DOI: 10.2352/ISSN.2470-1173.2016.8.MWSF-073

Attacks on Speaker Identification Systems Constrained to Speech-to-Text Decoding

Alireza Farrokh Baroughi and Scott Craver; Binghamton University; Binghamton, NY/USA
Daniel Douglas; Temple University, Philadelphia, PA/USA

Abstract

Speech processing is used to translate human speech to text and to identify speakers for applications in biometric systems. Speaker verification requires robust algorithms to prohibit an adversary from impersonating another speaker. Previous research has demonstrated that specially crafted additive noise can cause a misclassification of a speaker as a specific target. In this paper, we study whether targeted additive noise can thwart speaker verification without affecting speech-to-text decoding. Mel-frequency cepstral coefficients (MFCCs) and Gaussian mixture models (GMMs) are commonly used in both applications for encoding schemes. We attempt to induce a desired change in the probability of one speaker model used for speaker classification, while preserving likelihood under another speech model used for speech decoding.

Introduction

Speech processing is used both to translate human speech to text, and to identify speakers for applications in biometric systems. Speaker verification requires robust algorithms to prohibit an adversary impersonating another speaker. Previous research has demonstrated that specially crafted additive noise can cause a misclassification of a speaker. This could take the form of a smart phone application that can be held near a microphone, emitting noise while an attacker speaks, and possibly shaping that noise based on the attacker's speech. This noise would cause a deviation in the MFCC (Mel-frequency space cepstral coefficient) components of the user's speech, incurring a false match to a target user. This differs from an impersonation attack using a recorded voice or a synthesized voice; the direct additive attack confuses a detector on a more direct level, with acoustic signals that may not resemble speech at all.

The motivation for such an attack is that it can be applied instantaneously, and therefore circumvent safeguards against impersionation attacks. A simple safeguard against impersonation is to require a speaker to recite a randomly generated phrase displayed on a computer screen, and confirm that this phrase is uttered by the speaker; this prevents an impersonation attack that requires an impersonated voice signal to be recorded or computed in advance. A direct additive attack should evade this safeguard, since an attacker will simply read the phrase on display while playing an additive attack signal.

An open question, however, is whether such an attack can simultaneously misclassify a speaker without interfering with the speech decoding of the speaker's utterance. In this work we explore whether such attacks are possible.

An obvious attack on a speech recognition system is an *impersonation attack,* in which an attacker presents audio samples that sound like a target user. This could simply be a recording of a target user, or a speech sample processed so as to imitate a target user's vocal characteristics. We assert that such an attack is too simple and too clumsy to be of much concern: firstly, it can be made far more difficult with the use of a few simple safeguards, and secondly, it requires far more processing than is necessary. An attacker need not produce a voice sample that completely imitates a target user to human ears—the attacker need only produce a sample that induces the desired effect to a detection algorithm.

In this paper, we explore a specific safeguard to prevent impersonation attacks: constraining a test subject to reciting a specific phrase. If a subject is required to read a sample sentence displayed on a computer screen, a pre-recorded voice sample can not be used by an attacker. Neither can a recorded voice sample altered to imitate a victim user, if that alteration requires substantial computation—or if the speaker recognition environment prevents an attacker from recording a voice sample to alter. In such an environment, the attacker must provide an immediate speech signal, possibly altered or composed in real time by a portable device. The speech is then subject both to speaker classification, and to speech decoding to determine if the test subject has recited a given sentence.

In [1], the authors detailed an additive attack on speaker identification systems, in which specially constructed noise can be emitted by a portable audio player while an attacker speaks; this noise is intended to cause a specific misclassification of the attacker by producing desired modifications in the feature domain used by the speaker identification system. Here, we explore the question of whether such additive noise interferes with speech decoding, or if a different approach is necessary.

Speech Processing

Speaker identification and speech processing use similar feature sets, and similar approaches to modeling those features. In our exploration of attacks on speaker classification subject to speech decoding constraints, we targeted common algorithms for speaker classification and widely-used software for speech decoding, discovering to our surprise that both used not only the same feature set, but similar parameters such as feature dimension.

Feature extraction

The software audio processing library Kaldi [2] transforms user speech samples into frames of Mel-frequency cepstral coefficient (MFCC) features. Windows of user input are first subjected to an FFT, and then transformed into a Mel-frequency spectrum that both discards phase information and subjects the frequency scale to a nonlinear (logarithmic) distortion. The specific rela-

DOI: 10.2352/ISSN.2470-1173.2016.8.MWSF-073

tionship between the Mel scale and the audio frequency scale is

$$m = 2595 \log_{10}\left(1 + \frac{f}{700}\right)$$

...where f is an audio frequency in Hz. The specific mapping from the FFT domain into a Mel-frequency domain is achieved using triangular windows of varying width, producing a spectrum with fewer bins. This is then subject to a DCT, with the 13 coefficients of lowest frequency used as features. We thus have a set of core features consisting of 13-dimensional feature frames.

For both speech and speaker classification, subsequent processing is used for purposes of modeling. These frames can be normalized both for mean and variance over the entirety of a recorded utterance, and first- and second-order differences of the feature vectors are computed to capture variation in the recorded speech. For 13-dimensional feature vectors $\{c_k\}$, we have "velocity" vectors $\{v_k = c_k - c_{k-1}\}$ and "acceleration" vectors $\{a_k = v_k - v_{k-1}\}$, resulting in a 39-dimensional feature set.

In the speaker identification system explored here and in [1], a similar feature set is computed, using 13-dimensional MFCC coefficients augmented with first- and second-order differences to produce a 39-dimensional feature vector per audio frame.

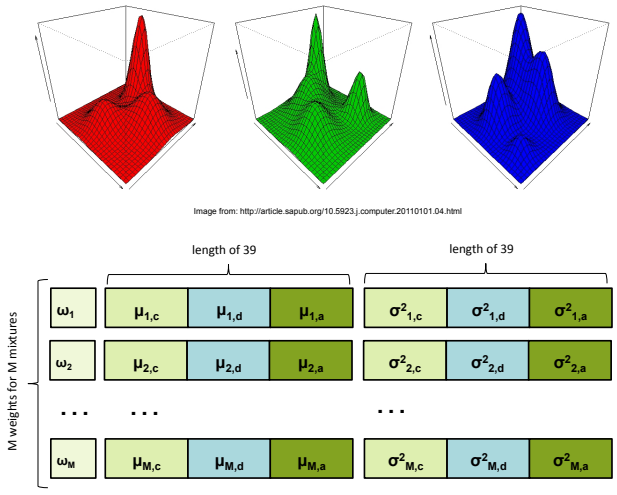

Figure 1. *Gaussian mixture modeling. Above, example densities of two-dimensional Gaussian mixture models. Below, representation of MFCC data with first- and second-order differences with mean and diagonal covariance matrices.*

Detection and modeling

Speaker classification is simpler than speech decoding, and is achieved by modeling a (39-dimensional) feature vector using a Gaussian mixture model. [4, 5] In this model, a speaker is represented as one of M mixtures, each described by a multivariate Gaussian distribution with mean μ_k and covariance matrix Σ_k, and assigned a weight w_k. The probability of a given vector x for a given user is the sum

$$f(x) = \sum_{k=1}^{m} w_k \cdot (2\pi)^{-n/2} |\Sigma_k|^{-1/2} e^{-\frac{1}{2}(x-\mu_k)' \Sigma_k^{-1} (x-\mu_k)}$$

...where $n = 39$ is the feature space dimension. A sequence of feature frame $\{x_t\}$ is scored according to their overall probability $\prod_t f(x_t)$, or more practically $\sum_t \log f(x_t)$.

The Gaussian mixture models for each user are derived from training samples of the user's voice. It is common for speech recognition systems to include a universal background model or UBM, a Gaussian mixture model trained on speech samples of all users, which can match speech samples that are sufficiently unlike any specific user in a speech recognition database. With a UBM, an attacker who wishes to be misclassified as a specific target user can not produce a signal so unusual that it is more likely under the UBM than under the target.

Gaussian mixture models are also used in speech decoding, but they merely form a foundation for a processing chain that is far more sophisticated. Speech decoding uses probabilistic models such as GMM models for individual phones, for example monophones, but these phones belong to an unknown time-varying sequence. These are commonly represented by hidden Markov model, whose hidden state is governed by higher-level features of spoken language.

In the application explored in this paper, a test user is required to recite an utterance, for example a randomly generated sentence displayed on a computer screen. In such a scenario, a speech decoding system has a great deal more information that can assist in decoding of the test user's utterance. This is possible through forced alignment of the underlying hidden Markov model; however, we consider our attacker constrained to conventional speech decoding. Our reasoning is that if a noise attack is subtle enough not to interfere with conventional decoding, then it should pass a decoder that is more informed and therefore able to decode speech with a greater detector power.

If we observe that a speaker detection system and speech decoding uses Gaussian mixture models for representing both speakers and phones, it is tempting to consider an attack that optimally alters a frame's classification under one set of mixture models, while minimizing a change in score under a second set of mixture models. However, the attacker does not know in advance what phones are expected by the speech decoding system. We therefore focus on a simpler problem: altering a user's voice so as to incur a minimal necessary change in the MFCC features.

Classes of attacks

In [1], a speaker classification system was foiled through the use of additive noise, unrelated to the speaker's utterance. This noise was composed so as to cause certain MFCC features to appear that would be classified as another speaker. It was not clear that such an injection of noise would be ignored by a speech decoding system. In an initial exploration we produced speech samples with this added noise, at the power levels listed by the authors as producing successful attacks, and played the resulting sounds into speech decoding software. The decoded results were dramatically different from the spoken utterance, and often could not be decoded at all.

This led us to consider an additive attack that would be shaped to the user's utterance, in order to produce a more controllable effect in the MFCC domain. In this attack, a portable device such as a smart phone uses both its microphone and speaker, playing a filtered version of the user's utterance in real time. In a conventional cepstrum without Mel-frequency scaling, and with-

DOI: 10.2352/ISSN.2470-1173.2016.8.MWSF-073

out additional pre-processing of samples, this produces an additive effect in the feature domain.

For a signal $x(t)$ with a frequency spectrum $X(\omega)$, we imagine a device that instantaneously plays a filtered version $[h \star x](t)$, whose speaker is so placed relative to a detector's microphone that we can model the input as a signal $y(t)$ with spectrum

$$Y(\omega) = (1 + \alpha H(\omega))X(\omega)$$

...where H denotes the frequency response of the filter. In the log-cepstral domain, we have

$$\begin{aligned} \log Y(\omega) &= \log(1 + \alpha H(\omega)) + X(\omega) \\ &\sim \alpha H(\omega) + X(\omega) \end{aligned}$$

...where the second line is justifiable for values of $\alpha H(\omega)$ of small magnitude. If the filter's effects are not so subtle as to justify this approximation, it doesn't matter: we have the result that in the log-spectral domain, a filtering effect produces an additive signature that can be used to doctor cepstral features before a detector. A final frequency transform preserves the additive relationship between the input cepstrum and output cepstrum.

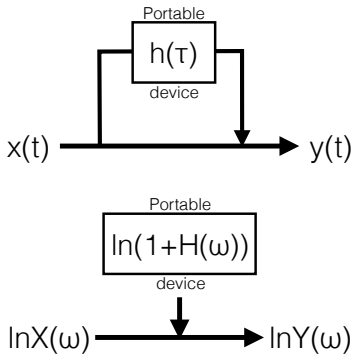

Figure 2. *Addition of a filtered signal, and the effect on the log spectrum.*

A complication comes from the conversion of the spectrum to a Mel-frequency scale using overlapping triangular windows. Only magnitudes are used as input to this transformation, and the overlapping produces a constraint that presents some difficulty in inducing a specific log-spectral signature. One way around this is to produce a filter whose frequency response is limited to the peak frequencies of the triangular filters; if the kth Mel-frequency bin w_k is derived from an overlapping triangular window of peak frequency f_k, then we can imagine a filter H of frequency response

$$H(\omega) = \sum_k w_k \delta(\omega - f_k)$$

This can be achieved in practice by sampling the attacker's speech, computing energy at specific peak frequencies, and then playing a sinusoid of appropriate amplitude in order to augment the energy at that frequency.

The problem with such an approach is that it neglects all of the energy placed in each bin, and energy that is not uniformly distributed within the domain of a triangular window may be missed by such a specific filtering effect. In this paper we do not focus on producing an ideal filtering effect, but rather explore the effect of simple filters, e.g. echoes, on the MFCC domain, and compare this effect to the level of energy that can foil a speaker identification system without derailing speech decoding.

Objective and Method

Our goal is to characterize the effect of simple additive attacks on speech decoding, specifically to determine if an additive attack can properly derail a speaker identification system without effecting the word error rate (WER) of a speech-to-text system. In our hypothetical arrangement, a speaker identification system and speech-to-text system are used in combination to admit a user. A user is required to utter an arbitrary phrase, and is admitted only if the speech is properly decoded as that phrase. Essentially we want to craft a signal that induces a false match in one detector without effecting the decision of a second detector.

The systems under test both compute Mel-spaced Frequency Cepstral Coefficient (MFCC) vectors, combined with first- and second-order time differences of MFCC coefficients, from a user's utterance. MFCC coefficients are commonly used in both speech applications; these are processed in different ways, but the similar transform domain allows us to restrict our focus to optimal tampering of MFCC coefficients. Thus the objective of this research is to explore the feasible region of MFCC vector modifications that induce a small WER whilst incurring a chosen speaker misclassification.

For speech recognition we use Kaldi, which transforms an utterance into a sequence of 39-dimensional vector of 13 MFCC features, 13 first-order differences (delta-cepstral features), and 13 second-order differences (delta-delta-cepstral features). Our speaker identification system uses these same feature vectors. The speaker identification system models a speaker's feature vector using a time-independent Gaussian Mixture Model (GMM), and a speaker model is scored by the combined likelihood of all frames taken from a recorded utterance. The Kaldi speech-to-text system is far more complicated, employing a hidden Markov model for phoneme sequences and Gaussian Mixture Models of individual phonemes. In practice, a biometric system can decode speech more robustly than a generic algorithm because the user must recite a phrase known to the system in advance, using forced alignment of the hidden Markov model. However, the ability to preserve a low WER in a generic decoding should imply the efficacy of attacks under more robust speech decoding.

We assume the attacker can not know and should therefore ignore how MFCC vectors are classified by a speech-to-text system. However, the attacker does have information about the target user and about his or her own voice. In ideal circumstances the attacker may know the parameters of the GMMs used to represent the target user, and can make an informed attempt to alter MFCC vectors to make them optimally more likely under the target user's model.

While the system uses first- and second-order time differences of MFCC values, the features extracted from the audio consist only of the 13-dimensional MFCC vectors. We restrict our attack to those vectors; if we want to induce effects in successive time differences it will be necessary for us to induce a time-varying effect in our attack signal.

DOI: 10.2352/ISSN.2470-1173.2016.8.MWSF-073

Injection of noise

Kaldi is a very complicated software package, consisting of a suite of audio processing programs chained together with scripts. Altering computed features before classification is not a simple matter of altering the software. To achieve this, we identified a point in the Kaldi processing pipeline after which MFCC coefficients are created, and wrote a utility in C called `feature_alter` to sit within this pipeline. Our utility is able to extract 13-dimensional MFCC vectors computed by the Kaldi software and subject them to an arbitrary callback function, allowing them to be stored, analyzed, or altered before they are passed on for speech processing.

To achieve this, it was necessary to reverse-engineer the Kaldi file format. We observed that at this stage in the processing pipeline, only raw MFCC features existed in the stream; first- and second-order differences were not stored in the file. The source code for the `feature_alter` program can be found at `www.binghamton.edu/~scraver/feature_alter.c`, or upon request to the authors.

For our experiment, we modified MFCC feature vectors vectors of iid coefficients of value ± 1. This vector was added subject to a scaling factor and the magnitude of each feature vector: a vector v_k, therefore, was amended to $v_k + \alpha\|n_k\|$. By varying the value of α, we could then follow the Kaldi processing chain through to the end and assess the effect of this noise power on the word error rate.

The results of this experiment are shown below. For an α value around 0.1, word error rate drops significantly, suggesting that any attack that surgically modifies MFCC features should attempt to inflict a modification constrained to this power or below. Note that an $\alpha = 0.1$ implies an additive noise vector whose magnitude is $\|v_k\|\alpha\sqrt{13} \sim 0.36\|v_k\|$.

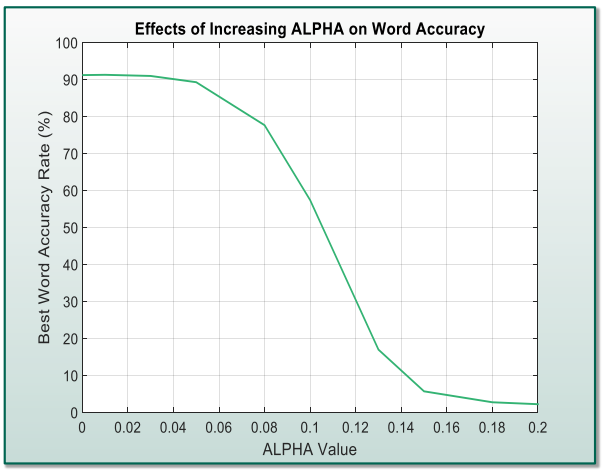

Figure 3. *Word error rate as a function of α parameter used to inflict additive noise in the MFCC domain.*

Analysis of attack methods

Armed with this knowledge, we first explored the effect in the MFCC domain of the additive noise attacks employed in [1]. Although the additive noise was designed so as to induce a specific cepstral effect, it was not designed to surgically alter an attacker's MFCC coefficients. We therefore expected the MFCC "noise," amounting to the difference in MFCC coefficients before and after noise injection, would have a significant magnitude.

Using the five speakers from [1], each with additive noise to imitate each user, we computed the average magnitude difference over all frames. This was computed by taking MFCC vectors v_k and w_k representing the samples without and with noise, and computing the ratio $r_k = \|w_k - v_k\|/\|v_k\|$. The results are summarized below, and show a significant difference. The reader will note that if two MFCC vectors are of equal magnitude and so unrelated as to be orthogonal, their magnitude ratio will equal $\sqrt{2}$; in our data, magnitude ratios are very close to this value.

We therefore conclude that these uninformed additive attacks induce a degree of noise far beyond what is reasonable for preserving speech decoding. A different approach is necessary.

Average MFCC magnitude ratios from additive attacks

Attacker	Victim (energy level 0.1)				
	1	2	3	4	5
1	–	1.33	1.34	1.36	1.34
2	1.15	–	1.19	1.25	1.21
3	0.97	1.01	–	1.09	1.04
4	0.98	0.99	1.00	–	1.02
5	1.10	1.12	1.13	1.18	–
Attacker	Victim (energy level 0.25)				
1	–	1.36	1.37	1.38	1.38
2	1.22	–	1.25	1.29	1.26
3	1.08	1.10	–	1.18	1.13
4	1.08	1.08	1.10	–	1.11
5	1.20	1.19	1.22	1.27	–
Attacker	Victim (energy level 0.5)				
1	–	1.39	1.40	1.41	1.39
2	1.27	–	1.29	1.34	1.30
3	1.15	1.15	–	1.24	1.19
4	1.15	1.16	1.17	–	1.19
5	1.25	1.27	1.28	1.31	–
Attacker	Victim (energy level 1.0)				
1	–	1.40	1.42	1.43	1.42
2	1.32	–	1.34	1.36	1.34
3	1.22	1.22	–	1.29	1.26
4	1.22	1.21	1.23	–	1.25
5	1.33	1.32	1.35	1.40	–

Analysis of echo injection

As a more surgical approach we experimented with the injection of echoes in utterance samples from the speaker recognition database. An echo of the form

$$y(t) = x(t) + \beta x(t-\tau)$$

...will produce a filtered version of the input of the form

$$Y(\omega) = X(\omega) \cdot \left(1 + \beta e^{-j\omega\tau}\right)$$

...inducing an additive signature in the log-spectral domain. It is worth noting that for small β, $\log|1+\beta e^{-j\omega\tau}| \sim \cos(\omega\tau)$, which causes the signature of an echo to contain a significant amount of

DOI: 10.2352/ISSN.2470-1173.2016.8.MWSF-073

energy in a single frequency. This was the original justification for adding an extra frequency transform in the computation of a cepstrum: cepstral analysis was invented to detect echoes in seismic data, and an echo in particular produces a visible peak in a cepstrum [3]. However, the Mel-frequency scaling used to produce MFCC coefficients quickly removes any visible effects that would come from a simple additive signature in the log-spectral domain.

The use of echoes allows us to alter both the strength of the echoes, and alter the delay so as to admit a family of additive signatures. Given such a family it is possible to inflict an attacker's speech with a filter representing a chain of echoes of various delays and gains, in order to produce a desired additive signature in the log-spectral domain. Again, we do not concern ourselves in this paper with the specific signal we wish to inflict, only exploring the effect of the alteration in the MFCC domain and its expected effect on speech decoding.

Using utterances from the speaker recognition database, we again computed MFCC coefficient vectors v_k and w_k before and after alteration by echoes of varying delays and gains, and imagined the "noise" vector $n_k = w_k - v_k$. Echoes were inflicted with gains varying from 0.1 to 1.0 relative to the original signal, and with relative delays of 0.05, 0.1, 0.5, and 1.0 times the audio frame length. Example noise vectors are shown below in figure 4, for varying gains and delays. We observe that increased echo gain produces a more powerful noise vector, but also that increased echo delay produces a similar effect. The effect of gain is more strongly pronounced if the MFCC feature vectors are first preprocessed to normalize them with respect to mean and variance.

Using this data, we computed the average amplitude ratio observed for echoes of various delays and gains. The reader will note that a relative magnitude of approximately 0.36 is equivalent to an $\alpha = 0.1$, which we consider a necessary constraint in order to preserve the WER of speech decoding. This confirms that inflicting of an echo can induce stable effects in the MFCC domain.

Inducing of effects in a Gaussian mixture model

The above data tells us how we can potentially inflict mild noise in the MFCC domain; it does not tell us whether that noise will be effective in inducing a misclassification. To that end we must determine the typical magnitude of MFCC vectors, and the magnitude of noise signal that must be added in order to induce a misclassification. To this end, we can examine the Gaussian mixture models for users; if these are suitable models for a user's MFCC coefficients, then they should give us some idea of the magnitude of vectors, as well as the vector differences between an attacker and a victim user.

Table 2 shows the mean vector of maximum weight for the users in our dataset, along with the Euclidean distance between the maximum-weight mean vectors for each user, restricted to the 13-dimensional MFCC coefficients.

One may also consider the average, weighted mean vector for each user, whose value is $\sum_k w_k \mu_k$ over all of the user's mixtures; this could be taken as a rough estimate of the average mean vector for a given user. However, this mean is of very small magnitude, on the order of 10^{-4}, owing to the pre-processing of MFCC frames that normalizes with respect to mean and variance. The average MFCC frame is close to the zero vector, and so this value is not very informative.

Magnitudes and distances of highest-weight mean vectors

	weight	$\|\mu\|$	distance				
			1	2	3	4	5
1	0.266	0.62	0	2.50	1.58	2.25	2.89
2	0.162	2.10	2.50	0	3.37	2.64	2.04
3	0.172	1.93	1.58	3.37	0	3.45	3.59
4	0.142	1.83	2.25	2.64	3.45	0	3.09
5	0.196	2.50	2.89	2.04	3.59	3.09	0

These values give us some idea of the kind of vectors an attacker may wish to inject in order to impersonate a target user; however, the mean magnitude does not reflect the actual vector magnitude of MFCC coefficients. This is determined by considering both mean and variance from the model. The models in our dataset have diagonal covariance matrices, giving the following expression for the expected squared magnitude of a single user:

$$E\|X\|^2 = \sum_k w_k \sum_{i=1}^{13} \left(\mu_{ki}^2 + \sigma_{ki}^2\right)$$

Examining the GMMs produced for the five speakers, we can compute the expected squared magnitude for the MFCC coefficients in each model. Because of variance normalization prior to fitting the MFCC data to a GMM, the expected squared magnitude is close to 1 for each index, and therefore close to 13 overall. We therefore consider modifying each frame with an additive attack vector of magnitude at most $0.1\sqrt{13}\|v\|$, or 1.3, in order to cause a misclassification.

Our question is, then: if an electronic device contributes an additive signal to a user's voice at the point of speaker identification, and this can be so engineered as to induce an additive effect in the MFCC domain, what should that additive effect be? A constant filtering effect that produces a constant additive effect in the MFCC domain is problematic on two counts: first, any constant offset in the MFCC domain may be removed by preprocessing that removes the mean from MFCC frames; and second, that the attacker's MFCC vector will vary significantly, and this vector plus a constant offset may not always have a high likelihood under a victim user's model. The situation is analogous to hitting a target by firing an arrow in the same direction every time, despite appearing in a random position with respect to the target.

We suggest a way around this: if an electronic device can determine the MFCC coefficients of the attacker's voice as it is emitting a filtered version or a noise signal based on that voice, then it may be able to induce an MFCC offset based on the coefficients of each frame. In this case, a device that has the GMM of a victim user can amend each frame in a way that moves the attacker's MFCC components toward one of the victim user's mean vectors, one that requires the smallest contribution of energy. Table 3 shows, for each user pair in our data set, the expected Euclidean distance from an attacker's mean to the closest mean vector of a target user.

These numbers are greater than the power constraint dictated by our experiments with speech decoding, but not much greater. We note that an attacker does not need to move his or her MFCC vector exactly to the mean of a target user, and moreover that an attacker's utterance consists of a large number of frames, where a

DOI: 10.2352/ISSN.2470-1173.2016.8.MWSF-073

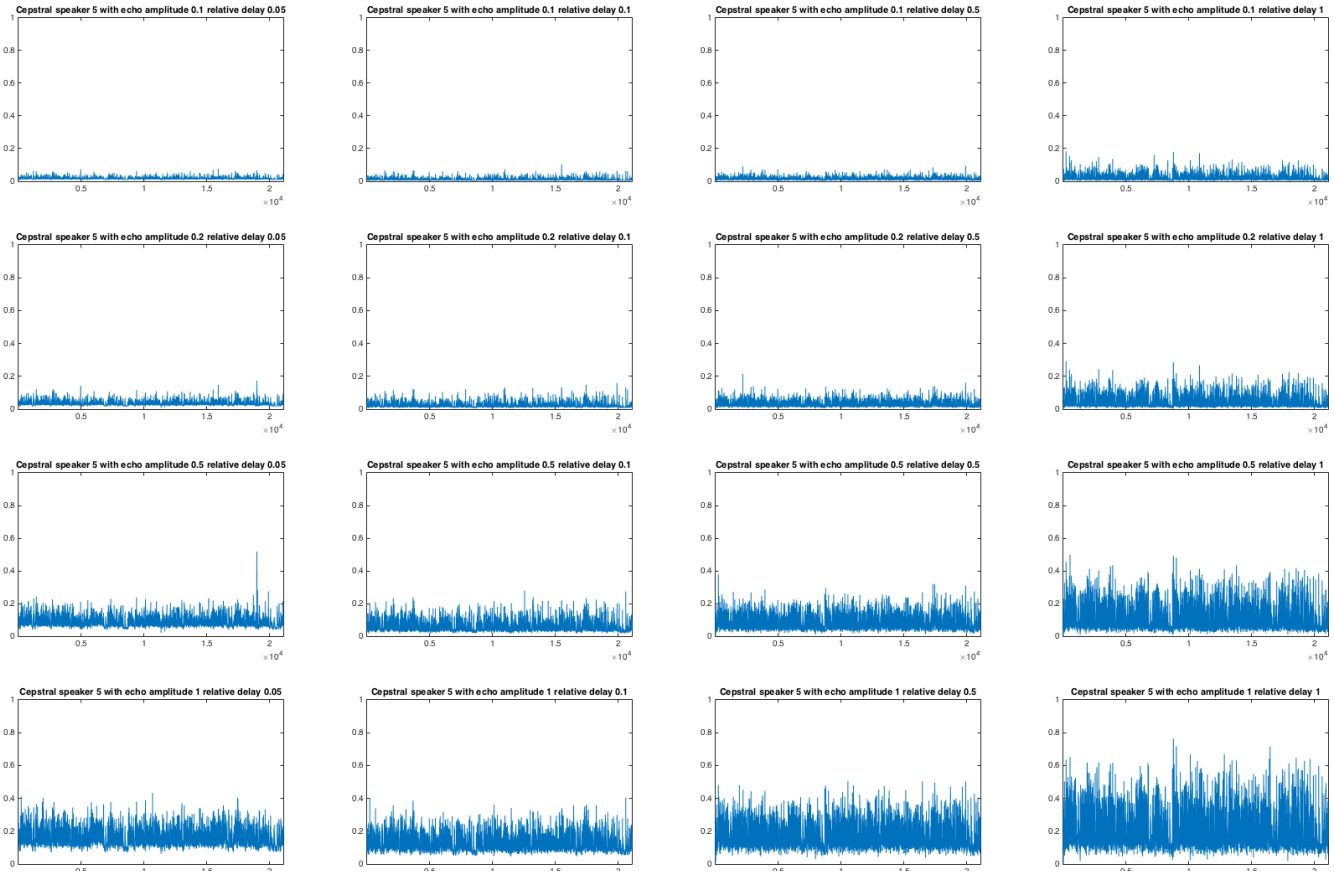

Figure 4. *Example MFCC differences induced by echoes of varying gains and delays.*

Expected distances of closest mean

User	distance				
	1	2	3	4	5
1	0.00	2.23	1.93	2.18	2.02
2	2.26	0.00	1.71	1.83	1.70
3	1.64	1.62	0.00	1.73	1.33
4	2.02	1.56	1.68	0.00	1.39
5	1.62	1.26	1.30	1.37	0.00

small increase of likelihood per frame can accumulate to a false positive.

Taken together, then, these numbers suggest that the inter-mean distances between users are small enough, relative to average MFCC vector magnitudes, that a modest noise injection should be possible without sacrificing the word error rate of speech decoding.

Discussion and conclusions

We envision an application that will perform miminal processing of an attacker's voice, playing a filtered or processed version thereof alongside the attacker, and induce a misclassification in a speaker recognition system. Our findings indicate that such an application could, if designed to surgically alter MFCC features, induce a false classification without affecting the word error rate of speech decoding. However, a straightforward additive attack as seen in [1] produces a degree of MFCC alteration that does not satisfy this constraint.

How would such a feature be implemented? We suggest the following approach, which is the subject of future work: instead of simply filtering user speech input with such as an echo, a smart phone or other programmable device will capture windows of audio and compute the FFT and then Mel-frequency components of each frame. The energy in each bin can be modified by playing a sinusoidal tone at the peak frequency of the triangular window for each Mel-frequency component.

As long as these values are computed, at the cost of an FFT and a single pass over FFT magnitudes to apply the overlapping triangular windows of the Mel-scale frequency scaling, the audio device need not produce sound corresponding to a constant alteration in the MFCC domain. It can, instead, store a mixture model of a victim user, choose one of the victim's mean vectors nearest

DOI: 10.2352/ISSN.2470-1173.2016.8.MWSF-073

to that of the spectrum just computed, and then emit noise in order to push the resulting MFCC coefficients in that direction.

Our investigation implies that such an attack should be feasible, although more complex than originally proposed in [1]. In order to trigger a desired misclassification without affecting speech decoding, we can not simply inflict significant additive noise independent of an attacker's utterance. Due to specific aspects of the processing chain used to compute and classify MFCC features, a portable device should produce an additive signal as a function of an attacker's spoken utterance, ideally informed by the GMM of a target user used by a speaker classification system.

References

[1] Alireza Farrokh Baroughi, Scott Craver, "Additive attacks on speaker recognition," Proc. SPIE 9028, Media Watermarking, Security, and Forensics 2014, 90280Q (19 February 2014); doi: 10.1117/12.2040872.

[2] Povey, Daniel and Ghoshal, Arnab and Boulianne, Gilles and Burget, Lukas and Glembek, Ondrej and Goel, Nagendra and Hannemann, Mirko and Motlicek, Petr and Qian, Yanmin and Schwarz, Petr and Silovsky, Jan and Stemmer, Georg and Vesely, Karel, "The Kaldi Speech Recognition Toolkit," IEEE 2011 Workshop on Automatic Speech Recognition and Understanding, Hilton Waikoloa Village, Big Island, Hawaii, US, IEEE Catalog No.: CFP11SRW-USB.

[3] B.P. Bogert, M.J.R. Healy and J.W. Tukey, "The Quefrency Analysis of Time Series for Echoes: Cepstrum, Pseudo-Autocovariance, Cross-cepstrum and Saphe Cracking", in Proceedings of the Symposium on Time Series Analysis, by M. Rosenblatt, (Ed.), Wiley N.Y. 1963, pp. 209-243.

[4] D. Reynolds and R. Rose, "Robust text-independent speaker identification using Gaussian mixture speaker models." Speech and Audio Processing, IEEE Transactions on, 3(1):72 83, jan 1995.

[5] R. Togneri and D. Pullella, "An overview of speaker identification: Accuracy and robustness issues." Circuits and Systems Magazine, IEEE, 11(2):23 61, second quarter 2011.

Author Biography

Alireza Farrokh Baroughi received his BS in electrical engineering from University of Tabriz (2006), his MS in electrical engineering from Iran University of Science and Technology (2009), and is pursuing his PhD in electrical engineering at Binghamton University. His research has focused on the security of Biometrics, especially speaker recognition and face recognition.

Scott Craver received his BS in Computer Science at Northern Illinois University and his PhD in electrical engineering from Princeton University. He is an associate professor of Electrical and Computer Engineering at Binghamton University. His research interests include information hiding and multimedia security.

Daniel Douglas is currently an fourth year student of Electrical and Computer Engineering at Temple University. His research is focused on Signal Denoising, Image Classification, and Speech Recognition. He is interested in Computer Network topologies and Control Theories, and develops several software applications as a member of the Neural Engineering Data Consortium at Temple University. After completing his undergraduate career, Daniel will pursue a PhD in Electrical Engineering.

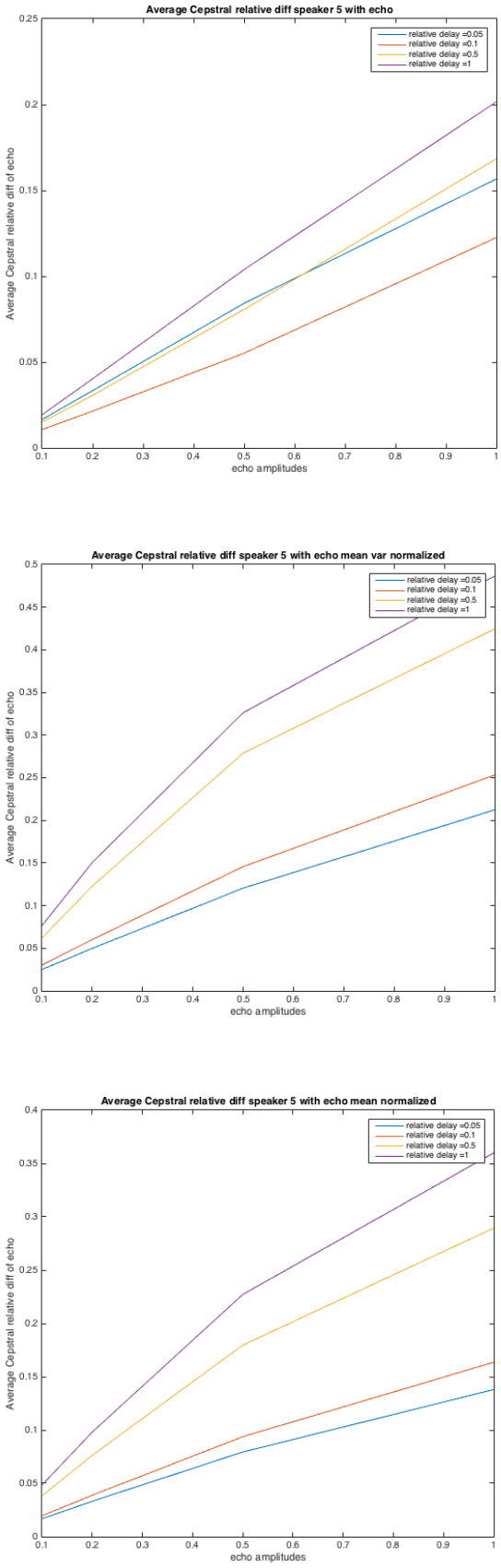

Figure 5. *Average magnitude ratio for echoes for MFCC data.*