Quantifying the Breakability of Voice Assistants

Maliheh Shirvanian
Visa Research ⋆
Palo Alto, CA, USA
mshirvan@visa.com

Summer Vo
Massachusetts Institute of Technology ⋆
Cambridge, MA, USA
summer.vo@outlook.com

Nitesh Saxena
University of Alabama at Birmingham
Birmingham, AL, USA
saxena@uab.edu

***Abstract*—In this paper, we present a thorough study of voice impersonation attacks that can compromise the security of voice authentication technology deployed in several popular, state-of-the-art Android and iOS apps. Our study is based on our formulated *Sneakers* attack system that comprises a variety of well-known as well as newly designed attacks: (1) recorded and replayed voice of the authorized user (*replay attack*); (2) reordered and played-back voice of the authorized user (*reorder attack*); and (3) synthesized voice generated — based on *voice conversion* techniques — using an unauthorized user's voice (*standard conversion attack*), or using a noise-free recording from a text-to-speech engine (*TTS conversion attack*). Taking Sneakers as a basis, we report on a carefully designed study to examine a variety of real-world voice authentication apps for their vulnerability against *malicious authentication*.**

Our study follows a two-phase methodology. In the preliminary phase, we analyze 8 popular mobile apps against standard simplistic attack setups. Our results show that, while the tested apps seem to resist the reorder attack and the standard conversion attack, they are highly vulnerable to the replay attack. In the main phase of the study, we comprehensively assess 5 of the above apps against more advanced newly designed attack setups. Like in the preliminary phase, the apps prove to be *highly vulnerable to the replay attack*. More seriously, the apps also turn out to be *highly insecure against our advanced attack setups*, i.e., the reorder attack with coordinated timing and the TTS conversion attack, yielding success rates of 82%–98%. These malicious authentication measurement results are highly pertinent in practice because, we demonstrate that the apps generally work well in the *benign authentication* scenario to reliably "accept" an authorized user and "reject" an unauthorized user.

Our work shows that many standard attacks that prior work demonstrated to be effective against standalone voice authentication algorithms do not work against current voice authentication apps. Yet, our new attack designs could still compromise these apps. Overall, our work highlights a serious vulnerability of real-world voice authentication apps, which seems very challenging to mitigate at a fundamental level.

***Index Terms*—Voice Authentication, Speaker Verification, Voice Synthesis Attack**

I. Introduction

Voice authentication is getting deployed in real-world scenarios at a rapid pace. Many smartphones now incorporate the "voice unlock" feature, along with the traditional PINs or passwords, to make it easier for the users to authenticate to their phones [1]. Virtual personal assistants offer voice authentication option to listen to the commands only issued by the authorized users [2], [3]. Other apps are available that provide secure out-of-band authentication based on voice authentication [4] or provide secure access to secret data [5]. Banks and financial institutions, such as Barclays [6], HSBC [7] and Wells Fargo, have started using voice authentication for mobile and phone banking [8], [9].

Given the rise in the deployment of voice authentication, a natural concern about these systems is their security. In this paper, we study and quantify the (in)security of a number of real-world, popular smartphone apps, which authenticate the users by means of their voices. First, we design a system, called *Sneakers*[1], comprising a variety of well-known as well as newly designed voice impersonation attacks. Second, we report on a study to examine mobile apps in a *black-box model* (without the knowledge of the underlying algorithms) with respect to their susceptibility to the presented attacks.

Design of the Sneakers System: The first aspect of our work lies in the design of Sneakers, consisting of the following attacks to be used as the bases of our study:

1) *Replay Attack:* In this attack, the attacker collects samples of the victim's voice speaking the predefined passphrase and attempts to attack the apps with these samples.
2) *Reorder Attack:* This form of attack requires a pre-recorded collection of some words or numbers spoken by the victim, but not necessarily in the same order as they appear in the passphrase. The attacker may shuffle the collected words to match the passphrase. We define a slightly advanced variant of the reorder attack, *the time-synced reorder attack*, in which the delay between the audio playback and the voice authentication is adjusted for higher attack success.
3) *Conversion Attack:* In this attack, the adversary has access to some recordings of the victim's voice, but not the exact passphrase or words and numbers used in the passphrase. The adversary uses these recordings to generate a synthesized voice based on the *voice conversion/synthesis* techniques (e.g., [10]–[12]), using as the source of conversion an unauthorized user's voice (*standard conversion*), or a noise-free recording from a text-to-speech (TTS) engine (*advanced conversion*), a new variant defined in our work.

A Security Study of Voice Authentication Apps: The second aspect of our work constitutes the design of a formal study that

⋆ Work has been done at UAB.

[1] In the 1992 movie Sneakers, an interesting sequence involves hacking into a voice authentication system with a replayed voice of the victim speaker. About 25 years later, when voice authentication is a reality, this paper brings this classic sequence to life by highlighting the vulnerability of real-world voice authentication apps to different forms of voice impersonation attacks.

978-1-5386-9148-9/19/$31.00 ©2019 IEEE

assesses a number of real-world, already popular mobile apps under the attacks defined by Sneakers. For the benign setting, we analyze whether the authorized users can successfully authenticate to the apps. Also, as the baseline for our Sneakers' attacks, we consider a "zero-effort different speaker attack", in which the app is trained with the voice of the victim user but is tested against the voice of a different speaker (e.g., the attacker's own physical voice). We follow a two-phase study methodology in the lab setting to comprehensively investigate the performance/security of the apps. Table I briefly summarizes the studies and the results.

- *Phase I—Preliminary Study:* In this phase, we evaluated 8 apps[2] in the benign and standard attack settings, with two live users. The result of this study showed a high accuracy of a majority of the apps in recognizing the authorized speaker's voice. The apps performed equally well in distinguishing the voice of a different speaker (different gender). Interestingly, apps showed resistance to the standard reordering and standard conversion attacks but not the replay attack.
- *Phase II—Main Study:* In the second phase, we more comprehensively examined the reliability and security of 5 of the above 8 apps with 10 participants. Here, we discarded the unsuccessful attacks from the first phase and tested the apps against our advanced attacks. As part of the advanced attack setup, we replaced the attacker's voice with "noise-free" audio samples collected from a "text-to-speech" (TTS) tool to improve the success of the standard conversion attack. To prepare the noise-free samples, we recorded the audio spoken via the IBM TTS tool [13] by feeding the TTS directly to the audio recorder using an *audio cable*. We also coordinated the time between playing audio and starting voice authentication in the time-synced reorder attack to increase the success of the reorder attack (possibly by presenting audio samples that carry similar noise background to defeat liveness detection).

  The results of the study reiterate that all apps can successfully authenticate an authorized user and provide a reasonable security level in recognizing the baseline zero-effort different speaker attack (at least 75%) for different gender and same gender. However, the apps fail to resist our replay attack yielding an attack success rate of 100%, and the other advanced attack setups resulting an average attack success rate of about 82% when confronted with time-synced reorder attack and above 95% for synthesized voice converted from a noise-free recorded samples of a TTS engine (TTS conversion attack).

Novel Contributions of Our Work: Our work provides two main novel contributions:

First, our study constitutes the first methodical investigation of the vulnerability of mobile apps that deploy the voice authentication technology. We note that a considerable number of voice authentication *algorithms* have been studied independently and shown to be vulnerable to replay attack, voice mimicking, conversion and synthesis attacks [14]–[20]. However, none of the prior studies has evaluated the security of real-world smartphone apps, which is fundamentally different from studying algorithms in isolation. Smartphones usually have limited resources compared to powerful servers running these algorithms. Accordingly, attacks applicable to one setting may not work on the other. Moreover, many of the apps do not reveal the underlying algorithms and therefore in studying the apps, reverse engineering or tweaking of the parameters based on the deployed algorithm is not possible. Finally, the apps receive the input from a live or played-back sample, traversing through the air to reach the device's microphone. Hence, the input signal may degrade, experience loss or become noisy in the process. In contrast, static "audio files" can be simply fed to the system without being subject to any degradation. Due to the same reason, isolated algorithms can be tested using publicly available voice datasets, as used in many prior studies cited above, while our study had to be conducted with real human users. Therefore, we believe analyzing the real-world mobile apps is inherently different from the previous studies.

TABLE I
SUMMARY OF OUR WORK. PRELIMINARY STUDY IS THE MOTIVATION FOR THE MAIN STUDY AS THE FORMER SHOWED THE TESTED APPS WERE ROBUST TO STANDARD ATTACKS (EXCEPT REPLAY). MAIN STUDY SHOWS THAT ALL APPS CAN BE BROKEN WITH ADVANCED ATTACK VARIATIONS.

	Preliminary Study	**Main Study**
Replay Attack	Succeeded	Succeeded
Standard Reorder Attack	Failed	-
Standard Conversion Attack	Failed	-
Time-Sync Reorder Attack	-	Succeeded
TTS Conversion Attack	-	Succeeded

Second, we subjected the apps to: (1) *standard attacks* previously tested against voice authentication algorithms (*not apps*), namely, replay attack, standard reorder attack, and standard conversion attack, and (2) *two novel attacks* specifically designed for mobile apps, namely, time-synced reorder attack and TTS conversion attack. Our work demonstrates that many standard attacks (reordering and conversion) that prior work demonstrated to be effective against standalone voice authentication algorithms actually fail against current voice authentication apps. Nevertheless, our new attack designs could still succeed at compromising the security of these apps.

Overall, we believe the combination of novel variations of attack techniques and the comprehensive black-box evaluation of the mobile apps makes our work significantly different from previous work.

Responsible Disclosure: We notified the vendors of the apps studied in the second phase of the study about the vulnerabilities. Two of the vendors explicitly requested us to anonymize the references to their apps and the companies. Since we have not heard back from other vendors, we decided to anonymize all the 8 apps and their vendors in this submission. We believe that this level of anonymization may help to protect

[2] The app and vendor names are anonymized as discussed in "Responsible Disclosure".

the individual business interests of the vendors in light of the reported vulnerability, while still exposing the extent of the vulnerability in *aggregate* terms. As canonical names, we adopt *shades* as the *app names* and the corresponding *colors* as the *vendor names* (e.g., *Green Mint* refers to the app named *Mint* offered by the vendor named *Green*).

II. Background and Prior Work

A. Voice Authentication

Voice authentication refers to the process of verifying a user by analyzing a spoken sample of the user's voice [21], [22]. Such systems first register a speaker's "voice biometrics template" or "voiceprint". This can be done by requiring the user to speak a certain phrase once or multiple times. After the noise reduction, the system extract voice features (e.g., Mel-frequency cepstral coefficients, Linear Predictive Coding, or Fast Fourier Transform features) and builds a model of the speaker's voice. In the testing phase, after noise reduction, the features are extracted from the input signal and the feature vectors are then used to compute the similarity to the model using a likelihood function and classification techniques (e.g., Gaussian Mixture Model, or Hidden Markov Model).

Three types of voice authentication systems are commonly used in the security application: text-dependent, text-independent, and text-prompted systems. In text-dependent systems, the user speaks a "pre-determined" fixed phrase (e.g., a passphrase) for authentication, while text-independent systems allow the user to converse naturally during verification. A text-prompted system is a form of a text-dependent system, in which the user should speak a random phrase displayed by the app (e.g., in the form of challenge-response).

B. Voice Authentication Apps

Practical Use: Smartphones have added voice authentication to unlock the phone in addition to numeric passcodes, unlock patterns, and fingerprint. Also, personal assistant apps such as Ok Google [1], Siri [2], and Dragon Mobile Assistant [3], which were only being used for speech recognition now offer the voice authentication feature to add a layer of security measure when executing user's commands. Some other apps protect the user's private data (e.g., access to apps and data) by voice authentication.

Passphrases in Training and Testing Phases: The training process for most apps involves speaking a passphrase multiple times. Some apps allow the user to pick any passphrase of their choice, while others may have pre-defined phrases, or only a few phrases that the user can choose from. The passphrases are typically short sentences, a sequence of words, or numbers.

During the authentication phase (the testing phase), apps may display the phrase used in the training phase or wait for the user to speak the passphrase they have picked during the training (without displaying it). Those apps that let the user choose the passphrase, authenticate the user not only by matching their voice to the trained model, but also they compare the spoken phrase with the trained passphrase using speech recognition techniques. The challenge-response text-prompted apps display random phrases (e.g., possibly not used in the training phase) and ask the users to speak the phrases.

Synchronization: Apps may provide an input button to initiate the voice authentication process, or they may wake up by the voice of the authorized user who speaks the correct passphrase. The passphrase may be followed by a command in virtual assistant apps. The voice authentication process ends after the user finishes speaking the passphrase/command, or after a timer defined by the app, times out.

C. Voice Synthesis

A class of approaches synthesizing naturally sounded voice using a small training dataset is referred to as voice conversion (e.g., [10]). Voice conversion modifies a source speaker's voice to sound like a target speaker by mapping between the features of their voices. Such systems create a model of the transition from a source speaker's voice to a target speaker's voice during a training phase. The model can then be used during the testing phase to generate a voice that sounds similar to the target speaker, while carrying intonation in the source speech. Compared to other voice synthesis approaches, voice conversion requires less training data and therefore is a suitable tool to attack someone's voice. The training samples and testing samples can be completely different allowing for the conversion system to generate arbitrary speech that the victim has possibly never spoken earlier. Other recent approaches (e.g., [12]) are also available that rely on deep learning techniques to extract features of the speaker's voice from only a few minutes of the speech and produce naturally sounding samples in the speaker's voice.

D. Related Prior Work

Several prior work has studied the security and reliability of voice biometrics algorithms in isolation. Other than classical professional or amateur human-based impersonation attacks [14], [15], automated voice conversion and synthetic speech can also be used to attack voice biometric [17], [23].

In [24], [25], the vulnerability of advanced voice biometrics systems to synthetic speech has been studied, and possible defenses for attacks have been proposed. In [26], the vulnerability of voice authentication against artificial signals has been demonstrated.

It has been shown in [19], [27] that playing a pre-recorded audio sample from the target or victim (a.k.a. replay attack) can be an effective way to spoof text-independent voice authentication systems. A replay attack is a low-technology attack easily accessed by any potential attacker since it does not require much specialized knowledge of speech processing. Furthermore, the availability of inexpensive high-quality recording equipment suggests that this attack may be effective and difficult to detect. However, replay is not flexible in generating phrases not spoken by the victim before.

Conversion attack has attracted more interest over a decade in the context of vulnerabilities in systems and apps. Conversion attack applies to text-dependent, text-independent and

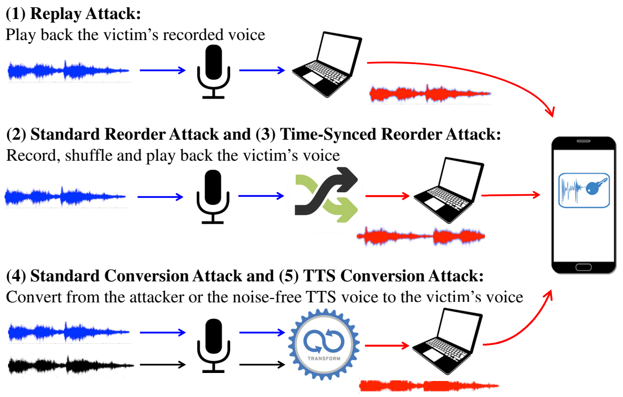
(1) Replay Attack:
Play back the victim's recorded voice
(2) Standard Reorder Attack and (3) Time-Synced Reorder Attack:
Record, shuffle and play back the victim's voice
(4) Standard Conversion Attack and (5) TTS Conversion Attack:
Convert from the attacker or the noise-free TTS voice to the victim's voice
TRANSFORM

Fig. 1. Our standard and advanced variants of voice impersonation attacks

text-prompted systems. Mukhopadhyay et al. [20] focused on investigating the effect of conversion attack created by an off-the-shelf voice conversion tool and showed the vulnerability of different algorithms to this type of attack. Further, Kinnunen et al. [16] have studied the vulnerabilities of text-independent voice authentication systems against voice conversion based on telephonic speech. They implemented a voice conversion system and tested different speaker verification systems against the attack. Wu et al. [28] conducted a study on text-dependent systems which have a predetermined phrase to verify the user. In this study, joint density Gaussian mixture model and unit-selection methods were tested with conversion attack.

As mentioned in Section I, although previous work has shown the vulnerability of voice authentication algorithms, they had attacked the systems based on the knowledge of algorithms. Also, none of the previous work considered algorithms used by mobile apps. Therefore, the main difference between these lines of work and our work lies in the black-box evaluation of the algorithms that are incorporated in real-world mobile apps. The study of real-world apps is inherently different from the study of independent algorithms. This is because, in studying real apps, the voice samples should be live or played back, which means the samples may experience loss and carry the background noise. While, in contrast, in independent evaluation of voice authentication algorithms, audio file samples can be directly inputted to the systems without any noise or signal loss. Such differences result in different behavior of the system even to well-known standard attacks, as will be shown in our studies.

III. Sneakers Attack Design

A. Benign and Baseline Settings

Benign Setting: The apps are expected to grant access to an authorized user who trained the system in the enrollment phase and may test the system later on. We refer to this setting as the benign setting.

Zero-Effort Different Speaker Attack: This attack is considered as the baseline for the evaluation of the attacks in Sneakers. Here, the attacker follows a rather naive approach by speaking to the voice authentication app in his/her voice. Depending on the similarity of the attacker's voice to the victim's voice, the attacker might be able to authenticate to the app successfully. In this form of attack, the attacker does not need or have access to prior samples of the victim's voice. This attack could be used against text-dependent, text-prompted, and text-independent voice authentication apps. If a voice authentication app works well, we would expect this baseline attack to be detected by the app with a high probability.

B. Sneakers Sub-Attacks

Sneakers consists of three types of attacks, including their previously introduced standard and newly designed advanced variations as shown in Figure 1 and described next.

Replay Attack: The attacker collects the victim's voice speaking the passphrase of the voice authentication app, for example, while the victim is authenticating to the apps or while speaking the same phrases in a daily conversation. The attacker can even use social engineering tricks to encourage the victim into speaking the passphrases in a conversation or a phone call, as shown in recent scams [29]. The attacker who has access to the phone can play the previously recorded samples of the original speaker's speech in an attempt to gain access. Replay can be applied to the text-dependent pre-determined passphrase apps.

Standard and Time-Synced Reorder Attacks: Phrases in a given voice authentication challenge consist of certain words or numbers. Although the attacker may not have access to the exact combination and arrangement of these words or numbers as expected by the app, s/he may have obtained the same words and digits in a different ordering through social engineering or via publicly available speech samples of the victim. The attacker can extract the words and numbers individually and rearrange them into a meaningful form as expected by the app and play it back to the app to authenticate on behalf of the user. We refer to this type of attack as reorder attack. Reorder can be used against text-prompted voice authentication apps.

In the standard variation of the attack, the attacker plays the samples after starting the voice authentication process. In the time-synced reorder attack, the attacker prepares and plays the attacked audio samples such that it starts with a brief period of silence, followed by the speech, and ended with another moment of silence (as illustrated in Figure 2). The voice authentication functionality should start right before the speech starts. The motivation behind designing this attack is to feed the same background noise to the voice authentication

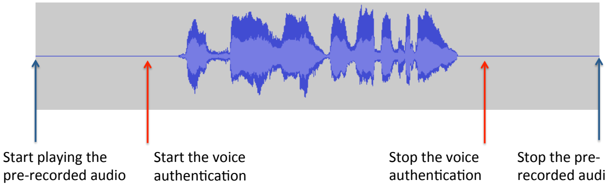
Start playing the pre-recorded audio
Start the voice authentication
Stop the voice authentication
Stop the pre-recorded audio

Fig. 2. Timing between the time-synced reorder attackand starting the voice authentication for the apps starting by a button

app throughout the authentication process and thereby, to bypass any possible liveness detection schemes that work by comparing/removing the background noise.

Standard and TTS Conversion Attacks: In this type of attack, the attacker trains a voice conversion tool by feeding a few minutes of the victim's voice to the tool (e.g., either by recording the voice in proximity or by collecting publicly available audio files). The attacker can then synthesize the victim's voice offline or on the fly and present them to the apps in an authentication attempt. Since the attacker can generate any given phrase in the victim's voice, this attack is suitable for all types of voice authentication (text-dependent, text-prompted, and even text-independent). However, the attacker needs samples of the victim's voice for conversion training.

We introduce two variations of this attack. In the standard variation, the voice conversion tool is trained with the voice of an unauthorized human user as the source speech. In the advanced variation, TTS conversion attack, the source speech is replaced with noise-free high-quality audio samples generated by TTS tools to generate higher quality audio (i.e., automatically generated voice is used as the source of conversion). The higher quality audio has a higher chance of defeating the speaker recognition system since it more accurately reflects features of the speaker's voice.

IV. Study Preliminaries and Goals

A. Objectives and Metrics

The robustness of voice authentication is evaluated using False Rejection Rate (FRR) and False Acceptance Rate (FAR). FRR is the likelihood of rejecting a benign case in which an authorized user incorrectly gets rejected by the system. FRR may force the users to restart the authentication process, and therefore it indirectly impacts user's experience of the system, i.e., the lower the FRR, the higher the perceived usability of the scheme. FAR is the measure of the likelihood that the system accepts an attacked case in which an unauthorized user incorrectly gets accepted by the system. FAR implies the success of the attack and compromise the security of the system. Therefore, FAR indicates the robustness of the system in the face of the attacks.

B. Selected Mobile Apps

To study the robustness of the mobile voice authentication, we selected several apps running on Android and iOS platforms. To cover different types of applications (i.e., phone unlock, personal assistance, and vault) and different types of phrases used during the authentication process (i.e., fixed as well as random numerical and word phrases), we selected several popular apps, in the phase one of the study based on their ranking and the number of downloads on Google Play and Apple App Store. This selection includes *Grey Silver, Pink Rose, Red Lava, Blue Iris, Brown Tan, Black Jet, Orange Rust, and Green Mint*. Then, we downselected a smaller set to be tested in more detail in phase two of the study. This selection includes Rose, Tan, Jet, Rust, and Mint. Some vendors including Pink, Brown and Black offer paid as well as free apps. However, all three companies declared that the same speaker recognition engine/algorithms are used in their free and paid apps. Therefore, we only evaluated the free versions. We believe the results would not have been different if paid versions were to be evaluated. The apps tested in the two studies are listed below (summarized in Appendix Table II).

(1) Grey Silver (*tested on Android in the preliminary study*): Silver is a tool developed to demonstrate how the voice biometric works. Since this app is only for educational purposes, we did not test it in the main study.

In the training phase, the user can set the features and parameters based on their preferences. To train the system, the user presses the "Train" button and speaks the training phrase to record and train the system. In this app, there are no pre-determined phrases to choose from, and the user can speak any phrase of his/her choice (e.g., in our study "spoke in gibberish"). After speaking the phrase, the user can either press the stop button or let the system time out to finish recording the audio. This process is only done once. In the testing phase, the user presses the "Test" button and speaks the same phrase used in the training. Once finished recording, the user can press the stop button or let the system time out. The app analyzes and displays the result of the voice authentication attempt.

(2) Pink Rose (*tested on Android and iOS in both studies*[3]): Rose is a voice authentication software built on Pink's voice biometric technology, and is delivered for Android and iOS using Amazon Web Services hosting. Pink claims Rose has a successful 99.99% rejection and a 97% acceptance rate.

To train, first, the user should create an account on the app and select one of the phrases listed by the app (text-dependent fixed phrase) or choose to authenticate with numeric passphrases (text-prompted numeric phrase). The user then presses the record button and speaks the phrase displayed on the app three times to train the system. As a backup security measure, the user then creates a lock pattern. In the testing phase, the user opens the app and selects the account created during the training phase. After choosing to log in, the user presses the record button and speaks the displayed phrase in the given time slot and the app verifies the user.

(3) Red Lava (*tested on Android and iOS in the preliminary study*): Lava is an app to store secret information such as username and passwords on the phone.

To train the app, the user speaks numbers in the order of 1 through 9, three times. After training is done, the user selects the enroll button. The system processes the samples by sending it to the Lava server. In the testing phase, the user speaks the displayed sequence of digits. Three sequences of 3-digit numbers are displayed one after another that the user is expected to speak in the given time slot. The app then transmits the samples to the server for analysis. We excluded this app from the second study due to the high false rejection in the first study.

[3] The tested version seems to have been taken down after we notified them.

(4) Blue Iris (*tested on iOS in the preliminary study*): Iris is a security solution that combines voice biometrics with PKI digital certificates to authenticate the users.

During training, the user creates an account with the system and speaks the presented sequence of digits. This is completed three times in total with the same sequence of digits. During testing, the users select which section of the app to access. If the section requires authentication, the user should speak a sequence of the displayed digits. The system either opens the section or notifies the user that the sample was rejected and shows the number of remaining trials. We excluded this app in the second study due to the similarity to Rose app (i.e., text-prompted random numeric passphrases are used and samples are sent to an online service for voice authentication).

(5) Brown Tan (*tested on Android in both studies*): Tan is a mobile authentication app that offers security through voice and facial recognition. Brown boasts that their speaker recognition engine provides a 99.999% successful rejection rate and a 98% acceptance rate.

To train the app, the user "selects" a passphrase and speaks it while facing the camera so that the app can capture facial and vocal features. Then, the user selects an alternative method of authentication (e.g., face, or face and voice). After training, the user can select the apps that should be protected and choose the verification method (e.g., voice). For testing, the user opens an app that is protected by Tan and speaks the phrase (or face the camera or both, based on the preferred verification method). The app opens if the authentication is successful.

(6) Black Jet (*tested on Android in both studies*): Jet is a virtual personal assistant app based on the vendor's voice recognition technology running from any screen by speaking a wake-up command.

To set the voice authentication feature, the user should select the option to set a voiceprint in the setting. To train the system, the user speaks and records the wake-up phrase three times. To test the system, the user speaks the wake-up phrase to the app. If the system recognizes the voice, the microphone icon enlarges and waits for a command to execute.

(7) Orange Rust (*tested on Android in both studies*): Rust was instituted as an optional security feature as part of the lock suite. The feature allows users to unlock their phone by speaking a passphrase.

To set-up the app, the user should select the Rust feature from the phone lock setting. Then the user trains the system by speaking and recording the passphrase three times. If the voice authentication feature is set, the user can say the passphrase from any screen or from the search bar. If the authentication is successful, the user is prompted by the app that it is listening and then waits for the commands. This feature can also be used to unlock the phone.

(8) Green Mint (*tested on iOS in both studies*): Mint is a virtual assistant for iPhone smartphones that uses a natural language user interface to interpret and execute the user's commands. Mint can be personalized to accept only the voice of the owner.

To set the voice authentication feature, the user should turn on Mint and train with the phrases prompted. In the testing phase, the user can say the passphrase when the device is locked or unlocked. If the system recognizes the voice as valid, it opens the Mint display and the app waits for a command.

C. Study Assumptions and Hypotheses

Our hypothesis is that the tested voice authentication apps do not provide a security level expected by the users, i.e., they may have high FAR when subjected to the Sneakers' voice impersonation attacks (e.g., replay attack, reorder attack, and conversion attack), although they may be successful in authenticating the user in a benign setting and rejecting a different speaker's voice (zero-effort attack).

We use the different speaker attack as the baseline of the study that would have the lowest FAR, because of the inherent difference between the voice of two people. However, replay attack and reorder attack may have the highest FAR since the app is presented with the voice that is the same as the voice of the benign user. The voice conversion attack may also have a high FAR since the synthesized voice captures the characteristics of the victim's voice. The FARs of all of the Sneakers' attacks should be significantly higher than the FAR of the baseline different speaker attack.

D. Threat Model

Our focus is to attack real-world popular apps deploying voice biometrics for the purpose of user authentication. We consider a scenario where the attacker has physical access to the phone either permanently (stealing), or temporarily (lunch-time attack). The attacker's goal is to unlock the phone or to access secret data by authenticating to the apps via speaking or playing the samples that are potentially accepted by apps. We conservatively assume that the attacker can launch the attack only once (within one attempt or trial). In practice, an attacker may be able to increase his chances of success by trying multiple times, as multiple attempts may be allowed by the apps to improve system's usability (i.e., to reduce FRR).

Our attacks are designed based on a black-box model in which the attacker does not have any knowledge of the underlying voice biometric algorithm used by the apps. The apps may or may not detect the liveness of the samples.

To prepare the attacked samples, we assume that the attacker can collect the victim's audio sample using standard audio recording tools (e.g., recording software and smartphone audio recording apps) with minimal background noise (e.g., a quiet office). We can also assume that the attacker might have access to recordings of the user publicly posted online (e.g., teaching lectures, seminar talks, social media posts). We expect the attacker to have access to personal audio recording devices (e.g., a smartphone), and off-the-shelf audio playing and voice conversion tools to launch his attacks. This assumption is a crucial choice for our attacks to be implemented with low-cost and high convenience, and to be realistic.

Although we do not aim to attack the "speech recognition" feature on the apps, since the apps deploy text-dependent speaker recognition, for a successful attack both the "speech" and "speaker" recognition should accept the attacked samples.

Finally, even though some of the apps offer the speaker recognition only as an optional feature, we assume that the user has the speaker recognition feature enabled.

V. Attack Implementation and Study Design

A. Experimental Setup

To test our hypothesis, we designed a two-phase study: Phase I, the preliminary study, and Phase II, the main study. In the first study, we selected several (8) voice authentication apps and tested the benign and simplistic attack cases against two users who acted as the victims (two other users acted as the attackers for the different speaker attack). After this initial study, we ran the second phase of the study with a smaller number of apps (5) with a larger number of users ($N = 10$), acting as both victims and attackers, and against more advanced attack set-ups.

1) Devices and Tools: In the preliminary study, we tested the apps on a Samsung Galaxy Stellar, a Samsung Galaxy S5, and an iPhone 6s. In the main study, we tested the apps on a Samsung Galaxy S5 and an iPhone 7.

In the preliminary study, we ran the apps on both iOS and Android platform, as available. However, in the second phase, for all the apps compatible with both Android and iOS, we ran the app only on Android platform. We assume that the underlying voice biometrics algorithm is the same across different platforms. Hence, evaluating the app only on one platform would be sufficient.

To record and play back the audio samples, as part of our different attack settings, we used the built-in microphone and speaker of an HP Pavilion with an Intel core i7 and an Apple MacBook Air with an Intel core i5. We recorded audio using Audacity 2.1.2 and played back the audio with Groove Music on the HP Pavilion and with iTunes on MacBook Air laptop.

2) Data Collection: To test the apps, we collected audio samples in a quiet environment in the stereo sound system with a sampling rate of 44100 Hz using Audacity. This captures a real-world attack setting, as it is fair to assume that users often train and perhaps test the systems in less noisy conditions to reduce FRR. Also, the attacker can record the victims in such settings (e.g., during lectures or in quiet shared offices) using high-quality recorders. To test the attack settings, we asked all the participants in the two studies to speak a 700-word article once to train the conversion tool, numbers from 0 to 9, and the apps' passphrases to test the attack and benign settings.

Quality of the "source" samples has a direct effect on the acceptance of converted voice. Based on this key intuition, we replaced the source with a high-quality TTS voice to achieve higher attack accuracy in our advanced conversion attack. TTS-generated voices have previously been used in other security contexts (e.g., [30]), however, our use of TTS here is to create less-noisy voice conversions. To collect this data, we used the IBM Watson Text to Speech demo tool to speak and record numbers, phrases, and the same article as spoken by the participants in our study. We opened the demo application on a Dell desktop and entered the text to produce the synthesized speech in the text-to-speech voice, and recorded the output. To eliminate the background noise, we fed the output of the computer's speaker to the built-in microphone input with a *4 conductor audio cable*, then ran the text to speech tool to play the synthesized speech and at the same time recorded the audio. We used these collected samples as the attacker's voice in the second study to train the voice conversion tool.

3) Voice Dataset Preparation: After collecting the data, we split the spoken article into 23 samples of around 5 seconds long, which contained the same phrase spoken by each person. These samples were labeled from 0 to 22. We changed the samples to mono sound system and 16000 Hz rate, and exported them each as a signed 16-bit PCM wav file. Each digit and each phrase were similarly exported to one single audio file.

Zero-effort Different Speaker Attack Samples: In the first study, we trained the system with the voice of the user and used the voice of the attacker as the different speaker's voice. For the second study, we trained the system with the voice of the victim user and used the samples we collected from the other study users, to attack the system. In the first study, the user and the attacker are of different genders, while the second study has the user and the attacker of both the same and different genders.

Replay Attack Samples: These audio files are the samples we collected from the authorized user who trained the voice authentication app. To test the replay attack, these samples are played back in front of the voice authentication app.

Reorder Attack Samples: For the apps that display randomized numerical phrases, we used Audacity to create the random numeric phrases from the pre-recorded digits. Then we played back the constructed phrase to attack the system.

Conversion Attack Samples: We used Festvox [10], [31] to create the synthesized (converted) audio. Festvox has been widely used and developed over the past 18 years and is a well recognized tool for voice conversion. We trained the tool with the collected samples to convert the source voice to the target voice. Once trained, we used the system to generate the audio files expected by the voice authentication apps (e.g., random digit) in the victim's voice. The training dataset was not used in the testing phase. The voice of the target in the voice conversion training was the victim user who trained the voice authentication system. In the first study, we input the attacker's voice to train the voice conversion and to generate the synthesized samples, while in the second study, we used the TTS voice as the attacker's voice (conversion source).

B. Study Protocol

1) Phase I: Preliminary Study: Figure 3 shows the flow of the study protocol in the preliminary study as follows:

Step 1–Data Collection: We first collected the data following the procedure explained in Section V-A2.

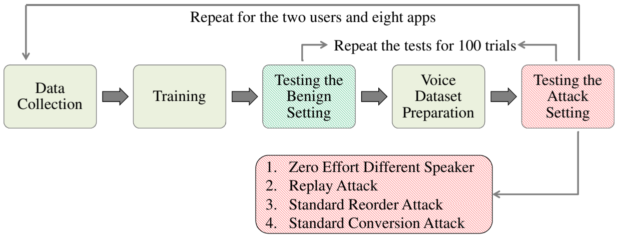
Repeat for the two users and eight apps
Repeat the tests for 100 trials
Data Collection
Training
Testing the Benign Setting
Voice Dataset Preparation
Testing the Attack Setting
1. Zero Effort Different Speaker
2. Replay Attack
3. Standard Reorder Attack
4. Standard Conversion Attack

Fig. 3. Protocol flow in the preliminary study.

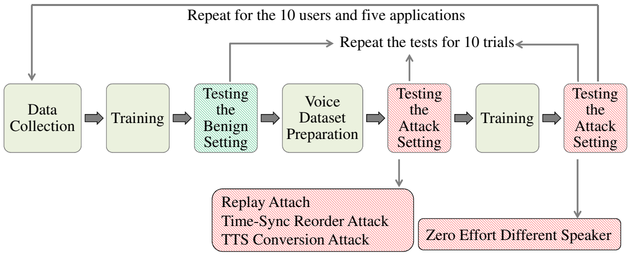
Repeat for the 10 users and five applications
Repeat the tests for 10 trials
Data Collection
Training
Testing the Benign Setting
Voice Dataset Preparation
Testing the Attack Setting
Training
Testing the Attack Setting
Replay Attach
Time-Sync Reorder Attack
TTS Conversion Attack
Zero Effort Different Speaker

Fig. 4. Protocol flow in the main study.

Step 2–Training: We asked the users to train each of the apps following the training instruction and procedures provided and required by the apps. The training was performed along the data collection.

Step 3–Testing the Benign Case: The first test was with the authorized user's voice in a quiet room with little background noise to confirm if the app can correctly recognize the authorized user. The authorized user sat at a desk while speaking the passphrases for each of the apps on the phones. This test was repeated for 100 trials by each of the two users.

Step 4–Voice Dataset Preparation: We created the samples required to test the apps as defined in Section V-A3.

Step 5–Testing Attacks: For the replay attack, conversion attack, and zero-effort different speaker attack, we tested the system with samples of the authorized user, converted voice, and voice of an unauthorized speaker, respectively.

We opened the audio files of the authorized voice, the converted voice, and the different speaker's voice on the music player software on a laptop and held the phone close to the laptop. While playing the audio samples, we ran the voice authentication app to recognize the voice. We performed each of these tests for 100 trials. We recorded all the response results generated by the voice authentication apps for further analysis presented in Section VI-A.

2) *Phase II: Main Study:* In the second study, we had to train the system with the voice of each of the 10 users and collect the data required for attacking the system. However, since we could only train the system with the voice of one user, we had to repeat the training step twice: the first time to test the benign case, the replay attack and the conversion attack, and the second time after we collected the data from all the users to test the zero-effort different speaker attack. We followed the procedure described next (and Figure 4):

Step 1–Data Collection: In a quiet room, with little background noise, we collected data from the users (Section V-A2).

Step 2–Training: We asked the users to follow the steps given by each of the apps to train the system.

Step 3–Testing the benign Case: We asked the user to test the voice authentication app while holding the phone in their hand. We repeated this test for 10 trials.

Step 4–Voice Dataset Preparation: After the online session, we prepared the dataset for the attack setting (i.e., training the conversion system, preparing the reordered numbers and replayed samples). We followed the same procedure as in the first study to create the attacked samples. However, we tested the TTS conversion attack in this phase of the study.

Step 5–Testing the replay attack, reorder attack, and conversion attack offline: In a quiet room, we played back the audio collected from the users for the replay attack, reordering of the user's voice for the reorder attack (i.e., for Rose), and converted voice for the advanced conversion attack. We repeated this test for 10 trials for each of the attack scenarios.

We repeated Step 1 to 5 for each of the 10 users, then followed the study protocol as below:

Step 6–Training: We called each user to train the system for the second time to test the zero-effort different speaker attack.

Step 7–Testing the zero-effort different speaker attack In a quiet room, we played back the different speaker's audio samples. For each of the users, we played the audio collected from the 9 other users and also another audio sample of one of the researchers (i.e., we tested the different speaker attack for 10 attackers). For each user, we played back the audio once.

We repeated step 6 and 7 for all the 10 users.

The 10 participants in this study were 5 female and 5 male users from the employees of our university. The participation in the study was strictly voluntary and approved by our IRB.

VI. Results and Analysis

A. Preliminary Study Analysis

Analysis of Benign Setting. Figure 5 shows the rates of rejecting an authorized user (benign setting) for various apps tested in the preliminary study (averaged over the two users and 100 trials). Most of the apps showed low false rejection rates indicating that generally, the apps are successful in authenticating the authorized user.

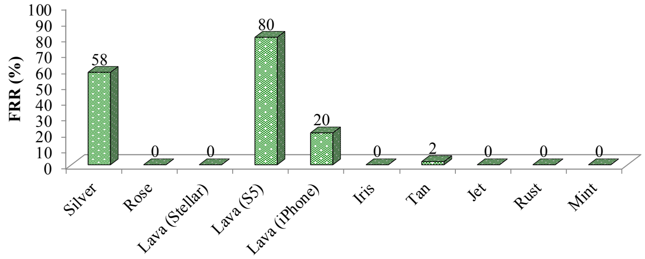

Fig. 5. The average false rejection rates for the apps tested in the benign setting in the preliminary study. Except for Silver and Lava, other apps show high accuracy in authenticating the authorized user.

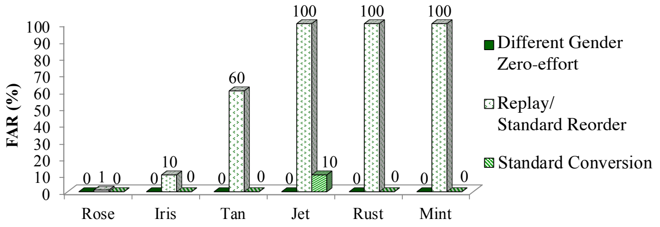

Fig. 6. The average false acceptance rates for the apps tested in the attack settings in the **preliminary study**. The two text-prompted apps, i.e, Rose and Iris, are tested only against the standard reorder attack, the rest, i.e, Tan, Jet, Rust, and Mint, are text-dependent and are tested against replay attack. We see that all apps can resist the baseline zero-effort attacks very well. Also, most of the apps can defeat the standard reorder and standard conversion attacks, but are highly vulnerable to the replay attack.

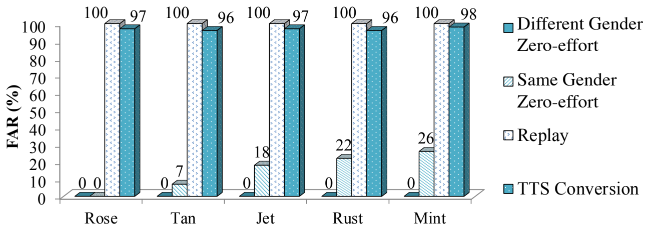

Fig. 7. The average false acceptance rates for the tested apps in the attack settings in the **main study**. We see that while the evaluated apps offer relatively high robustness to the baseline zero-effort different speaker attack, they are highly vulnerable to the replay attack and TTS conversion attack. The time-synced reorder attack was applicable to only Rose app and also showed a high success rate of 82% (not shown in this figure).

Rose, Tan, Jet, and Rust showed 0% FRR (i.e., authenticated all the authorized users' samples) on the Android platform. On iOS platform, both Iris and Mint showed a 0% error rate.

Lava exhibited different behavior on different devices showing 0% FRR on Galaxy Stellar, 80% on Galaxy S5, and 20% on iPhone. Silver, which is an educational tool, did not show a reliable performance in the benign setting (the FRR of this app was 58%). Due to this poor performance in authenticating the authorized user, we discarded these two apps from the rest of the study.

Analysis of Zero-effort Different Speaker Attack. This attack is the simplest form of attack against voice authentication apps and is used as the baseline in our studies. In this attack, the attacker simply speaks to the apps to get authenticated. Based on the differences between the features of the authorized and unauthorized user's voice, the apps are expected to reject this form of attack and provide a false acceptance rate of 0% (rejecting all the instances of the different speaker's voice). As hypothesized, the results show that the apps rejected all instances of the voice of a *different gender speaker*, as shown in Figure 6.

Analysis of Replay Attack. In this test, we played back the pre-recorded samples of the voice of the authorized speaker to the apps that accept a fixed passphrase. The results are presented in the last four grouped bars of Figure 6. The four apps that work with a fixed passphrase (Tan, Jet, Rust, and Mint) were highly susceptible to our replay attack with a false acceptance rate of 60% for Tan and 100% for the other three apps. This shows that if an attacker can collect the audio sample of an authorized user speaking the same passphrase, s/he could easily authenticate posing as the victim user.

Analysis of Reorder Attack. For the text-prompted apps (Rose and Iris), we reordered the arrangement of numbers as displayed by the app for each attempt and played back the reordered audio. The text-prompted app Rose, showed resistance against the standard reorder attack in the preliminary study while Iris showed 10% FAR against the same attack as shown in the first two grouped bars of Figure 6.

Analysis of Conversion Attack. In this attack, we converted the voice of the attacker to the voice of the victim speaking the phrases expected by the apps. For the challenge-response apps, we first converted numbers from 0 to 9 to the victim's voice and then reordered them to generate the phrases. As presented in Figure 6, this attack was not successful in the preliminary study and most of the apps did not accept the converted voice samples. This demonstrates the resilience of these apps against a standard form of conversion attack, which actually had succeeded against isolated voice biometrics algorithms with audio file input as shown in [16], [20], [28].

B. Main Study Analysis

In the main study, for all the 5 apps, we tested the benign scenario, zero-effort different speaker attack against the same and different gender voice, replay attack, and TTS conversion attack. Rose was the only app that offered both fixed phrases, and challenge response features. Hence, Rose was tested against time-synced reorder attack for the challenge-response option, and against replay attack and TTS conversion attack for the fixed-phrase option.

Analysis of Benign Setting. In the benign setting averaged over all the 10 users and 10 trials, the FRR was 0% for Mint, Rust, Jet, and Tan. This observation indicates that all these apps could successfully authenticate the authorized user. Rose was the only app that could not authenticate all the authorized user's attempts but the FRR for this app was still very low at 3%. The low FRR gives us the confidence that the apps function properly in a benign setting, further justifying their popularity among users and the need for our security study.

Analysis of Zero-effort Different Speaker Attack. We further divide the different speaker test into the same gender and different gender cases denoted as "same gender zero-effort" and "different gender zero-effort" respectively (see Figure 7). Due to the differences between the voices of people of the different genders, we expect the apps to perform better in distinguishing voices of different genders. Indeed, the average FPR for this attack was 0% for all the apps when the apps were tested against the different gender's voice. Although the apps showed a high success rate in recognizing the different genders, not all of them succeeded in recognizing voices belonging to the same gender. Among all the tested apps, Rose was the only app that rejected all instances of this attack (i.e., 0% FAR for the same gender different speaker attack). Other apps had FAR values ranging from 7% for Tan to 26% for Mint. This result shows that the apps offer a limited, although still reasonable level of security when a user with a different voice (but the same gender) tries to login to the apps.

Analysis of Replay Attack. The FAR value for the replay attack was 100%, averaged over the 10 users and the 10 trials for each app. This result shows that none of the apps under the test could detect the liveness of the samples and accepted the pre-recorded audio of the authorized user. This attack exposes a serious vulnerability in the apps, given that the attacker may record samples of the victim without being noticed by general-purpose audio recording devices such as smartphones.

Analysis of Reorder Attack. Rose, in its challenge-response mode, displays a 4 digit number that should be spoken by the user in a given time period. To run a successful attack against this app, we played the reordered audio and started voice authentication right after playing the sample as presented in Figure 2 (i.e., time-synced reorder attack). Using this approach, the attack succeeded with an FAR of 82% compared to 0% in the preliminary study.

Analysis of TTS Conversion Attack. By replacing the attacker's voice with noise-free high-quality TTS samples, we achieved an FAR of over 95% for all the apps averaged over all users for the TTS conversion attack as compared to 0% success of the standard conversion attack in the preliminary study. This attack shows the vulnerability of both text-dependent and text-independent voice authentication methods. As long as the attacker has access to "some" voice samples of the victim, she can create *any* text from the voice samples and authenticate to apps with a high probability by impersonating the victim.

C. Statistical Analysis.

Using Wilcoxon Signed-Rank Test, we compared the FARs of the Sneakers' attacks (averaged over all apps) with the FAR of zero-effort different speaker attack as the baseline for the existence of statistical differences. We report the results with a 95% confidence level. The results of the test indicate that the FAR for all the Sneakers' attacks in the main study is significantly higher than that of the baseline attack. The test shows that the replay attack was more successful than the zero-effort different speaker attack ($Z = -3.051, p = 0.002$). Similarly, the conversion attack showed a higher success rate compared to the zero-effort different speaker attack ($Z = -2.913, p = 0.004$). The reorder attack also performed better than baseline attack ($Z = -2.831, p = 0.005$).

D. Increasing the Number of Attempts

All the error rates reported so far were averaged over the number of experimental trials and therefore represent the error rate for *a single authentication attempt*. An adversary may repeat the attack multiple times to increase her success probability. Although none of the apps tested in our main study lock out the user after unsuccessful attempts (a weakness of these apps on its own), hypothetically we can assume that the apps at least allow three trials before locking an account. In this realistic context, for an attack with success rate of p for the first trial, the success rate would increase to $p^3 - 3p^2 + 3p$ after the third trial. For example, for an app with FAR = 96% against the TTS conversion attack in one trial, the attack will yield FAR = 99.99% in 3 trials. This suggests that almost all of our attacks could succeed with a very high probability for a limited number of attempts.

E. Other types of Attacks

The results of our study can be extended to an attacker who compromises the victim's phone using a malicious app that can automatically input the (imitated) voice samples to the voice authentication app. An example of this type of attack against *speech recognition* (*not* speaker verification) has been introduced in [32], in which an attack against the Android's built-in personal assistant app was launched to accept the prepared audio files (i.e., a voice command) from the malicious app. Diao et al. [32] suggest speaker verification as a defense mechanism against their remote attack (and possibly other attacks against speech recognition systems [33], [34]). However, note that the apps seem vulnerable to voice imitation attacks introduced in our work. Therefore, even if a speaker verification were to be deployed, a malicious app could remotely exploit the security of the system, without physical access to the phone.

VII. Conclusions

Voice biometrics is being used increasingly by mobile apps to authenticate the users to the phone or the apps. In this paper, we introduced Sneakers, a set of voice impersonation attacks, and ran a black box evaluation of several highly popular voice authentication apps against the attacks. We presented live samples of the authorized users to test the applications in the benign setting. As the baseline, we presented the apps with the voice of a different speaker. We defined three sub-attacks as part of Sneakers: the replay of the authorized user's voice, reordering of the user's voice and the conversion from an attacker's or TTS voice to the authorized user's voice.

The results show that while these apps might be mature enough to recognize an authorized user and a different speaker's voice, they fail when confronted with the replicated and synthesized voice. Given that users are often not concerned about speaking out loud or publishing their voice online, collecting samples of the users' voices is very easy and the attacks introduced in this paper do not take much effort. Also, the increasing availability of speech synthesis tools (initially introduced for text to speech) opens the door for the attackers to authenticate to the apps by the voice signals that may satisfy the biometric verification algorithm, which shows the significance of improving the speaker verification techniques and devising security measures to protect them against spoofing attacks before deployment in security applications.

Acknowledgment

We thank the anonymous reviewers for their constructive comments. We are also thankful to Manasa Chithrashekar and Asutosh Nanda for their help in the background study. This work was partially supported by the following grants: CNS-1714807, CNS-1526524 and CNS-1547350.

References

[1] (Accessed 05/06/2018) Set up your device for automatic unlock. [Online]. Available: https://goo.gl/iXNpJK

[2] (Accessed 05/06/2018) Apple adds individual voice recognition to Hey Siri in iOS 9. [Online]. Available: https://goo.gl/LWg4k5

[3] (Accessed 05/06/2018) Dragon Mobile Assistant – Simplify your mobile life. [Online]. Available: https://goo.gl/HrmNBu

[4] (Accessed 05/06/2018) Authentify xFA provides simple, secure primary authentication using digital certificates and voice biometrics. [Online]. Available: https://goo.gl/SJ9nQA

[5] (Accessed 05/06/2018) AppLock from Sensory Keeps Apps Safe with Face and Voice Biometrics. [Online]. Available: https://goo.gl/BKkWYl

[6] (Accessed 05/06/2018) Barclays rolls out voice biometrics for phone banking. [Online]. Available: https://goo.gl/rxjVSs

[7] (Accessed 05/06/2018) HSBC rolls out voice and touch ID security for bank customers. [Online]. Available: https://goo.gl/TR5FyJ

[8] (Accessed 05/06/2018) Banks turning to voice recognition. [Online]. Available: https://goo.gl/bVTm4J

[9] (Accessed 05/06/2018) More banks turn to biometrics to keep an eye on security. [Online]. Available: https://goo.gl/JcM5wo

[10] (Accessed 05/06/2018) TRANSFORM: Flexible Voice Synthesis Through Articulatory Voice Transformation. [Online]. Available: http://goo.gl/ZrRtXG

[11] (Accessed 05/06/2018) CandyVoice Website. [Online]. Available: https://www.candyvoice.com/

[12] (Accessed 05/06/2018) Copy the voice of anyone. [Online]. Available: https://lyrebird.ai/

[13] (Accessed 05/06/2018) IBM Watson Speech to Text - Convert human voice into written word. [Online]. Available: https://ibm.co/2r0ZUKg

[14] Y. W. Lau, M. Wagner, and D. Tran, "Vulnerability of speaker verification to voice mimicking," in *Proceedings of 2004 International Symposium on Intelligent Multimedia, Video and Speech Processing*, 2004.

[15] M. Blomberg, D. Elenius, and E. Zetterholm, "Speaker verification scores and acoustic analysis of a professional impersonator," in *Proc. FONETIK*, vol. 29, 2004, pp. 58–63.

[16] T. Kinnunen, Z.-Z. Wu, K. A. Lee, F. Sedlak, E. S. Chng, and H. Li, "Vulnerability of speaker verification systems against voice conversion spoofing attacks: The case of telephone speech," in *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2012, pp. 4401–4404.

[17] Z. Wu, N. Evans, T. Kinnunen, J. Yamagishi, F. Alegre, and H. Li, "Spoofing and countermeasures for speaker verification: a survey," *Speech Communication*, vol. 66, 2015.

[18] Z. Wu, A. Khodabakhsh, C. Demiroglu, J. Yamagishi, D. Saito, T. Toda, and S. King, "Sas: A speaker verification spoofing database containing diverse attacks," in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2015.

[19] Z. Wu and H. Li, "Voice conversion and spoofing attack on speaker verification systems," in *Signal and Information Processing Association Annual Summit and Conference (APSIPA), 2013 Asia-Pacific*, 2013.

[20] D. Mukhopadhyay, M. Shirvanian, and N. Saxena, "All your voices are belong to us: Stealing voices to fool humans and machines," in *European Symposium on Research in Computer Security*, 2015, pp. 599–621.

[21] S. Furui, "Research of individuality features in speech waves and automatic speaker recognition techniques," *Speech communication*, vol. 5, no. 2, pp. 183–197, 1986.

[22] D. Bhattacharyya, R. Ranjan, F. Alisherov, M. Choi *et al.*, "Biometric authentication: A review," *International Journal of u-and e-Service, Science and Technology*, vol. 2, no. 3, pp. 13–28, 2009.

[23] Z. Wu, S. Gao, E. S. Cling, and H. Li, "A study on replay attack and anti-spoofing for text-dependent speaker verification," in *2014 Annual Summit and Conference Asia-Pacific Signal and Information Processing Association (APSIPA)*. IEEE, 2014, pp. 1–5.

[24] P. L. De Leon, M. Pucher, and J. Yamagishi, "Evaluation of the vulnerability of speaker verification to synthetic speech," 2010.

[25] P. L. De Leon, V. R. Apsingekar, M. Pucher, and J. Yamagishi, "Revisiting the security of speaker verification systems against imposture using synthetic speech," in *2010 IEEE International Conference on Acoustics Speech and Signal Processing (ICASSP)*, 2010.

[26] F. Alegre, R. Vipperla, N. Evans, and B. Fauve, "On the vulnerability of automatic speaker recognition to spoofing attacks with artificial signals," in *EUSIPCO 2012, 20th European Signal Processing Conference*, 2012.

[27] N. Evans, T. Kinnunen, J. Yamagishi, Z. Wu, F. Alegre, and P. De Leon, "Speaker recognition anti-spoofing," in *Handbook of Biometric Anti-Spoofing*. Springer, 2014, pp. 125–146.

[28] Z. Wu, A. Larcher, K.-A. Lee, E. Chng, T. Kinnunen, and H. Li, "Vulnerability evaluation of speaker verification under voice conversion spoofing: the effect of text constraints." in *INTERSPEECH*, 2013.

[29] (Accessed 05/06/2018) Stealing Voice Prints. [Online]. Available: https://goo.gl/4pZiNy

[30] G. Zhang, C. Yan, X. Ji, T. Zhang, T. Zhang, and W. Xu, "Dolphinattack: Inaudible voice commands," in *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*. ACM, 2017, pp. 103–117.

[31] T. Toda, A. W. Black, and K. Tokuda, "Spectral Conversion Based on Maximum Likelihood Estimation Considering Global Variance of Converted Parameter," in *Proc. ICASSP*, vol. 1, 2005.

[32] W. Diao, X. Liu, Z. Zhou, and K. Zhang, "Your voice assistant is mine: How to abuse speakers to steal information and control your phone," in *Proceedings of the 4th ACM Workshop on Security and Privacy in Smartphones & Mobile Devices*. ACM, 2014.

[33] 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*. ACM, 2015.

[34] P. J. Young, J. H. Jin, S. Woo, and D. H. Lee, "Badvoice: Soundless voice-control replay attack on modern smartphones," in *International Conference on Ubiquitous and Future Networks (ICUFN)*. IEEE, 2016.

Appendix

A. Additional Figures and Tables

The anonymized list of the apps used in the two studies is summarized in Table II.

TABLE II
Evaluated Apps In The Studies. ⋆ Apps Tested In Both Studies; † Developers Offer Paid And Unpaid Apps, Using The Same Speaker Recognition Engine/algorithm In Both.

App Name	Developer	Version	Category	Passphrase	Device
Silver	Grey	x.x	Text-dependent	User-selected	Samsung Galaxy Stellar and S5
Rose ⋆	Pink †	x.x, x.x	Text-prompted	User-selected/random numeric	Samsung Galaxy Stellar and S5
Lava	Red	x.x	Text-prompted	Random numeric	Samsung Galaxy Stellar, S5, iPhone 6s, 7
Iris	Blue	x.x	Text-prompted	Random numeric	iPhone 6s
Tan ⋆	Brown †	x.x	Text-dependent	User-selected	Samsung Galaxy Stellar and S5
Jet ⋆	Black †	x.x	Text-dependent	Fixed 2-word	Samsung Galaxy Stellar and S5
Rust ⋆	Orange	x.x	Text-dependent	Fixed 2-word	Samsung Galaxy Stellar and S5
Mint ⋆	Green	x.x	Text-dependent	Fixed 2-word	iPhone 6s/iPhone 7