Who is Real Bob? Adversarial Attacks on Speaker Recognition Systems

Guangke Chen*†‡, Sen Chen¶§‖, Lingling Fan§, Xiaoning Du§, Zhe Zhao*, Fu Song*‡‡✉ and Yang Liu§
*ShanghaiTech University, †Shanghai Institute of Microsystem and Information Technology, Chinese Academy of Sciences
‡University of Chinese Academy of Sciences, §Nanyang Technological University
¶College of Intelligence and Computing, Tianjin University
‡‡Shanghai Engineering Research Center of Intelligent Vision and Imaging, ‖Co-first Author

2021 IEEE Symposium on Security and Privacy (SP) | 978-1-7281-8934-5/21/$31.00 ©2021 IEEE | DOI: 10.1109/SP40001.2021.00004

***Abstract*—Speaker recognition (SR) is widely used in our daily life as a biometric authentication or identification mechanism. The popularity of SR brings in serious security concerns, as demonstrated by recent adversarial attacks. However, the impacts of such threats in the practical black-box setting are still open, since current attacks consider the white-box setting only.**

In this paper, we conduct the first comprehensive and systematic study of the adversarial attacks on SR systems (SRSs) to understand their security weakness in the practical black-box setting. For this purpose, we propose an adversarial attack, named FAKEBOB, to craft adversarial samples. Specifically, we formulate the adversarial sample generation as an optimization problem, incorporated with the confidence of adversarial samples and maximal distortion to balance between the strength and imperceptibility of adversarial voices. One key contribution is to propose a novel algorithm to estimate the score threshold, a feature in SRSs, and use it in the optimization problem to solve the optimization problem. We demonstrate that FAKEBOB achieves 99% targeted attack success rate on both open-source and commercial systems. We further demonstrate that FAKEBOB is also effective on both open-source and commercial systems when playing over the air in the physical world. Moreover, we have conducted a human study which reveals that it is hard for human to differentiate the speakers of the original and adversarial voices. Last but not least, we show that four promising defense methods for adversarial attack from the speech recognition domain become ineffective on SRSs against FAKEBOB, which calls for more effective defense methods. We highlight that our study peeks into the security implications of adversarial attacks on SRSs, and realistically fosters to improve the security robustness of SRSs.

I. INTRODUCTION

Speaker recognition [1] is an automated technique to identify a person from utterances which contain audio characteristics of the speaker. Speaker recognition systems (SRSs) are ubiquitous in our daily life, ranging from biometric authentication [2], forensic tests [3], to personalized service on smart devices [4]. Machine learning techniques are the mainstream method for implementing SRSs [5], however, they are vulnerable to adversarial attacks (e.g., [6], [7], [8]). Hence, it is vital to understand the security implications of SRSs under adversarial attacks.

Though the success of adversarial attack on image recognition systems has been ported to the speech recognition systems in both the white-box setting (e.g., [9], [10]) and black-box setting (e.g., [11], [12]), relatively little research has been done on SRSs. Essentially, the speech signal of an utterance consists of two major parts: the underlying text and the characteristics of the speaker. To improve the performance, speech recognition will minimize speaker-dependent variations to determine the underlying text or command, whereas speaker recognition will treat the phonetic variations as extraneous noise to determine the source of the speech signal. Thus, adversarial attacks tailored to speech recognition systems may become ineffective on SRSs.

An adversarial attack on SRSs aims at crafting a sample from a voice uttered by some source speaker, so that it is misclassified as one of the enrolled speakers (untargeted attack) or a target speaker (targeted attack) by the system under attack, but still correctly recognized as the source speaker by ordinary users. Though current adversarial attacks on SRSs [13], [14] are promising, they suffer from the following three limitations: (1) They are limited to the white-box setting by assuming the adversary has access to the information of the target SRS. Attacks in a more realistic black-box setting are still open. (2) They only consider either the close-set identification task [13] that always classifies an arbitrary voice as one of the enrolled speakers [15], or the speaker verification task [14] that checks if an input voice is uttered by the unique enrolled speaker or not [16]. Attacks on the open-set identification task [17], which strictly subsumes both close-set identification and speaker verification, are still open. (3) They do not consider over-the-air attacks, hence it is unclear whether their attacks are still effective when playing over the air in the physical world. Therefore, in this work, *we investigate the adversarial attack on all the three tasks of SRSs in the practical black-box setting*, in an attempt to understand the security weakness of SRSs under adversarial attack in practice.

In this work, we focus on the black-box setting, which assumes that the adversary can obtain at most the decision result and scores of the enrolled speakers for each input voice. Hence attacks in the black-box setting is more practical yet more challenging than the existing white-box attacks [13], [14]. We emphasize that the scoring and decision-making mechanisms of SRSs are different among recognition tasks [18]. Particularly, we consider 40 attack scenarios (as demonstrated in Fig. 2) in total differing in attack types (targeted vs. untargeted), attack channels (API vs. over the air), genders of source and target speakers, and SR tasks (cf. §II-B). We demonstrate our attack on 16 representative attack scenarios.

DOI 10.1109/SP40001.2021.00004

To launch such a practical attack, two technical challenges need to be addressed: (C1) crafting adversarial samples as less imperceptible as possible in the black-box setting, and (C2) making the attack practical, namely, adversarial samples are effective on an unknown SRS, even when playing over the air in the physical world. In this paper, we propose a practical black-box attack, named FAKEBOB, which is able to overcome these challenges.

Specifically, we formulate the adversarial sample generation as an optimization problem. The optimization objective is parameterized by a confidence parameter and the maximal distortion of noise amplitude in L_∞ norm to balance between the strength and imperceptibility of adversarial voices, instead of using noise model [10], [19], [20], due to its device- and background-dependency. We also incorporate the score threshold, a key feature in SRSs, into the optimization problem. To solve the optimization problem, we leverage an efficient gradient estimation algorithm, i.e., the natural evolution strategy (NES) [21]. However, even with the estimated gradients, none of the existing gradient-based white-box methods (e.g., [22], [23], [10], [24]) can be directly used to attack SRSs. This is due to the score threshold mechanism, where an attack fails if the predicated score is less than the threshold. To this end, we propose a novel algorithm to estimate the threshold, based on which we leverage the Basic Iterative Method (BIM) [23] with estimated gradients to solve the optimization problem.

We evaluate FAKEBOB for its attacking capabilities, on 3 SRSs (i.e., ivector-PLDA [25], GMM-UBM [16] and xvector-PLDA [26]) in the popular open-source platform Kaldi [27] in the research community and 2 commercial systems (i.e., Talentedsoft [28] and Microsoft Azure [29]) which are proprietary without any publicly available information about the internal design and implementations, hence completely black-box. We evaluate FAKEBOB using 16 representative attack scenarios (out of 40) based on the following five aspects: (1) effectiveness/efficiency, (2) transferability, (3) practicability, (4) imperceptibility, and (5) robustness.

The results show that FAKEBOB achieves 99% targeted attack success rate (ASR) on all the tasks of ivector-PLDA, GMM-UBM and xvector-PLDA systems, and 100% ASR on the commercial system Talentedsoft within 2,500 queries on average (cf. §V-B). To demonstrate the transferability, we conduct a comprehensive evaluation of transferability attack on ivector-PLDA, GMM-UBM and xvector-PLDA systems under cross-architecture, cross-dataset, and cross-parameter circumstances and the commercial system Microsoft Azure. FAKEBOB is able to achieve 34%-68% transferability (attack success) rate except for the speaker verification of Microsoft Azure. The transferability rate could be increased by crafting high-confidence adversarial samples at the cost of increasing distortion. To further demonstrate the practicability and imperceptibility, we launch an over-the-air attack in the physical world and also conduct a human study on the Amazon Mechanical Turk platform [30]. The results indicate that FAKEBOB is effective when playing over the air in the physical world against both the open-source systems and the open-set identification task of Microsoft Azure (cf. §V-D) and it is hard for humans to differentiate the speakers of the original and adversarial voices (cf. §V-E).

Finally, we study four defense methods that are reported promising in speech recognition domain: audio squeezing [10], [31], local smoothing [31], quantization [31] and temporal dependency-based detection [31], due to lacking of domain-specific defense solutions for adversarial attack on SRSs. The results demonstrate that these defense methods have limited effects on FAKEBOB, indicating that FAKEBOB is a practical and powerful adversarial attack on SRSs.

Our study reveals that the security weakness of SRSs under black-box adversarial attacks. This weakness could lead to lots of serious security implications. For instance, the adversary could launch an adversarial attack (e.g., FAKEBOB) to bypass biometric authentication on the financial transaction [2], [32] and smart devices [4], as well as high-security intelligent voice control systems [33] so that follow-up voice command attacks can be launched, e.g., CommanderSong [10] and hidden voice commands [34]. For the voice-enabled cars using Dragon Drive [33], the attacker could bypass its voice biometrics using FAKEBOB so that command attacks can be launched to control cars. Even for commercial systems, it is a significant threat under such a practical black-box adversarial attack, which calls for more robust SRSs. To shed further light, we discuss the potential mitigation and further attacks to understand the arm race in this topic. In summary, our main contributions are:

- To our knowledge, this is the first study of targeted adversarial attacks on SRSs in the black-box setting. Our attack is launched by not only using gradient estimation based methods, but also incorporating the score threshold into the adversarial sample generation. The proposed algorithm to estimate the score threshold is unique in SRSs.
- Our black-box attack addresses not only the speaker recognition tasks considered by existing white-box attacks but also the more general task, open-set identification, which has not been considered by previous adversarial attacks.
- Our attack is demonstrated to be *effective* on the popular open-source systems and commercial system Talentedsoft, *transferable* and *practical* on the popular open-source systems and the open-set identification task of Microsoft Azure even when playing over the air in the physical world.
- Our attack is *robust* against four potential defense methods which are reported very promising in speech recognition domain. Our study reveals the security implications of the adversarial attack on SRSs, which calls for more robust SRSs and more effective domain-specific defense methods.

For more information of FAKEBOB, please refer to our website [35] which includes voice samples and source code.

II. BACKGROUND

In this section, we introduce the preliminaries of speaker recognition systems (SRSs) and the threat model.

A. Speaker Recognition System (SRS)

Speaker recognition is an automated technique that allows machines to recognize a person's identity based on his/her

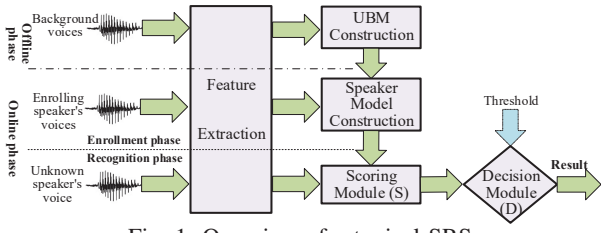


Fig. 1: Overview of a typical SRS

utterances using the characteristics of the speaker. It has been studied actively for four decades [18], and currently supported by a number of open-source platforms (e.g., Kaldi and MSR Identity [36]) and commercial solutions (e.g., Microsoft Azure, Amazon Alexa [37], Google home [38], Talentedsoft, and SpeechPro VoiceKey [39]). In addition, NIST actively organizes the Speaker Recognition Evaluation [40] since 1996.

Overview of SRSs. Fig. 1 shows an overview of a typical SRS, which includes five key modules: Feature Extraction, Universal Background Model (UBM) Construction, Speaker Model Construction, Scoring Module and Decision Module. The top part is an offline phase, while the lower two parts are an online phase composed of speaker enrollment and recognition phases.

In the offline phase, a UBM is trained using the acoustic feature vectors extracted from the background voices (i.e., voice training dataset) by the feature extraction module. The UBM, intending to create a model of the average features of everyone in the dataset, is widely used in the state-of-the-art SRSs to enhance the robustness and improve efficiency [1]. In the speaker enrollment phase, a speaker model is built using the UBM and feature vectors of enrolling speaker's voices for each speaker. During the speaker recognition phase, given an input voice x, the scores $S(x)$ of all the enrolled speakers are computed using the speaker models, which will be emitted along with the decision $D(x)$ as the recognition result.

The feature extraction module converts a raw speech signal into acoustic feature vectors carrying characteristics of the signal. Various acoustic feature extraction algorithms have been proposed such as Mel-Frequency Cepstral Coefficients (MFCC) [41], Spectral Subband Centroid (SSC) [42] and Perceptual Linear Predictive (PLP) [43]. Among them, MFCC is the most popular one in practice [1], [18].

Speaker recognition tasks. There are three common recognition tasks of SRSs: open-set identification (OSI) [17], close-set identification (CSI) [15] and speaker verification (SV) [16].

An OSI system allows multiple speakers to be enrolled during the enrollment phase, forming a speaker group G. For an arbitrary input voice x, the system determines whether x is uttered by one of the enrolled speakers or none of them, according to the scores of all the enrolled speakers and a preset (score) threshold θ. Formally, suppose the speaker group G has n speakers $\{1, 2, \cdots, n\}$, the decision module outputs $D(x)$:

$$D(x) = \begin{cases} \underset{i \in G}{\operatorname{argmax}} \, [S(x)]_i, & \text{if } \max\limits_{i \in G} [S(x)]_i \geq \theta; \\ \texttt{reject}, & \text{otherwise.} \end{cases}$$

where $[S(x)]_i$ for $i \in G$ denotes the score of the voice x that is uttered by the speaker i. Intuitively, the system classifies the input voice x as the speaker i if and only if the score $[S(x)]_i$ of the speaker i is the largest one among all the enrolled speakers, and not less than the threshold θ. If the largest score is less than θ, the system directly rejects the voice, namely, it is not uttered by any of the enrolled speakers.

CSI and SV systems accomplish similar tasks as the OSI system, but with some special settings. A CSI system never rejects any input voices, i.e., an input will always be classified as one of the enrolled speakers. Whereas an SV system can have exactly *one* enrolled speaker and checks if an input voice is uttered by the enrolled speaker, i.e., either *accept* or *reject*.

Text-Dependency. SRSs can be either text-dependent, where cooperative speakers are required to utter one of pre-defined sentences, or text-independent, where the speakers are allowed to speak anything. The former achieves high accuracy on short utterances, but always requires a large amount utterances repeating the same sentence, thus it is *only* used in the SV task. The latter may require longer utterances to achieve high accuracy, but practically it is more versatile and can be used in all tasks (cf. [18]). Therefore, in this work, we mainly demonstrate our attack on text-independent SRSs.

SRS implementations. ivector-PLDA [25], [44] is a mainstream method for implementing SRSs in both academia [27], [45], [46] and industries [47], [48]. It achieves the state-of-the-art performance for all the speaker recognition tasks [49], [50]. Another one is GMM-UBM based methods, which train a Gaussian mixture model (GMM) [16], [51] as UBM. Basically, GMM-UBM tends to provide comparative (or higher) accuracy on short utterances [52].

Recently, deep neural network (DNN) becomes used in speech [53] and speaker recognition (e.g., xvector-PLDA [26]), where speech recognition aims at determining the underlying text or command of the speech signal. However, the major breakthroughs made by DNN-based methods reside in speech recognition; for speaker recognition, ivector based methods still exhibit the state-of-the-art performance [5]. Moreover, DNN-based methods usually rely on a much larger amount of training data, which could greatly increase the computational complexity compared with ivector and GMM based methods [54], thus are not suitable for off-line enrollment on client-side devices. We denote by ivector, GMM, and xvector the ivector-PLDA, GMM-UBM, and xvector-PLDA, respectively.

B. Threat Model

We assume that the adversary intends to craft an adversarial sample from a voice uttered by some source speaker, so that it is classified as one of the enrolled speakers (untargeted attack) or the target speaker (targeted attack) by the SRS under attack, but is still recognized as the source speaker by ordinary users.

To deliberately attack the authentication of a target victim, we can compose adversarial voices, which mimic the voiceprint of the victim from the perspective of the SRSs. Reasonably, the adversary can unlock the smartphones [55],

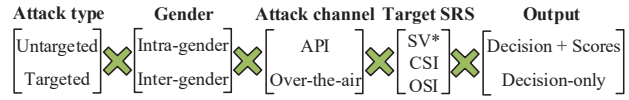


Fig. 2: Attack scenarios, where $*$ means that targeted and untargeted are the same on the SV task, as an SV system only has one enrolled speaker.

log into applications [56], and conduct illegal financial transactions [2]. Under untargeted attack, we can manipulate voices to mimic the voiceprint of any one of enrolled speakers. For example, we can bypass the voice-based access control such as iFLYTEK [57], where multiple speakers are enrolled. After bypassing the authentication, follow-up hidden voice command attacks (e.g., [10], [34]) can be launched, e.g., on smart car with Dragon Drive [33]. These attack scenarios are practically feasible, for example, when the victim is not within the hearable distance of the adversarial voice, or the attack voice does not raise the alertness of the victim due to the presence of other voice sources, either human or loudspeakers.

This paper focuses on the practical black-box setting where the adversary has access *only* to the recognition result (decision result and scores) of a target SRS for each test input, but not the internal configurations or training/enrollment voices. This black-box setting is feasible in practice, e.g., the commercial systems Talentedsoft [28], iFLYTEK, SinoVoice [58] and SpeakIn [59]. If the scores are not accessible (e.g., OSI task in the commercial system Microsoft Azure), we can leverage transferability attacks. We assume the adversary has some voices of the target speakers to build a surrogate model, while these voices are not necessary the enrollment voices. This is also feasible in practice as one can possibly record speeches of target speakers. To our knowledge, the targeted black-box setting renders all previous adversarial attacks impractical on SRSs. Indeed, all the adversarial attacks on SRSs are white-box [13], [14] except for the concurrent work [60], which performs only untargeted attacks.

Specifically, in our attack model, we consider five parameters: attack type (targeted vs. untargeted attack), genders of speakers (inter-gender vs. intra-gender), attack channel (API vs. over-the-air), speaker recognition task (OSI vs. CSI vs. SV) and output of the target SRS (decision and scores vs. decision-only) as shown in Fig. 2. Intra-gender (resp. inter-gender) means that the genders of the source and target speakers are the same (resp. different). API attack assumes that the target SRS (e.g., Talentedsoft) provides an API interface to query, while over-the-air means that attacks should be played over the air in the physical world. Decision-only attack means that the target SRS (e.g., Microsoft Azure) only outputs decision result (i.e., the adversary can obtain the decision result $D(x)$), but not the scores of the enrolled speakers. Therefore, targeted, inter-gender, over-the-air and decision-only attacks are the most practical yet the most challenging ones. In summary, by counting all the possible combinations of the parameters in Fig. 2, there are $48 = 2 \times 2 \times 2 \times 3 \times 2$ attack scenarios. Since targeted and untargeted attacks are the same on the SV task, there are $40 = 48 - 2 \times 2 \times 2$ attack scenarios. However, demonstrating all the 40 attack scenarios requires huge engineering efforts, we design our experiments to cover 16 representative attack scenarios (cf. Appendix B).

III. Methodology

In this section, we start with the motivations, then explain the design philosophy of our attack in black-box setting and the possible defenses, finally present an overview of our attack.

A. Motivation

The research in this work is motivated by the following questions: (Q1) How to launch an adversarial attack against all the tasks of SRSs in the practical black-box setting? (Q2) Is it feasible to craft robust adversarial voices that are transferable to an unknown SRS under cross-architecture, cross-dataset and cross-parameter circumstances, and commercial systems, even when played over the air in the physical world? (Q3) Is it possible to craft human-imperceptible adversarial voices that are difficult, or even impossible, to be noticed by ordinary users? (Q4) If such an attack exists, can it be defended?

B. Design Philosophy

To address Q1, we investigate existing methods for black-box attacks on image/speech recognition systems, i.e., surrogate model [61], gradient estimation [62], [21] and genetic algorithm [63], [64]. Surrogate model methods are proved to be outperformed by gradient estimation methods [62], hence are excluded. For the other two methods: it is known that natural evolution strategy (NES) based gradient estimation [21] requires much fewer queries than finite difference gradient estimation [62], and particle swarm optimization (PSO) is proved to be more computationally efficient than other genetic algorithms [63], [65]. To this end, we conduct a comparison experiment on an OSI system using NES as a black-box gradient estimation technique and PSO as a genetic algorithm. The result shows that the NES-based gradient estimation method obviously outperforms the PSO-based one (cf. Appendix A). Therefore, we exploit the NES-based gradient estimation.

However, even with the estimated gradients, none of the existing gradient based white-box methods (e.g., [22], [23], [66], [67], [10], [20], [19], [24]) can be directly used to attack SRSs. This is due to the threshold θ which is used in the OSI and SV tasks, but not in image/speech recognition. As a result, these methods will fail to mislead SRSs when the resulted score is less than θ. To solve this challenge, we incorporate the threshold θ into our adversarial sample generation and propose a novel algorithm to estimate θ in the black-box setting.

Theoretically, the adversarial samples crafted in the above way are effective if directly fed as input to the target SRS via exposed API. However, to launch a practical attack as in Q2, adversarial samples should be played over the air in the physical world to interact with a SRS that may differ from the SRS on which adversarial samples are crafted. To address Q2, we increase the strength of adversarial samples and the range of noise amplitude, instead of using noise model [10], [19],

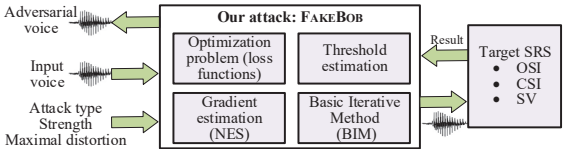


Fig. 3: Overview of our attack: FAKEBOB

[20], due to its device- and background-dependency. We have demonstrated that our approach is effective in transferability attack even when playing over the air in the physical world.

To address Q3, we should consider two aspects of the human-imperceptibility. First, the adversarial samples should sound natural when listened by ordinary users. Second, and more importantly, they should sound as uttered by the same speaker of the original one. As a first step towards addressing Q3, we add a constraint onto the perturbations using L_∞ norm, which restricts the maximal distortion at each sample point of the audio signal. We also conduct a real human study to illustrate the imperceptibility of our adversarial samples.

To address Q4, we should launch attacks on SRSs with defense methods. However, to our knowledge, no defense solution exists for adversarial attacks on SRSs. Therefore, we use four defense solutions for adversarial attacks on speech recognition systems: audio squeezing [10], [31], local smoothing [31], quantization [31] and temporal dependency detection [31], to defend against our attack.

C. Overview of Our Attack: FAKEBOB

According to our design philosophy, in this section, we present an overview (shown in Fig. 3) of our attack, named FAKEBOB, addressing two technical challenges (C1) and (C2) mentioned in §I. To address C1, we formulate adversarial sample generation as an optimization problem (cf. §IV-A), for which specific loss functions are defined for different attack types (i.e., targeted and untargeted) and tasks (i.e., OSI, CSI and SV) of SRSs (cf. §IV-B, §IV-C and §IV-D). To solve the optimization problem, we propose an approach by leveraging a novel algorithm to estimate the threshold, NES to estimate gradient and the BIM method with the estimated gradients. C2 is addressed by incorporating the maximal distortion (L_∞ norm) of noise amplitude and strength of adversarial samples into the optimization problem (cf. §IV-A, §IV-B, §IV-C and §IV-D).

IV. OUR ATTACK: FAKEBOB

In this section, we elaborate on the techniques behind FAKEBOB, including the problem formulation and attacks on OSI, CSI, and SV systems.

A. Problem Formulation

Given an original voice, x, uttered by some source speaker, the adversary aims at crafting an adversarial voice $\acute{x} = x+\delta$ by finding a perturbation δ such that (1) $\acute{x}$ is a valid voice [68], (2) δ is as human-imperceptible as possible, and (3) the SRS under attack classifies the voice $\acute{x}$ as one of the enrolled speakers or the target speaker. To guarantee that the adversarial voice $\acute{x}$ is a valid voice, which relies upon the audio file format (e.g., WAV, MP3 and AAC), our attack FAKEBOB first normalizes the amplitude value $x(i)$ of a voice x at each sample point i into the range $[-1, 1]$, then crafts the perturbation δ to make sure $-1 \leq \acute{x}(i) = x(i) + \delta(i) \leq 1$, and finally transforms $\acute{x}$ back to the audio file format which will be fed to the target SRS. Hereafter, we assume that the range of amplitude values is $[-1, 1]$. To be as human-imperceptible as possible, our attack FAKEBOB adapts L_∞ norm to measure the similarity between the original and adversarial voices and ensures that the L_∞ distance $\|\acute{x}, x\|_\infty := \max_i\{|\acute{x}(i) - x(i)|\}$ is less than the given maximal amplitude threshold ϵ of the perturbation, where i denotes sample point of the audio waveform. To successfully fool the target SRS, we formalize the problem of finding an adversarial voice $\acute{x}$ for a voice x as the following constrained minimization problem:

$$\begin{aligned} &\text{argmin}_\delta\, f(x+\delta) \\ &\text{such that } \|x+\delta, x\|_\infty < \epsilon \text{ and } x+\delta \in [-1,1]^n \end{aligned} \tag{1}$$

where f is a loss function. When f is minimized, $x + \delta$ is recognized as the target speaker (targeted attack) or one of enrolled speakers (untargeted attack). Our formulation is designed to be fast for minimizing the loss function rather than minimizing the perturbation δ, as done in [22], [23]. Some studies, e.g., [24], [7], formulate the problem to minimize both the loss function and perturbation. It remains to define the loss function and algorithm to solve the optimization problem. In the rest of this section, we mainly address them on the OSI system, then adapt the solution to the CSI and SV systems.

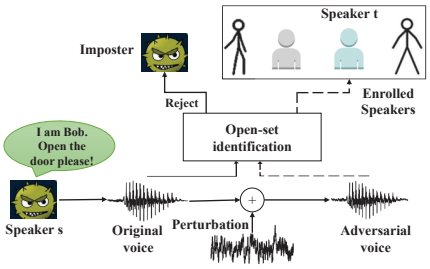


Fig. 4: Attack on OSI systems

B. Attack on OSI Systems

As shown in Fig. 4, to attack an OSI system, we want to craft an adversarial voice $\acute{x}$ starting from a voice x uttered by some source speaker (i.e., $D(x) = \texttt{reject}$) such that the voice $\acute{x}$ is classified as the target speaker $t \in G = \{1, \cdots, n\}$ by the SRS, i.e., $D(\acute{x}) = t$. We first present the loss function f and then show how to solve the minimization problem.

Loss function f. To launch a successful targeted attack on an OSI system, the following two conditions need to be satisfied *simultaneously*: the score $[S(x)]_t$ of the target speaker t should be (1) the maximal one among all the enrolled speakers, and (2) not less than the preset threshold θ. Therefore, the loss function f for the target speaker t is defined as follows:

$$f(x) = \max\left\{(\max\{\theta, \max_{i \in G\setminus\{t\}}[S(x)]_i\} - [S(x)]_t), -\kappa\right\} \tag{2}$$

where the parameter κ, inspired by [24], intends to control the strength of adversarial voices: the larger the κ is, the more confidently the adversarial voice is recognized as the target speaker t by the SRS. This has been validated in §V-C.

Our loss function is similar to the one defined in [24], but we also incorporate an additional threshold θ. Considering $\kappa = 0$, when $(\max\{\theta, \max_{i \in G \setminus \{t\}}[S(x)]_i\} - [S(x)]_t)$ is minimized, the score $[S(x)]_t$ of the target speaker t will be maximized until it exceeds the threshold θ and the scores of all other enrolled speakers. Hence, the system recognizes the voice x as the speaker t. When $\kappa > 0$, instead of looking for a voice that just barely changes the recognition result of x to the speaker t, we want that the score $[S(x)]_t$ of the speaker t is much larger than any other enrolled speakers and the threshold θ.

To launch an untargeted attack, the loss function f can be revised as follows:

$$f(x) = \max\{(\theta - \max_{i \in G}[S(x)]_i), -\kappa\}. \quad (3)$$

Intuitively, we want to find a perturbation δ such that the largest score of x is at least κ greater than the threshold θ.

Solving the optimization problem. To solve the optimization problem in Eq. (1), we use NES as a gradient estimation technique and employ the BIM method with the estimated gradients to craft adversarial examples. Specifically, the BIM method begins by setting $\acute{x}_0 = x$ and then on the i^{th} iteration,

$$\acute{x}_i = \texttt{clip}_{x,\epsilon}\{\acute{x}_{i-1} - \eta \cdot \texttt{sign}(\nabla_x f(\acute{x}_{i-1}))\}$$

where η is a hyper-parameter indicating the learning rate, and the function $\texttt{clip}_{x,\epsilon}(\acute{x})$, inspired by [23], performs per-sample clipping of the voice $\acute{x}$, so the result will be in L_∞ ϵ-neighbourhood of the source voice x and will be a valid voice after being transformed back into the audio file format. Formally, $\texttt{clip}_{x,\epsilon}(\acute{x}) = \max\{\min\{\acute{x}, 1, x + \epsilon\}, -1, x - \epsilon\}$.

We compute the gradient $\nabla_x f(\acute{x}_{i-1})$ by leveraging NES, which only depends on the recognition result. In detail, on the i^{th} iteration, we first create m (must be even) Gaussian noises $(u_1, ..., u_m)$ and add them onto $\acute{x}_{i-1}$, leading to m new voices $\acute{x}^1_{i-1}, ..., \acute{x}^m_{i-1}$, where $\acute{x}^j_{i-1} = \acute{x}_{i-1} + \sigma \times u_j$ and σ is the search variance of NES. Note that $u_j = -u_{m+1-j}$ for $j = 1, ..., \frac{m}{2}$. Then, we compute the loss values $f(\acute{x}^1_{i-1}), ..., f(\acute{x}^m_{i-1})$ by querying the target system (m queries). Next, the gradient $\nabla_x f(\acute{x}_{i-1})$ is approximated by computing

$$\frac{1}{m \times \sigma} \sum_{j=1}^{m} f(\acute{x}^j_{i-1}) \times u_j.$$

In our experiments, $m = 50$ and $\sigma = 1e-3$. Finally, we compute $\texttt{sign}(\nabla_x f(\acute{x}_{i-1}))$, a vector over the domain $\{-1, 0, 1\}$, by applying element-wise $\texttt{sign}$ mathematical operation to the gradient vector $\frac{1}{m \times \sigma} \sum_{j=1}^{m} f(\acute{x}^j_{i-1}) \times u_j$.

However, the BIM method with the estimated gradients alone is not sufficient to construct adversarial samples in the black-box setting, due to the fact that the adversary has no access to the threshold θ used in the loss function f. To solve this problem, we present a novel algorithm for estimating θ.

Estimating the threshold θ. To estimate the threshold θ, the main technical challenge is that the estimated threshold $\acute{\theta}$ should be no less than θ in order to launch a successful attack, but should not exceed θ too much, otherwise, the attack cost might become too expensive. Therefore, the goal is to compute a small $\acute{\theta}$ such that $\acute{\theta} \geq \theta$. To achieve this goal, we propose a novel approach as shown in Algorithm 1. Given an OSI system with the scoring S and decision D modules, and an arbitrary voice x such that $D(x) = \texttt{reject}$, i.e., x is uttered by an imposter, Algorithm 1 outputs $\acute{\theta}$ such that $\acute{\theta} \geq \theta$.

Algorithm 1 Threshold Estimation Algorithm

Input: The target OSI system with scoring S and decision D modules
An arbitrary voice x such that $D(x) = \texttt{reject}$
Output: Estimated threshold $\acute{\theta}$

1: $\acute{\theta} \leftarrow \max_{i \in G}[S(x)]_i$; ▷ *initial threshold*
2: $\Delta \leftarrow |\frac{\acute{\theta}}{10}|$; ▷ *the search step*
3: $\acute{x} \leftarrow x$;
4: **while** True **do**
5: $\quad \acute{\theta} \leftarrow \acute{\theta} + \Delta$;
6: $\quad f' \leftarrow \lambda x. \max\{\acute{\theta} - \max_{i \in G}[S(x)]_i, -\kappa\}$; ▷ *loss function*
7: $\quad$ **while** True **do**
8: $\quad\quad \acute{x} \leftarrow \texttt{clip}_{x,\epsilon}\{\acute{x} - \eta \cdot \texttt{sign}(\nabla_x f'(\acute{x}))\}$; ▷ *craft sample using* f'
9: $\quad\quad$ **if** $D(\acute{x}) \neq \texttt{reject}$ **then**; ▷ $\max_{i \in G}[S(\acute{x})]_i \geq \theta$
10: $\quad\quad\quad$ **return** $\max_{i \in G}[S(\acute{x})]_i$;
11: $\quad\quad$ **if** $\max_{i \in G}[S(\acute{x})]_i \geq \acute{\theta}$ **then break**;

In detail, Algorithm 1 first computes the maximal score $\acute{\theta} = \max_{i \in G}[S(x)]_i$ of the voice x by querying the system (line 1). Since $D(x) = \texttt{reject}$, we can know $\acute{\theta} < \theta$. At Line 2, we initialize the search step $\Delta = |\frac{\acute{\theta}}{10}|$, which will be used to estimate the desired threshold $\acute{\theta}$. $|\frac{\acute{\theta}}{10}|$ is chosen as a tradeoff between the precision of $\acute{\theta}$ and efficiency of the algorithm. The outer-while loop (Lines 4-11) iteratively computes a new candidate $\acute{\theta}$ by adding Δ onto it (Line 5) and computes the function $f' = \lambda x. \max\{\acute{\theta} - \max_{i \in G}[S(x)]_i, -\kappa\}$ (Line 6). f' indeed is the loss function for untargeted attack in Eq. (3), in which θ is replaced by the candidate $\acute{\theta}$. The function f' will be used to craft samples in the inner-while loop (Lines 7-11). For each candidate $\acute{\theta}$, the inner-while loop (Lines 7-11) iteratively computes samples $\acute{x}$ by querying the target system until the target system recognizes $\acute{x}$ as some enrolled speaker (Line 9) or the maximal score of $\acute{x}$ is no less than $\acute{\theta}$ (Line 11). If $\acute{x}$ is recognized as some enrolled speaker (Line 9), then Algorithm 1 terminates and returns the maximal score of $\acute{x}$ (Line 10), as $\max_{i \in G}[S(\acute{x})]_i \geq \theta$ is the desired threshold. If the maximal score of $\acute{x}$ is no less than $\acute{\theta}$ (Line 11), we restart the outer-while loop.

One may notice that Algorithm 1 will not terminate when $D(\acute{x})$ is always equal to $\texttt{reject}$. In our experiments, this never happens (cf. §V). Furthermore, it estimates a very close value to the actual threshold. Remark that the actual threshold θ, obtained from the open-source SRS, is used to evaluate the performance of Algorithm 1 *only*.

C. Attack on CSI Systems

A CSI system always classifies an input voice as one of the enrolled speakers. Therefore, we can adapt the attack on the OSI systems by ignoring the threshold θ. Specifically, the loss function for targeted attack on CSI systems with the target speaker $t \in G$ is defined as:

TABLE I: Dataset for experiments

Datasets	#Speaker	Details
Train-1 Set	7,273	Part of **VoxCeleb1** [69] and whole **VoxCeleb2** [70] used for training ivector and GMM
Train-2 Set	2,411	Part of **LibriSpeech** [71] used for training system C in transferability
Test Speaker Set	5	5 speakers from **LibriSpeech** 3 female and 2 male, 5 voices per speaker, voices range from 3 to 4 seconds
Imposter Speaker Set	4	Another 4 speakers from **LibriSpeech** 2 female and 2 male, 5 voices per speaker, voices range from 2 to 14 seconds

$$f(x) = \max\left\{(\max_{i \in G \setminus \{t\}} [S(x)]_i - [S(x)]_t), -\kappa\right\}$$

Intuitively, we want to find some small perturbation δ such that the score of the speaker t is the largest one among all the enrolled speakers, and $[S(x)]_t$ is at least κ greater than the second-largest score.

Similarly, the loss function for untargeted attack on CSI systems is defined as:

$$f(x) = \max\{([S(x)]_m - \max_{i \in G \setminus \{m\}} [S(x)]_i), -\kappa\}$$

where m denotes the true speaker of the original voice. Intuitively, we want to find some small perturbation δ such that the largest score among other enrolled speakers is at least κ greater than the score of the speaker m.

D. Attack on SV Systems

An SV system has exactly one enrolled speaker and checks if an input voice is uttered by the enrolled speaker or not. Thus, we can adapt the attack on OSI systems by assuming the speaker group G is a singleton set. Specifically, the loss function for attacking SV systems is defined as:

$$f(x) = \max\{\theta - S(x), -\kappa\}$$

Intuitively, we want to find a small perturbation δ such that the score of x being recognized as the enrolled speaker is at least κ greater than the threshold θ. We remark that the threshold estimation algorithm for SV systems should be revised by replacing the loss function f' at Line 6 in Algorithm 1 with the following function: $f' = \lambda x. \max\{\acute{\theta} - S(x), -\kappa\}$.

V. Attack Evaluation

We evaluate FakeBob for its attacking capabilities based on the following five aspects: effectiveness/efficiency, transferability, practicability, imperceptibility, and robustness.

A. Dataset and Experiment Design

Dataset. We mainly use three widely used datasets: VoxCeleb1, VoxCeleb2, and LibriSpeech (cf. Table I). To demonstrate our attack, we target the ivector and GMM systems from the popular open-source platform Kaldi, having 7,631 stars and 3,418 forks on Github [27]. The UBM model is trained using the *Train-1 Set* as the background voices. The OSI and CSI are enrolled by 5 speakers from the *Test Speaker Set*, forming a speaker group. The SV is enrolled by 5 speakers from the *Test Speaker Set*, resulting in 5 ivector and 5 GMM systems.

TABLE II: Metrics used in this work

Metric	Description
Attack success rate (ASR)	Proportion of adversarial voices that are recognized as the target speaker
Untargeted success rate (UTR) for CSI	Proportion of adversarial samples that are not recognized as the source speaker
Untargeted success rate (UTR) for OSI	Proportion of adversarial samples that are not rejected by the target system

We conducted the experiments on a server with Ubuntu 16.04 and Intel Xeon CPU E5-2697 v2 2.70GHz with 377G RAM (10 cores). We set $\kappa = 0$, max iteration=1,000, max/min learning rate η is 1e-3/1e-6, search variance σ in NES is 1e-3, and samples per draw m in NES is 50, unless explicitly stated.

Evaluation metrics. To evaluate our attack, we use the metrics shown in Table II. Signal-noise ratio (SNR) is widely used to quantify the level of signal power to noise power, so we use it to measure the distortion of the adversarial voices [10]. We use the equation, SNR(dB)$= 10\log_{10}(P_x/P_\delta)$, to obtain SNR, where P_x is the signal power of the original voice x and P_δ is the power of the perturbation δ. Larger SNR value indicates a (relatively) smaller perturbation. To evaluate efficiency, we use two metrics: *number of iterations* and *time*. (Note that the number of queries is the number of iterations multiplied by samples per draw m in NES and $m = 50$ in this work.)

Experiment design. We design five experiments. (1) We evaluate the *effectiveness* and *efficiency* on both open-source systems (i.e., ivector, GMM, and xvector) and the commercial system Talentedsoft. We also evaluate FakeBob under intra-gender and inter-gender scenarios, as inter-gender attacks are usually more difficult. (2) We evaluate the *transferability* by attacking the open-source systems with different architecture, training dataset, and parameters, as well as the commercial system Microsoft Azure. (3) We further evaluate the *practicability* by playing the adversarial voices over the air in the physical world. (4) For *human-imperceptibility*, we conduct a real human study through Amazon Mechanical Turk platform (MTurk) [30], a crowdsourcing marketplace for human intelligence. (5) We finally evaluate *defense methods*, local smoothing, quantization, audio squeezing, temporal dependency-based detection, to defend against FakeBob.

Recall that we demonstrate our attack on 16 representative attack scenarios out of 40 (cf. §II-B). In particular, we mainly consider targeted attack which is much more powerful and challenging than untargeted attack [9]. Our experiments suffice to understand the other four parameters of the attack model, i.e., inter-gender vs. intra-gender, API vs. over-the-air, OSI vs. CSI vs. SV, decision and scores vs. decision-only.

The OSI task can be seen as a combination of the CSI and SV tasks (cf. §II). Thus, we sometimes only report and analyze the results on the OSI task due to space limitation, which is much more challenging and representative than the other two. The missing results can be found in Appendix.

B. Effectiveness and Efficiency

Target model training. To evaluate the effectiveness and efficiency, we train ivector and GMM systems for the OSI,

TABLE III: Six trained SRSs

Task	Metrics	ivector	GMM
CSI	Accuracy	99.6%	99.3%
SV	FRR	1.0%	5.0%
	FAR	11.0%	10.4%
OSI	FRR	1.0%	4.2%
	FAR	7.9%	11.2%
	OSIER	0.2%	2.8%

TABLE IV: Results of threshold estimation

ivector			GMM		
θ	$\acute{\theta}$	Time (s)	θ	$\acute{\theta}$	Time (s)
1.45	**1.47**	**628**	**0.091**	**0.0936**	**157**
1.57	1.60	671	0.094	0.0957	260
1.62	1.64	686	0.106	0.1072	269
1.73	1.75	750	0.113	0.1141	289
1.84	1.87	804	0.119	0.1193	314

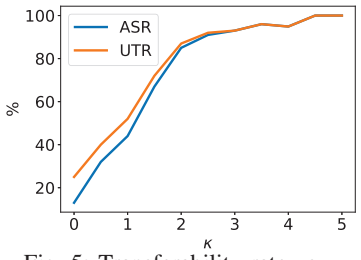


Fig. 5: Transferability rate vs. κ

CSI and SV tasks. The performance of these systems is shown in Table III, where accuracy is as usual, False Acceptance Rate (FAR) is the proportion of voices that are uttered by imposters but accepted by the system [18], False Rejection Rate (FRR) is the proportion of voices that are uttered by an enrolled speaker but rejected by the system [18], Open-set Identification Error Rate (OSIER) is the rate of voices that cannot be correctly classified [17]. Notice that the threshold θ is 1.45 for ivector and 0.091 for GMM, so that the FAR is close to 10%. Although the parameter θ in SV and OSI tasks can be tuned using Equal Error Rate, i.e., FAR is equal to FRR, we found that the results for SV and OSI tasks do not vary too much (cf. Table XVII in Appendix).

Setting. The parameter ϵ is one of the most critical parameters of our attack. To fine-tune ϵ, we study ASR, efficiency and distortion by varying ϵ from 0.05, 0.01, 0.005, 0.004, 0.003, 0.002, to 0.001, on ivector and GMM for the CSI task. The results are given in Appendix C. With decreasing of ϵ, both the attack cost and SNR increase, while ASR decreases. As a trade-off between ASR, efficiency, and distortion, we set $\epsilon = 0.002$ in this experiment.

The target speakers are the speakers from the Test Speaker Set (cf. Table I), the source speakers are the speakers, from the Test Speaker Set for CSI, and from the Imposter Speaker Set (cf. Table I) for SV and OSI. Ideally, we will craft 100 adversarial samples using FAKEBOB for each task, where 40 adversarial samples are intra-gender and 60 inter-gender for CSI, and 50 intra-gender and 50 inter-gender for SV and OSI. Note that to diversify experiments, the source speakers of CSI and SV/OSI are designated to be different.

Results. The results are shown in Table V. Since the OSI task is more challenging and representative than the other two, we only analyze the results of the OSI task here. We can observe that FAKEBOB achieves 99.0% ASR for both ivector and GMM. In terms of SNR, the average SNR value is 31.5 (dB) for ivector and 31.4 (dB) for GMM, indicating that the perturbation is less than 0.071% and 0.072%. Furthermore, the average numbers of iterations and execution time are 86 and 38.0 minutes on ivector. The average numbers of iterations and execution time are 38 and 3.8 minutes on GMM, much smaller than that of ivector. Due to space limitation, results of attacking xvector are given in Appendix D where we observe similar results. These results demonstrate the effectiveness and efficiency of FAKEBOB.

We can also observe that inter-gender attack is much more difficult (more iterations and execution time) than intra-gender attack due to the difference between sounds of male and female. Moreover, ASR of inter-gender attack is also lower than that of intra-gender attack. The result unveils that once the gender of the target speaker is known by attackers, it is much easier to launch an intra-gender attack.

For evaluation of the threshold estimation algorithm, we report the estimated threshold $\acute{\theta}$ in Table IV by setting 5 different thresholds. The estimation error is less than 0.03 for ivector and less than 0.003 for GMM. This shows that our algorithm is able to effectively estimate the threshold in less than 13.4 minutes. Note that our attack is black-box, and the actual thresholds are accessed *only* for evaluation.

Attacking the commercial system Talentedsoft [28]. We also evaluate the effectiveness and efficiency of FAKEBOB on Talentedsoft, developed by the constitutor of the voiceprint recognition industry standard of the Ministry of Public Security (China). We query this online platform via the HTTP post (seen as the exposed API). Since Talentedsoft targets Chinese Mandarin, to fairly test Talentedsoft, we use the Chinese Mandarin voice database aishell-1 [72]. Both FAR and FRR of Talentedsoft are 0.15%, tested using 20 speakers and 7,176 voices in total which are randomly chosen from aishell-1.

We enroll 5 randomly chosen speakers from aishell-1 as targeted speakers, resulting in 5 SV systems. Each of them is attacked using another 20 randomly chosen speakers and one randomly chosen voice per speaker. Our attack achieves 100% ASR within 50 iterations (i.e., 2,500 queries) on average. Remark that FAKEBOB is an iterative-based method. We can always set some time slot between iterations or queries so that such amount of queries do not cause heavy traffic burden to the server, hence our attack is feasible. This demonstrates the effectiveness and efficiency of FAKEBOB on commercial systems that are completely black-box.

C. Transferability

Transferability [7] is the property that some adversarial samples produced to mislead a model (called source system) can mislead other models (called target system) even if their architectures, training datasets, or parameters differ.

Setting. To evaluate the transferability, we regard the previously built GMM (A) and ivector (B) as source systems and build another 8 target systems (denoted by C,...,J respectively). C,...,I are ivector systems differing in key parameter and training dataset, and J is the xvector system. For details and performance of these systems, refer to Tables XIV and XV

TABLE V: Experimental results of FAKEBOB when $\epsilon = 0.002$, where #Iter refers to #Iteration.

Task	System								System (Intra-gender attack)								System (Inter-gender attack)							
	ivector				GMM				ivector				GMM				ivector				GMM			
	#Iter	Time (s)	SNR (dB)	ASR (%)	#Iter	Time (s)	SNR (dB)	ASR (%)	#Iter	Time (s)	SNR (dB)	ASR (%)	#Iter	Time (s)	SNR (dB)	ASR (%)	#Iter	Time (s)	SNR (dB)	ASR (%)	#Iter	Time (s)	SNR (dB)	ASR (%)
CSI	124	2845	30.2	99.0	40	218	29.3	99.0	92	2115	29.3	100.0	25	126	28.8	100.0	146	3340	30.8	98.0	50	278	29.62	98.0
SV	84	2014	31.6	99.0	39	241	31.4	99.0	31	751	31.7	98.0	30	185	31.7	100.0	135	3252	31.6	100.0	48	298	31.2	98.0
OSI	86	2277	31.5	99.0	38	226	31.4	99.0	32	833	31.3	98.0	31	178	31.5	100.0	140	3692	31.6	100.0	45	274	31.2	98.0

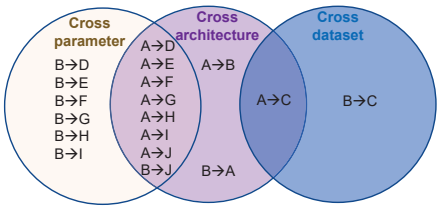


Fig. 6: Distribution of transferability attacks

in Appendix. We denote by $X \to Y$ the transferability attack where X is the source system and Y is the target system. The distribution of the transferability attacks is shown in Fig. 6 in terms of architecture, training dataset, and key parameters. We can see that some attacks belong to multiple scenarios. We set $\epsilon = 0.05$ and (1) $\kappa = 0.2$ (GMM) and $\kappa = 10$ (ivector) for the CSI task, (2) $\kappa = 3$ (GMM) and $\kappa = 4$ (ivector) for the SV task, (3) $\kappa = 3$ (GMM) and $\kappa = 5$ (ivector) for the OSI task. Remark that κ differs from architectures and tasks due to their different scoring mechanisms. We fine-tuned the parameter κ for ASR under the max iteration bound 1,000.

Results. The results of attacking OSI systems are shown in Table VI. All the attacks (except for $B \to A$) achieve 34%-68% ASR and 40%-100% UTR. For $B \to D$, $B \to E$, $B \to F$, $B \to G$, and $B \to H$ (all are ivector, but differ in one key parameter), FAKEBOB achieves 100% ASR and UTR, indicating that cross architecture reduces transferability rate. From $A \to B$ and $A \to C$ (where A is GMM, B and C are ivector but differ in training data), cross dataset also reduces transferability rate. The transferability rate of $B \to A$ is the lowest one and less than that of $A \to B$, indicating that transferring from the architecture ivector (B) to GMM (A) is more difficult. Compared with $A \to C$ (both cross dataset and architecture), $B \to C$ (cross dataset) achieves nearly 20% more ASR and UTR. This reveals that the larger the difference between the source and target systems is, the more difficult the transferability attack is. Due to space limitation, the results of attacking the CSI and SV systems are shown in Tables XVI and XVIII in Appendix. We can observe similar results. The average SNR is similar to the one given in Table VII.

To understand how the value of κ influences the transferability rate, we conduct $B \to F$ attack (OSI task) by fixing $\epsilon = 0.05$ and varying κ from 0.5 to 5.0 with step 0.5. In this experiment, the number of iterations is unlimited. The results are shown in Fig. 5. Both ASR and UTR increase quickly with κ, and reach 100% when $\kappa = 4.5$. This demonstrates that increasing the value of κ increases the probability of a successful transferability attack.

Attacking the commercial system Microsoft Azure [29]. Microsoft Azure is a cloud service platform with the second largest market share in the world. It supports both the SV and OSI tasks via HTTP REST API. Unlike Talentedsoft, Azure's API only returns the decision (i.e., the predicted speaker) along with 3 confidence levels (i.e., low, normal and high) instead of scores, so we attack this platform via transferability. We enroll 5 speakers from the Test Speaker Set to build an OSI system on Azure (called OSI-Azure for simplicity). Its FAR is 0% tested by the Imposter Speaker Set. For each target speaker, we randomly select 10 source speakers and 2 voices per source speaker from LibriSpeech, which are rejected by OSI-Azure. We set $\epsilon = 0.05$ and craft 100 adversarial voices on the GMM system, as it produces high tranferability rate in the above experiment. The ASR, UTR and SNR are 26.0%, 41.0% and 6.8 dB, respectively. They become 34.0%, 57.0% and 2.2 dB when we increase ϵ from 0.05 to 0.1.

We also demonstrate FAKEBOB on the SV task of Azure (SV-Azure) which is text-dependent with 10 supported texts. We recruited and asked 2 speakers to read each text 10 times, resulting in 200 voices. For each pair of speaker and text, we randomly select 3 enrollment voices for both GMM and SV-Azure, and the FARs of them are 0%. We attack SV-Azure using 200 adversarial samples crafted from GMM ($\epsilon = 0.05$, $\kappa = 3$). However, SV-Azure reports "error, too noisy" instead of "accept" or "reject" for 190 adversarial voices. Among the other 10 voices, one voice is accepted, leading to 10% ASR. To our knowledge, this is the first time that SV-Azure is successfully attacked. As Azure is proprietary without any publicly available information, it is very difficult to know the reason why SV-Azure outputs "error, too noisy". After comparing the SNR of the 190 voices with the other 10 voices (8.8 dB vs. 11.5 dB), we suspect that it checks each input and outputs "error, too noisy" without model classification if the noise of the input is too large. This check makes SV-Azure more challenging to attack, but we infer it may also reject normal voices when the background is noisy in practice.

D. Practicability for Over-the-Air Attack

To simulate over-the-air attack in the physical world, we first craft adversarial samples by directly interacting with API of the system (i.e., over the line), then play and record these adversarial voices via loudspeakers and microphones, and finally send recorded voices to the system via API to check their effectiveness. Our experiments are conducted in an indoor room (length, width, and height are 10, 4, 3.5 meters).

TABLE VI: Transferability rate (%) for OSI task, where S and T denote source and target systems respectively.

S \ T	A		B		C		D		E		F		G		H		I		J	
	ASR	UTR	ASR	UTR	ASR	UTR	ASR	UTR	ATR	UTR	ASR	UTR	ASR	UTR	ASR	UTR	ASR	UTR	ASR	UTR
A	—	—	62.0	64.0	48.0	48.0	55.2	56.9	68.0	68.0	64.0	64.0	52.0	54.0	68.0	68.0	38.0	40.0	34.0	42.0
B	5.0	5.0	—	—	67.5	67.5	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	72.5	75.0	40.0	41.7

TABLE VII: Results of different systems

System		SNR (dB)	Result (%)	
			Normal voices	Adversarial voices
ivector	CSI	6.6	Accuracy: 100	ASR: 80, UTR: 80
	SV	9.8	FAR: 0, FRR: 0	ASR: 76
	OSI	7.8	FAR: 4, FRR: 0, OSIER: 0	ASR: 100, UTR: 100
GMM	CSI	6.1	Accuracy: 85	ASR: 90, UTR: 100
	SV	7.9	FAR: 0, FRR: 62	ASR: 100
	OSI	8.2	FAR: 0, FRR: 65, OSIER: 0	ASR: 100, UTR: 100
Azure	OSI	6.8	FAR: 5, FRR: 2, OSIER: 0	ASR: 70, UTR: 70

To thoroughly evaluate FAKEBOB, the over-the-air attacks vary in systems, devices (loudspeakers and microphones), distance between loudspeakers and microphones, and acoustic environments. In total, it covers 26 scenarios. The overview of different settings is shown in Table XIX in Appendix. We consider all tasks of ivector and GMM, and the OSI-Azure only. We use the same parameters as in Section V-C, as over-the-air attack is more practical yet more challenging due to the noise introduced from both air channel and electronic devices which probably disrupts the perturbations of adversarial samples. For OSI-Azure, we use the adversarial voices crafted on GMM in Section V-C that are successfully transferred to OSI-Azure.

Results of different systems. We use portable speaker (JBL clip3 [73]) as the loudspeaker, iPhone 6 Plus (iOS) as the microphone with 1 meter distance between them. We attack all tasks of ivector and GMM, and the OSI-Azure in a relatively quiet environment. The results are shown in Table VII. We can observe that the FRR of GMM SV (resp. OSI) is 62% (resp. 65%), revealing that GMM is less robust than ivector for normal voices. FAKEBOB achieves (1) for the CSI task, 90% ASR (i.e., the system classifies the adversarial voice as the target speaker) and 100% UTR (i.e., the system does not classify the adversarial voice as the source speaker) on the GMM, and achieves 80% ASR and 80% UTR on the ivector; (2) for the SV task, at least 76% ASR; (3) for the OSI task, 100% ASR on both the GMM and ivector; (4) achieves 70% ASR on the commercial system OSI-Azure.

In terms of SNR, the average SNR is no less than 6.1 dB, and the average SNR is up to 9.8 dB on the ivector for the SV task, indicating that the power of the signal is 9.5 times greater than that of the noise. Moreover, the SNR is much better than the over-the-air attack in CommanderSong [10].

Results of different devices. For loudspeakers, we use 3 common devices: laptop (DELL), portable speaker (JBL clip3) and broadcast equipment (Shinco [74]). For microphones, we use built-in microphones of 2 mobile phones: OPPO (Android) and iPhone 6 Plus (iOS). We evaluate FAKEBOB against the OSI task of ivector with 1 meter distance in a relatively quiet environment. The results are shown in Table VIII.

TABLE VIII: Results of different devices (%), where L and M denote loudspeakers and microphones respectively.

L \ M	iPhone 6 Plus (iOS)					OPPO (Android)				
	Normal voices			Adv. voices		Normal voices			Adv. voices	
	FAR	FRR	OSIER	ASR	UTR	FAR	FRR	OSIER	ASR	UTR
DELL	10	0	0	100	100	13	6	0	78	80
JBL clip3	4	0	0	100	100	6	0	0	80	80
Shinco	8	5	0	89	91	14	0	0	75	75

We can observe that for any pair of loudspeaker and microphone, FAKEBOB can achieve at least 75% ASR and UTR. When JBL clip3 or DELL is the loudspeaker and iPhone 6 Plus is the microphones, FAKEBOB is able to achieve 100% ASR. When the loudspeaker is fixed, the ASR and UTR of attacks using IPhone 6 Plus are higher (at least 14% and 16% more) than that of using OPPO. Possible reason is that the sound quality of iPhone 6 Plus is better than that of OPPO phone. These results demonstrate the effectiveness of FAKEBOB on various devices.

Results of different distances. To understand the impact of the distance between loudspeakers and microphones, we vary distance from 0.25, 0.5, 1, 2, 4 to 8 meters. We attack the OSI task of ivector in a relatively quiet environment by using JBL clip3 as the loudspeaker and iPhone 6 Plus as the microphone.

The results are shown in Table IX. We can observe that FAKEBOB can achieve 100% ASR and UTR when the distance is no more than 1 meter. When the distance is increased to 2 meters (resp. 4 meters), ASR and UTR drop to 70% (resp. 40% and 50%). Although ASR and UTR drop to 10% when the distance is 8 meters, FRR also increases to 32%. This shows the effectiveness of FAKEBOB under different distances.

Results of different acoustic environments. We attack the OSI task of ivector using JBL clip3 and iPhone 6 Plus with 1 meter distance. To simulate different acoustic environments, we play different types of noises in the background using Shinco broadcast equipment. Specifically, we select 5 types of noises from Google AudioSet [75]: white noise, bus noise, restaurant noise, music noise, and absolute music noise. White noise is widespread in nature, while bus, restaurant, (absolute) music noises are representative of several daily life scenarios where FAKEBOB may be launched. For white noise, we vary its volume from 45 dB to 75 dB, while the volumes of other noises are 60 dB. Both adversarial and normal voices are played at 65 dB on average. The results are shown in Table X.

We can observe that FAKEBOB achieves at least 48% ASR and UTR when the volume of background noises is no more than 60 dB no matter the type of the noises. Although both ASR and UTR decrease with increasing the volume of white noises, the FRR also increases quickly. This demonstrates the effectiveness of FAKEBOB in different acoustic environments.

TABLE IX: Results of different distances (%)

Distance (meter)		0.25	0.5	1	2	4	8
Normal Voices	FAR	4	3	4	6	0	0
	FRR	0	0	0	5	10	32
	OSIER	0	0	0	0	0	0
Adversarial Voices	ASR	100	100	100	70	40	10
	UTR	100	100	100	70	50	10

TABLE X: Results of different acoustic environments (%)

Environment		Quiet	White (45 dB)	White (50 dB)	White (60 dB)	White (65 dB)	White (75 dB)	Bus (60 dB)	Rest. (60 dB)	Music (60 dB)	Abs. Music (60 dB)
Normal voices	FAR	4	0	6	0	0	10	0	0	0	4
	FRR	0	5	12	30	**40**	**97**	25	20	10	10
	OSIER	0	0	0	0	0	0	0	0	10	0
Adv. voices	ASR	100	75	70	57	20	2	50	50	66	48
	UTR	100	75	70	60	20	2	50	50	67	48

E. Human-Imperceptibility via Human Study

To demonstrate the imperceptibility of adversarial samples, we conduct a human study on MTurk [30]. The survey is approved by the Institutional Review Board (IRB) of our institutes.

Setup of human study. We recruit participants from MTurk and ask them to choose one of the two tasks and finish the corresponding questionnaire. We neither reveal the purpose of our study to the participants, nor record personal information of participants such as first language, age and region. The Amazon MTurk has designed Acceptable Use Policy for permitted and prohibited uses of MTurk, which prohibits bots or scripts or other automated answering tools to complete Human Intelligence Tasks [76]. Thus, we argue that the number of participants can reasonably guarantee the diversity of participants. The two tasks are described as follows.

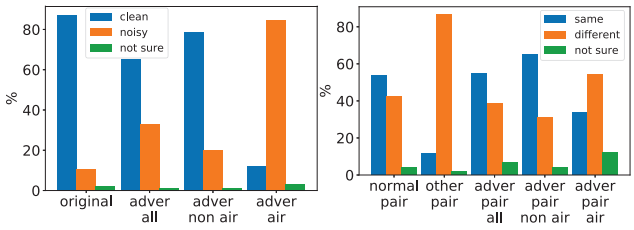
- *Task 1: Clean or Noisy.* This task asks participants to tell whether the playing voice is clean or noisy. Specifically, we randomly select 12 original voices and 15 adversarial voices crafted from other original voices, among which 12 adversarial voices are randomly selected from the voices which become non-adversarial (called ineffective) when playing over the air with $\epsilon = 0.002$ and low confidence, and the other 3 are randomly selected from the voices which remain adversarial (called effective) when playing over the air with $\epsilon = 0.1$ and high confidence. We ask users to choose whether a voice has any background noise (The three options are *clean*, *noisy*, and *not sure*).
- *Task 2: Identify the Speaker.* This task asks participants to tell whether the voices in a pair are uttered by the same speaker. Specifically, we randomly select 3 speakers (2 male and 1 female), and randomly choose 1 normal voice per speaker (called reference voice). Then for each speaker, we randomly select 3 normal voices, 3 distinct adversarial voices that are crafted from other normal voices of the same speaker, and 3 normal voices from other speakers. In summary, we build 27 pairs of voices: 9 pairs are *normal pairs* (one reference voice and one normal voice from the same speaker), 9 pairs are *other pairs* (one reference voice and one normal voice from another speaker) and 9 pairs are *adversarial pairs* (one reference voice and one adversarial voice from the same speaker). Among 9 adversarial pairs, 6 pairs contain effective adversarial samples when playing over the air, and 3 pairs do not. We ask the participants to tell whether the voices in each pair are uttered by the same speaker (The three options are *same*, *different*, and *not sure*).

To ensure the quality of our questionnaire and validity of our results, we filter out the questionnaires that are randomly chosen by participants. In particular, we set three simple questions in each task. For task 1, we insert three silent voices as a concentration test. For task 2, we insert three pairs of voices, where each pair contains one male voice and one female voice as a concentration test. Only when all of them are correctly answered, we regard it as a valid questionnaire, otherwise, we exclude it.

Results of human study. We finally received 135 questionnaires for task 1 and 172 questionnaires for task 2, where 27 and 11 questionnaires are filtered out as they failed to pass our concentration tests. Therefore, there are 108 valid questionnaires for task 1 and 161 valid questionnaires for task 2. The results of the human study are shown in Fig. 7.

For task 1, as shown in Fig. 7(a), 10.7% of participants heard noise on normal voices, while 20.2% and 84.8% of participants heard noise on ineffective and effective adversarial voices (when played over-the-air) respectively. We can see that 78.8% of participants still believe that ineffective voices are clean. For effective voices, we found that 84.8% is comparable to the recent white-box adversarial attack (i.e., 83%) that tailors to craft imperceptible voices against speech recognition systems [20]. (We are not aware of any other adversarial attacks against SRSs that have done such human study.)

For task 2 which is more interesting (in Fig. 7(b)), 86.5% of participants believe that voices in each *other pair* are uttered by different speakers, indicating the quality of collected questionnaires. For the *adversarial pairs*, 54.6% of participants believe that voices in each pair are uttered by the same speaker, very close to the baseline 53.7% of *normal pairs*, indicating that humans cannot differentiate the speakers of the normal and adversarial voices. The prior work [14] conducted an ABX testing on adversarial samples crafted by white-box attacks against SV systems. The ABX test first provides to users two voices A and B, each being either the original (reconstructed) voice or an adversarial voice; then provides the third voice X which was randomly chosen from $\{A, B\}$; finally asks the users to decide if X is A or B. The ABX testing of [14] shows that 54% of participants correctly classified the adversarial voices, which is very close to ours. For the *adversarial pairs* which contain ineffective adversarial voices, 64.9% of participants believed that the two voices are from the same speakers, much greater than the baseline 53.7%, thus more imperceptible. For the *adversarial pairs* which contain effective adversarial voices, 54.0% of participants can definitely differentiate the speaker, not too larger than the baseline 42.2% of *normal pairs*.



(a) Task 1: clean or noisy (b) Task 2: identify the speaker

Fig. 7: Results of human study, where air (resp. non air) denotes voices that are effective (resp. ineffective) for over-the-air attack

The results unveil that the adversarial voices crafted by FAKEBOB can make systems misbehave (i.e., making a decision that the adversarial voice is uttered by the target speaker), while most of ineffective adversarial samples are classified clean and cannot be differentiated by ordinary users, and the results of effective ones are comparable to existing related works. Hence, our attack is reasonably human-imperceptible.

F. Robustness of FAKEBOB *against Defense Methods*

As mentioned in Section III-B, we study four defense methods: local smoothing, quantization, audio squeezing and temporal dependency detection. We evaluate on the OSI task of the GMM system unless explicitly stated using 100 seed voices. The FRR, FAR, ASR and UTR of the system without defense is 4.2%, 11.2%, 99% and 99%, respectively. We consider two settings: (S1) crafting adversarial voices on the system without defense and attacking the system with defense, and (S2) directly attacking the system with defense. S1 follows from CommanderSong [10]. An effective defense method should be able to mitigate the perturbation or detect the adversarial voices in S1. Thus, we will use the UTR metric. In S2, an effective defense method should increase the overhead of the attack and decrease the attack success rate, thus we will use the ASR metric. We set $\epsilon = 0.002$, a very weak attacker capacity. Increasing ϵ will make FAKEBOB more powerful.

We found that the local smoothing can increase attack cost, but is ineffective in terms of ASR, audio squeezing is ineffective in terms of both attack cost and ASR, while the other two are not suitable for defending our attack. Due to space limitation, details are given in Appendix E.

VI. DISCUSSION OF THE POSSIBLE ARM RACE

This section discusses the potential mitigation of our attacks and possible advanced attacks.

Mitigation of FAKEBOB. We have demonstrated that four defense methods have limited effects on FAKEBOB although some of them are reported promising in the speech recognition domain. This reveals that more effective defense methods are needed to mitigate FAKEBOB. We discuss several possible defense methods as follows.

Various liveness detection methods have been proposed to detect spoofing attacks on SRSs. Such methods detect attacks by exploiting the different physical characteristics of the voices generated by the human speech production system (i.e., lungs, vocal cords, and vocal tract) and electronic loudspeaker. For instance, Shiota et al. [77] use pop noise caused by human breath, VoiceLive [78] leverages time-difference-of-arrival of voices to the receiver, and VoiceGesture [79] leverages the unique articulatory gesture of the user. Adversarial voices also need to be played via loudspeakers, hence liveness detection could be possibly used to detect them. An alternative detection method is to train a detector using adversarial voices and normal voices. Though promising in image recognition domain [80], it has a very high false-positive rate and does not improve the robustness when the adversary is aware of this defense [81]. Another scheme to mitigate adversarial images is input transformation such as image bit-depth reduction and JPEG compression [82]. We could mitigate adversarial voices by leveraging input transformations such as bit-depth reduction and MP3 compression. However, Athalye et al. [83] have demonstrated that input transformation on images can be easily circumvented by strong attacks such as Backward Pass Differentiable Approximation. We conjecture that bit-depth reduction and MP3 compression may become ineffective for high-confidence adversarial voices.

Finally, one could also improve the security of SRSs by using a text-dependent system and requiring users to read dynamically and randomly generated sentences. By doing so, the adversary has to attack both the speaker recognition and the speech recognition, hence incurring attack costs. If the set of phrases to be uttered is relatively small, we could also attack the system by iteratively querying the target system using the voice corresponding to the generated phrase. While our attack will fail when the set of phrases to be uttered is very large or even infinite. However, this also brings the challenge for the recognition system, as the training data may not be able to cover all the possible normal phrases and voices.

We will study the above methods [77], [78], [79], [82], [83], [84], [85], [86] for adversarial attacks in future. We next discuss possible methods on improving adversarial attacks.

Possible advanced attacks. For a system that outputs the decision result and scores, FAKEBOB can directly craft adversarial voices via interacting with it. However, for a system that only outputs the decision result, we have to attack it by leveraging transferability. When the gap between source and target systems is larger, the transferability rate is limited. One possible solution to improve FAKEBOB is to leverage the boundary attack, which is proposed to attack decision-only image recognition systems by Brendel et al. [87].

Our human study shows that our attack is reasonably human-imperceptible. However, many of effective adversarial voices are still noisier than original voices (human study task 1), and some of effective adversarial voices can be differentiated from different speakers by ordinary users (human study task 2), there still has space for improving imperceptibility in future. One possible solution is to build a psychoacoustic model and limit the maximal difference between the spectrum of the original and adversarial voices to the masking threshold

(hearing threshold) of human perception [88], [20].

VII. Related Work

The security issues of intelligent voice systems have been studied in the literature. In this section, we discuss the most related work on attacks over the intelligent voice systems, and compare them with FakeBob.

Adversarial voice attacks. Gong et al. [13] and Kreuk et al. [14] respectively proposed adversarial voice attacks on SRSs in the white-box setting, by leveraging the Fast Gradient Sign Method (FGSM) [22]. The attack in [13] addresses DNN-based gender recognition, emotion recognition and CSI systems, while the attack in [14] addresses a DNN-based SV system. Compared to them: (1) Our attack FakeBob is black-box and more practical. (2) FakeBob addresses not only the SV and CSI, but also the more general OSI task. (3) We demonstrate our attack on ivector, GMM and DNN-based systems in the popular open-source platform Kaldi. (4) FakeBob is effective on the commercial systems, even when playing over the air, which was not considered in [13], [14].

In a concurrent work, Abdullah et al. [60] proposed a poisoning attack on speaker and speech recognition systems, that is demonstrated on the OSI-Azure. There are three key differences: (1) Their attack crafts an adversarial voice from a voice uttered by an *enrolled* speaker A such that the adversarial voice is neither rejected nor recognized as the speaker A. Thus, their attack neither can choose a specific source speaker nor a specific target speaker to be recognized by the system, consequently, they cannot launch targeted attack or attacks against the SV task. Whereas our attack goes beyond their attack. (2) They craft adversarial voice by decomposing and reconstructing an input voice, hence, achieved a limited untargeted success rate and cannot be adapted to launch more interesting and powerful targeted attacks. (3) We evaluate over-the-air attacks in the physical world, but they did not.

We cannot compare the performance (i.e., effectiveness and efficiency) of our attack with the three related works above [13], [14], [60] because all of them are not available. We are the first considering the threshold θ in adversarial attack. Adversarial attacks on speech recognition systems also have been studied [11], [9], [89]. Carlini et al. [9] attacked DeepSpeech [90] by crafting adversarial voices in the white-box setting, but failed to attack when playing over the air. In the black-box setting, Rohan et al. [11] combined a genetic algorithm with finite difference gradient estimation to craft adversarial voices for DeepSpeech, but achieved a limited success rate with strict length restriction over the voices. Alzantot et al. [89] presented the first black-box adversarial attack on a CNN-based speech command classification model by exploiting a genetic algorithm. However, due to the difference between speaker recognition and speech recognition, these works are orthogonal to our work and cannot be applied to ivector and GMM based SRSs.

Other types of voice attacks. Other types of voice attacks include hidden voice attack (both against speech and speaker recognition) and spoofing attack (against speaker recognition).

Hidden voice attack aims to embed some information (e.g., command) into an audio carrier (e.g., music) such that the desired information is recognized by the target system without catching victims' attention. Abdullah et al. [91] proposed such an attack on speaker and speech recognition systems. There are two key differences: (1) Based on characteristics of signal processing and psychoacoustics, their attack perturbed a sample uttered by an *enrolled speaker* such that it is still correctly classified as the *enrolled speaker* by the target system but becomes incomprehensible to human listening. While our attack perturbed a sample uttered by an *arbitrary speaker* such that it is misclassified as a target speaker (targeted attack) or another enrolled speaker (untargeted attack) but the perturbation is imperceptible to human listening. This means their attack addresses a different attack scenario compared with ours. (2) They did not demonstrate over-the-air attack on SRSs and their tool is not available, hence it is unclear how effective it is on SRSs. DolphinAttack [92], CommanderSong [10] and the work done by Carlini et al. [34] proposed hidden voice attacks on SRSs. Carlini et al. launched both black-box (i.e., inverse MFCC) and white-box (i.e., gradient decent) attacks on GMM based speech recognition systems. DolphinAttack exploited vulnerabilities of microphones and employed the ultrasound as the carrier of commands to craft inaudible voices. However, it can be easily defended by filtering out the ultrasound from voices. CommanderSong launched white-box attacks by exploiting a gradient descent method to embed commands into music songs.

Another attack type on SRSs is spoofing attack [93] such as mimic [94], replay [95], [96], recorder attack [97], [96], voice synthesis [98], and voice conversion [99], [100], [101], [96] attacks. Different from adversarial attack [14], [102], spoofing attack aims at obtaining a voice such that it is correctly classified as the target speaker by the system, and also sound like the *target speaker* listened by ordinary users. When anyone familiar with the victim (including the victim) cannot hear the attack voice, both spoofing and adversarial attacks can be launched. However, if someone familiar with the victim (including the victim) can hear the attack voice, he/she may detect the spoofing attack. Whereas, adversarial attack could be launched in this setting as discussed in Section II-B.

VIII. Conclusion

In this paper, we conducted the first comprehensive and systematic study of adversarial attack on SRSs in a practical black-box setting, by proposing a novel practical adversarial attack FakeBob. FakeBob was thoroughly evaluated in 16 attack scenarios. FakeBob can achieve 99% targeted attack success rate on both open-source and the commercial systems. We also demonstrated the transferability of FakeBob on Microsoft Azure. When played over the air in the physical world, FakeBob is also effective. Our findings reveal the security implications of FakeBob for SRSs, calling for more robust defense methods to better secure SRSs against such practical adversarial attacks.

ACKNOWLEDGMENTS

This research was partially supported by National Natural Science Foundation of China (NSFC) grants (No. 61532019 and No. 61761136011), National Research Foundation (NRF) Singapore, Prime Ministers Office under its National Cybersecurity R&D Program (Award No. NRF2014NCR-NCR001-30 and No. NRF2018NCR-NCR005-0001), National Research Foundation (NRF) Singapore, National Satellite of Excellence in Trustworthy Software Systems under its Cybersecurity R&D Program (Award No. NRF2018NCR-NSOE003-0001), and National Research Foundation Investigatorship Singapore (Award No. NRF-NRFI06-2020-0001).

REFERENCES

[1] T. Kinnunen and H. Li, "An overview of text-independent speaker recognition: From features to supervectors," *Speech Commun.*, 2010.
[2] TD Bank voiceprint. https://www.tdbank.com/bank/tdvoiceprint.html.
[3] S. Nand, "Forensic and automatic speaker recognition system," *IJCEE*, 2018.
[4] H. Ren, Y. Song, S. Yang, and F. Situ, "Secure smart home: A voiceprint and internet based authentication system for remote accessing," in *ICCSE*, 2016.
[5] D. Ribas and E. Vincent, "An improved uncertainty propagation method for robust i-vector based speaker recognition," in *ICASSP*, 2019.
[6] B. Biggio, I. Corona, D. Maiorca, B. Nelson, N. Srndic, P. Laskov, G. Giacinto, and F. Roli, "Evasion attacks against machine learning at test time," in *ECML/PKDD*, 2013.
[7] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. J. Goodfellow, and R. Fergus, "Intriguing properties of neural networks," in *ICLR*, 2014.
[8] Y. Lei, S. Chen, L. Fan, F. Song, and Y. Liu, "Advanced evasion attacks and mitigations on practical ml-based phishing website classifiers," *arXiv preprint arXiv:2004.06954*, 2020.
[9] N. Carlini and D. Wagner, "Audio adversarial examples: Targeted attacks on speech-to-text," in *IEEE S&P Workshops*, 2018.
[10] X. Yuan, Y. Chen, Y. Zhao, Y. Long, X. Liu, K. Chen, S. Zhang, H. Huang, X. Wang, and C. A. Gunter, "Commandersong: A systematic approach for practical adversarial voice recognition," in *USENIX Security*, 2018.
[11] R. Taori, A. Kamsetty, B. Chu, and N. Vemuri, "Targeted adversarial examples for black box audio systems," in *IEEE S&P Workshops*, 2019.
[12] S. Khare, R. Aralikatte, and S. Mani, "Adversarial black-box attacks for automatic speech recognition systems using multi-objective genetic optimization," *CoRR*, vol. abs/1811.01312, 2018.
[13] Y. Gong and C. Poellabauer, "Crafting adversarial examples for speech paralinguistics applications," in *DYNAMICS*, 2018.
[14] F. Kreuk, Y. Adi, M. Cisse, and J. Keshet, "Fooling end-to-end speaker verification with adversarial examples," in *ICASSP*, 2018.
[15] T. Liu and S. Guan, "Factor analysis method for text-independent speaker identification," *JSW*, 2014.
[16] D. A. Reynolds, T. F. Quatieri, and R. B. Dunn, "Speaker verification using adapted gaussian mixture models," *Digit. Signal Process.*, 2000.
[17] J. Fortuna, P. Sivakumaran, A. Ariyaeeinia, and A. Malegaonkar, "Open-set speaker identification using adapted gaussian mixture models," in *INTERSPEECH*, 2005.
[18] H. Beigi, *Fundamentals of Speaker Recognition*. Springer, 12 2011.
[19] H. Yakura and J. Sakuma, "Robust audio adversarial example for a physical attack," in *IJCAI*, 2019.
[20] Y. Qin, N. Carlini, G. W. Cottrell, I. J. Goodfellow, and C. Raffel, "Imperceptible, robust, and targeted adversarial examples for automatic speech recognition," in *ICML*, 2019.
[21] A. Ilyas, L. Engstrom, A. Athalye, and J. Lin, "Black-box adversarial attacks with limited queries and information," in *ICML*, 2018.
[22] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," in *ICLR*, 2015.
[23] A. Kurakin, I. J. Goodfellow, and S. Bengio, "Adversarial examples in the physical world," in *ICLR*, 2017.
[24] N. Carlini and D. Wagner, "Towards evaluating the robustness of neural networks," in *IEEE S&P*, 2017.
[25] N. Dehak, P. J. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet, "Front-end factor analysis for speaker verification," *IEEE Trans. on Audio, Speech, and Language Processing*, 2010.
[26] D. Snyder, D. Garcia-Romero, G. Sell, A. McCree, D. Povey, and S. Khudanpur, "Speaker recognition for multi-speaker conversations using x-vectors," in *IEEE ICASSP*, 2019.
[27] Kaldi. https://github.com/kaldi-asr/kaldi.
[28] Talentedsoft. http://www.talentedsoft.com.
[29] Microsoft Azure, https://azure.microsoft.com.
[30] Amazon Mechanical Turk Platform. https://www.mturk.com.
[31] Z. Yang, B. Li, P. Chen, and D. Song, "Characterizing audio adversarial examples using temporal dependency," in *ICLR*, 2019.
[32] 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/#7b01dea1109c.
[33] The voice-enabled car of the future. https://tractica.omdia.com/user-interface-technologies/the-voice-enabled-car-of-the-future.
[34] N. Carlini, P. Mishra, T. Vaidya, Y. Zhang, M. Sherr, C. Shields, D. Wagner, and W. Zhou, "Hidden voice commands," in *USENIX Security*, 2016.
[35] Fakebob. https://sites.google.com/view/fakebob.
[36] MSR Identity. https://www.microsoft.com/en-us/download/details.aspx?id=52279.
[37] Amazon Alexa. https://developer.amazon.com/en-US/alexa.
[38] Google Home. https://store.google.com/product/google˙home.
[39] Speechpro. https://speechpro-usa.com.
[40] NIST. National institute of standards and technology speaker recognition evaluation. https://www.nist.gov/itl/iad/mig/speaker-recognition.
[41] L. Muda, M. Begam, and I. Elamvazuthi, "Voice recognition algorithms using mel frequency cepstral coefficient (MFCC) and dynamic time warping (dtw) techniques," *Journal of Computing*, 2010.
[42] N. P. H. Thian, C. Sanderson, and S. Bengio, "Spectral subband centroids as complementary features for speaker authentication," in *ICB*, 2004.
[43] H. Hermansky, "Perceptual linear predictive (PLP) analysis of speech," *The Journal of the Acoustical Society of America*, vol. 87, no. 4, 1990.
[44] M. K. Nandwana, L. Ferrer, M. McLaren, D. Castan, and A. Lawson, "Analysis of critical metadata factors for the calibration of speaker recognition systems," in *INTERSPEECH*, 2019.
[45] P. S. Nidadavolu, V. Iglesias, J. Villalba, and N. Dehak, "Investigation on neural bandwidth extension of telephone speech for improved speaker recognition," in *ICASSP*, 2019.
[46] K. A. Lee, Q. Wang, and T. Koshinaka, "The CORAL+ algorithm for unsupervised domain adaptation of PLDA," in *ICASSP*, 2019.
[47] Tencent VPR. https://cloud.tencent.com/product/vpr.
[48] Fosafer VPR. http://caijing.chinadaily.com.cn/chanye/2018-06/06/content˙36337667.htm.
[49] G. Heigold, I. Moreno, S. Bengio, and N. Shazeer, "End-to-end text-dependent speaker verification," in *ICASSP*, 2016, pp. 5115–5119.
[50] S. Sremath Tirumala and S. R. Shahamiri, "A review on deep learning approaches in speaker identification," in *ICSPS*, 2016.
[51] D. A. Reynolds and R. C. Rose, "Robust text-independent speaker identification using gaussian mixture speaker models," *IEEE Trans. Speech and Audio Processing*, vol. 3, no. 1, pp. 72–83, 1995.
[52] V. Vestman, D. Gowda, M. Sahidullah, P. Alku, and T. Kinnunen, "Speaker recognition from whispered speech: A tutorial survey and an application of time-varying linear prediction," *Speech Commun.*, 2018.
[53] D. Amodei, S. Ananthanarayanan, R. Anubhai, J. Bai, E. Battenberg, C. Case, J. Casper, B. Catanzaro, Q. Cheng, G. Chen *et al.*, "Deep speech 2: End-to-end speech recognition in english and mandarin," in *International conference on machine learning*, 2016, pp. 173–182.
[54] C. Li, X. Ma, B. Jiang, X. Li, X. Zhang, X. Liu, Y. Cao, A. Kannan, and Z. Zhu, "Deep speaker: an end-to-end neural speaker embedding system," *CoRR*, vol. abs/1705.02304, 2017.
[55] "Android app which enables unlock of mobile phone via voice print," http://app.mi.com/details?id=com.jie.lockscreen.
[56] "Social software wechat adds voiceprint lock login function," https://kf.qq.com/touch/wxappfaq/1208117b2mai141125YZjAra.html.
[57] VPR of iFLYTEK. https://www.xfyun.cn/services/isv.
[58] Sinovoice voice print recognition. http://doc.aicloud.com/sdk5.2.8.
[59] Speakin vpr. http://www.speakin.mobi/devPlatform.html.

[60] H. Abdullah, M. S. Rahman, W. Garcia, L. Blue, K. Warren, A. S. Yadav, T. Shrimpton, and P. Traynor, "Hear "no evil", see "kenansville": Efficient and transferable black-box attacks on speech recognition and voice identification systems," *CoRR*, vol. abs/1910.05262, 2019.
[61] N. Papernot, P. McDaniel, I. Goodfellow, S. Jha, Z. B. Celik, and A. Swami, "Practical black-box attacks against machine learning," in *AsiaCCS*, 2017, pp. 506–519.
[62] P.-Y. Chen, H. Zhang, Y. Sharma, J. Yi, and C.-J. Hsieh, "Zoo: Zeroth order optimization based black-box attacks to deep neural networks without training substitute models," in *AISec*, 2017, pp. 15–26.
[63] M. Sharif, S. Bhagavatula, L. Bauer, and M. K. Reiter, "Accessorize to a crime: Real and stealthy attacks on state-of-the-art face recognition," in *ACM CCS*, 2016, pp. 1528–1540.
[64] M. Alzantot, Y. Sharma, S. Chakraborty, H. Zhang, C. Hsieh, and M. B. Srivastava, "Genattack: practical black-box attacks with gradient-free optimization," in *GECCO*, 2019, pp. 1111–1119.
[65] L. M. Rios and N. V. Sahinidis, "Derivative-free optimization: a review of algorithms and comparison of software implementations," *Journal of Global Optimization*, vol. 56, no. 3, pp. 1247–1293, 2013.
[66] Y. Dong, F. Liao, T. Pang, H. Su, J. Zhu, X. Hu, and J. Li, "Boosting adversarial attacks with momentum," in *CVPR*, 2018, pp. 9185–9193.
[67] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, "Towards deep learning models resistant to adversarial attacks," in *ICLR*, 2018.
[68] Y. Duan, Z. Zhao, L. Bu, and F. Song, "Things you may not know about adversarial example: A black-box adversarial image attack," *CoRR*, vol. abs/1905.07672, 2019.
[69] A. Nagrani, J. S. Chung, and A. Zisserman, "Voxceleb: a large-scale speaker identification dataset," in *INTERSPEECH*, 2017.
[70] J. S. Chung, A. Nagrani, and A. Zisserman, "Voxceleb2: Deep speaker recognition," in *INTERSPEECH*, 2018.
[71] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: An asr corpus based on public domain audio books," in *ICASSP*, 2015.
[72] H. Bu, J. Du, X. Na, B. Wu, and H. Zheng, "Aishell-1: An open-source mandarin speech corpus and a speech recognition baseline," in *2017 20th Conference of the Oriental Chapter of the International Coordinating Committee on Speech Databases and Speech I/O Systems and Assessment (O-COCOSDA)*, Nov 2017, pp. 1–5.
[73] JBL clip3 portable speaker. https://www.jbl.com/bluetooth-speakers/JBL+CLIP+3.html.
[74] Shinco broadcast equipment. https://item.jd.com/5009202.html.
[75] J. F. Gemmeke, D. P. W. Ellis, D. Freedman, A. Jansen, W. Lawrence, R. C. Moore, M. Plakal, and M. Ritter, "Audio set: An ontology and human-labeled dataset for audio events," in *ICASSP*, 2017.
[76] Amazon mechanical turk acceptable use policy. https://www.mturk.com/acceptable-use-policy.
[77] S. Shiota, F. Villavicencio, J. Yamagishi, N. Ono, I. Echizen, and T. Matsui, "Voice liveness detection algorithms based on pop noise caused by human breath for automatic speaker verification," in *INTERSPEECH*, 2015.
[78] L. Zhang, S. Tan, J. Yang, and Y. Chen, "VoiceLive: A phoneme localization based liveness detection for voice authentication on smartphones," in *ACM CCS*, 2016.
[79] L. Zhang, S. Tan, and J. Yang, "Hearing your voice is not enough: An articulatory gesture based liveness detection for voice authentication," in *ACM CCS*, 2017.
[80] Z. Gong, W. Wang, and W.-S. Ku, "Adversarial and clean data are not twins," *arXiv preprint arXiv:1704.04960*, 2017.
[81] N. Carlini and D. Wagner, "Adversarial examples are not easily detected: Bypassing ten detection methods," in *AISec*, 2017.
[82] C. Guo, M. Rana, M. Cisse, and L. Van Der Maaten, "Countering adversarial images using input transformations," *arXiv preprint arXiv:1711.00117*, 2017.
[83] A. Athalye, N. Carlini, and D. Wagner, "Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples," *arXiv preprint arXiv:1802.00420*, 2018.
[84] X. Du, X. Xie, Y. Li, L. Ma, Y. Liu, and J. Zhao, "Deepstellar: Model-based quantitative analysis of stateful deep learning systems," in *ESEC/FSE*, 2019.
[85] X. Zhang, X. Xie, L. Ma, X. Du, Q. Hu, Y. Liu, J. Zhao, and M. Sun, "Towards characterizing adversarial defects of deep learning software from the lens of uncertainty," in *ICSE*, 2020.
[86] Y. Liu, L. Ma, and J. Zhao, "Secure deep learning engineering: A road towards quality assurance of intelligent systems," in *ICFEM*, 2019.
[87] W. Brendel, J. Rauber, and M. Bethge, "Decision-based adversarial attacks: Reliable attacks against black-box machine learning models," *arXiv preprint arXiv:1712.04248*, 2017.
[88] L. Schönherr, K. Kohls, S. Zeiler, T. Holz, and D. Kolossa, "Adversarial attacks against automatic speech recognition systems via psychoacoustic hiding," in *NDSS*, 2019.
[89] M. Alzantot, B. Balaji, and M. B. Srivastava, "Did you hear that? adversarial examples against automatic speech recognition," *CoRR*, vol. abs/1801.00554, 2018.
[90] A. Y. Hannun, C. Case, J. Casper, B. Catanzaro, G. Diamos, E. Elsen, R. Prenger, S. Satheesh, S. Sengupta, A. Coates, and A. Y. Ng, "Deep speech: Scaling up end-to-end speech recognition," *CoRR*, vol. abs/1412.5567, 2014.
[91] H. Abdullah, W. Garcia, C. Peeters, P. Traynor, K. R. B. Butler, and J. Wilson, "Practical hidden voice attacks against speech and speaker recognition systems," in *NDSS*, 2019.
[92] G. Zhang, C. Yan, X. Ji, T. Zhang, T. Zhang, and W. Xu, "Dolphinattack: Inaudible voice commands," in *ACM CCS*, 2017.
[93] Z. Wu, N. Evans, T. Kinnunen, J. Yamagishi, F. Alegre, and H. Li, "Spoofing and countermeasures for speaker verification: A survey," *Speech Commun.*, 2015.
[94] R. G. Hautamäki, T. Kinnunen, V. Hautamäki, T. Leino, and A.-M. Laukkanen, "I-vectors meet imitators: on vulnerability of speaker verification systems against voice mimicry." in *INTERSPEECH*, 2013.
[95] Z. Wu, S. Gao, E. S. Cling, and H. Li, "A study on replay attack and anti-spoofing for text-dependent speaker verification," in *APSIPA*, 2014.
[96] M. Shirvanian, S. Vo, and N. Saxena, "Quantifying the breakability of voice assistants," in *PerCom*, 2019.
[97] M. Shirvanian and N. Saxena, "Wiretapping via mimicry: Short voice imitation mitm attacks on crypto phones," in *ACM CCS*, 2014.
[98] P. L. De Leon, M. Pucher, J. Yamagishi, I. Hernaez, and I. Saratxaga, "Evaluation of speaker verification security and detection of hmm-based synthetic speech," *IEEE/ACM Trans. Audio, Speech & Language Processing*, 2012.
[99] Z. Wu and H. Li, "Voice conversion and spoofing attack on speaker verification systems," in *APSIPA*, 2013.
[100] D. Mukhopadhyay, M. Shirvanian, and N. Saxena, "All your voices are belong to us: Stealing voices to fool humans and machines," in *ESORICS*, 2015.
[101] M. Shirvanian, N. Saxena, and D. Mukhopadhyay, "Short voice imitation man-in-the-middle attacks on crypto phones: Defeating humans and machines," *Journal of Computer Security*, 2018.
[102] S. Chen, M. Xue, L. Fan, S. Hao, L. Xu, H. Zhu, and B. Li, "Automated poisoning attacks and defenses in malware detection systems: An adversarial machine learning approach," *Computers & Security*, 2018.
[103] R. Eberhart and J. Kennedy, "A new optimizer using particle swarm theory," in *MHS*, 1995.
[104] J. Sohn, N. Kim, and W. Sung, "A statistical model-based voice activity detection," *IEEE Signal Processing Letters*, 1999.
[105] A TensorFlow implementation of Baidu's DeepSpeech architecture. https://github.com/mozilla/DeepSpeech.

APPENDIX

A. Comparison of our FAKEBOB and PSO-based Method

W compare our attack FAKEBOB over a PSO-based method. We reduce the finding of an adversarial sample as an optimization problem (cf. §IV-A), then solve the optimization problem via the PSO algorithm. PSO solves the optimization problem by imitating the behaviour of a swarm of birds [103]. Each particle is a candidate solution, and in each iteration, the particle updates itself by the weighted linear combination of three parts, i.e., inertia, local best solution and global best solution. The related weights are initial inertia factor w_{init}, final inertia factor w_{end}, acceleration constant c_1 and c_2.

We implement a PSO-based attack following the algorithm of Sharif et al. [63] which is used to fool face recognition systems. After fine-tuning the above hyper-parameters, we conduct the experiment using the PSO-based method with 50 particles for a maximum of 35 epochs, and we set the iteration

TABLE XI: Our attack FAKEBOB vs. the PSO-based method, where $[S(x_0)]_t$ denotes the initial score of input voice of the speaker t, and $*$ denotes that only one adversarial attack succeeds.

	$-\infty < [S(x_0)]_t < \infty$		$[S(x_0)]_t \leq -0.5$		$-0.5 < [S(x_0)]_t \leq 0$		$0 < [S(x_0)]_t \leq 0.5$		$0.5 < [S(x_0)]_t \leq 1$		$1 < [S(x_0)]_t \leq 1.5$	
	FAKEBOB	**PSO**	**FAKEBOB**	**PSO**	**FAKEBOB**	**PSO***	**FAKEBOB**	**PSO**	**FAKEBOB**	**PSO**	**FAKEBOB**	**PSO**
#Iteration	86	136	187	—	84	72	61	147	17	297	4	24
Time (s)	2277	2524	4409	—	1947	1311	1384	2715	357	5517	77	449
SNR (dB)	31.5	31.9	31.4	—	30.5	22.8	31.5	31.6	32.4	32.3	31.8	32.2
ASR (%)	99.0	33.0	96.3	0.0	100.0	5.3	100.0	17.6	100.0	60.0	94.1	100.0

TABLE XII: Experimental results of FAKEBOB on xvector system

Task	**All**								**Intra-gender attack**				**Inter-gender attack**			
	Targeted Attack				**Untargeted Attack**				**Targeted Attack**				**Targeted Attack**			
	#Iter	**Time (s)**	**SNR (dB)**	**ASR (%)**	**#Iter**	**Time (s)**	**SNR (dB)**	**ASR (%)**	**#Iter**	**Time (s)**	**SNR (dB)**	**ASR (%)**	**#Iter**	**Time (s)**	**SNR (dB)**	**ASR (%)**
CSI	117	575	30.1	100.0	73	499	29.6	100.0	89	444	29.3	100	135	662	30.7	100.0
SV	92	702	31.8	100.0	-	-	-	-	44	340	31.9	100.0	136	1035	31.7	100.0
OSI	95	995	32.0	100.0	26	171	31.5	100.0	51	601	32.0	100.0	138	1380	32.0	100.0

limitation of each epoch to 30, w_{init} to 0.9, w_{end} to 0.1, c_1 to 1.4961 and c_2 to 1.4961. The experiment is conducted on the ivector system for the OSI task.

The results are shown in Table XI. For comparison purposes, we also report the results of our attack FAKEBOB in Table XI. Overall, the PSO-based method achieves 33% targeted attack success rate (ASR), only one-third of FAKEBOB, indicating that FAKEBOB is much more effective than the PSO-based method. Specifically, the PSO-based method is less effective for input voices whose initial scores are low.

- When $[S(x_0)]_t \leq -0.5$, the PSO-based method fails to launch attack for all the voices.
- When $-0.5 < [S(x_0)]_t \leq 0$ and $0 < [S(x_0)]_t \leq 0.5$, the ASR is very low, i.e., 5.3% and 17.6%, respectively.

Whereas our attack FAKEBOB is more effective no matter the initial scores of input voices.

In terms of efficiency, FAKEBOB takes less number of iterations and execution time than the PSO-based method, except for the case $-0.5 < [S(x_0)]_t \leq 0$ on which the PSO-based method is able to launch a successful attack for one voice *only*. Specifically, the higher the initial score of the input voice is, the more efficient of our attack FAKEBOB is compared to the PSO-based method. For instance, when $0.5 < [S(x_0)]_t \leq 1$, the number of iterations (resp. execution time) of the PSO-based method is 17 times (resp. 15 times) larger than the one of FAKEBOB.

In summary, the experimental results demonstrate that our attack FAKEBOB is much more effective and efficient than the PSO-based method.

B. 16 Attack Scenarios

All of following combinations are evaluated in this work, where D.&S. denotes decision and scores.

$$\left\{ \begin{array}{c} \left(\begin{array}{c}\text{targeted}\\ \text{untargeted}\end{array}\right) \times \left(\begin{array}{c}\text{intra-gender}\\ \text{inter-gender}\end{array}\right) \times \text{API} \times \left(\begin{array}{c}\text{OSI}\\ \text{CSI}\\ \text{SV}\end{array}\right) \times \text{D.\&S.} \\ + \\ \text{targeted} \times \left(\begin{array}{c}\text{OSI}\\ \text{CSI}\\ \text{SV}\end{array}\right) \times \text{API} \times \text{decision-only} \\ + \\ \text{targeted} \times \left(\begin{array}{c}\text{OSI}\\ \text{CSI}\\ \text{SV}\end{array}\right) \times \text{over-the-air} \times \text{D.\&S.} \\ + \\ \text{targeted} \times \text{OSI} \times \text{over-the-air} \times \text{decision-only} \end{array} \right\}$$

C. Results of Tuning the Parameter ϵ

Table XIII shows the results of tuning the parameter ϵ on both ivector and GMM systems for the CSI task. To choose a suitable ϵ, we need to trade off the imperceptibility and the attack cost. Smaller ϵ contributes to less perturbation (i.e, higher SNR), but also give rise to the attack cost (i.e, more iterations and execution time and lower success rate). We found that 0.002 is a more suitable value of ϵ for two reasons: (1) compared with other ϵ values, the average SNR of adversarial voices when $\epsilon = 0.002$ is higher, indicating that $\epsilon = 0.002$ introduces less perturbation, while the success rate of 0.002 is merely 1% lower than that of other ϵ values. (2) $\epsilon = 0.001$ introduce less perturbation than $\epsilon = 0.002$, but the success rate of $\epsilon = 0.001$ drops to 41% for ivector and 87% for GMM, 58% and 12% lower than that of $\epsilon = 0.002$. Moreover, the attack cost increases more sharply when decreasing ϵ from 0.002 to 0.001 compared with decreasing ϵ from 0.003 to 0.002. That is, the number of iterations and execution time of $\epsilon = 0.002$ are 1.6 times and 1.4 times than that of $\epsilon = 0.003$, while the number of iterations and execution time of $\epsilon = 0.001$ are 2.2 times and 2.4 times than that of $\epsilon = 0.002$.

TABLE XIII: Results of tuning ϵ on the CSI task

ϵ	**ivector**				**GMM**			
	#Iter	**Time** (s)	**SNR** (dB)	**ASR** (%)	**#Iter**	**Time** (s)	**SNR** (dB)	**ASR** (%)
0.05	18	422	12.0	100	18	91	16.7	100
0.01	23	549	16.2	100	16	81	19.1	100
0.005	44	1099	21.8	100	19	102	22.3	100
0.004	56	1423	23.8	100	21	104	24.0	100
0.003	76	2059	26.3	100	27	124	26.1	100
0.002	124	2845	30.2	99	40	218	29.3	99
0.001	276	6738	36.4	41	106	551	35.7	87

D. Experiment results of FAKEBOB on xvector system

We demonstrate the effectiveness and efficiency of FAKEBOB against a state-of-the-art DNN-based SRS [26], called xvector system, in which xvector is extracted from DNN

TABLE XIV: Details of source and target systems for transferability attacks, where DF denotes Dimension of feature, FL/FS denotes Frame length/Frame step, ♯GC denotes the number of gaussian components, DV denotes Dimension of ivector (xvector), and xvector is a DNN-based SRS from [26].

System ID	A	B	C	D	E	F	G	H	I	J
Architecture	**GMM**	**ivector**	ivector	ivector	ivector	ivector	ivector	ivector	ivector	**xvector**
Training set	Train-1 Set	Train-1 Set	**Train-2 Set**	Train-1 Set	Train-1 Set	Train-1 Set	Train-1 Set	Train-1 Set	Train-1 Set	Train-1 Set
Feature	MFCC	MFCC	MFCC	**PLP**	MFCC	MFCC	MFCC	MFCC	**PLP**	MFCC
DF	24×3	24×3	24×3	24×3	**13×3**	24×3	24×3	24×3	**13×3**	**30**
FL/FS (ms)	25/10	25/10	25/10	25/10	25/10	**50/10**	25/10	25/10	**50/10**	25/10
♯GC	2048	2048	2048	2048	2048	2048	**1024**	2048	**1024**	**–**
DV	–	400	400	400	400	400	400	**600**	**600**	**512**

TABLE XV: The performance of the target systems C,...,J

Task \ System		C	D	E	F	G	H	I	J
CSI	**Accuracy**	99.8%	99.4%	99.2%	99.8%	99.6%	99.8%	99.2%	99.2%
SV	**FAR**	10.0%	9.8%	9.4%	10.0%	11.2%	9.8%	10.4%	10.2%
	FRR	1.2%	0.6%	1.6%	1.2%	0.8%	1.0%	2.2%	0.8%
OSI	**FAR**	9.1%	8.8%	10.9%	9.2%	8.5%	8.1%	11.0%	7.7%
	FRR	1.4%	0.6%	1.6%	1.4%	1.2%	0.8%	2.2%	0.8%
	OSIER	0.0%	0.2%	0.2%	0.0%	0.2%	0.0%	0.4%	0.2%

TABLE XVI: Results of transferability attack for CSI task (%), where S denotes source system and T denotes target system.

T / S	A		B		C		D		E		F		G		H		I		J	
	ASR	**UTR**	**ASR**	**UTR**	**ASR**	**UTR**	**ASR**	**UTR**	**ATR**	**UTR**	**ASR**	**UTR**	**ASR**	**UTR**	**ASR**	**UTR**	**ASR**	**UTR**	**ASR**	**UTR**
A	—	—	76.9	76.9	89.7	89.7	64.1	71.8	87.2	89.7	84.6	84.6	76.9	87.2	76.9	84.6	48.7	69.2	28.2	38.5
B	30.7	88.0	—	—	93.3	96.0	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	88.0	89.3	**100.0**	**100.0**	73.3	80.0	25.3	38.7

TABLE XVIII: Results of transferability attack for SV task (%), where S: source system and T: target system.

T / S	A	B	C	D	E	F	G	H	I	J
	ASR	**ASR**	**ASR**	**ASR**	**ASR**	**ASR**	**ASR**	**ASR**	**ASR**	**ASR**
A	—	57.9	49.1	54.4	64.9	61.4	52.6	66.7	36.8	33.3
B	5.0	—	67.5	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	80.0	38.3

TABLE XVII: Results of FAKEBOB when θ is tuned based on Equal Error Rate. The Equal Error Rate and corresponding threshold θ for ivector (resp. GMM) are 2.2% and 1.75 (resp. 5.8% and 0.103), and $\epsilon = 0.002$.

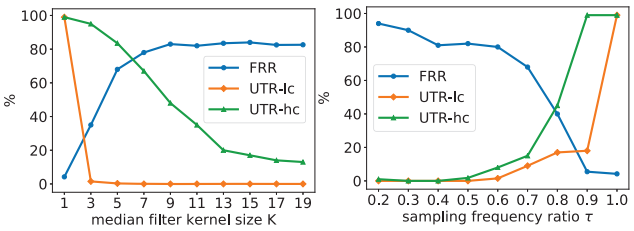
Task	ivector				GMM			
	#Iter	**Time (s)**	**SNR (dB)**	**ASR (%)**	**#Iter**	**Time (s)**	**SNR (dB)**	**ASR (%)**
SV	120	2297	31.7	99.0	46	273	31.4	99.0
OSI	125	2786	32.1	99.0	54	334	31.9	99.0

networks. We use the pre-trained xvector model from SITW recipe of Kaldi and construct OSI, CSI and SV systems. We use the same settings as in Section V-B. The baseline performance of the resulting systems is shown in Column J of Table XV. Moreover, we also conduct untargeted attacks against these systems. The results are shown in Table XII. Our attack is able to achieve 100% ASR, indicating FAKEBOB is also effective and efficient against DNN-based SRSs.

E. Robustness of FAKEBOB *against Defense Methods*

Local smoothing. It mitigates attacks by applying the mean, median or gaussian filter to the waveform of a voice. Based on the results in [31], we use the median filter. A median filter with kernel size k (must be odd) replaces each audio element x_k by the median of k values $[x_{k-\frac{k-1}{2}}, \ldots, x_k, \ldots, x_{k+\frac{k-1}{2}}]$. In S1, we vary k from 1 to 19 with step 2. The results are shown in Fig. 8a. We can see that the defense is ineffective against high-confidence (hc) adversarial voices. For low-confidence (lc) adversarial voices, though the UTR drops from 99% to nearly 0%, the minimal FRR of normal voices increases to 35%, significantly larger than the baseline 4.2%. We also tested median with $k = 3$ on ivector. The FRR of normal voices only increases by 7%. It seems that ivector is more robust than GMM. In S2, we fix k=7 as [31] did. The results are shown in Fig. 9a. Although the median filter increases the attack cost slightly, FAKEBOB can quickly achieve 90% ASR using 250 max iteration bound, where the baseline is 90. To solve other few voices (9%), the max iteration bound should be 15,000. Though ivector is more robust than GMM, the similar result is observed (cf. Fig. 9b).

We conclude that the local smoothing (at least median filter) can increase attack cost, but is ineffective in terms of ASR.



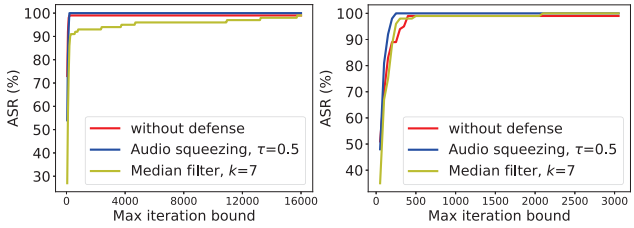
(a) Median filter (b) Audio squeezing

Fig. 8: Results of median filter and audio squeezing in S1, where UTR-lc denotes UTR of low-confidence adversarial voices (κ=0), and UTR-hc denotes UTR of high-confidence adversarial voices ($0 < \kappa < 5$).

Audio squeezing. It down-samples voices and applies signal recovery to disrupt perturbations. In S1, we vary τ (the ratio between new and original sampling frequency) from 0.1 to

TABLE XIX: Settings of the over-the-air attacks, where *x meter (y dB)* means when the microphone is kept x meters away from the loudspeaker, the average volume of voices reaches y dB, and *white noise (z dB)* means the acoustic environment is degraded with a white-noise generator playing at z dB.

	System	Loudspeaker	Microphone	Distance	Acoustic Environment
Different Systems	**GMM OSI/CSI/SV** **ivector OSI/CSI/SV** **Azure OSI**	JBL clip3 portable speaker	IPhone 6 Plus (iOS)	1 meter (65 dB)	relatively quiet
Different Devices	ivector OSI	**DELL laptop** **JBL clip3 portable speaker** **Shinco brocast equipment**	**IPhone 6 Plus (iOS)** **OPPO (Android)**	1 meter (65 dB)	relatively quiet
Different Distances	ivector OSI	JBL clip3 portable speaker	IPhone 6 Plus (iOS)	**0.25 meter (70 dB)** **0.5 meter (68 dB)** **1 meter (65 dB)** **2 meters (62 dB)** **4 meters (60 dB)** **8 meters (55 dB)**	relatively quiet
Different Acoustic Environments	ivector OSI	JBL clip3 portable speaker	IPhone 6 Plus (iOS)	1 meter (65 dB)	**white noise (45/50/60/65/75 dB)** **bus noise (60 dB)** **restaurant noise (60 dB)** **music noise (60 dB)** **absolute music noise (60 dB)**



(a) GMM system (b) ivector system

Fig. 9: Attack cost of median filter and audio squzzeing

1.0, the same as [10]. The results are shown in Fig. 8b. We can observe that when $\tau = 0.9$, (1) the FRR of normal voices is 6%, close to the baseline 4.2%, (2) the UTR of the low-confidence adversarial voices is 17%, smaller than the baseline 99%, (3) however, the UTR of the high-confidence adversarial voices is the same as the baseline. In S2, we fix τ=0.5 as [31] did. The results are shown in Fig. 9a and Fig. 9b. Unexpectedly, the defense decreases the overhead of attack and increases ASR. For instance, FAKEBOB achieves 100% ASR using 200 max iteration bound on the system with defense, while can only achieve 99% ASR even using 16,000 max iteration bound on the unsecured system. It is possibly because audio squeezing ($\tau = 0.5$) sacrifices the performance of SRSs.

We conclude that the audio squeezing is ineffective against FAKEBOB in terms of both attack cost and ASR.

Quantization. It rounds the amplitude of each sample point of a voice to the nearest integer multiple of factor q to mitigate the perturbation. In S1, we vary q from 128, 256, 512 to 1024 as [31] did. However, the system did not output any result on adversarial and normal voices. An in-depth analysis reveals that all the frames of voices are regarded as unvoiced frame by the Voice Activity Detection (VAD) [104] component. This demonstrates that quantization is not suitable for defending against FAKEBOB. Due to this, we do not consider S2.

Temporal dependency Detection. For a given voice v, suppose a speech-to-text system produces text $t(v)$. Given a parameter $0 \leq k \leq 1$, let v_k (resp. t_k) denote the k percent prefix of the voice v (resp. text t). The temporal dependency detection uses the distance between the texts $t(v)_k$ and $t(v_k)$ to determine whether v is an adversarial voice, as the distance of adversarial voices is greater than that of normal voices. We use this method to check adversarial voices crafted by FAKEBOB using k=$\frac{4}{5}$ and the Character Error Rate distance metric, the best one in [31]. We do not test different values of k as the result will not vary too much as mentioned in [31]. We use Baidu's DeepSpeech model as the speech-to-text system, which is implemented by Mozilla on Github [105] with more than 13k stars.

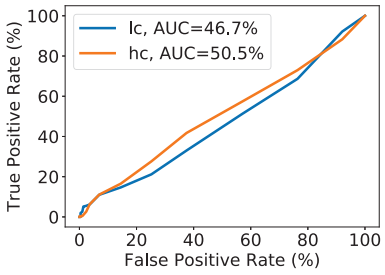


Fig. 10: ROC curves of Temporal Dependence Detection

Fig. 10 shows the ROC curves of this method distinguishing low-confidence and high-confidence adversarial samples. It obtains 50% true positive rate at about 50% false positive rate. The AUC values are 46.7% and 50.5%, close to random guess, indicating it fails to detect adversarial samples. This is because FAKEBOB does not alter the transcription of the voices, thus the temporal dependency is preserved. Due to this, we do not consider S2.