: Transformation from signal to MFCC.

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.

3.2 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 (Reynolds et al., 2000) 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 (Reynolds et al., 2000) where the authors found that the equal false positive and false negative rates saw very little improvement beyond 256 components. Another important results of (Reynolds et al., 2000) 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.

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 and 5 show the distribution for 12 MFCC components of 138 different people. Figure 4 uses 256 component GMM and Figure 5 uses 4 component GMM. It should be clear that the general shape and peaks of the distributions are pre-

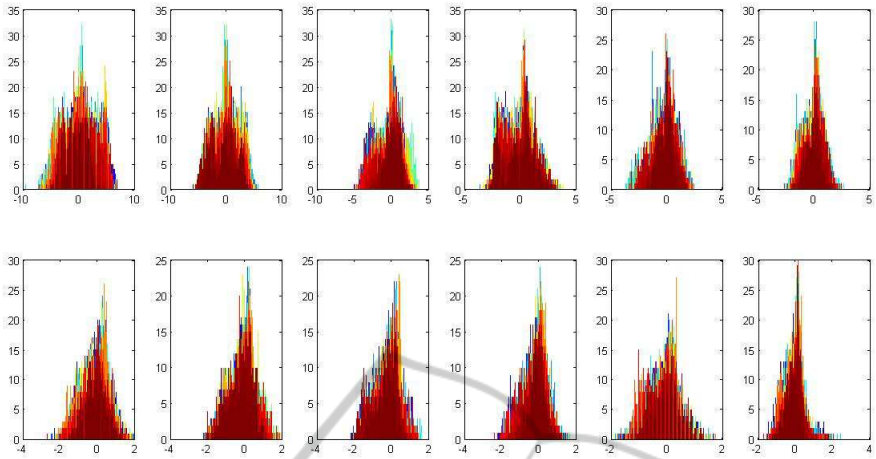

Figure 4: 12 MFCC features each modeled using 256 GMM components with each color representing one of 138 people.

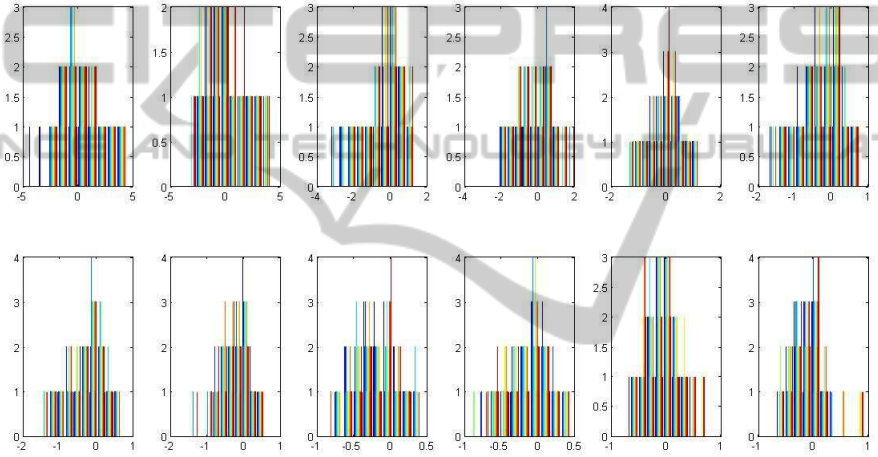

Figure 5: 12 MFCC features each modeled using 4 GMM components with each color representing one of 138 people.

served 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 and 5 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 representing 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.

3.3 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 (Ellis, 2005). 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 1.

Table 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.

4 EXPERIMENTAL RESULTS

Our experiments utilize the YOHO database which contains voice samples collected from 138 different speakers with a sampling frequency of 8 kHz (Higgins et al., 1989). 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 speech).

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

4.1 Voice Authentication Setup

As explained in the previous section, our voice authentication system is composed of two parallel submodules, 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 (Microsoft Corporation, 2011). 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 (Kinnunen, 2003) have shown that dynamic (derivative) features contribute far less to the speaker verification performance than normal features do. Furthermore, research in (Soong and Rosenberg, 1988) 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 (Reynolds et al., 2000):

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

$$D_j(\text{MFCC}_x) = \log\left(\frac{p(\text{MFCC}_x|\lambda_j)}{p(\text{MFCC}_x|\lambda_b)}\right) . \quad (6)$$

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% [2]

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

4.2 Hybrid Signal Setup

Now we introduce our setup for the 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 [3], 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.

4.3 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 to the speech signal S_1. The larger the q parameter is, the more dominate the mock signal part is. From a 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 an 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 2 summarizes the results of the attack. For select q values these results are also plotted in Figure 6.

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 security of voice password authentication systems.

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

[3] 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 2: Success rate of Attacking with different parameters. Assume speech signal $S1$ is with a ratio of 1.

% of passing people

mock signal ratio (q)	# of 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

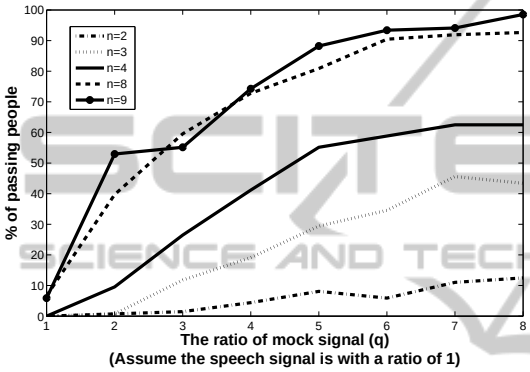

Figure 6: Attack success rate for select q values.

4.4 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.

5 CONCLUSIONS

In this paper 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.

REFERENCES

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

Brummer, N. and Strasheim, A. (2009). AGNITIO's Speaker Recognition System for EVALITA 2009. In *The 11th Conference of the Italian Association for Artificial Intelligence*.

Davis, S. and Mermelstein, P. (1980). Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences. *Acoustics,*

Speech and Signal Processing, IEEE Transactions on, 28(4):357–366.

Ellis, D. P. W. (2005). PLP and RASTA (and MFCC, and inversion) in Matlab. www.ee.columbia.edu/∼dpwe/resources/matlab/rastamat.

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

Heck, L. and Mirghafori, N. (2000). On-line unsupervised adaptation in speaker verification. In *Sixth International Conference on Spoken Language Processing*.

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

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

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

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

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

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

Marco F, H., Tim, B., Hugh, D.-W., and Uwe D., H. (2008). On entropy approximation for Gaussian mixture random vectors. In *IEEE International Conference on In Multisensor Fusion and Integration for Intelligent Systems*.

Microsoft Corporation (2011). System.speech programming guide for .net framework 4.0. Microsoft Developer Network (MSDN).

Miller, D. and Top, D. (2010). Voice biometrics 2010: A transformative year for voice-based authentication.

Mirghafori, N. and Heck, L. (2002). An adaptive speaker verification system with speaker dependent a priori decision thresholds. In *Seventh International Conference on Spoken Language Processing*.

Monrose, F., Reiter, M., Li, Q., and Wetzel, S. (2001a). Cryptographic key generation from voice. *sp*, page 0202.

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

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

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

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

Teunen, R., Shahshahani, B., and Heck, L. (2000). A model-based transformational approach to robust speaker recognition. In *Sixth International Conference on Spoken Language Processing*.

Vergin, R., O'Shaughnessy, D., and Gupta, V. (1996). 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. IEEE Computer Society.