Attacking Speaker Recognition Systems with Phoneme Morphing

Henry Turner, Giulio Lovisotto, and Ivan Martinovic

University of Oxford, UK {firstname.lastname}@cs.ox.ac.uk

Abstract. As voice interfaces become more widely available they increasingly implement speaker recognition, to provide both personalized functionalities and security via authentication. In this paper, we present a method that transforms the voice of one person so that it resembles the voice of a victim, such that it can be used to deceive speaker recognition systems into believing an utterance was spoken by the victim. The transformation only requires short pieces of audio recordings from the source and victim voices, and does not require specific words to be spoken by the victim. We show that the attack can be improved by using a population of source voices and we provide a metric to identify promising source voices, from within such a population.
We evaluate our attack along a set of dimensions, including: varying quantity, quality and types of known victim audio, verification and identification systems, white- and black-box models and both over-the-wire and over-the-air access. We test the audio transformation on two different proprietary models: (i) the Azure Speaker Recognition API and (ii) the Siri voice activation of an Apple iPhone, showing that individuals can easily be impersonated by obtaining as little as one minute of their audio, even when such audio is recorded in noisy conditions. With attempts from only three source voices, our attack achieves success rates of over 40% in the weakest assumption scenario against the Azure Verification API and rates of over 80% in all scenarios against Siri.

Keywords: voice conversion · speaker · authentication · biometrics

1 Introduction

As voice interfaces become more popular, voice-based devices are now adding speaker recognition to their capabilities, so they can understand both *what* has been said (speech recognition) and *who* has said it (speaker recognition). Speaker recognition allows for customized functionality, as well as authentication, removing the burden of other less user-friendly authentication approaches (e.g., PINs or passwords). Nowadays, speaker recognition is available in commercial products such as Google Home [16] or Apple Siri [2]. Additionally, speaker recognition is increasingly been deployed for over the phone authentication by companies in the financial sector (e.g., HSBC [18], Lloyds Bank [25]).

Recent studies have focused on analyzing the security of speech recognition systems [7, 8, 33, 35, 36], often with attacks which use adversarial machine learning techniques to craft malicious audio. These studies have shown that many

voice assistants are vulnerable to these types of attacks, and that commands can inconspicuously be injected or hidden in other sounds, such as songs [35]. However, as speaker recognition becomes widely available and used for sensitive functions, additional investigations are needed to assess its security guarantees. When targeting a speaker recognition system, rather than inconspicuously inject voice commands, the adversary's goal is to unlock access to a voice-protected device by replaying audio that resembles the device owner's voice.

Previous work in this area focused on creating complex and expensive models for synthesizing audio or converting one's own voice into the voice of the victim [10, 13, 19, 21, 26, 31]. Generally the training audio is collected in a well-isolated studio environment and the ultimate goal of the generated audio is to deceive a human listener. However, from an adversarial perspective, obtaining audio of spoken utterances could be suspicious or unfeasible for certain victims. The unavailability of long samples of victim audio brings two limitations in re-creating the victim's voice: (i) models based on parallel datasets for voice conversion can not be used and (ii) synthesizers or conversion methods based on deep models do not reach sufficient accuracy, as intra-user variability is not efficiently captured. A detailed analysis of related work is given in Section 2.

In this paper, we present a voice conversion attack that manipulates individual phonemes from a source voice into sounding like those of a target voice. This attack is not aimed at deceiving humans, but at deceiving speaker recognition systems. The transformation is based on morphing phoneme-related features in the Mel frequency cepstrum space [27], which is a representation of sound commonly used as feature inputs for voice recognition systems. Our transformation only requires knowledge of the number of phonemes in the target language and a piece of audio from the victim. We show that such an attack can be improved by using a population of candidate source voices (easily available online), as some voices are better transformed into others. We provide a method of identifying which source voices are likely to succeed in impersonating a target voice.

We evaluate the transformation on different speaker recognition systems. We use a white-box model to learn how to improve the voice conversion and we show that the attack can successfully fool black-box models in both *over-the-wire* and *over-the-air* access. We conduct our evaluation across a different set of assumptions for the adversary, including (i) amount of known audio and (ii) recording noise. We further show that our transformation can be used on verification systems unlocked with a *text-dependent* keyphrase, as well as *text-independent* voice identification systems.

The main contributions of this paper are the following:

- We propose a new voice conversion method based on the creation of a phoneme mapping function between a source and a target voice in MFCC space, which only requires knowledge of the language's number of phonemes.
- We analyze the effectiveness of the transformation across different assumptions regarding quantity and quality of training data, identification and verification use-cases. We show that only few authentication attempts can deceive the proprietary models of the Azure Speaker Recognition APIs.

– We further test the attack over-the-air on the Siri assistant of Apple iOS, achieving success rates of over 50% with only one authentication attempt in the worst case.

2 Related Work

Prior work in this area focused on two different domains. The first consists of attacks on speech recognition, where the aim is to inconspicuously inject malicious commands into voice assistants (e.g., "delete my contact list"). This is done both by generating new audio that is inaudible to the user or by modifying existing audio in a way that is non-perceivable by a human. The second domain consists of attacks on speaker recognition, where an adversary attempts to impersonate a victim's voice. Impersonation attacks are particularly threatening in verification (authentication) scenarios, where a system uses voice-based access control, but can also be dangerous in identification use-cases, where an adversary could fool the system about their identity. We cover both domains in this section.

2.1 Speech Recognition

For speech recognition, several works attempted to create inconspicuous ways to embed or hide speech commands which trigger specific actions on voice assistants. Vaidya et al. [33] introduced obfuscated voice commands, which are recognized and executed by voice systems but unintelligible to a human listener. This is done through repeated application of a Mel Frequency Cepstrum Coeficient (MFCC) based audio mangler, which applies and then inverts the MFCC transform. This work is extended in [7], showing that an attacker can perform this attack even in more realistic conditions, such as black-box models, and in the presence of background noise, while retaining the non-intelligibility of the commands for humans. Yuan et al. [35] demonstrate a technique which allows an adversary to embed speech commands into songs, which allows an attack to be conducted in front of a victim with greater ease. In [8] this is extended to allow any pre-chosen command to be embedded into a given audio sample, such that a human does not perceive the command. This allows a human to hear one utterance but the system to believe it is has heard something entirely different. A slightly different approach is taken in [36], where Zhang et al. demonstrate embedding commands entirely in (human-)inaudible frequency range, but which are still accepted by voice assistants.

Our Work. As opposed to speech recognition, we specifically focus on the security of speaker recognition systems. This means that our work is related to speech recognition, but does not overlap directly. An adversary could design a way to combine our method with a speech recognition attack to bypass impersonation while still achieving inconspicuous malicious transcription. Compared to speech recognition attacks, we **do not require** the audio to have non-intelligible or inaudible properties. Since the adversary wants to be able to interact with the

voice-based system directly to perform some malicious activity, they require temporary unsupervised access to the device, and as such nothing is gained by the audio being imperceptible. We also retain the goal of **correct transcription**, so that keyphrases are still accepted by the system.

2.2 Speaker Recognition

The goal of attacks on speaker recognition systems is to impersonate users, that is to create audio that is incorrectly interpreted by the system as belonging to a specific user. Attacks against speaker recognition systems can be divided into four categories: (i) *impersonation*, (ii) *replay*, (iii) *speech synthesis* and (iv) *voice conversion* [14]. In *mimicry* attacks, human impersonators attempt to alter their own voice in order to mimic another person's voice [23]. Replay attacks involve replaying (with a loudspeaker) audio samples to the system, either in whole or by merging parts of other audio files together [24]. Speech synthesis aims to create a model for generating completely artificial speech. In [10] De Leon et al. proposed a technique based on a Hidden Markov Model (HMM), which adapts a background model in order to derive an audio synthesizer. The analysis show that such a synthesizer can impersonate users in the well known Gaussian Mixture Model (GMM) [3] 81% of the time.

The method presented in this paper falls in the voice conversion category, where the goal is to convert the perceived speaker identity of a given utterance. Many works have addressed the problem of voice conversion and recently the popularity of the Voice Conversion Challenge [32] gave way to a numerous set of works [13, 19, 21, 26, 31]. Approaches such as [26] use a probabilistic mapping of vocal tract models to convert between speakers, where as [21] use a GMM trained on aligned audio from victim and attacker, which can then be applied to the source audio.

Both speech synthesis and voice conversion approaches have been shown to achieve good results in re-creating a person's voice. However, these approaches are designed for non-adversarial scenarios, where large volumes of high-quality audio for each speaker are available to train models. Additionally, voice conversion approaches often require labelled *parallel* training data: both source and target speaker uttering the same known sentences, so that a model can be trained by mapping them on a one-to-one basis. Furthermore, these approaches are generally targeted at fooling human listeners, imposing many constraints on how realistic the voice sounds.

Our Work. We address the voice conversion problem from an adversarial perspective, which brings **limited availability of audio**: both in quantity and in quality (noise). Additionally, we focus on **fooling systems** rather than human listeners. Fooling a human listener would not provide an advantage to the adversary in our case (the system is unsupervised at the time of the attack).

In our approach we learn a phoneme-specific transformation from a source voice to the victim voice, guaranteeing that each phoneme from the source can be transformed to a similarly-sounding phoneme for the victim. This allows the correct transcription to be retained for the speech recognition, while at the same

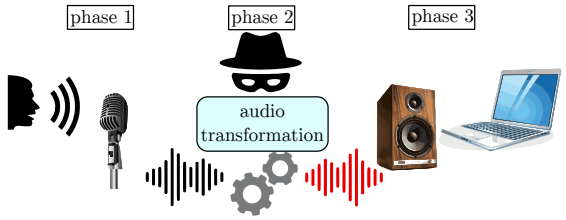


Fig. 1: Threat model. The adversary initially records audio of the victim. This is used to create a transformation, which is applied to some source audio and replayed to the device (e.g., with a loudspeaker).

time transforms the source voice so that it resembles the victim's voice. The method works with limited amounts of text-independent voice data, and even in the presence of noise, and only requires knowledge of the approximate number of phonemes in a language. The goal of our analysis is to evaluate the security of speaker recognition systems in realistic threat scenarios.

3 Threat Model

The phases of an attack are shown in Figure 1. The adversary first records the victim speaking and then constructs a mapping function between another individual's voice and the victim's voice. They then use this function to transform the other individual's voice into the victim's and replays the transformed audio into the system, with the goal of impersonating the victim.

Background. Users interact with a speaker recognition system, either verification or identification. In the case of verification, the system requires users to utter a specific sentence, hereafter *keyphrase*, and uses such utterance to recognize users, in either authentication or identification use cases. As an example, a laptop could use speaker authentication with the "Hey Siri, it's me" keyphrase in order to be unlocked (rather than typing a password). The keyphrase could either be fixed or contain a challenge, such as asking to speak today's date or utter a set of numbers being shown on the screen at the time of authentication.

Capabilities. Adversaries can: (i) record audio of the victims talking, (ii) replay audio to the voice recognition system (e.g., with a loudspeaker).

Knowledge. Following from the capabilities, adversaries have some knowledge of the victims voice trait (from recording audio of them talking). Additionally, the adversary has a set of audio samples containing spoken words for a population of individuals. This can be achieved easily by utilizing free speech datasets such as VoxForge [34]. However, adversaries are limited along the following dimensions:

1. *black-box model*: adversaries do not know what voice processing and recognition algorithms are in place, and thus cannot optimize their attack for a specific method.

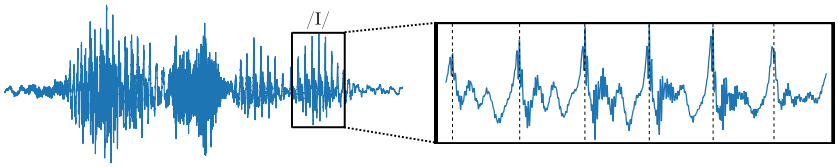


Fig. 2: Sound wave of the utterance "Hey Siri". Within the same phoneme (/ɪ/) a wave pattern repeats itself, depending on the fundamental frequency of voice [15].

2. *recorded utterances*: adversaries cannot record victims uttering the exact keyphrase required for authentication, nor its individual words (not all of them). This is straightforward when the keyphrase includes a challenge, but also reasonable when it does not. Keyphrases are typically designed so that they do not occur in normal day-to-day speech, to avoid unwanted authentications.
3. *audio quality*: adversaries may only be able to record audio in public settings. This means that the recorded audio would have poor quality, as it involves a combination of (i) background noise, (ii) recording from a distance, (iii) recorded audio being emitted by loudspeakers rather than victims themselves.
4. *audio duration*: adversaries can only record the victim for a limited amount of time before raising suspicion. Consequently, they might have a weak representation of victims vocal characteristics, increasing modelling difficulty.

Scenarios. Following from the considerations of the previous paragraphs, we define three different scenarios that represent realistic attack situations.

- *Conference*: the attacker is attending a conference where the victim is giving a talk, and records the victim speaking during their talk. The recorded audio is not of the victim directly, but is a recording of the room speakers connected to the victim's microphone.
- *Cafe*: the attacker is at the same cafe where the victim is enjoying a coffee while having a conversation with other people. The victim's audio is recorded from a distance and is subject to background noise.
- *Ideal*: the attacker obtains high quality audio of the victim from the internet and uses it for their attack. The audio is extracted from a source such as a podcast, or a video of the victim speaking.

All attackers finalize the attack by playing their generated audio to the device. If attackers want to avoid detection, depending on the scenario, they can wait for the device to be left unattended before replaying audio to the device. These adversaries guide our experimental design. We further discuss how we model them in Section 5.

4 Attack Method

Overview. We construct the attack using the concept of phonemes, which are the individually perceivable units of sound in spoken language. We show in Figure 2 how phonemes appear in an audio wave of a spoken word: each phoneme

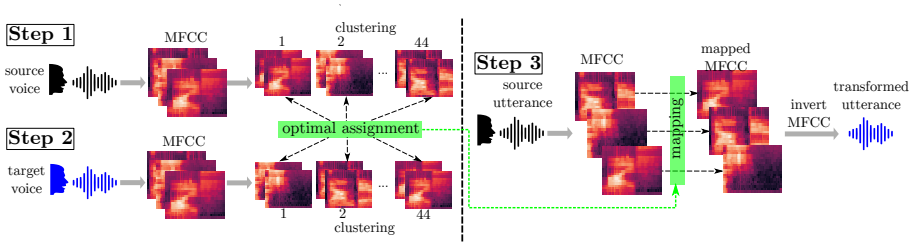


Fig. 3: Steps to craft transformed utterances. In the first and second step the adversary computes the optimal mapping between the source and the target phonemes, in the third step they use the mapping to transform a specific utterance from the source.

is composed of a repeating wave pattern. The attack aims to transform each of these phoneme-related patterns so that they closely resemble the victim's. This is done by deriving a function which maps phonemes spoken by a known speaker into phonemes that resembles those spoken by the victim. The strength of using such an approach is that all knowledge requirements about the structure of the spoken language are removed. This way, an attacker can also afford to ignore the relationship between these phonemes and utterances (i.e., whether a particular phoneme occurs in an audio sample). In fact, no phoneme extraction is necessary, knowing the approximate number of phonemes for the language is sufficient (in spoken British English there are 44 phonemes [17]). We construct the mapping in the MFCC domain, as opposed to modifying the raw audio wave. We show that mapping the outputs of the MFCC extraction and reconstructing the audio wave afterwards is sufficient for the transformation to work.

4.1 Formulation

Given two speakers S and T (*source* and *target*) and a set of known audio recordings produced by them s_i, t_j, the transformation works as follows. Initially, the audio recordings are transformed into the MFCC spectrum, for a single audio file a we obtain a set of samples (due to the windowing process) as follows:

$$\text{MFCC}(a) = \{m_0^{(a)}, \ldots, m_n^{(a)}\} \tag{1}$$

where the number of points n depends on the audio length. We extract MFCC features for all audio recordings a_i belonging to a speaker. Then we use K-means clustering, where K is the number of phonemes in the language, ($K = 44$ in our case) on all the samples (separately for T and S) to infer the clusters $C_k^{(S)}, C_k^{(T)}$, where each cluster represents a phoneme. With the clusters, we also obtain the cluster centroids $C_S = \{s_1, \ldots, s_K\}$ and $C_T = \{t_1, \ldots, t_K\}$.

Afterwards, we compute an optimal mapping between individual cluster centroids from the two sets C_S, C_T. We formulate the optimization as an assignment problem, which we solve with the Hungarian algorithm [22]. We use l_1 as the distance function between two centroids. The output of the mapping consists in

a set of pairs (k, j) where $k, j \in \{1, \ldots, K\}$ and the pair (k, j) indicates that points belonging to cluster k for speaker S should be transformed into points belonging to cluster j for the speaker T to maximize the similarity.

We implement the above transformation using a linear shift in MFCC space. Given $m_i^{(a)} \in C_k^{(S)}$ and given the optimal mapping for cluster k, pair (k, j), we compute a transformed sample $o_i^{(a)}$ as follows:

$$o_i^{(a)} = m_i^{(a)} + t_j - s_k. \tag{2}$$

For an entire audio recording a, Equation 2 is applied sequentially to each sample $m_i^{(a)}$ in $\text{MFCC}(a)$, resulting in a set of transformed samples $\{o_0^{(a)}, \ldots, o_n^{(a)}\}$. Finally we invert the MFCC transformation using the method shown by Ellis [12], to give the transformed audio a^*:

$$a^* = \text{MFCC}^{-1}(\{o_0^{(a)}, \ldots, o_n^{(a)}\}) \tag{3}$$

4.2 Attack Execution

There are three steps to generate the attack audio, shown in Figure 3. In the first step, adversaries compute the phoneme clustering for a source voice, which can be their own. In the second step, they obtain a recording of the target's voice and compute clustering for this data. Immediately afterwards, the adversary can compute the optimal phoneme mappings between the source and the target clusters. In the final step the adversary selects a source utterance, usually the keyphrase or a voice command used by the system, applies the transformation in Equation 2 and creates a transformed utterance audio to be played to the system. The first step can always be computed *offline*, that is before the adversary selects a target, while the remaining steps depend on when the adversary is able to record the victim speaking and when they obtain physical access to the system.

Choice of Source Speaker. We found that the selection of source speaker greatly affects the quality of the transformation, meaning that certain voices can be more accurately mapped to certain targets. We therefore extend our attack to consider a population of individuals as sources, that the adversary can obtain by downloading online voice datasets, or recruiting a population of people to provide a set of potential source voices. This way, the adversary can compute mappings for each individual in the population, and later has several candidates to choose as the source utterance in the last phase of the attack (see Figure 3). As it is reasonable for adversaries to limit the number of *failed attempts* (i.e., playing an attack utterance and being rejected or wrongly classified by the system), one strategy is to estimate the chance of successful impersonation based on the mapping output.

Following these considerations, given a mapping composed of a set of pairs $(k_1, j_1), \ldots, (k_K, j_K)$ we use the sum of the L_1 norm of paired cluster centroids as an indicator:

$$\epsilon = \sum_{i=1}^{K} ||s_{k_i} - t_{j_i}||_1 \tag{4}$$

Intuitively, the lower the distance (error, ϵ) between the mapped clusters, the more accurate the transformation becomes. Therefore, whenever the adversary carries out an attack they sort the possible source voices based on increasing ϵ and use them as sources in this order.

5 Experimental Design

In this section we describe our data collection method, then present how we model the adversaries of Section 3 and describe the target systems considered for the evaluation.

5.1 Data Collection

Collection Procedure. We collected audio data from 20 male native English speakers, recruited mainly through social media and mailing lists. Participants were mostly from southern England, and aged between 18 and 30. Recording sessions took place in an isolated room in a university building, taking approximately 30 minutes. Recordings were conducted using an AmazonBasics Portable USB Condenser Microphone, connected to a Windows laptop. Recordings used the inbuilt "Voice Recorder" software. Participants were instructed to keep the distance between themselves and the microphone between 5 and 15cm. The data collection was approved by an ethical review board at our University, Reference: SSD/CUREC1A_CS_C1A_18_032. Participants were informed of the purpose of the study and informed consent was obtained from them prior to commencing any recording sessions. As voice is personally identifying information, we do not publicly share the voice dataset.
Transcripts. The participants were required to utter sentences from four different categories: (i) conference transcripts, (ii) conversation transcripts, (iii) commands and (iv) enrollment transcripts. Each utterance source is designed to re-create the scenarios mentioned in Section 3. The enrollment and commands transcripts are identical for every participant, while for conference and conversation, to increase the dissimilarity of spoken words, we randomly assign one out of five transcripts to each user[1]. Transcripts were split into utterances of roughly equal length, with an utterance typically containing a single sentence.

5.2 Adversary Modelling

Conference Attacker. This attacker only obtains audio samples coming from utterances from the conference transcripts. In order to recreate the "conference" effect (the recorded audio coming from distant loudspeakers), we apply the following processing to the original audio. First we apply the Freeverb [30] algorithm to generate reverberation in the audio (following *data augmentation* practices used in Kaldi [29]). To simulate recording from a distance, we apply a

[1] Transcript summaries are available in appendix A

low-pass filter (with cutoff at 8Khz) to attenuate higher frequencies, and scale the amplitude of the signal to reduce the volume.

Cafe Attacker. This attacker only obtains audio samples coming from utterances from the conversation transcripts. In order to recreate the "cafe" effect (recording from a distance plus background chatter and noise), we apply the same processing used for Conference Attacker (with less reverberation). Additionally, we mix the audio file with common cafe background noise[2] (the overlaid noise segment is chosen randomly per sample).

Ideal Attacker. This attacker uses the clean recorded audio from the data collection, with no post-processing or noise applied to it. The Ideal Attacker represents a worst-case scenario where the adversary obtains good quality audio samples, and we use it as an indication of the empirical upper bound for the attacker's success rate.

Audio Duration. In order to evaluate the effect of different amounts of audio on the attack success, we model two different audio durations in our experiments: *all* and *one minute*. The *all* case represents the case where we use all audio collected for a given scenario (either conference or cafe). The audio quantity averages 317.7 seconds for the Conference Attacker and 330.5 seconds for the Cafe Attacker, including pre- and post- speech silence. Ideal Attacker uses all the audio available for that victim, giving an average of 648.2 seconds per victim. In the *one minute* case, we randomly sample utterances from the related transcripts until we reach a cumulative total of 60 seconds of audio, including silence parts. We choose to systematically analyze each combination of these, creating six different scenarios (three attackers, two audio lengths).

5.3 Target Systems

We evaluate our experiments against speaker recognition systems, both in the identification and verification use-case. We use three different systems for the evaluation: (i) Spear [20] (ii) Azure Speaker Recognition APIs[3] and (iii) Apple iOS Siri ("Hey Siri"). The Spear toolbox is a set of libraries used to train and evaluate speaker recognition models, which we download and train locally with the VoxForge [34] dataset. Meanwhile, Azure Speaker Recognition only offers online (subscription-based) API access. Microsoft reported that the verification API has performance "competitive with the best published number" and that the identification API has "high precision (above 90%) [which] is obtained at around a 5% rejection rate" [28][4]. Apple iOS Siri provides a real world test of the attack against a widely deployed system, which is used for accessing functions on iOS devices. Apple reports that the end-to-end performance of the system has an imposter acceptance rate of 3.2% [2] and an EER of 4.3% on the speaker recognition task alone (i.e not including keyphrase matching). In all cases, we treat the system as a black-box model: we never change nor adapt the method of Section 4.

[2] https://youtube.com/watch?v=BOdLmxy06H0

[3] https://azure.microsoft.com/en-us/services/cognitive-services/

[4] We conducted our experiments against the Microsoft APIs in January 2019.

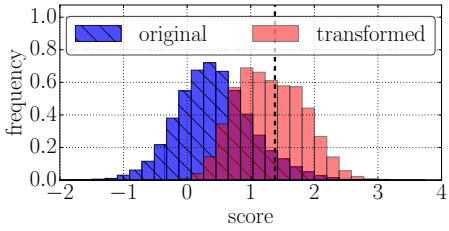


Fig. 4: Frequency distribution of scores for identification before and after the phoneme transformation. Scores move towards the decision boundary after the application of transformation. (Spear).

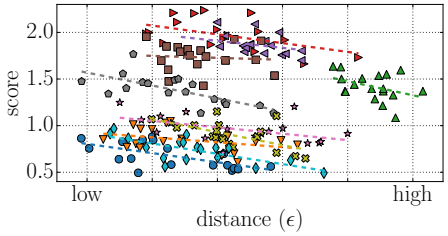


Fig. 5: Similarity score between transformed audio and target user templates, computed by the classifier, as a function of distance between voices. Reduced distance leads to an increase in score (Spear).

6 Experimental Evaluation

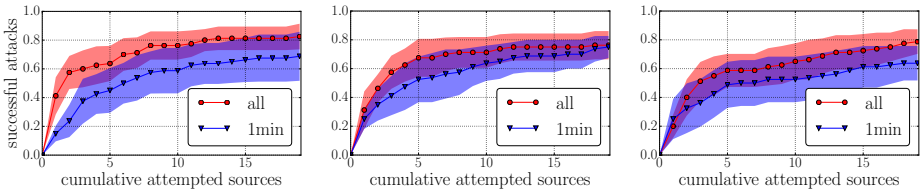
In this section we first show some preliminary results on the Spear system, then show the results on the Azure Speaker API and finally on the Apple iPhone Siri.

6.1 Spear Toolkit

Setup. We use the Spear toolkit to train a GMM-based classifier, with 20 MFCC features plus their first and second derivatives as input features. Throughout our Spear experiments we use audio data obtained from the VoxForge [34] database. Specifically, we use data from users who define themselves as speaking "American English" and take the 63 users with the longest total amount of recorded audio. The users are then randomly split into three groups of 15 plus one of 18: (i) one group for training the background model, (ii) one for refining the model parameters (development set), (iii) one enrolled into the system (test set), and we use the larger (iv) fourth group as voice sources for the attack.

The classifier decides whether an input audio file belongs to certain enrolled user by computing a similarity score between the audio and the enrolled template for every user (identification), with larger scores being closer matches. We compute the EER on the development set by varying the score threshold for acceptance, we find EER to be 7% corresponding to a decision boundary threshold of 1.38. We use the learned threshold on the (unseen) test set to compute the system recognition rates, which leads to a false accept rate of 3.7% and a false reject rate of 0%. Since we are using Spear as a baseline system to quickly evaluate the attack, we only consider the Ideal attacker in this section.

Results. Figure 4 shows two frequency distributions of distance scores from the acceptance decision boundary (vertical dashed line, set at the EER). The original distribution corresponds to distances obtained by testing an impersonation attack with non-modified voice samples (zero-effort attack), all possible source-target pairs (15×18) are used for the visualization. The transformed distribution



(a) Ideal Attacker. (b) Cafe Attacker. (c) Conference Attacker.

Fig. 6: Results of the different attackers on ASV, considering different amounts of audio. Shaded areas show results within one standard deviation, averaged over the four keyphrases.

shows the distance scores for the same samples, but when applying the transformation of Section 4, no population is used in this case. Figure 4 shows that applying the transformation greatly increases the likelihood of the sample lying above the decision threshold and therefore being accepted.

Figure 5 shows how the mapping accuracy affects the success rate of the attack. The figure reports the distance from the decision boundary (score) of transformed samples, as a function of the error ϵ measuring the mapping (in)accuracy (see Section 4.2): lower error is correlated with higher matching score ($r = .48$). In Figure 5, each marker identifies all the data points related to a particular source voice (i.e., for source voice i, each $i \rightarrow j$ transformation with j being a target voice). For each source, we fit a linear regression curve to highlight this trend and we can see that as the distance (error) ϵ increases, the score of transformed samples decreases. Figure 5 also shows how some victim voices are more vulnerable to being impersonated than others, with clusters of higher scoring points belonging to some victims. In the next section we build on these results to evaluate the attack against the Azure APIs.

6.2 Azure Speaker Verification

Setup. The idea behind this experiment is to see whether the attack can be successfully conducted against a commercially available API, with a proprietary model for speaker verification. The Azure Speaker Verification API (hereafter ASV) is text-dependent and has a set of keyphrases that can be used with it. We collected audio of five of these keyphrases, which we require each participant to speak four times. Each user is enrolled using four samples of a given phrase (ASV requires at least three samples). There are no parameters within ASV to modify its performance, and as such no way to adjust any thresholds associated with acceptance or rejection[5].

We generate attack samples for these keyphrases using each of our participants as a victim, and using all the remaining participants as source voices, for

[5] We had to remove one phrase, "Houston we have had a problem", as participants spoke the phrase as "Houston we have a problem", a popular misconception.

	Ideal		Conference		Cafe	
Keyphrase	**1 min**	**all**	**1 min**	**all**	**1 min**	**all**
KP_1:"my voice is stronger than passwords"	26.3%	52.6%	47.4%	57.9%	42.1%	57.9%
KP_2:"my password is not your business"	68.4%	94.7%	84.2%	89.5%	89.5%	89.5%
KP_3:"apple juice tastes funny after toothpaste"	21.1%	42.1%	15.8%	31.6%	21.1%	42.1%
KP_4:"you can activate security system now"	63.2%	73.7%	31.6%	52.6%	47.4%	73.7%

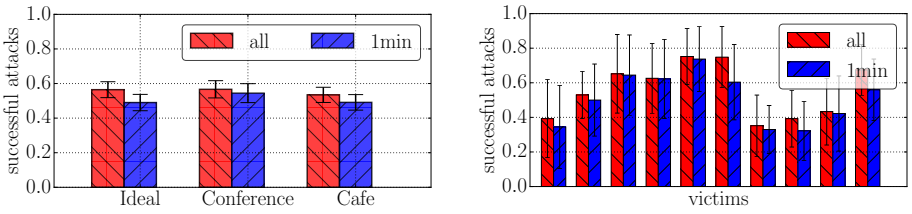
Table 1: Percentage of successful attacks using up to three source voices on ASV, computed for all scenarios and keyphrases.

each of our scenarios in turn. As we have four repetitions of each keyphrase, the attacker performs four authentication attempts for one source before moving to the next source. We submit each of these attack samples to ASV and receive a reject/accept response. Across all scenarios we create and evaluate a total of 38,400 attack samples, which we use to evaluate the performance of our attack.

Results. Table 1 shows the results of verification experiments, for each scenario and keyphrase. The values in Table 1 are the percentages of successful impersonation attacks, which are calculated in the following way: the adversary attempts impersonation with the first three sources in the ϵ-ranked list (see Section 4), if any of these are successful then we count this as a successful attack.

There is significant variability in the results between different keyphrases: KP_2 obtains the highest success rate on average (85%), while KP_3 performs the lowest (28%). This might be related to the mapping accuracy of the phonemes that form these utterances, which degrades when some phonemes are under-represented (i.e., they occur in low number) in the known victim audio. For example, the phonemes [dʒ], [ʊ] and [θ] all occur in KP_3, and are the 7^{th}, 4^{th} and 3^{rd} least common phonemes respectively [6], and therefore likely to be under represented. We see differing success rates across scenarios and amount of known audio, with the one minute audio scenario performing consistently worse (-16%) than the all audio scenario. Ideal Attacker performs the best, but even the noisy audio of Conference and Cafe Attacker achieves high success rates.

Figure 6 shows the cumulative successful attacks as the adversary attempts impersonation with each source voice in his dataset (sources are ranked by ϵ). Unexpectedly, Cafe and Conference attacker do not seem to greatly suffer from the additional audio noise in comparison to Ideal. This suggests that even noisy recordings of the victim audio might carry sufficient information about his vocal tracts and further confirms that most of the distinctiveness of one's voice comes from lower frequencies, which best survive noise during the recording. The plots additionally show how one minute of audio is also sufficient (though with a slight decrease in success rate when compared to all audio) to re-create one's voice. The curve slope indicates that the ranking of possible sources brings a greater percentage of successes in the beginning, where promising sources are tested first. We can see that at around three attempted sources (corresponding to 12 authentication attempts), the adversary can get up to 60% success rate depending on the scenario. Even if there are only marginal increments in the successful attacks after testing 15 sources, using a larger population of sources



(a) Scenario analysis. (b) Per-victim Analysis.

Fig. 7: Average successful impersonations on ASI. Results show that changes in audio quantity and quality only have small effects on success rate. Plots show the successful attacks for each scenario.

would increase overall attack effectiveness, as this increases the likelihood of having promising source voices, which can be mapped accurately to the victim.

6.3 Azure Speaker Identification

Setup. The Azure Speaker Identification API (hereafter ASI) is text-independent and requires a set of users to be enrolled, which are candidate users for who is speaking. In this case, enrollment requires a minimum of 30 seconds of audio per speaker, once silence is removed. To enroll users, we use audio specifically collected for this purpose, enrolling half of our participants in the system (see Appendix A for details). This gives us 10 potential victims, and 10 attackers, for a total of 100 source-victim pairs.

ASI accepts an audio sample as input and replies with the inferred identity from the list of the 10 enrolled users, or an *empty* reply when an audio sample does not match any of them. It is not possible to adjust any threshold for ASI, and as such there is no way of adjusting the threshold for when empty set is returned. We send all command utterances that have been transformed between a particular source and victim to ASI, but we concatenate audio files together into groups of four to obtain audio samples of approximately 8 seconds. This is because ASI is designed for longer audio samples, and without this concatenation the system returns none, as the samples are too short to make a decision. In total we submit 5,400 requests to ASI to conduct our experiments.

Results. Figure 7a shows the overall success rate for ASI for the three attackers and the two audio length combinations. In this use case each success corresponds to a submitted audio sample that is identified by the system as belonging to the victim. We see that the performance is broadly consistent across the scenarios, with a slight worsening of the recognition rates for the Cafe Attacker in particular (though not statistically significant, averaged over the 100 source-victim pairs) The performance slightly decreases in the one minute of audio case, but again with a minimal effect in the overall success.

Interestingly our results also reveal more information about ASI and its sensitivity. Table 2 shows that for all scenarios ASI was more likely to assign a

	Ideal		Conference		Cafe	
	1 min	**all**	**1 min**	**all**	**1 min**	**all**
Misclassified	31.1%	27.6%	28.1%	26.4%	31.9%	29.2%
None	19.9%	16.0%	17.4%	16.9%	19.0%	17.3%

Table 2: Percentage of incorrect and *empty* responses for the experiment on ASI, for each attacker and audio duration. ASI returns *empty* when the provided audio does not match any of the enrolled users. We report Misclassified whenever the returned identity does not match the target victim.

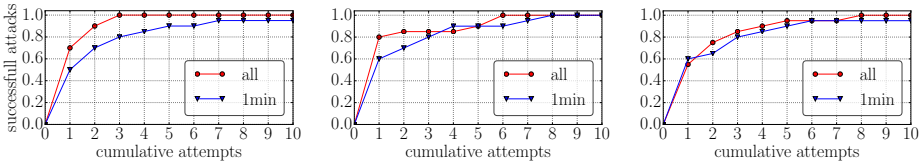
speaker to an incorrect label than it was to return the empty user classification. This suggests that the decision boundaries across different users are not very conservative and that generally they can not deal well with outliers.

Figure 7b shows how the successful attacks distribute over different victims. The plot highlights that certain voices are more vulnerable to this type of attack than others: comparing the hardest to attack with the easiest to attack victim we get a difference of around 40% in the success rate. Similar uneven distributions of rates have been noticed before in previous work [1, 11]. This suggests that some voices might be inherently harder to replicate, however, in our data, this might be due to a sample bias: some voices might significantly differ from our "average voice". A larger dataset would be required to investigate further whether this is the case.

6.4 Apple iPhone's Siri

Setup. In order to measure the capability of the attack of being conducted over-the-air we test the samples against the voice activation functionality of the Apple's Siri digital assistant on an iPhone 6S, running iOS version 12.2. We use the collected voice recordings of each of our 20 participants to enrol them onto the device. For both enrolment and attacks, we use a Bose SoundLink Mini 2 speaker to replay the participants audio samples. The speaker is placed 6 centimeters away from the smartphone in an office environment. Initial enrolment requires the user to pronounce four different phrases, which we construct by combining the original recordings of the collected "Hey Siri" utterance with the remaining words of the enrollment utterance added by splicing together audio from other recordings of the same individual. Siri speaker recognition updates the user template after a successful access [2]. Therefore, after a successful attack we erase the user profile and repeat the enrollment process.

We test the system along the same dimensions as our previous experiments. When conducting the attack, we play a single transformed utterance of the keyphrase ("Hey Siri"), from each source voice, in the order suggested by our error function (nearest to furthest). If Siri activates, i.e., the voice is recognized as belonging to the legitimate user, we consider the attack successful and we do not present further samples. At the time of writing, Apple claimed Siri had an imposter accept rate of 3.2% [2].



(a) Ideal Attacker. (b) Cafe Attacker. (c) Conference Attacker.

Fig. 8: Results for different attackers on Siri. Plots show ratio of successful impersonations as the adversary consecutively attempts the attack with different source voices.

Results. Figure 8 shows the percentage of victims successfully impersonated after a given number of attempts, for each attacker and the two known audio amounts. The results show that performance is consistent with previous experiments, in that the differing scenarios lead to slightly worse success rate, and that performance is also worse in the one minute audio case. Our results demonstrate that the Siri voice activation is easily fooled by our attack. For all scenario and amount of known audio combinations over 70% of victims can be attacked in three attempts or fewer. Excluding one individual in two of the one minute scenario-time combinations, who could not be impersonated, all other attacks were successfully conducted in 8 attempts or fewer. In our dataset an utterance of "Hey Siri" took approximately 2 seconds, meaning that in most cases 20 seconds would suffice to successfully carry the attack out.

7 Discussion

Implications. The attack presented in this paper shows that a minimal amount of voice from a victim can be sufficient for an adversary to impersonate that victim with a high success rate. The attack's only requirement is to obtain a recording of the victim talking. Sources such as social media, podcasts and recordings of public speaking events are all easily available sources of such audio. Consequently, the audio becomes even easier to gather for higher profile targets. The ease of collection of voice samples in adversarial scenarios brings an inherent security vulnerability of voice-based systems, as highlighted by our analysis. We point out this vulnerability in order to raise awareness of the limitations of such authentication mechanisms, so that they can be accounted for during the the design of voice-based systems.

Our analysis highlights the weaknesses of voice-based authentication (and identification) in adversarial scenarios. This is not strictly due to the recognition algorithms themselves but rather to the availability and ease of collection of voice biometric samples. We point out this vulnerability in order to raise awareness to the limitations of such authentication mechanisms, so that these can be accounted for during the the design of voice-based systems.

Replay Detection. Similarly to other voice-based attacks, our method involves replaying audio to the system microphone via a speaker. This is necessary for

all attacks on voice systems that use only over-the-air interaction, and do not require harder to obtain over-the-wire access. A set of works have addressed the detection of replay attacks on such systems [4,5,9,13,14,37]. Some detection techniques rely on a combination of better hardware (e.g., multiple microphones) or require additional interactions from the user. Often replay detection evolves into an arms race with the adversaries improving their audio sample to present the features required to bypass detection. This work is orthogonal to replay detection, which could be bypassed with enough investment from an attacker.

Rate-Limiting. Oftentimes in verification systems, the number of failed authentication attempts can be used to temporarily block the authentication or swap it with more secure alternatives. For example, in Apple FaceID the face recognition is disabled after five failed authentication attempts, at which point a PIN is required to unlock the phone. We find that even if the 5-attempts limit were the same for Siri, a high percentage of victims would still be attack-able (90% in the 1 minute ideal scenario). Keeping the number of sequentially allowed failures low before locking the system becomes an immediate an effective way to prevent our and other population-based attacks.

8 Conclusions

In this paper, we describe a method to transform a source voice into a victim's voice to deceive speaker recognition systems. The transformation maps individual phonemes between the source and target voices and only requires knowledge of the number of language phonemes, a set of source voices (easily available online) and an audio sample of the victim speaking. Furthermore, we identify a metric for determining which voice among a group of voices is most likely to lead to a successful authentication.

We evaluate the attack under a set of scenarios that include different amounts and quality of victim audio and different systems. We test our attack on both the Azure Speaker Recognitions APIs and the Apple iOS Siri voice assistant. On Azure, for verification, we show that 12 authentication attempts are sufficient to successfully impersonate victims in 40% up to 68% of cases, using just one minute of victim audio for training, even in noisy recordings conditions. For identification, the method achieves much higher success rates reaching over 50% on average with a single attempt. We demonstrate that high success rates can be obtained even when testing the attack over-the-air on Siri: 80% of victims can be impersonated within three attempts, which correspond to only 8 seconds of audio in total.

Compared to previous work, these findings reveal that limited quantity and quality of audio have only limited impacts on the overall success of this attack. Given the increasing availability of potential victim's audio, our analysis highlights the vulnerability of using voice as a biometric for access control in adversarial settings, suggesting that such weakness should be included in the design phase of such systems.

Acknowledgements This work was supported by a grant from Mastercard and the Engineering and Physical Sciences Research Council [grant numbers EP/N509711/1 and EP/P00881X/1].

References

1. Allix, K., Bissyandé, T.F., Klein, J., Le Traon, Y.: Are your training datasets yet relevant? In: International Symposium on Engineering Secure Software and Systems. pp. 51–67. Springer (2015)
2. Apple Siri Team: Personalized Hey Siri - Apple (2018), `https://machinelearning.apple.com/2018/04/16/personalized-hey-siri.html`, [Accessed 2019-07-08]
3. Bimbot, F., Bonastre, J.F., Fredouille, C., Gravier, G., Magrin-Chagnolleau, I., Meignier, S., Merlin, T., Ortega-García, J., Petrovska-Delacrétaz, D., Reynolds, D.A.: A tutorial on text-independent speaker verification. EURASIP Journal on Advances in Signal Processing (4), 101962 (2004)
4. Blue, L., Abdullah, H., Vargas, L., Traynor, P.: 2MA: Verifying Voice Commands via Two Microphone Authentication. In: Proceedings of the 13th on Asia Conference on Computer and Communications Security. pp. 89–100. ACM (2018)
5. Blue, L., Vargas, L., Traynor, P.: Hello, is it me you're looking for?: Differentiating between human and electronic speakers for voice interface security. In: Proceedings of the 11th Conference on Security & Privacy in Wireless and Mobile Networks. pp. 123–133. ACM (2018)
6. Blumeyer, D.: Relative frequencies of english phonemes (2012), `https://cmloegcmluin.wordpress.com/2012/11/10/relative-frequencies-of-english-phonemes/`, [Accessed 2019-04-27]
7. Carlini, N., Mishra, P., Vaidya, T., Zhang, Y., Sherr, M., Shields, C., Wagner, D., Zhou, W.: Hidden voice commands. In: Proceedings of the 25th USENIX Security Symposium. pp. 513–530 (2016)
8. Carlini, N., Wagner, D.: Audio adversarial examples: Targeted attacks on speech-to-text. In: IEEE Security and Privacy Workshops. pp. 1–7. IEEE (2018)
9. Chen, S., Ren, K., Piao, S., Wang, C., Wang, Q., Weng, J., Su, L., Mohaisen, A.: You can hear but you cannot steal: Defending against voice impersonation attacks on smartphones. In: Proceedings of the 37th International Conference on Distributed Computing Systems. pp. 183–195. IEEE (2017)
10. De Leon, P.L., Pucher, M., Yamagishi, J., Hernaez, I., Saratxaga, I.: Evaluation of speaker verification security and detection of HMM-based synthetic speech. Transactions on Audio, Speech and Language Processing (2012)
11. Eberz, S., Rasmussen, K.B., Lenders, V., Martinovic, I.: Evaluating Behavioral Biometrics for Continuous Authentication. In: Proceedings of the 12th Asia Conference on Computer and Communications Security. pp. 386–399 (2017)
12. Ellis, D.P.W.: PLP and RASTA (and MFCC, and inversion) in Matlab (2005), `http://www.ee.columbia.edu/~dpwe/resources/matlab/rastamat/`, [Accessed 2019-07-08]
13. Ergünay, S.K., Khoury, E., Lazaridis, A., Marcel, S.: On the vulnerability of speaker verification to realistic voice spoofing. In: Proceedings of the 7th International Conference on Biometrics Theory, Applications and Systems. pp. 1–6. IEEE (2015)
14. Evans, N., Kinnunen, T., Yamagishi, J.: Spoofing and countermeasures for automatic speaker verification. Proceedings of the Annual Conference of the International Speech Communication Association pp. 925–929 (2013)

15. Fant, G.: Acoustic theory of speech production: with calculations based on X-ray studies of Russian articulations. No. 2, Walter de Gruyter (1970)
16. Google: Set up Voice Match on Google Home - Google Home Help (2018), https://support.google.com/googlehome/answer/7323910, [Accessed 2019-07-08]
17. Helland, T., Kaasa, R.: Dyslexia in english as a second language. Dyslexia **11**(1), 41–60 (2005)
18. HSBC: Voice ID — HSBC UK (2018), https://www.hsbc.co.uk/1/2/voice-id, [Accessed 2019-07-08]
19. Hsu, C.C., Hwang, H.T., Wu, Y.C., Tsao, Y., Wang, H.M.: Voice conversion from non-parallel corpora using variational auto-encoder. In: Proceedings of the Signal and Information Processing Association Annual Summit and Conference. pp. 1–6. IEEE (2016)
20. Khoury, E., El Shafey, L., Marcel, S.: Spear: An open source toolbox for speaker recognition based on Bob. In: Proceedings of the International Conference on Acoustics, Speech and Signal Processing. pp. 1655–1659. IEEE (2014)
21. Kinnunen, T., Wu, Z.Z., Lee, K.A., Sedlak, F., Chng, E.S., Li, H.: Vulnerability of Speaker Verification Systems Against Voice Conversion Spoofing Attacks: the Case of Telephone Speech. In: Proceedings of the International Conference on Acoustics, Speech and Signal Processing. pp. 4401–4404. IEEE (2012)
22. Kuhn, H.W.: The hungarian method for the assignment problem. Naval research logistics quarterly **2**(1-2), 83–97 (1955)
23. Lau, Y.W., Tran, D., Wagner, M.: Testing Voice Mimicry with the YOHO Speaker Verification Corpus. In: Proceedings of the 9th International Conference on Knowledge-Based Intelligent Information And Engineering Systems. vol. 3584, pp. 15–21 (2005)
24. Lindberg, J., Blomberg, M.: Vulnerability in speaker verification-a study of technical impostor techniques. In: Proceedings of the 6th European Conference on Speech Communication and Technology (1999)
25. Lloyds Bank: Voice ID — Lloyds Bank (2019), https://www.lloydsbank.com/contact-us/voice-id.asp, [Accessed 2019-07-08]
26. Matrouf, D., Bonastre, J.F., Fredouille, C.: Effect of speech transformation on impostor acceptance. In: Proceedings of the 31st International Conference on Acoustics Speech and Signal Processing. vol. 1. IEEE (2006)
27. Mermelstein, P.: Distance measures for speech recognition, psychological and instrumental. Pattern recognition and artificial intelligence **116**, 374–388 (1976)
28. Microsoft ML Blog Team: Now available: Speaker & video apis from microsoft project oxford, https://blogs.technet.microsoft.com/machinelearning/2015/12/14/now-available-speaker-video-apis-from-microsoft-project-oxford/
29. Povey, D., Ghoshal, A., Boulianne, G., Burget, L., Glembek, O., Goel, N., Hannemann, M., Motlicek, P., Qian, Y., Schwarz, P., et al.: The kaldi speech recognition toolkit. In: Proceedings of the 2011 workshop on automatic speech recognition and understanding. IEEE (2011)
30. Smith, J.O.: Physical audio signal processing, https://ccrma.stanford.edu/~jos/pasp/Freeverb.html, [Accessed 2019-07-08]
31. Sun, L., Li, K., Wang, H., Kang, S., Meng, H.: Phonetic posteriorgrams for many-to-one voice conversion without parallel data training. In: Proceedings of the 2016 International Conference on Multimedia and Expo. pp. 1–6. IEEE (2016)
32. Toda, T., Chen, L.H., Saito, D., Villavicencio, F., Wester, M., Wu, Z., Yamagishi, J.: The voice conversion challenge 2016. In: Proceedings of the Annual Conference of the International Speech Communication Association (2016)

33. Vaidya, T., Zhang, Y., Sherr, M., Shields, C.: Cocaine noodles: exploiting the gap between human and machine speech recognition. In: Proceedings of the 9th USENIX Workshop on Offensive Technologies (2015)
34. Voxforge Dataset: Free speech... recognition, http://www.voxforge.org/, [Accessed 2019-07-08]
35. Yuan, X., Chen, Y., Zhao, Y., Long, Y., Liu, X., Chen, K., Zhang, S., Huang, H., Wang, X., Gunter, C.A.: Commandersong: A systematic approach for practical adversarial voice recognition. In: Proceedings of the 27th USENIX Security Symposium. pp. 49–64 (2018)
36. Zhang, G., Yan, C., Ji, X., Zhang, T., Zhang, T., Xu, W.: Dolphinattack: Inaudible voice commands. In: Proceedings of the 24th SIGSAC Conference on Computer and Communications Security. pp. 103–117. ACM (2017)
37. Zhang, L., Tan, S., Yang, J., Chen, Y.: Voicelive: A phoneme localization based liveness detection for voice authentication on smartphones. In: Proceedings of the 23rd SIGSAC Conference on Computer and Communications Security. pp. 1080–1091. ACM (2016)

A Audio Collected

A.1 Commands

Command data was sourced as both utterances that could be presented to systems in existence, as well as commands used specifically by the Azure Speaker recognition system for verification. The utterances recorded were as follows:

1. Hey Siri (Repeated 4 times)
2. Ok Google (Repeated 4 times)
3. What is the weather like?
4. What time is it?
5. Who am I?
6. How tall is the shard?
7. My voice is stronger than passwords (Repeated 4 times)
8. My password is not your business (Repeated 4 times)
9. Apple juice tastes funny after toothpaste (Repeated 4 times)
10. Houston we have had a problem (Repeated 4 times)
11. You can activate security system now (Repeated 4 times)
12. My voice is my password (Repeated 4 times)

A.2 Conference

Conference talk transcripts were obtained from popular TED talks. The transcripts were shortened, so that they contained approximately the first 6 minutes of a given talk. The transcripts were then split into individual utterances, with each utterance being recorded as a separate audio file by the participant. Five different conference talk transcripts were used, which are the following:

1. Do schools kill creativity? by Sir Ken Robinson - www.ted.com/talks/ken_robinson_says_schools_kill_creativity/transcript

2. Your body language may shape who you are by Amy Cuddy - www.ted.com/talks/amy_cuddy_your_body_language_shapes_who_you_are/transcript
3. What makes a good life? by Robert Waldinger - www.ted.com/talks/robert_waldinger_what_makes_a_good_life_lessons_from_the_longest_study_on_happiness/transcript
4. How great leaders inspire action by Simon Sinek - www.ted.com/talks/simon_sinek_how_great_leaders_inspire_action/transcript
5. The power of vulnerability by Brené Brown - www.ted.com/talks/brene_brown_on_vulnerability/transcript

A.3 Cafe

Our conversation audio is derived from TED talks where two people are having a conversation. A single speakers audio was extracted from each transcript, and the transcript was shortened until it was approximately 6 minutes in length. Five different conversation transcripts were used, which were dervied from the following talks:

1. SpaceX's plan to fly you across the globe in 20 minutes - Gwynne Shotwell - https://www.ted.com/talks/gwynne_shotwell_spacex_s_plan_to_fly_you_across_the_globe_in_30_minutes/transcript
2. How Netflix changed entertainment - Reed Hastings - https://www.ted.com/talks/reed_hastings_how_netflix_changed_entertainment_and_where_it_s_headed/transcript
3. Mammoths resurrected, geoengineering and other thoughts from a futurist - Stewart Brand - https://www.ted.com/talks/stewart_brand_and_chris_anderson_mammoths_resurrected_geoengineering_and_other_thoughts_from_a_futurist/transcript
4. The future we're building and boring - Elon Musk - https://www.ted.com/talks/elon_musk_the_future_we_re_building_and_boring/transcript
5. What everyday citizens can do to claim power on the internet - Fadi Cehadé - https://www.ted.com/talks/fadi_chehade_what_everyday_citizens_can_do_to_claim_power_on_the_internet/transcript

A.4 Enrolment

Enrolment audio was used to enroll individual speakers with the Azure Speaker Recognition API for identification. Participants were asked to read the first 6 paragraphs of the speech given by UK Prime Minister David Cameron at the start of the London 2012 Olympics. The speech can be found on the UK government speeches website at the following URL: https://www.gov.uk/government/speeches/pms-speech-at-olympics-press-conference