2019 IEEE Security and Privacy Workshops (SPW)

You Talk Too Much: Limiting Privacy Exposure via Voice Input

Tavish Vaidya
Georgetown University

Micah Sherr
Georgetown University

***Abstract*—Voice synthesis uses a voice model to synthesize arbitrary phrases. Advances in voice synthesis have made it possible to create an accurate voice model of a targeted individual, which can then in turn be used to generate spoofed audio in his or her voice. Generating an accurate voice model of target's voice requires the availability of a corpus of the target's speech.**

This paper makes the observation that the increasing popularity of voice interfaces that use cloud-backed speech recognition (e.g., Siri, Google Assistant, Amazon Alexa) increases the public's vulnerability to voice synthesis attacks. That is, our growing dependence on voice interfaces fosters the collection of our voices. As our main contribution, we show that voice recognition and voice accumulation (that is, the accumulation of users' voices) are separable. This paper introduces techniques for locally *sanitizing* voice inputs before they are transmitted to the cloud for processing. In essence, such methods employ audio processing techniques to remove distinctive voice characteristics, leaving only the information that is necessary for the cloud-based services to perform speech recognition. Our preliminary experiments show that our defenses prevent state-of-the-art voice synthesis techniques from constructing convincing forgeries of a user's speech, while still permitting accurate voice recognition.

I. Introduction

A person's voice is an integral part of his or her identity. It often serves as an implicit authentication mechanism to identify a remote but familiar person in a non-face-to-face setting such as a phone call. The ability to identify a known person based on their voice alone is an evolutionary skill (e.g., enabling a child to quickly locate its parents) and is an ingrained and automated process that requires little conscious effort [31].

That humans regularly authenticate each other based solely on voice lends to a number of potential impersonation attacks, which notably include voice spearphishing and various other forms of social engineering. The ease at which such attacks can be conducted has increased due to advances in speech synthesis. Emerging services such as Adobe Voco [1], Lyrebird.ai [15, 16] and Google WaveNet [13] aim to produce artificial speech in a person's voice that is indistinguishable from that person's real voice. Surprisingly, producing believable synthetic speech does not require a large corpus of audio data. For example, it has been reported that Adobe Voco can mimic a person's speech with as little as 20 minutes of the targeted speaker's recordings [1, 2], and Lyrebird.ai can create a digital version of a voice from a one minute speech sample.

Advances in voice synthesis open up a large number of potential attacks. An adversary who has access to a speech sample of a target victim could apply voice synthesis to authenticate as the victim to banks and other commercial entities that rely on voice authentication [19–21]. Forged speech could also be used to impugn reputations (e.g., for political gain) or plant false evidence. In general, voice synthesis poses a significant security threat wherever voice is used as an authenticator.

A core requirement of such attacks is that the adversary must have access to a corpus of voice recordings of its target.

The ability to obtain such samples is buoyed by the rising popularity of voice input. Voice input has become ubiquitous and a common method of computer-human interaction, in no small part because it is a natural (to humans) method of communication. Smartphones, tablets, wearables and other IoT devices often come equipped with voice assistants (VAs) such as Alexa, Siri, Google Now and Cortana. Dedicated VA devices such as Amazon Echo and Google Home have found their way into living rooms, constantly listening to users' voice input and providing quick responses. Users of these devices regularly surrender their voice data, making them more vulnerable to future voice synthesis attacks.

Currently, only the voice assistant service providers have access to the voice samples of a user. However, it is unclear due to conflicting reports whether the application developers will get access to user's voice samples [3, 8]. For example, it had been reported that Google Home allowed access to raw voice command audio to application developers while Amazon Echo also plans to do so in the future [8]. Thus, the increased use of voice input increases the opportunities to gain access to raw voice samples of the users.

This paper aims to reduce the threat of voice synthesis attacks for ordinary users. We concede that much voice data is already in the public domain—certainly, it is not difficult to obtain audio recordings of celebrities and politicians, or of ordinarily users who post their own video or audio content to publicly accessible social media (e.g., YouTube). Such users are already vulnerable to voice synthesis attacks and the techniques that we propose in this paper unfortunately do not attempt to protect them. Rather, our aim is to present wide-scale vulnerability to voice synthesis attacks by changing the norm – that is, by permitting the use of voice-based services (e.g., VAs) while preventing the collection of users' raw (unmodified) voice inputs.

We propose a defense that prevents an adversary with access to recordings of voice commands, issued by users to VAs, from

DOI 10.1109/SPW.2019.00026

building a voice model of a targeted user's voice. Our proposal is based on the following two observations:

1) *A user does not need to sound like herself to use a voice assistant.* The first step in generating a response to a user's voice command is conversion of speech to text, i.e., speech recognition. Modern speech recognition systems are oblivious to unique characteristics of a person's voice, and thus, are able to transcribe audio from thousands of users. Therefore, altering a user's voice so that it does not sound like the user herself does not prevent her from using VAs[1].
2) *Speech recognition systems do not need all the information present in spoken audio.* The first step in speech recognition is usually a feature extraction step that converts the high dimensional input audio into low dimensional feature vectors which are then used as inputs to machine learning models for transcribing the audio. Removing some of the information from the high dimensional audio, that is anyway thrown away during the feature extraction, will not affect the speech recognition process but can be used to alter the voice characteristics of the audio.

In brief, our proposed defense extracts audio information from voice commands that are relevant for speech recognition while perturbing other features that represent unique characteristics of a user's voice. Put plainly, we strip out identifying information in audio, which significantly hinders (if not makes impossible) the task of speech synthesis. Our approach could be applied locally—in particular, on smartphones and smartspeaker devices—as a "security filter" that prevents third parties (whether they be the speech recognition service itself, third-party developers, or even network eavesdroppers) from being able to construct convincing synthesized voices. Additionally, our proposed defense has the benefit that it does not require any modifications to the cloud-based speech recognition systems.

In what follows, we describe our initial design and prototype of our defense. Our preliminary experiments, including a small (IRB-approved) user-study, show that our proposed approach prevents the constructing of convincing voice synthesis models while imposing minimal effects on the accuracy of speech recognition.

II. Related Work

We believe we are the first to propose filtering raw voice audio data for the purposes of thwarting voice synthesis attacks. However, existing work has proposed several approaches for achieving *privacy-preserving* voice recognition:

Smaragdis et al. [37] propose a privacy-preserving speech recognition system as an instance of secure multiparty computation, where one party (the transcriber) has a private model for performing speech recognition while the other parties have private audio data that need to be transcribed without revealing the audio content to the transcriber. However, their work does not describe the performance or accuracy of such a system and is limited to HMM-based speech recognition systems. Additionally, secure-multiparty computation is computationally expensive and requires both parties to cooperate. In contrast, our approach can be deployed locally and does not require any changes to existing speech recognition services.

Pathak et al. [33] provide a number of techniques for privacy-preserving speech processing. They describe various frameworks that aim to make conventional speech processing algorithms based on statistical methods, such as HMM, privacy-preserving by computing various operations via secure operations such as secure multiparty computations, additive secret sharing, and secure logsum. Their techniques are impressive, but suffer from practical limitations due to their dependence on computationally expensive cryptography. Their framework also does not achieve good speech recognition accuracy; in contrast, our defense is intended for advanced and (arguably) accurate services such as Google's, Apple's, and Microsoft's cloud-based speech recognition systems.

Ballesteros and Moreno propose scrambling of a private speech message to a non-secret target speech signal using a secret key, which the receiver unscrambles using the same shared secret [27]. The target speech signal's plaintext is different from that of the secret message, so as to fool an eavesdropping adversary. However, the technique requires both cooperation between the sender and receiver of the scrambled signal as well as out-of-band key sharing.

More generally, techniques that prevent speech recognition services from learning the transcription (e.g., via secure multiparty computation) are not applicable to our problem, since in our setting, transcriptions are *required* by the service provider to respond to voice commands. All major existing VA systems (including Google Home, Amazon Alexa, and Siri) use proprietary, cloud-based speech recognition; it is unlikely that these services would choose to deploy expensive and poorly scalable cryptographic-based protocols. In contrast, our proposed defense aims to improve the privacy of voice data for existing and already deployed systems that are widely used by millions of users worldwide without requiring any changes to the speech recognition systems.

Most relevant to this paper are recent studies by Vaidya et al. [38] and Carlini et al. [29]; there, the authors show the feasibility of specially crafting audio that is intelligible to computer speech recognition services but not to human listeners. Our work borrows the use of MFCCs to extract audio information from their approach, and removes additional audio features that provide uniqueness to a person's voice.

III. Threat Model

We consider an adversary whose goal is to impersonate a targeted individual's voice. The adversary achieves its goal of generating spoofed audio in the target's voice by building an

[1]The speaker based personalization supported by various VAs is not hampered by such alteration, since the speaker detection is done locally on the client device (e.g., smartphone) and only applies to the activation keywords for the voice assistants.

accurate model of his voice by using speech synthesis services such as Adobe Voco or Lyrebird.ai. Crucially, to be successful, the adversary needs to first collect a corpus of the target user's speech.

Acquiring voice samples: Our threat model assumes that the adversary requires high quality voice speech samples of the target to build its voice model. As an example means of collecting voice samples, an adversary could create a (legitimate) voice application[2] for a voice assistant, which provides raw voice command audio data to the application. Alternatively, a speech recognition service may itself be malicious and/or sell users' speech data to other parties. Finally, if speech is transmitted unencrypted (which hopefully is a rarity) during a voice-over-IP call, a network eavesdropper could trivially collect a corpus.

We emphasize that in this paper, we explicitly do not consider voice collection from in-person conversations, postings of audio on social media websites (e.g., YouTube), broadcast media (e.g., TV), or other sources. We acknowledge that highly skilled and committed adversaries can likely obtain audio of a specific person, for example, by physically planting a listening device near the target. Our goal is to change the norm such that the collection of ordinary users' audio is much more difficult. Specifically, we want to enable ordinary users to use VAs while minimizing their risk to voice synthesis attacks.

Generating voice models: Our threat model assumes that the adversary has access to services such as Adobe Voco or Lyrebird.ai that can be used to create a voice model of a person's voice from the acquired voice samples.

Lyrebird.ai, at its current state of deployment, is able to create a voice model of a person's voice and synthesize arbitrary audio that share the voice characteristics of that person. We tested how well Lyrebird.ai is able to imitate a person's speech by replaying the synthesized phrases against the *speaker recognition* (as opposed to speech recognition) systems that are built into personal voice assistants. Siri and Google Assistant both employ speaker recognition on their respective activation phrases "Hey Siri" and "Ok Google" to identify the active user and to provide a more personalized experience based on the user's identity [11, 18]. One of the authors trained both Google Assistant and Siri on a Google Home and iPhone 8 Plus, respectively, with his voice. To ensure that the VAs were not tricked by another speaker's voice, we successfully verified the voice assistants did not accept the respective activation phrases generated by MacOS' `say` text-to-speech command. We then created a voice model of the first author's voice using Lyrebird.ai and used the service to synthesize the activation keywords. Both of the synthesized phrases were successfully able to trick Siri and Google Assistant into believing that the phrases were spoken by the registered user. Although this is an admittedly small experiment and we acknowledge that much more sensitive voice authentication systems exist, it demonstrates the feasibility of defeating widely deployed speaker recognition systems—in particular, those that guard our smartphone devices.

[2]These are sometimes called *skills*.

IV. Strawman Solution: Client-side speech recognition

We can trivially prevent an adversary from getting access to voice data by performing only *client-side* speech recognition. However, there are various practical challenges that prohibit such a solution:

Cloud-based speech recognition allows for large, complex models to be trained, deployed, and updated transparently without affecting client-facing services. Such speech recognition models require significant computing power since state-of-the-art systems rely heavily on computationally expensive deep neural networks. Cloud deployment also allows for constant improvements in speech recognition without requiring updates to client-side software or any service downtime for clients. Sending raw audio to remote servers also allows service providers to gather more data for improving the performance of their speech recognition systems. The majority of commercially deployed speech recognition systems use supervised machine learning techniques [7, 12] that can potentially benefit from access to more data for training or testing. In particular, Alexa, Siri, Google Assistant and Cortana all reportedly use recorded voice commands to improve thee performance and accuracy of their voice-based service offerings [22–25].

Additionally, existing open source client-side speech recognition tools (e.g., CMU Sphinx [26] and Mozzila's DeepSpeech [17] generally have worse accuracy compared to current cloud-based speech recognition services [28]. Client devices such as smartphones and in-home assistants are usually too resource constrained to employ the better performing speech recognition techniques that are used by cloud-based services.

Aside from the technical benefits of cloud-based speech recognition, service providers may also consider their speech recognition models to be intellectual property. Pushing such models to client devices would increase the risk of reverse engineering and could, in turn, lead to the leakage of trade secrets. We posit that the the ability to maintain speech recognition as a closed, cloud-based, black-box service is likely a powerful motivator for service providers.

V. Audio Sanitizer

Our high-level approach to reducing the threat of voice synthesis attacks is to make it more difficult to collect corpora of ordinary users' voices. We introduce the concept of an *audio sanitizer*, a software audio processor that filters and modifies the voice characteristics of the speaker from audio commands before they leave the client device. Altering such features transforms the voice in the audio commands that is available to the adversary, making it difficult to extract the original voice characteristics of the speaker and reducing the accuracy of the speaker's voice model.

The unique characteristics of a person's voice can be attributed to the anatomy of various organs involved in the

process of generating the voice. To identify the audio features that capture the uniqueness of a person's voice, we identify features used in speaker recognition to identify a speaker from his voice. Since the goal of speaker recognition is to tell users apart from each other based on their voice characteristics, we believe that modifying the features used for speaker recognition provides a good starting point for the audio sanitizer.

Speaker recognition system typically employ the following three types of features [39]:

1) Short-term spectral features: These features are extracted from short overlapping frames and correlate to voice timbre. Common spectral features include Mel-frequency cepstral coefficients (MFCCs) and linear predictive cepstral coefficients (LPCCs).
2) Prosodic and spectro-temporal features: These features include pitch, rhythm, tempo, pause and other segmental information and capture the speaking style and intonation.
3) High level features: These features represent speaker behavior or lexical clues and are usually extracted using a lexicon.

We focus on a subset of these features—namely MFCCs, pitch, tempo and pause—and modify them to alter the voice characteristics of the spoken audio. Our perturbations are random, but are applied consistently for the audio of a given individual speaker. (Otherwise, if our modifications were randomly chosen per sample, then an adversary who collects a sufficient number of samples could recover the underlying voice characteristics by "subtracting away" the mean of the applied random distribution.)

In addition to modifying the identifying features of a speakers voice, we also remove the extraneous information present in the audio that is not required for speech recognition. Recall that the first step in speech recognition is feature extraction, which converts high dimensional, raw audio to low dimensional feature vectors. To preserve the acoustic information relevant for speech recognition, we first compute the MFCCs of the input audio and then convert the MFCCs back to audio signal by adding white noise [29]. Importantly, performing an MFCC and then inverting the MFCC back to an audio signal is a lossy operation that cannot be reversed (since information is lost). Here, our goal is to keep only the audio that is required for speech recognition while losing information that is useful to construct accurate voice models.

As we discuss in more detail in the next section, for each speaker, we choose a parameter for each feature such that the resulting sanitized audio has minimal voice characteristics of the speaker and is accurately transcribed by the speech recognition service.

VI. Evaluation

We evaluate our proposed audio sanitizer by analyzing the degree to which it can degrade the quality of voice models to conduct speech synthesis attacks while simultaneously enabling accurate speech recognition.

Feature	Modification
Pitch	Shift up or down by 0 - $\frac{1}{5}$th octave
Tempo	Change by 85% - 115%
Pause	Introduce 0 - 15ms of pause at random 1% positions.
MFCCs	Nbands: 100, Numcep: 100, Wintime: 0.025s, Hoptime: 0.01s

TABLE I: Modifications performed to various features by the audio sanitizer.

A. Impact on Speech Recognition

We evaluate the impact of sanitizing audio (audio output from the audio sanitizer) by comparing the transcription accuracy of the unsanitized (unmodified) and the sanitized audio. Ideally, sanitized audio should provide identical accuracy to the baseline unsanitized audio.

We choose a random subset of 500 audio samples from the West Point Company English speech data corpus from the University of Pennsylvania's Linguistic Data Consortium (LDC) [32]. The LDC corpus consists of both male and female, American English-language speakers, each speaking short, multiple sentences. Our subset is comprised of 130 different speakers, with 53 females and 77 males. We measure our impact on speech recognition quality using Google's and IBM's cloud-based speech recognition services [14, 34]. To quantify the accuracy of speech recognition systems, we consider the Levenshtein edit distance between the words of the correct, expected transcription and the best transcription provided by the speech recognition service. We report the normalized word edit distance by dividing Levenshtein edit distance by the number of words in the baseline transcription.

For each audio sample in the corpus, we first transcribe the unsanitized audio file to establish the baseline accuracy using the online speech recognition services. Each file is then sanitized using the audio sanitizer, which modifies the features that provide unique characteristics to a speaker's voice (see §V). To permanently remove the extraneous audio not required for speech recognition, we compute the MFCCs for each audio and then invert those MFCCs and add white noise to generate the sanitized audio [30]. The audio sanitizer first modifies the pitch, tempo and pause features followed by the lossy MFCC step to produce the sanitized audio. Finally, we transcribe the sanitized audio file generated by the audio sanitizer using the online speech recognition services.

Table I shows the features and the level of modifications performed to each of those features for each audio file. For example, for male speakers, we increase the pitch by 0 to $\frac{1}{5}$th octave, randomly choosing the octave value in the specified range. To modify the tempo, we multiply the tempo of the audio by a number chosen uniformly at random from $[0.85, 1.15]$.

Figure 1 shows the cumulative distribution (CDF) of the normalized edit distances for the unsanitized and sanitized audio samples when using Google's and IBM's speech recognition services. For Google's speech recognition service, the best-case accuracy (i.e., having a perfect transcription and a normalized edit distance of zero) drops from 83.2% to 60.4%

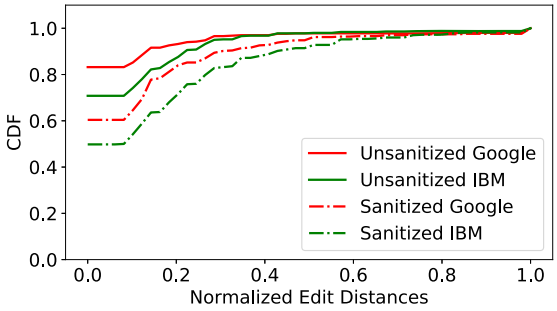

Fig. 1: Impact of audio sanitizer on transcription accuracy.

	Same speaker	Different speaker
Baseline	47	45
Unsanitized audio	23	25
Sanitized audio	22	20

TABLE II: Number of participants assigned to each baseline condition and each of the four test conditions.

when the audio sanitizer is used. In the case of IBM's speech recognition service, sanitizing the audio decreases the accuracy from 70.8% to 50.1%.

Our initial implementation of the audio sanitizer shows promise: in the worst case, transcription is perfect more than half of the time. However, we anticipate that accuracy could be significantly increased by more intelligently performing voice modifications. In particular, in our initial version of the audio sanitizer, we use a fixed set of modifications (see Table I) for all speakers. Given significant variations in people's voices, we can likely achieve improved accuracy results by analyzing individual voice characteristics and choosing specific parameter ranges on a per-speaker basis. We posit that by moving away from a one-size-fits-all model and performing per-speaker audio sanitization, we can make our sanitizer less coarse and more focused by removing only the information that makes an individual speaker's voice distinctive.

B. Privacy Gain

To conduct a speech synthesis attack, the attacker requires a corpus of the targeted user's speech. We evaluate the efficacy of the audio sanitizer by comparing attacks' effectiveness when the corpus is based on unmodified speech (the current norm) and speech that has been filtered by the audio sanitizer. More concretely, we examine the adversary's ability to successfully launch an attack—that is, cause actual human listeners to conflate a synthesized voice with a legitimate recording of a speaker. To perform such an evaluation, we conduct a small user study to measure how well the attacker is able to fool human listeners when (i) using a voice model created from unsanitized voice commands and (ii) comparing that to the case in which the voice is is based on sanitized audio.

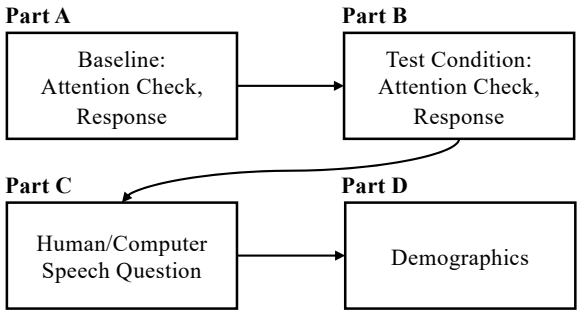

Fig. 2: Sections and flow of the user study.

Metric	Percentage	Metric	Percentage
Gender		**Age**	
Female	41.3%	18-29 years	38.0%
Male	54.3%	30-49 years	47.8%
Other	2.1%	50-64 years	9.8%
		65+ years	1.0%
Ethnicity		**Education**	
Caucasian	75.0%	H.S. or below	9.8%
African American	8.7%	Some college	32.6%
Hispanic	4.3%	B.S. or above	55.4%
Asian	7.7%		
Other	4.3%		

TABLE III: Participant demographics for the user study. Percentages may not add to 100% due to non-response or selection of multiple options.

a) User Study: Our user study is designed to determine the success rate of an attacker when attempting to trick human evaluators with synthesized audio. The user study presents the participants with different pairs of audio samples and asks them to specify whether they think the audio samples were spoken by the same person.

Figure 2 illustrates the design of our online user survey. In Part A of the survey, participants listen to two short audio samples with different speech content and are then asked about the content of the first audio as an attention check. The two audio samples are normal speech samples either from the same speaker or two different speakers, shown evenly to the participants. On the next page of the survey, participants are asked to describe the relationship between the speakers of both audio samples using a five-point scale, from "definitely spoken by same speaker (person)" to "definitely spoken by different speaker (person)". Part A was designed to establish a baseline accuracy of how well survey participants are able to correctly identify whether two voice samples reflect the same or different speakers.

Part B of the study measures whether participants can determine the relationship between the speakers of two audio samples, when one of the audio is synthesized from a voice model. The survey participants listen to two short audio samples with different speech content. The first audio is always a normal speech audio from a single speaker, the second audio is always a synthesized audio generated from a voice model chosen based on the following two factors:

1) Voice model: the voice model is generated by either using

unsanitized audio or sanitized audio.
2) Speaker: the speaker can either be same or different speaker with respect to the first audio.

Using a full factorial design, we consider the four conditions based on the above two factors for choosing the second audio in Part B as shown in Table II. All voice synthesis was performed using the Lyrebird.ai service. Participants are first asked about the content of the first audio as an attention check. On the next survey page, participants are asked to describe the relationship between the speakers of both audio samples, again on a five-point scale ranging from "definitely spoken by same speaker (person)" to "definitely spoken by different speaker (person)".

Part B was designed to answer our primary condition of interest: i.e., while using synthesized audio constructed from a corpus of sanitized audio data, were the participants less able to identify whether the speakers were the same or different? We compare this to the case in which synthesized audio is based on normal, unmodified audio. Put simply, we determine whether the voice synthesis attacks are less convincing when they are forced to train models based only on sanitized audio samples.

In Part C, the participants again listen to the same pair of audio from Part B. They are then asked about the speech in both of the audio samples with options: "both are human voices", "first in human voice but second is a computer generated voice","first is computer generated voice but second is a human voice", "both are computer generated voices" and "not sure". The goal of Part C was to indirectly measure how well users can identify speech that is synthesized using Lyrebird.ai.

The online survey concludes in Part D with demographic questions about education, gender, ethnicity and age.

b) Recruitment: We used Amazon's Mechanical Turk (MTurk) crowdsourcing service to recruit participants for the user study. We required participants to be at least 18 years old and located in the United States. To improve data quality, we also required participants to have at least 95% HIT approval rate [35]. Participants were paid $1.00 for completing the study, which was approved by the IRB at Georgetown University. The demographics of our participants are summarized in Table III.

c) Results: In total, 104 MTurk workers participated and completed our study. Table II shows the number of responses across the baseline conditions and the four test conditions. We exclude 11 responses as duplicates based on their originating IP addresses and only consider their first response and also exclude three responses that failed the attention checks. For the remainder of the paper, we refer to the remaining 90 participants.

Table IV summaries the results of the user study and shows the percentage of users that reported a given relationship between the speakers of the two audio samples for the given condition. For the baseline response, 65.2% of the participants correctly identified the relationship between the speakers from Part A of the survey; 76.6% correctly identified the same speaker whereas 53.3% were able to correctly differentiate between two different speakers. This shows that the majority of the participants were able to correctly identify whether or not two audio samples are from the same speaker.

	Same Speaker		**Different Speaker**	
	Same	*Different*	*Same*	*Different*
Baseline	76.6%	21.3%	40.0%	53.3%
Unsanitized	30.4%	60.9%	12.0%	72.0%
Sanitized	9.1%	81.8%	0.0%	95.0%

TABLE IV: Summary of responses from the user study for various conditions. Each cell shows the percentage of participants that reported a given relationship (excluding the "not sure" response) between the speakers of the two audio samples for the given condition. For Baseline, the two audio were human speech, for Unsanitized and Sanitized, the first audio was human speech while the second was generated by Lyrebird.ai.

We next focus on the case in which the survey participants are tasked with identifying whether two samples originate from the same speaker, when one of the samples is synthetically generated using Lyrebird.ai. When the synthetic voice was produced using a corpus of the first speaker's unmodified (non-sanitized) voice, 30.4% of the participants correctly identified that voices were from the same speaker. This corresponds to the attacker's success rate in impersonating the targeted individual by spoofing his voice using synthesized speech generated from his voice model built using unsanitized speech audio. However, when the synthetic voice was produced using a corpus of the first speaker's modified (sanitized) voice, only 9.1% of the participants believed that the voices were from the same speaker while 81.8% reported them to be from different speakers. Our results show that the audio sanitizer is able to significantly reduce the efficacy of the attack; that is, the attack is far less successful when the attacker only has access to sanitized speech audio samples.

In the case of different speakers, when the synthesized voice was generated using a corpus of another speaker's unmodified (non-sanitized) voice, 72.0% of the participants correctly identified the voices to be from different speakers while 12.0% reported them to be from the same speaker. However, the use of a sanitized audio corpus to synthesize the audio for another speaker resulted in 95.0% of the participants correctly identifying the voices to be from different speakers and none of the participants reporting the voices to be from the same speaker.

In summary, the results from our user study show that given the current quality of Lyrebird.ai's voice synthesis, an attacker with access to unmodified speech audio samples of the targeted individual can synthesize convincing spoofed speech samples in the target's voice. However, sanitization of the audio to remove the voice characteristics prevents the attacker from generating an accurate voice model, resulting in synthesized spoofed audio that are far less convincing.

VII. Discussion

We conclude by discussing in more detail the benefits, limitations, and deployment considerations surrounding our audio sanitizer defense.

Detection of Computer Generated Audio by Humans. In Part C of the online survey, we asked the survey participants to identify whether the two audio samples presented to them were spoken by a human or were computer generated. 76.7% of the participants correctly identified the first audio to be human generated speech while the second one being computer generated across all four conditions. This shows that the users, with the current state of voice synthesis, are able to correctly identify computer generated voices. However, this does not diminish the threat posed by the collection of voice data for the purpose of building voice models for malicious purposes, since further improvements in the underlying technologies for voice synthesis and conversational voice assistants will increase privacy risks. Additionally, the availability of more training data for creating a voice model is likely to improve the accuracy of the synthesized voice. For example, the synthesized audio used in our user study were generated from voice models, with each model built using 40 short audio samples. As stated by the Lyrebird.ai voice synthesis service, providing more training samples improves the quality of the voice model and the synthesized speech.

Practical Deployment. A major goal of our proposed audio sanitizer is to improve the privacy of users without requiring any support from various transcription services. Our defense requires only the manipulation of audio on client devices before it is sent to the remote transcription services. Any device that accepts voice commands and does not perform speech recognition locally can leverage the audio sanitizer. To be effective, the audio sanitizer has to be placed somewhere in the path between the user and the transcription service so that it can intercept and sanitize the audio that comprises the voice command.

One possible point of interception is the communication link between the client device and the remote service. The audio sanitizer can capture the voice command from network packets and then forward it to the service after sanitization. However, the communication between the client device and the remote transcription service usually happens over an encrypted channel[3].

A more practical point of interception of audio data is within the client device itself, after the audio has been recorded by the microphone(s) and before it leaves the device. Relatedly, mechanisms for tracking and intercepting sensor data before delivering it to applications have previously been explored [36, 40]. For example, Xu and Zhu [40] propose a framework for Android smartphones that allows users to control the data generated by various sensors based on user-defined policies for the requesting application before forwarding the sensor data to that application. In particular, for audio data recorded by a microphone, their approach allows replacement of actual audio with mock data or with the addition of random noise. Thus, we can leverage such existing mechanisms on devices running Android to allow the audio sanitizer to intercept and sanitize the audio from the microphone before it is delivered to the application.

For in-home assistants with dedicated hardware such as Amazon Echo or Google Home, our defense can be deployed in a less subtle way. An ideal scenario would be to allow the user to run custom software on these devices. That way, we can directly integrate the audio sanitizer on such devices. A motivating example is the Amazon Echo that runs FireOS, which is an Android-based operating system [9, 10] and thus, can possibly use the same strategy as other Android devices.

Additional Benefits. In addition to thwarting an adversary's attempt to build an accurate voice model of targeted speaker, the audio sanitizer also allow service providers to make stronger claims about user privacy. Current devices such as Amazon Echo and Google Home record and transmit any sound they hear after the activation word. Thus, any accidental triggering of such always-on voice assistants during confidential conversations poses a significant threat to user privacy [6]. As highlighted by recent events, governments can subpoena service providers for any such recordings [4, 5], which can harm a provider's efforts to alleviate public concern about the privacy risks of installing always-on listening devices. Service provides may opt to build audio sanitizers into their appliances and applications, as a way of assuaging privacy concerns.

[3]We verified that Google Home and Amazon Echo use encrypted TLS connection to send voice commands to remote servers.

Acknowledgments

We thank Daniel Votipka for providing feedback on the user study for the paper. This work has been partially supported by the National Science Foundation under grant CNS-1718498. The opinions, findings, and conclusions or recommendations expressed in this work are strictly those of the authors and do not necessarily reflect the official policy or position of any employer or funding agency.

References

[1] Adobe Voco 'Photoshop-for-voice' causes concern. http://www.bbc.com/news/technology-37899902, . Last Accessed February 10, 2019.

[2] Adobe demos "photoshop for audio," lets you edit speech as easily as text. https://arstechnica.com/information-technology/2016/11/adobe-voco-photoshop-for-audio-speech-editing/, . Last Accessed February 10, 2019.

[3] Get audio file of input speech. https://forums.developer.amazon.com/questions/65952/get-audio-file-of-input-speech.html, . Last Accessed February 10, 2019.

[4] A Murder Case Tests Alexa's Devotion to Your Privacy. https://www.wired.com/2017/02/murder-case-tests-alexas-devotion-privacy/, . Last Accessed February 10, 2019.

[5] Judge orders Amazon to produce Echo recordings in double murder case. https://www.cbsnews.com/news/amazon-echo-judge-orders-company-produce-alexa-recordings-double-murder-case-2018-11-12/, . Last Accessed February 9, 2019.

[6] Is Alexa Listening? Amazon Echo Sent Out Recording of Couple's Conversation. https://www.nytimes.com/2018/05/25/

business/amazon-alexa-conversation-shared-echo.html, . Last Accessed February 9, 2019.
[7] The iBrain Is Here—and It's Already Inside Your Phone. https://www.wired.com/2016/08/an-exclusive-look-at-how-ai-and-machine-learning-work-at-apple/. Last Accessed February 10, 2019.
[8] Amazon may give app developers access to Alexa audio recordings. https://www.theverge.com/2017/7/12/15960596/amazon-alexa-echo-speaker-audio-recordings-developers-data, . Last Accessed February 10, 2019.
[9] Exploring the Amazon Echo Dot, Part 1: Intercepting firmware updates. https://medium.com/@micaksica/exploring-the-amazon-echo-dot-part-1-intercepting-firmware-updates-c7e0f9408b59, . Last Accessed February 10, 2019.
[10] Fire OS 5. https://developer.amazon.com/android-fireos. Last Accessed February 10, 2019.
[11] Google Home now recognizes your individual voice. https://money.cnn.com/2017/04/20/technology/google-home-voice-recognition/index.html, . Cited January 27, 2019.
[12] Google voice search: faster and more accurate. https://research.googleblog.com/2015/09/google-voice-search-faster-and-more.html, . Last Accessed February 10, 2019.
[13] WaveNet: A Generative Model for Raw Audio. https://deepmind.com/blog/wavenet-generative-model-raw-audio/, . Last Accessed February 10, 2019.
[14] IBM Speech to Text. https://www.ibm.com/watson/services/speech-to-text/. Last Accessed January 22, 2019.
[15] Copy the voice of anyone. https://lyrebird.ai, . Last Accessed January 23, 2019.
[16] Lyrebird is a voice mimic for the fake news era. https://techcrunch.com/2017/04/25/lyrebird-is-a-voice-mimic-for-the-fake-news-era/, . Last Accessed February 10, 2019.
[17] Mozilla DeepSpeech. https://github.com/mozilla/DeepSpeech. Last Accessed February 10, 2019.
[18] Personalized Hey Siri. https://machinelearning.apple.com/2018/04/16/personalized-hey-siri.html. Last Accessed February 10, 2019.
[19] How your voice can protect you from credit card fraud. http://money.cnn.com/2015/11/02/pf/voice-biometrics-customer-fraud/index.html, . Last Accessed February 10, 2019.
[20] Banking on the power of speech. https://wealth.barclays.com/en_gb/home/international-banking/insight-research/manage-your-money/banking-on-the-power-of-speech.html, . Last Accessed February 10, 2019.
[21] Citi Uses Voice Prints To Authenticate Customers Quickly And Effortlessly. https://www.forbes.com/sites/tomgroenfeldt/2016/06/27/citi-uses-voice-prints-to-authenticate-customers-quickly-and-effortlessly/#716e21b4109c, . Last Accessed February 10, 2019.
[22] Alexa Terms of Use. https://www.amazon.com/gp/help/customer/display.html?nodeId=201809740, . Last Accessed February 10, 2019.
[23] Cortana and privacy. https://privacy.microsoft.com/en-us/windows-10-cortana-and-privacy, . Last Accessed February 10, 2019.
[24] Data security & privacy on Google Home. https://support.google.com/googlehome/answer/7072285?hl=en, . Last Accessed February 10, 2019.
[25] Apple Finally Reveals How Long Siri Keeps Your Data. https://www.wired.com/2013/04/siri-two-years/, . Last Accessed February 10, 2019.
[26] CMU Spinx. http://cmusphinx.sourceforge.net/, 2015. Last Accessed February 10, 2019.
[27] D. M. Ballesteros L and J. M. Moreno A. Speech scrambling based on imitation of a target speech signal with non-confidential content. 2014.
[28] G. Bohouta and V. Këpuska. Comparing speech recognition systems (microsoft api, google api and cmu sphinx). 2248-9622:20–24, 03 2017.
[29] N. Carlini, P. Mishra, T. Vaidya, Y. Zhang, M. Sherr, C. Shields, D. Wagner, and W. Zhou. Hidden Voice Commands. In *USENIX Security Symposium (Security)*, 2016.
[30] D. P. W. Ellis. PLP and RASTA (and MFCC, and inversion) in Matlab. http://www.ee.columbia.edu/~dpwe/resources/matlab/rastamat/, 2005.
[31] S. Mathias and K. von Kriegstein. How do we recognise who is speaking? *Frontiers in Bioscience (Scholar edition)*, 6:92, 2014.
[32] J. Morgan, S. LaRocca, S. Bellinger, and C. C. Ruscelli. West Point Company G3 American English Speech. Linguistic Data Consortium, item LDC2005S30. University of Pennsylvania. Available at https://catalog.ldc.upenn.edu/LDC2005S30, 2005.
[33] M. A. Pathak, B. Raj, S. D. Rane, and P. Smaragdis. Privacy-preserving speech processing: cryptographic and string-matching frameworks show promise. *IEEE Signal Processing Magazine*, March 2013.
[34] T. Payton. Google Speech API: Information and Guidelines. http://blog.travispayton.com/wp-content/uploads/2014/03/Google-Speech-API.pdf.
[35] E. Peer, J. Vosgerau, and A. Acquisti. Reputation as a Sufficient Condition for Data Quality on Amazon Mechanical Turk. *Behavior Research Methods*, 46(4):1023–1031, Dec 2014.
[36] G. Petracca, Y. Sun, T. Jaeger, and A. Atamli. Audroid: Preventing attacks on audio channels in mobile devices. In *Proceedings of the 31st Annual Computer Security Applications Conference*, ACSAC 2015, 2015.
[37] P. Smaragdis and M. Shashanka. A framework for secure speech recognition. *IEEE Transactions on Audio, Speech, and Language Processing*, 2007.
[38] T. Vaidya, Y. Zhang, M. Sherr, and C. Shields. Cocaine noodles: Exploiting the gap between human and machine speech recognition. In *9th USENIX Workshop on Offensive Technologies (WOOT 15)*, 2015.
[39] Z. Wu, N. Evans, T. Kinnunen, J. Yamagishi, F. Alegre, and H. Li. Spoofing and countermeasures for speaker verification: A survey. *speech communication*, 66:130–153, 2015.
[40] Z. Xu and S. Zhu. Semadroid: A privacy-aware sensor management framework for smartphones. In *Proceedings of the 5th ACM Conference on Data and Application Security and Privacy*, 2015.