More Simplicity for Trainers, More Opportunity for Attackers: Black-Box Attacks on Speaker Recognition Systems by Inferring Feature Extractor

Yunjie Ge[1,*], Pinji Chen[1,*], Qian Wang[1,†], Lingchen Zhao[1,†], Ningping Mou[1],
Peipei Jiang[1,2], Cong Wang[2], Qi Li[3], and Chao Shen[4]

[1]*School of Cyber Science and Engineering, Wuhan University*
[2]*Department of Computer Science, City University of Hong Kong*
[3]*Institute of Network Sciences and Cyberspace, Tsinghua University*
[4]*School of Cyber Science and Engineering, Xi'an Jiaotong University*

Abstract

Recent studies have revealed that deep learning-based speaker recognition systems (SRSs) are vulnerable to adversarial examples (AEs). However, the practicality of existing black-box AE attacks is restricted by the requirement for extensive querying of the target system or the limited attack success rates (ASR). In this paper, we introduce VoxCloak, a new targeted AE attack with superior performance in both these aspects. Distinct from existing methods that optimize AEs by querying the target model, VoxCloak initially employs a small number of queries (e.g., a few hundred) to infer the feature extractor used by the target system. It then utilizes this feature extractor to generate any number of AEs locally without the need for further queries. We evaluate VoxCloak on four commercial speaker recognition (SR) APIs and seven voice assistants. On the SR APIs, VoxCloak surpasses the existing transfer-based attacks, improving ASR by 76.25% and signal-to-noise ratio (SNR) by 13.46 dB, as well as the decision-based attacks, requiring 33 times fewer queries and improving SNR by 7.87 dB while achieving comparable ASRs. On the voice assistants, VoxCloak outperforms the existing methods with a 49.40% improvement in ASR and a 15.79 dB improvement in SNR.

1 Introduction

Our voices not only convey "what we speak" but also reflect "who we are". Based on this phenomenon, speaker recognition (SR) techniques have been widely applied in various biometric authentication systems [1–3]. The core of current speaker recognition systems (SRSs) is deep neural networks (DNNs) [4, 5]. Unfortunately, SRSs also inherit various vulnerabilities from DNNs. One of the primary security concerns is adversarial examples (AEs), where an adversary can bypass SRSs by adding small perturbations to audio inputs [6–8].

The AE attacks can be categorized into the white-box attacks and the black-box attacks based on the capacity of the attacker. In the white-box setting, where the internal details, such as the parameters and architecture of the underlying DNN, are accessible, the adversary can easily generate AEs using the gradient descent method [9, 10]. In the black-box setting where the adversary cannot access the internal details, it is still possible to generate AEs through query-based attacks or transfer-based attacks. Query-based attacks [7, 11] operate by estimating the gradients of the target model via the analysis of how variations in model inputs affect outputs, as shown in Figure 1 (a). Transfer-based attacks leverage the transferability of AEs, using examples that can deceive a known model to fool an unknown model [12, 13], as seen in Figure 1 (b). However, the assumption of the white-box attacks is too strong, as attackers typically cannot access the internal information of the target system in practice. Both black-box attack approaches also have inherent limitations. Specifically, query-based attacks require extensive queries to the target model to estimate gradients, while transfer-based attacks face restricted attack success rates, as summarized in Table 1. These constraints hinder their practicability in reality.

In this paper, we take a different approach from prior works to explore the vulnerability of SRSs against AE attacks. Instead of generating AEs targeting the underlying DNNs, we focus on disrupting the feature extractor (FE) to realize the attack. Our primary insight is that since the DNN relies on the output from the FE as its input, disrupting the FE will naturally impact the output of the DNN. Compared to complex DNNs, the FEs are easier to manipulate intuitively because their structures are simpler. Besides, we observe that most existing SRSs utilize a few common feature extraction algorithms, such as Mel-frequency Cepstral Coefficients (MFCC) [14], D-Vector [15], and X-Vector [16], to construct their FEs [17]. This fact inspires us to attempt to infer the FE used by the target system, thereby enabling a white-box attack to simplify the optimization of adversarial perturbations. This approach is similar to prior works, which initially establish local substitute models closely resembling the target model through

[*] The first two authors contributed equally to this work.

[†] Corresponding authors.

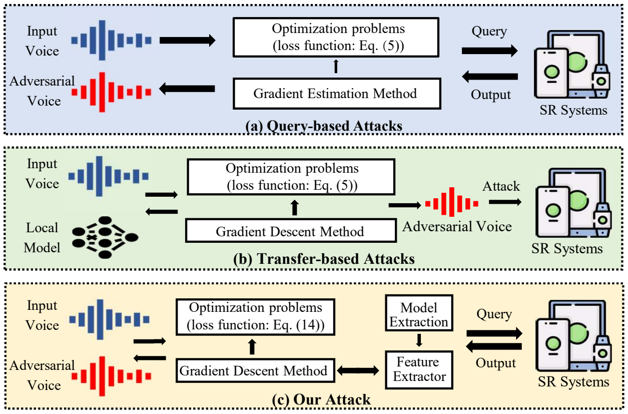


Figure 1: Overview of query-based attacks, transfer-based attacks, and our attack.

model extraction attacks [18, 19], and then utilize the substitute models to optimize AEs. However, compared to the prior works that aim to extract complex black-box DNNs, determining which feature extraction algorithm the target system uses could be much easier.

Based on the above considerations, we propose VoxCloak, a new targeted AE attack against black-box SRSs, as shown in Figure 1 (c). VoxCloak comprises two stages. The first is inferring the FE of the target black-box SRS from a set of candidates. We utilize the fact that each FE defines unique feature spaces, and the subsequent DNN will output similar results for similar feature vectors. Specifically, we generate AEs for each candidate FE. Then, we can determine the target FE by analyzing which AEs successfully compromise the target system. In addition, to address the problem where the target system outputs identical results for multiple AEs associated with different FEs, making it hard to determine the target FE, we design a genetic algorithm to eliminate incorrect candidates to improve the accuracy of the inference results.

The second is generating high-quality AEs based on the inferred FE. After inferring the target FE with a small number of queries, we can generate an arbitrary number of AEs locally without the need for any further queries. Therefore, compared to previous attacks that may require tens of thousands of queries to craft a single AE [6], our method significantly reduces the cost of the attack. We also enhance the imperceptibility and real-world robustness of the AEs by incorporating the psychoacoustic masking effect and the room impulse response into the AE generation process.

Compared to prior works, we highlight the three advantages of VoxCloak: (1) *Low Cost*: VoxCloak only requires a small number of queries to the target system, such as a few hundred, which is a one-time cost. After this, the attacker can generate any number of AEs without the need for further queries (unless the target system updates its used FE). (2) *Imperceptibility*: The AEs generated by VoxCloak are less detectable to the human ear, offering improved imperceptibility. (3) *Robustness*: VoxCloak could successfully attack various SRSs in the physical world with an effective attacking distance of up to 4 meters. This surpasses the state-of-the-art method with a maximum distance of 2 meters.

Our main contributions are summarized as follows:

- To the best of our knowledge, we are the first to utilize the vulnerability stemming from the reuse of FEs to realize targeted AE attacks against SRSs. This vulnerability allows adversaries to bypass user identity authentication mechanisms.
- We propose VoxCloak, a new targeted AE attack against black-box SRSs. Our attack requires a one-time, small number of queries to the target SRS for inferring the employed FE. This design facilitates the offline generation of an arbitrary number of AEs.
- We evaluate VoxCloak on four commercial SR APIs and seven smart devices equipped with SR techniques. The results show that VoxCloak achieves an average attack success rate close to 100% and 70%, with a signal-to-noise ratio of 21.47 dB and 21.38 dB in the digital and physical domains, respectively.

2 Background

2.1 Speaker Recognition Systems

SRSs are designed to authenticate the identities of speakers based on the unique features of their voices. The typical workflow of an SRS is shown in Figure 2, comprising five main steps: feature extraction, universal background model modeling, speaker modeling, pattern matching, and score decision. These steps can be divided into three phases: the offline training phase (top part), followed by the online enrollment and online recognition phases (lower parts) [24].

In the offline training phase, an FE is employed to extract acoustic feature vectors from a set of voices. These feature vectors are then utilized to train a UBM [5], which captures the general features inherent in human speech audio. In the online enrollment phase, the SRS constructs a user-specific model based on the voices uploaded by the user, leveraging the previously established background model. In the online recognition phase, the SRS calculates a score to quantify the similarity between the input voice and the voices stored in the system. The authentication result of the identity is then determined based on a predefined threshold about the score.
Remark. All three phases involve an independent feature extraction process for reducing the dimensionality of the raw speech signal. Most current SRSs rely on existing FEs, such as MFCC, Mel spectrum (Mel) [25], X-Vector, D-Vector, etc. Detailed information regarding these FEs is provided in Appendix A.2. As these FEs are open-sourced and time-proven,

Table 1: Overview of the state-of-the-art adversarial attacks against SRSs.

Method	Threat Scenario	Attack Type	Task	Commercial	Target	Carrier	Query	ASR
Li *et al.* [20]	White-box	Digital	CSI	×	✓	Speech	-	<100%
VMASK [12]	Black-box	Digital	SV	×	✓	Speech	500	100%
	Black-Box	Physical	SI	✓	✓		0	67%
Advpules [21]	White-box	Digital	SI	×	✓	Speech	-	90%
FakeBob [7]	Black-box	Digital	OSI, CSI, SV	×	✓	Speech	3000	100%
Occam [6]	Black-Box	Digital	SI, SV	✓	✓	Song	10000	100%
Abdullah *et al.* [22]	Black-Box	Digital	SI	×	×	Speech	0	100%
		Physical		✓				42%
NRI-FGSM [23]	Black-box	Digital	OSI, CSI	×	✓	Speech	0	<100%
Our	Black-box	Digital	OSI, CSI, SV	✓	✓	Speech	$\sim$300*	$\sim$100%
		Physical	SV				0	70%

$*$: The query cost is one-time, where our attack only queries the black-box model to infer its FE in the first stage and then generates AEs without querying the model in the second stage.

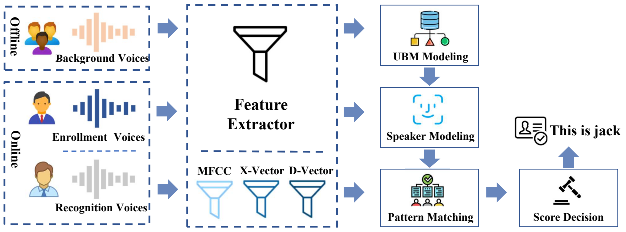


Figure 2: The workflow of an SRS contains three phases: offline training, online enrollment, and online recognition.

developers can easily build their own SRSs by directly invoking them, requiring minimal additional effort.

In terms of specific functionalities, SRSs typically address three types of sub-tasks: close-set identification (CSI) [26], speaker verification (SV) [27], and open-set identification (OSI) [28]. The differences among these sub-tasks lie in their methods of determining whether the current speaker initiating the authentication request is indeed the enrolled user. CSI identifies which enrolled user matches the current speaker. SV checks whether the current speaker can be identified as the target enrolled user. OSI determines whether the current speaker is an enrolled user or an unknown user.

Formally, let $S(\cdot)$ be the similarity score between the enrolled user and the current speaker x. For a CSI system, it outputs the identity of the enrolled user with the highest score. The decision module $D(x)$ can be formulated as follows:

$$D(x) = \underset{i \in T}{\arg\max}\ [S(x)]_i, \tag{1}$$

where the set T includes m enrolled users $\{1, 2, \ldots, m\}$. For an SV system, T only contains one enrolled user. It determines whether an input voice x belongs to the user according to the score and a predefined threshold θ. Its decision module output $D(x)$ can be formulated as follows:

$$D(x) = \begin{cases} S(x), & \text{if } S(x) \geq \theta, \\ \text{reject}, & \text{otherwise.} \end{cases} \tag{2}$$

The OSI system is a combination of CSI and SV. It utilizes the output of Eq. (1) and the threshold for decision as follows:

$$D(x) = \begin{cases} \underset{i \in T}{\arg\max}\ [S(x)]_i, & \text{if } \underset{i \in T}{\max}\ [S(x)]_i \geq \theta, \\ \text{reject}, & \text{otherwise.} \end{cases} \tag{3}$$

2.2 Adversarial Example Attacks

AE attacks are one of the primary security threats faced by DNNs [29–31]. Attackers can manipulate the output of the target model by introducing subtle noises into the inputs.

AE attacks can be categorized into white-box attacks [21, 32] and black-box attacks [33–35] based on the capabilities of the adversary. In the white-box setting, the adversary has complete access to the target model, including its architecture and parameters. Such information enables the adversary to generate AEs relatively easily using gradient-based methods [9, 10]. However, in the black-box setting, the adversary is limited to accessing only the outputs of the target model, making the attacks more challenging than in the white-box setting [6]. Furthermore, most real-world SRSs, such as commercial APIs and smart devices with authentication capabilities, operate as black boxes. They only disclose final results to users without any internal details, making black-box attacks more impactful than white-box attacks in practice.

Depending on the objective of the attack, AE attacks can also be categorized into untargeted and targeted attacks. Untargeted AEs [36] can deceive the target model into misclassifying the inputs into a random category, while targeted AEs [37] are designed to manipulate the target model into classifying the inputs into a category specified by the attacker.

Formally, let $\mathcal{X}$ be the set of inputs and $\mathcal{Y}$ be the set of outputs. Given the target model $f(x) : x \in \mathcal{X} \mapsto y \in \mathcal{Y}$, an untargeted AE x^* can be represented as:

$$x^*, \quad s.t. \;\; f(x^*) \neq y \text{ and } \|x^* - x\| \leq \varepsilon, \tag{4}$$

where ε is a parameter used to limit the magnitude of the adversarial perturbation, and $\|\cdot\|$ is a distance function, e.g., l_2-norm. Similarly, a targeted AE with the target class y^* can be formulated as follows:

$$x^*, \quad s.t. \;\; f(x^*) = y^* \text{ and } \|x^* - x\| \leq \varepsilon. \tag{5}$$

In this paper, we focus on generating targeted AEs in the black-box setting, as they potentially pose a greater real-world threat compared to untargeted AEs. Moreover, since targeted attacks are more challenging to achieve, the investigation of these attacks can provide deeper insights into the vulnerabilities of SRSs [38].

2.3 Model Extraction Attacks

Model extraction attacks are for creating a model highly similar to the target model by analyzing the returned query results [18, 19, 39]. One of the purposes of this attack is to serve as an auxiliary method to facilitate black-box AE attacks [40]. While obtaining a substitute model with similar functionality to the target model, the adversary can use it to generate AEs locally via white-box attack methods, thus saving the cost of querying the target model.

To our knowledge, there have been no prior model extraction attacks specifically targeting SRSs. Due to the inherent complexity of SRSs and the diversity of their training data collected from various users, it is difficult for the adversary to extract a model that is sufficiently similar to the target model.

2.4 Threat Model

Our goal is to achieve a practical attack capable of deceiving commercial SRSs, including online SR services in the digital domain, as well as smart devices with identity authentication mechanisms in the physical domain.

We outline the following essential properties for an effective real-world attack: (1) It should be a black-box attack, as most commercial SRSs, including APIs and smart devices, only provide users with the final recognition results; (2) The number of queries to the target system should be as few as possible to reduce the cost of the attack and enhance its efficiency; (3) The attacker should not be able to access the target device, such as a smartphone, physically; (4) The generated AEs should be highly imperceptible, ensuring that they remain undetected by the victim.

We make two assumptions about the target systems and the capabilities of the attackers in real-world scenarios. Firstly, we assume that the target system follows the typical workflow employed by most existing SRSs. As the SRSs usually consist of the five components as introduced in Section 2.1, we consider that the assumption is practical. Secondly, the attacker could obtain a single voice sample of the enrolled user, which is long enough for authentication by the SRS, through alternative attacks, such as eavesdropping, phishing, or social engineering. This assumption aligns with traditional replay attacks [41], voice impersonation attacks [42], and prior AE attacks against SRSs [6, 12, 43].[1]

3 Methodology

3.1 Key Insights

Our first insight is that *once the target FE is known, generating AEs becomes easier.* SRSs typically operate by initially employing the FE to derive feature vectors fed into a subsequent DNN. Therefore, if the target system extracts the same feature vector from the AE as the enrolled voice, it will naturally lead to incorrect recognition results. Previous studies have suggested generating AEs targeting some specific FEs [13]. However, given the nearly infinite space of voice samples, searching for an AE that meets this requirement is extremely difficult. We observe that most current SRSs achieve identification according to the similarity between the output of the universal background model and the enrolled voice rather than requiring them to be identical (which is inherently difficult in the physical world due to environmental interferences). Hence, we transform the problem of searching for AEs into an optimization problem that focuses on optimizing a voice sample to make its feature vector as close as possible to the voice of the legitimate user.

The second insight is that *to infer the FE used by the target system, it is only necessary to correctly select from among the candidates.* The previous research revealed that current SRSs typically employ only a few common FEs [24]. This observation simplifies the task from "extracting" the specific FE used by the target system to "determining" which FE it uses. A natural method for inferring the target FE is to analyze the differences in the outputs of the target system to various inputs. However, the output of an SRS is typically limited to "accept" or "reject", which is insufficient to achieve the attack. For example, if the attacker queries the target system using normal samples, the system will always return correct recognition results (as long as it has high accuracy), regardless of the FE it uses. Therefore, our idea is to generate AEs for each candidate FE and then identify which examples can successfully deceive the target system. Intuitively, due to the differences among various FEs, the effectiveness of AEs indicates that the target FE is identical or at least similar to the FE corresponding to the AEs.

[1] In some practical situations, AE attacks are more feasible than replay attacks and voice impersonation attacks because such attacks might be easier to notice by the target user.

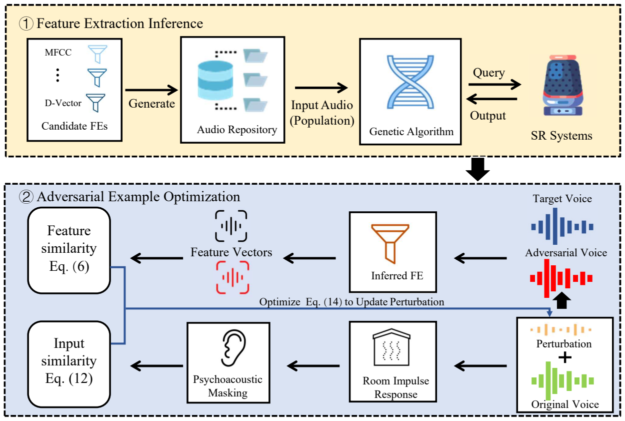


Figure 3: Overview of VoxCloak. The first step is to infer the FE of the target SRS using a genetic algorithm method. The second step utilizes the inferred FE to generate imperceptible and robust AEs through psychoacoustic masking and room impulse responses.

3.2 Overview

Building upon the two insights above, we propose VoxCloak, a black-box targeted attack focusing on the feature extractors. Specifically, VoxCloak comprises two main steps.

Step 1: Feature Extractor Inference. This step aims to infer the FE used by the target SRS, as illustrated in the upper part of Figure 3. To achieve this goal, we design a genetic algorithm. Initially, we generate AEs for all candidate FEs, respectively, by minimizing the distance between the voice of the attacker and the enrolled voice of the victim. These AEs serve as the initial population for the genetic algorithm. The fitness value of each individual (i.e., an AE) is evaluated according to the recognition results of the target SR. Based on the query results, we can filter out individuals with lower fitness. After multiple queries, we can infer the target FE according to the surviving individuals.

Step 2: Adversarial Example Optimization. This step aims to further improve the imperceptibility and robustness of the AE by utilizing the inferred FE, as shown in the bottom half of Figure 3. During the optimization process, in addition to minimizing the distance between the AE and the voice example of the victim, we incorporate psychoacoustic masking to make the perturbation almost inaudible. Moreover, we use room impulse responses to simulate the absorption and reverberation during over-the-air transmission, thereby improving the robustness of our attack. Note that this step is entirely carried out locally using the inferred FE without the need to query the target system.

4 Design of VoxCloak

4.1 Feature Extractor Inference

We first present how to generate AEs based on FEs and the specially designed genetic algorithm for inferring the FE utilized by the target SRS.

4.1.1 Generating AEs against Feature Extractor

As discussed in Section 3.1, we have transferred the challenge of extracting the FE employed by the target SRS into determining the FE from a set of candidates. A trivial approach is to compare the representation space of the candidate FEs with that of the target. Specifically, the attacker first constructs the common FEs locally (using open-source implementations). For each candidate FE, the attacker employs a personal input voice to generate AEs by minimizing the distance between the feature vector of this voice and the voice of the victim. Then, the attacker queries the target SRS by the AEs and analyzes the responses. Intuitively, AEs generated using an FE similar to the target FE would yield higher attack success rates due to their closer resultant feature vectors.

We approximate the computational process of the FE as a differentiable form, thereby minimizing the distance between the feature vector of the AE and that of the legitimate voice. Considering an audio example x of the attacker, an audio example x_t of the victim, and a candidate FE $g(\cdot)$, by introducing a perturbation δ on x, the formulation of optimizing a targeted AE $x+\delta$ can be defined as follows:

$$\arg\min_{\delta} \|g(x+\delta), g(x_t)\|_2 . \tag{6}$$

We select representative audio samples for each candidate FE from our dataset to construct an AE repository. This repository will then serve as the initial population for the subsequent genetic algorithm.

4.1.2 Genetic Algorithm

Theoretically, we can infer the information about the target FE according to the output of the AEs. However, such a straightforward method presented in Section 4.1.1 has a limitation: For each candidate FE, if we only generate one AE for the inference attack, then once multiple AEs can successfully deceive the target system, it becomes difficult to determine which candidate FE is correct. However, if we generate multiple AEs for each candidate FE, it will lead to a significant increase in the number of queries. To address this issue, we design a genetic algorithm to heuristically search for the FE most similar to the target FE from the candidate set.[2] Our

[2] To validate the effectiveness of the genetic algorithm, we have also attempted to infer the target FE using a straightforward grid search method. The results indicate that while this approach is feasible, it requires a higher number of queries. More detailed results are presented in Appendix B.1.

design includes two parts: encoding each candidate into a chromosome and realizing the basic operations of genetic algorithms, e.g., selection, crossover, and mutation operators, along with an apt fitness function. This approach is designed to mimic biological evolution, with the ultimate goal of accurately identifying the correct FE.

Encoding and Decoding. This step establishes a mapping relationship between the candidate FEs and the chromosomes of the genetic algorithm. We use an n-bit one-hot vector c^n to represent a possible solution, which can be formulated as:

$$c^n = [c_1, c_2, \cdots, c_n],\\ s.t. \begin{cases} c_i = 0, 1, \\ \sum_{i=1}^{n} c_i = 1. \end{cases} \tag{7}$$

Each element in the vector represents a candidate FE. To encode the i-th FE into a chromosome, we could set $c_i = 1$.

Fitness Function. This function assigns a fitness value to each chromosome to evaluate the quality of the candidate FE based on the authentication result outputted by the target SRS. Given an example x, the SRS can authenticate the identity of the speaker and return a score $S(x)$ or a decision $D(x)$.

For a score-based SRS that returns the decision scores to the user, the range of the score might be [0,1] or [0,100], depending on its configuration. Initially, we transform the score to fall within [0,100]. Then, we amplify the distance between different FEs by squaring the score. This amplification is also helpful for accelerating the convergence of the genetic algorithm. We formulate the fitness function $h(x)$ as $h(x) = S(x)^2$.

For a decision-based SRS that only returns the decision results to the user, we map the results to values within the [0,100] range to ensure the fitness function is differentiable. Specifically, since the OSI and CSI tasks might misidentify the input voice as belonging to another legitimate user, there are three possible decisions: reject, misidentify, and accept. We define the fitness function for these decisions as follows:

$$h(x) = \begin{cases} 10, & fail, \\ 50, & misidentify, \\ 100, & pass. \end{cases} \tag{8}$$

For the SV task, as it will only make two decisions, i.e., pass or fail, the fitness function sets the values of the two decisions to 10 and 100, respectively.

Selection, Crossover, and Mutation. The three operations are the basic components of a genetic algorithm. Selection involves choosing the most fit chromosomes and ensuring their genes are inherited in the subsequent generation. Crossover is combining two chromosomes to produce two new offspring chromosomes. The goal of mutation is to maintain the diversity of the population.

For the selection operation, we select chromosomes according to their fitness values based on the roulette wheel method. Moreover, we utilize the elitism selection approach, which transfers the superior chromosomes from previous populations to a new population. This approach ensures the selection of chromosomes exhibiting high performance across various populations while preventing the algorithm from erroneously identifying AEs, which succeed occasionally and are derived from inaccurate FEs, as the optimal solution. We do not implement the crossover operation because our encoding method renders the crossover meaningless and might even produce additional invalid chromosomes. Thus, after the selection process, we immediately undertake the mutation. The mutation operation is realized by randomly modifying a position c_i in the original one-hot vector to 1. It can prevent the genetic algorithm from converging on local optima.

Workflow. We hereby summarize the workflow of the genetic algorithm-based method for inferring the target FE. The attacker first establishes a population and encodes all candidate FEs into chromosomes. Secondly, AEs generated for candidate FEs are fed into the target SRS, yielding recognition results and associated fitness values. Thirdly, the attacker sequentially conducts the selection and mutation operations to create new populations. Then, the second and third steps are iteratively executed to update the population until the algorithm terminates. The termination criteria are set as identifying the best chromosome, one that persists across five consecutive generations, or the genetic algorithm running for over 30 generations.

4.2 Adversarial Example Optimization

After obtaining the inferred FE, the attacker can use it to generate AEs in a simpler way. This subsection details how to generate effective, imperceptible, and robust AEs.

4.2.1 Psychoacoustic Masking

Eq. (6) demonstrates a basic approach for generating an AE. The goal of the attacker is to make the feature vector of the AE as close as possible to that of the legitimate audio voice. However, such AEs generated by this unconstrained approach may contain significant perturbations, making them easily detectable by human ears and rendering the attack impractical.

Minimizing the l_p distance between two examples is a common way to improve the visual imperceptibility of image AEs. However, this approach is less feasible in the audio domain because human ears are highly sensitive to minor changes in amplitude and frequency in audio signals. Merely limiting the magnitude of the perturbation might still make it perceptible. To address this issue, we employ psychoacoustic masking, which exploits the limitations of human auditory perception to optimize adversarial perturbations.

Psychoacoustic masking refers to the phenomenon wherein louder sounds can mask quieter ones, thereby making them less noticeable [44]. It occurs when two sounds with close frequencies are present simultaneously. For example, if a louder sound at 1000 Hz is played alongside a quieter sound at 1010 Hz, the latter becomes less noticeable. In other words, the louder sound creates a "masking threshold" in the frequency domain. Any signals that fall below this threshold become less perceptible. Hence, we limit the magnitude of the perturbations and confine them to specific frequency ranges. This ensures that the perturbation does not significantly alter the energy and spectral shape of the audio.

Given an audio input, the first step is to calculate its frequency masking threshold through the normalized log-magnitude power spectral density (PSD) estimation [45]. It is achieved by performing a short-time Fourier transform on the raw audio signal, segmenting it into multiple frames, and then calculating their respective spectrums.[3] We represent the k-th bin of the spectrum for each frame x as $S_x(k)$. The PSD is calculated as follows:

$$p_x(k) = 10\log_{10}\left|\frac{1}{N}S_x(k)\right|^2. \tag{9}$$

Then, we normalize $p_x(k)$ to a sound pressure level (SPL) of 96 dB [46]

$$\bar{p}_x(k) = 96 - \arg\max_k\{p_x(k)\} + p_x(k). \tag{10}$$

Following the approach outlined in the work [46], we then calculate the frequency masking threshold $\theta_x(k)$. By analyzing the normalized PSD estimate of the perturbation $\bar{p}_x(k)$, we can determine which frequencies have sufficient energy to mask adjacent frequencies. Then, we calculate the frequency masking using a standard psychoacoustic model, such as the Bark scale, to identify which frequencies can be masked at certain energy levels to remain inaudible post-masking. For more details, please refer to the works [45, 46].

For an adversarial perturbation δ added to the audio input x, its normalized PSD estimate can be calculated as:

$$\bar{p}_\delta(k) = 96 - \arg\max_k\{p_x(k)\} + p_\delta(k). \tag{11}$$

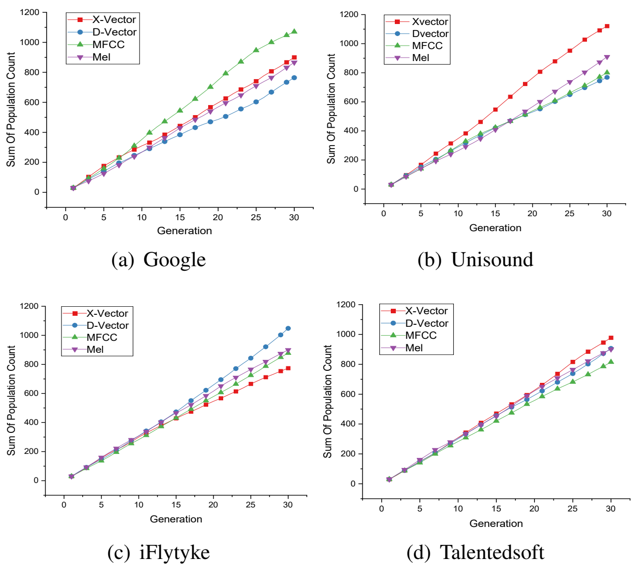
According to the principles of psychoacoustic masking, if the normalized PSD estimate of the perturbation δ falls below the global masking threshold $\theta_x(k)$, the perturbation will be less perceptible to human ears. It is achieved by minimizing the loss function defined as follows:

$$Psy(x,\delta) = \frac{1}{\left\lfloor\frac{N}{2}\right\rfloor + 1}\sum_{k=0}^{\left\lfloor\frac{N}{2}\right\rfloor} max\{\bar{p}_\delta(k) - \theta_x(k), 0\}. \tag{12}$$

Upon integrating the distance constraint, the optimization goal is:

$$\arg\min_\delta\{\|g(x+\delta), g(x_t)\|_2 + \alpha \cdot Psy(x,\delta)\}, \tag{13}$$

where α is a hyper-parameter to balance the effectiveness and the imperceptibility of the attack.



(a) Google

(b) Unisound

(c) iFlytyke

(d) Talentedsoft

Figure 4: Inferring FEs with genetic algorithm.

4.2.2 Room Impulse Response

An additional challenge arises in physical environments where audio signals might be distorted due to factors like absorption and reverberation, resulting in the ineffectiveness of the attack. To enhance the robustness of the AEs, we introduce a room impulse response (RIR) component during the optimization process to simulate environmental distortions and counteract their effects.

Specifically, we established a dataset of RIRs, using FAST_RIR [57] to generate 50,000 medium-sized room impulse responses. Then, we adjust the optimization objective as follows:

$$\arg\min_\delta\{\|g(r\otimes(x+\delta)), g(x_t)\|_2 + \alpha \cdot Psy(x,\delta)\}, \tag{14}$$

where r represents the RIR sampled from the set, and $\otimes$ represents the convolution operation.

5 Evaluation

5.1 Ethical Considerations

Throughout the experimental process, we rigorously follow the ethical guidelines below:

Strictly Controlled Experiments. In our tests with commercial APIs, we only used publicly available datasets, ensuring that no data concerning personal privacy or commercial secrets were involved. Furthermore, upon the completion of successful attacks, no unauthorized actions were executed. All devices used in our experiments were under the personal ownership of the research team.

[3] Here, we use a modified Hann function with a window size of 2048 and a hop size of 512 to segment an audio file into multiple frames.

Table 2: Details of the commercial speaker recognition systems.

SRS	Task	Text	Type	Over-the-Air	Query	Return Results	FRR	FAR	FE⋆
Google [2]	SV	Independent	API	×	√	D	1/10	0/10	MFCC
Unisound [47]	SI	Independent	API	×	√	D‡+S≀	0/10	0/10	X-Vector
iFlytyke [48]	SI	Independent	API	×	√	D+S	0/10	0/10	D-Vector
Talentedsoft [49]	SI	Independent	API	×	√	D+S	0/10	1/10	X-Vector
Apple Siri [50]	SV	Dependent	Device	√	×	Action†	0/10	1/10	X-Vector
Tmall Genie [51]	SI	Dependent	Device	√	×	Action	0/10	0/10	X-Vector
Millet Xiaoai [52]	SI	Dependent	Device	√	×	Action	0/10	0/10	X-Vector
Google Assistant [53]	SV	Dependent	Device	√	×	Action	0/10	0/10	X-Vector
Samsung Bixby [54]	SV	Dependent	Device	√	×	Action	0/10	0/10	X-Vector
Huawei Xiaoyi [55]	SV	Dependent	Device	√	×	Action	0/10	0/10	X-Vector
OPPO Breeno [56]	SV	Dependent	Device	√	×	Action	0/10	0/10	X-Vector

‡: "D" means that the SRS returns the final decision, e.g., accept or reject. ≀: "S" means that the SRS returns the confidence score. †: "Action" means that the SRS adopts the wake-up mechanism. ⋆: "FE" means the feature extractor used in our experiments.

Responsible Disclosure. We have informed all companies involved in our experiments about this potential vulnerability through official reporting channels or direct emails. We provided detailed explanations and solutions for mitigating this risk.

5.2 Experiment Setup

Target Systems. To evaluate the performance of our attack, we select several popular and representative commercial SRSs, including four online SR APIs and seven commercial voice assistants, as the targets (shown in Table 2).[4] Table 9 in Appendix A provides detailed information about the voice assistants.

Datasets. We utilize two widely-used datasets: VoxCeleb V1 (Vox) [58] and CMU_arctic (Cmu) [59], to evaluate the performance of VoxCloak. In addition, we recruited 20 volunteers for voice data collection and named the constructed dataset "WakeUp" because the purpose of the recorded voice commands, like "Hey Siri" and "Hey Google", are designed to activate the voice assistants. We randomly selected a pair of volunteers from all participants to serve as the attacker and victim in each experiment. For the experiments about commercial APIs, ten pairs of volunteers were selected. For voice assistants, 15 pairs were selected. Details about the datasets are provided in Appendix A.1.

Metrics. We utilize the following three metrics: (1) *Attack Success Rate (ASR)*: This measures the proportion of AEs that can successfully deceive the target SRSs. A higher ASR indicates that the attack is more effective. (2) *Signal-to-noise Ratio (SNR)*: This quantifies the level of adversarial perturbation, calculated using the formula $SNR = 10\log_{10}\frac{\mathcal{P}_x}{\mathcal{P}_\delta}$, where $\mathcal{P}_x$ and $\mathcal{P}_\delta$ indicate the average power of the signal and the perturbation, respectively. A higher SNR means a lower noise level, i.e., the perturbation is less perceptible. (3) *Number of Queries (NoQ)*: This refers to the number of queries required for generating an effective AE. Fewer queries imply that the attack is more efficient.

Moreover, we use the false rejection rate (FRR) and the false acceptance rate (FAR) to measure the function of the SRSs. The FRR is the proportion of audio samples from legitimate users that are erroneously rejected, and the FAR is the proportion of audio samples from unauthorized users that are incorrectly accepted.

Benchmarks. We compare VoxCloak with three state-of-the-art attacks: two query-based black-box attacks, namely FakeBob [7] and Occam [6], along with a transfer-based attack, VMask [12]. In addition, we established a baseline comparison, termed Vanilla, which involves the samples created by directly overlaying the audio waveforms of the attackers onto those of the legitimate users.

5.3 Evaluation on Online SR APIs

Effectiveness. The results of VoxCloak and baselines are presented in Table 3. Overall, AEs generated by VoxCloak achieved nearly a 100% ASR with an average SNR as high as 21.47 dB, marking a considerable improvement over existing works. Specifically, in comparison to decision-based attacks, VoxCloak outperforms FakeBob [7] with an ASR of 99.96% and an SNR of 8.04 dB and outperforms Occam [6] with a close ASR and an SNR of 7.86 dB. Compared to the transfer-based attack, VoxCloak outperforms VMask [12] with an ASR of 76.25% and an SNR of 13.46 dB. Figure 7 in Appendix B.3 shows the waveforms and spectrograms of both the original audio examples and the AEs produced by VoxCloak and various baselines. We can observe that the adversarial audio examples generated by VoxCloak more closely resemble the original audio compared to those from the baselines. This indicates that the AEs generated by VoxCloak are closer to normal examples, thereby exhibiting better imperceptibility.

Efficiency. As shown in Figure 4, the ASR of AEs related to

[4] We have surveyed more commercial SRSs, as detailed in Table 10 in the Appendix A. Unfortunately, several of them do not provide the interface for personal users. For example, Microsoft Azure only provided the service to their partners and rejected our request to use the service.

Table 3: Experimental results on commercial online SR APIs.

SRS	Dataset	Vanilla*		VMask*		FakeBob			Occam			VoxCloak		
		ASR	SNR	ASR	SNR	ASR	SNR	NoQ#	ASR	SNR	NoQ#	ASR	SNR	NoQ
Google	Cmu	0/10	0.01	3/10	9.15	0/10	N/A	3000	**10/10**	14.15	10,000	9/10	**22.45**	∼**300**
	Vox	0/10	0.02	1/10	6.14	1/10	15.3	750	**10/10**	13.25	10,000	**10/10**	**21.73**	
Unisound	Cmu	0/10	0.01	4/10	9.32	0/10	N/A	3000	**10/10**	14.31	10,000	**10/10**	**19.28**	∼**300**
	Vox	0/10	0.02	1/10	6.66	1/10	13.4	980	**10/10**	14.59	10,000	**10/10**	**19.47**	
iFlytyke	Cmu	0/10	0.01	2/10	9.98	0/10	N/A	3000	**10/10**	13.78	10,000	**10/10**	**23.82**	∼**300**
	Vox	1/10	0.02	2/10	5.95	1/10	11.6	2540	**10/10**	14.47	10,000	**10/10**	**21.26**	
TalentedSoft	Cmu	1/10	0.01	3/10	9.74	0/10	N/A	3000	**10/10**	11.25	10,000	**10/10**	**23.31**	∼**300**
	Vox	2/10	0.02	2/10	6.18	0/10	N/A	3000	**10/10**	12.96	10,000	**10/10**	**21.26**	

∗: The method does not query the target system. #: The value of NoQ is the average number of queries for generating one AE. N/A indicates "not available". Since FakeBob [7] produces no effective AE, its SNR cannot be calculated. VoxCloak only requires about 300 queries to infer the target FE and does not require further queries to generate AEs. We set the max NoQ for FakeBob [7] and Occam [6] as 3000 and 10,000, respectively.

a specific FE becomes significantly higher than others after running the genetic algorithm for 20 generations, indicating a high similarity between this FE and the target FE. We initially selected four FEs as candidates and created 25 AEs for each candidate FE, requiring one query to the target system per example. This equates to a total of 100 queries in the initial phase. Subsequently, in each generation of the genetic algorithm, we set the mutation probability to 10%, selecting 10 AEs for querying per generation. After running for 20 generations, this results in an additional 200 queries. Therefore, the entire process involves approximately 300 queries to the target system. This NoQ is significantly lower than prior decision-based attacks like Occam, which requires up to 10,000 queries for generating one AE, and Fakebob, which requires an average NoQ of 1423 for generating one AE. Furthermore, for Fakebob, we observe that it is challenging to generate a successful AE even after 3000 query attempts. We note that our results for FakeBob [7] differ from those in the original publication [7] but are consistent with results obtained by Zheng *et al.* [6]. This discrepancy is likely attributable to updates in the APIs.

In addition, as long as the target system does not update its extractor, VoxCloak can generate any number of AEs without additional queries. This not only lowers the operational costs of VoxCloak but also increases its ability to evade detection by defense mechanisms that analyze query patterns [60].

Generalization. We have also evaluated the performance of VoxCloak across three different SR tasks: CSI, SV, and OSI. The results in Table 4 demonstrate that VoxCloak achieves a near 100% average ASR and a 20.89 dB average SNR on CSI tasks. For SV tasks, it accomplishes an average ASR of 86.67% and an average SNR of 21.77 dB. However, for OSI tasks, there is a slight decline, with the average ASR reducing to 78.33% and the average SNR to 21.18 dB. We consider that this decrease may be attributed to the threshold settings in the OSI tasks, where a higher threshold may prevent some weaker AEs from misleading the decision-making process.

Table 4: Experimental results on different SR tasks.

SRS	Dateset	CSI		SV		OSI	
		ASR	SNR	ASR	SNR	ASR	SNR
Unisound	Cmu	10/10	19.28	8/10	22.54	8/10	22.54
	Vox	10/10	19.48	7/10	19.14	7/10	19.14
iFlytyke	Cmu	10/10	23.82	7/10	23.23	7/10	23.23
	Vox	10/10	19.48	10/10	21.26	7/10	19.44
TalentedSoft	Cmu	10/10	23.82	10/10	23.21	10/10	23.31
	Vox	10/10	19.48	10/10	21.26	8/10	19.44

Nevertheless, the high ASR of VoxCloak across the three tasks still reveals the potential security risks.

5.4 Evaluation on Voice Assistants

We evaluated the performance of VoxCloak using seven commercial smart devices equipped with voice assistants. Details of the experimental setup are provided in Figure 5 (a). All experiments took place in a closed indoor environment, with the loudspeaker and the voice-controlled device positioned about 0.8 meters apart. We used two types of devices, the JBL Clip 3 and the ThinkPad X1 8th, to play the AEs. Notably, neither VoxCloak nor the baseline methods are allowed to query the target system. An attack is considered successful if it manages to activate the voice assistants with the AE in no more than two attempts.

As it is unable to query the target system when attacking voice assistants in the physical world, for FakeBob [7] and Occam [6], we first adapted their approaches to generate AEs against the Unisound API, an online SR service that allows free queries. The generated examples were then used to attack the target system. For VoxCloak, we consistently used X-Vector as the FE for generating AEs. We choose X-Vector for two reasons. (1) Experimental results in Appendix B.2 show that AEs generated based on X-Vector possess a degree

Table 5: Experimental results on voice assistants.

SRS	Loudspeaker	Vanilla		Vmask		FakeBob		Occam		VoxCloak	
		ASR	SNR	ASR	SNR	ASR	SNR	ASR	SNR	ASR	SNR
Apple Siri	JBL Clip3	6/15	0.01	3/15	9.09	2/15	2.31	2/15	14.14	**12/15**	**17.63**
	ThinkPad	4/15	0.02	3/15	9.09	1/15	4.13	2/15	13.14	**12/15**	**17.63**
Tmall Genie	JBL Clip3	6/15	0.01	3/15	8.38	4/15	4.71	3/15	2.78	**14/15**	**21.71**
	ThinkPad	6/15	0.01	2/15	8.85	3/15	5.29	3/15	2.78	**13/15**	**22.02**
Millet Xiaoai	JBL Clip3	5/15	0.01	5/15	8.62	0/15	N/A	3/15	2.36	**13/15**	**20.89**
	ThinkPad	5/15	0.01	4/15	10.06	0/15	N/A	2/15	2.78	**11/15**	**21.26**
Google Assistant	JBL Clip3	6/15	0.01	4/15	8.38	3/15	4.11	2/15	11.89	**11/15**	**22.84**
	ThinkPad	5/15	0.01	4/15	8.38	3/15	4.11	3/15	11.75	**12/15**	**23.21**
Samsung Bixby	JBL Clip3	6/15	0.01	1/15	7.35	1/15	5.21	1/15	12.68	**7/15**	**22.45**
	ThinkPad	5/15	0.01	2/15	6.83	1/15	5.21	2/15	13.02	**6/15**	**22.57**
Huawei Xiaoyi	JBL Clip3	5/15	0.01	3/15	9.13	2/15	3.13	2/15	12.25	**11/15**	**20.89**
	ThinkPad	4/15	0.01	2/15	9.45	2/15	3.13	2/15	12.25	**10/15**	**20.43**
OPPO Breeno	JBL Clip3	6/15	0.01	2/15	7.82	2/15	2.87	4/15	13.26	**8/15**	**23.48**
	ThinkPad	6/15	0.01	1/15	7.82	1/15	3.91	3/15	12.87	**7/15**	**22.27**

N/A denotes "not available". All attacks are conducted in the physical world, and the attacker does not query the SRSs. For FakeBob [7] and Occam, we find that the AEs with higher SNR cannot attack SRSs, so we try to find a successful example by lowering the SNR.

Table 6: Results under different attack distances (meter).

Distance		0.25	0.5	1	2	4	8
Apple Siri	FRR	0/15	0/15	0/15	0/15	1/15	2/15
	FAR	0/15	1/15	1/15	1/15	1/15	0/15
	ASR	14/15	14/15	12/15	11/15	7/15	2/15
Tmall Genie	FRR	0/15	0/15	1/15	3/15	5/15	7/15
	FAR	0/15	0/15	0/15	0/15	0/15	0/15
	ASR	14/15	14/15	12/15	5/15	2/15	1/15
Millet Xiaoai	FRR	0/15	0/15	0/15	2/15	6/15	6/15
	FAR	0/15	0/15	0/15	0/15	0/15	9/15
	ASR	13/15	13/15	12/15	46.67	1/15	0/15

of transferability to other FEs. (2) X-Vector has better performance than other common FEs [16], suggesting it is more likely to be used in target systems.

Effectiveness. The results in Table 5 demonstrate that VoxCloak outperforms baseline methods in both ASR and SNR across all tested voice assistants. For instance, when targeting Apple Siri with the JBL Clip3, VoxCloak achieves an 80% ASR and an SNR of 17.63 dB. In contrast, neither FakeBob [7] nor Occam [6] could effectively generate AEs capable of a successful attack with a comparable SNR. However, the ASRs on Samsung Bixby and OPPO Breeno are markedly lower than those of other systems, implying substantial differences between the FEs employed by these two systems and X-Vector. Nevertheless, the performance of VoxCloak on these two systems still outperforms the baselines.

Robustness. To evaluate the effect of attack distance on performance, we conducted tests with varying distances between the speaker (JBL Clip3) and voice-controlled devices, specifically at 0.25, 0.5, 1, 2, 4, and 8 meters. The results, as shown in Table 6, align with the intuition: the ASR decreases while the distance increases due to the signal attenuation caused by air transmission and environmental interferences. For example, when targeting Apple Siri, the ASR at an attack distance of 0.5 meters is 93.33%; however, it decreases to 73.33% at 2 meters and further drops to just 13.33% at 8 meters. Despite this decrease, VoxCloak consistently outperforms baseline methods in the physical world.

5.5 Evaluation on Different Acoustic Environments

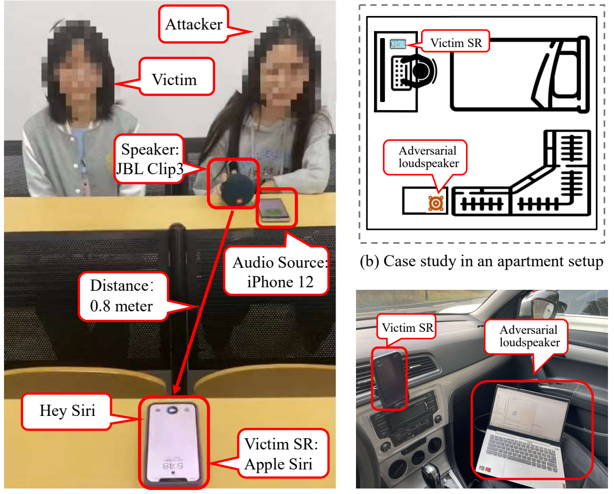
In this subsection, we evaluate the effectiveness of VoxCloak across various acoustic environments, including two real-world environments and four simulated environments augmented with background noise. A JBL Clip3 speaker was consistently used for all experiments. We detail the test environments as follows.

Apartment Scenario. The experiments are conducted in an apartment scenario (Figure 5 (b)). The dimensions of the apartment are approximately 7.5 meters × 3 meters, with the attack distance set at 0.5 meters. The primary environmental noises originated from a computer fan and the acoustics of the room, due to wall absorption and reverberation. These noises could potentially impact the audio received by the microphone compared to that emitted by the speakers.

Table 7: Results under different acoustic environments.

Environment		White (30dB)	White (45dB)	White (50dB)	White (60dB)	White (65dB)	Bus (60dB)	AirCon. (60dB)	Neighbor (60dB)	Office (64dB)	InsideCar (75dB)
Apple Siri	FRR	0/15	0/15	0/15	1/15	2/15	2/15	7/15	2/15	2/15	1/15
	FAR	2/15	3/15	2/15	2/15	2/15	3/15	1/15	2/15	2/15	2/15
	ASR	15/15	15/15	15/15	12/15	11/15	13/15	6/15	8/15	10/15	12/15
Tmall Genie	FRR	0/15	0/15	0/15	0/15	0/15	0/15	3/15	1/15	0/15	0/15
	FAR	1/15	0/15	0/15	1/15	1/15	3/15	2/15	0/15	2/15	1/15
	ASR	15/15	15/15	15/15	13/15	10/15	7/15	9/15	6/15	12/15	13/15
Millet Xiaoai	FRR	0/15	0/15	0/15	0/15	3/15	0/15	4/15	3/15	1/15	0/15
	FAR	1/15	1/15	1/15	0/15	1/15	1/15	1/15	1/15	1/15	1/15
	ASR	15/15	15/15	13/15	7/15	7/15	7/15	6/15	9/15	9/15	10/15

“White” indicates white noise.“Bus” means the noise from bus noise class. “AirCon.” indicates the noise from the air conditioning Class. “Neighbor” means the noise from the neighbor speaking noise class.



(a) Case study in an office setup

(b) Case study in an apartment setup

(c) Case study in an inside-vehicle setup

Figure 5: Demonstration of acoustic environments.

Inside-car Scenario. The experiments are conducted in a Volkswagen Gran Lavida (interior depicted in Figure 5 (c)). The AEs were played while the car was parked and the engine was running. Hence, the environmental noises mainly stemmed from the engine and the reverberation. The speaker was placed about 0.2 meters from the target devices due to spatial limitations.

Simulated Scenarios. By utilizing the Microsoft Scalable Noisy Speech Dataset, we simulated environments with four types of noises: white noise, bus noise, air conditioning noise (AirCon.), and noise from neighboring conversations. The white noise volume is varied between 30 and 65 dB. In the simulated office environment, one JBL Clip3 is used to broadcast the AEs and another to play the noise. Both are positioned around 0.5 meters from the target device.

The results are presented in Table 7. VoxCloak achieves high ASRs across various acoustic environments. For Apple Siri, TMall Genie, and Millet Xiaoai, the average ASRs in these ten environments were 78.00%, 76.67%, and 65.33%, respectively. Note that even when the white noise reached 60 dB, close to the volume of the AEs (approximately 60 dB), our attack still maintains an ASR of 69.63%. Besides, we observe that some voice assistants could be activated by clean voices from unauthorized users. We attribute this to two potential causes: (1) the voiceprint features of the unauthorized user might be similar to those of an authorized user, and (2) the underlying SRS may have a relatively low threshold setting. While a lower threshold can improve the user experience by reducing authentication failures for authorized users, it also poses potential security risks.

5.6 Evaluation on Human Perception

SNR is a common metric for quantifying audio adversarial perturbation. However, it does not fully reflect the imperceptibility of AEs to the human ear. Two AEs with identical SNRs might be perceived differently because human auditory perception is impacted not just by sound intensity but also by frequency. In this subsection, we evaluate the imperceptibility of VoxCloak in terms of human perception.

5.6.1 Subjective Evaluation

For our human studies, we recruit 31 volunteers aged between 18 and 25 with normal hearing, including 18 males and 13 females.[5]

Study 1. The objective of this study is to assess whether volunteers could correctly discern if two audio clips were spoken by the same person. Volunteers are asked to listen to multiple audio clips and respond with “yes”, “no”, or “uncertain”. We present them with 20 audio pairs in the experimental group, each comprising one AE and one normal speech clip. The control group consists of two sets of audio triples, each con-

[5] The ethical clearance for this work was obtained from our institution.

taining a normal clip as a reference, a clip from the same speaker as the reference, and a clip from a different speaker.

In the control group, our results show that 77.42% of the audio pairs from the same speaker are correctly identified, and 74.19% of pairs from different speakers were correctly identified. However, in the experimental group, 78.55% of the audio pairs are incorrectly identified as being from the same speaker. This demonstrates that the AEs generated by VoxCloak can effectively deceive humans, indicating a high level of auditory deception.

Study 2. The objective of this study is to assess whether volunteers could correctly recognize the identity of the speaker. Volunteers are first provided with an audio clip as a reference. Then, they are presented with two additional clips and asked to identify the one spoken by the same speaker with the reference clip. We conduct 20 experiments as the experimental group, each using an AE as the reference, and ask volunteers to distinguish between normal voices from the target speaker and the original speaker used to generate the AEs. The control group involved two sets of audio triples, each containing a normal clip as a reference, a clip from the same speaker, and a clip from a different speaker.

The results show that in the control group, 93.55% of the audio clips are correctly identified. However, in the experimental group, only 4.68% of the clips are identified as belonging to the target speaker, with 89.07% identified as the original speaker and 6.45% as "uncertain". This implies that volunteers generally associate the AEs more with their original speakers, highlighting the deception of VoxCloak.

Study 3. The objective of this study is to evaluate the human perception of the audio quality of AEs. Volunteers are initially provided with normal audio clips as a reference for quality. They are then asked to evaluate and rank 20 sets of AEs, each including clips generated by VMask [12], FakeBob [7], Occam [6], and VoxCloak.

The results show that more than 80% of the volunteers rate the AEs generated by VoxCloak as having the highest quality, with 10% ranking them second and only 5% ranking them third or fourth. These results indicate that the AEs generated by VoxCloak are acoustically similar to normal samples, thereby demonstrating the imperceptibility of VoxCloak.

5.6.2 Objective Evaluation

We also utilize the Perceptual Evaluation of Speech Quality (PESQ) [61], as recommended by the International Telecommunication Union, for an objective assessment of the imperceptibility of VoxCloak. The PESQ score ranges from 1 to 5, with 1 indicating poor audio quality and 5 indicating excellent audio quality. The score is impacted by various factors, such as audio sharpness, background noise, variable latency, and audio interference.

We evaluate 80 AEs generated by VoxCloak against four online SR APIs, achieving an average PESQ score of 2.93. This performance outperforms that of several baselines, i.e., Occam [6] (2.19), FakeBob [7] (2.81), and VMask [12] (2.35), where each method is also evaluated with 80 AEs against the same four SR APIs.

6 Related Work

6.1 Adversarial Example Attacks

Researchers initially explored AE attacks against SR systems in the white-box setting. Li *et al.* [20] employed the Fast Gradient Sign Method to generate AEs against SV systems. Gong *et al.* [10] proposed an attack targeting CSI systems by perturbing the raw audio recording. NRI-FGSM generates subsecond-level universal adversarial perturbations, which can add adversarial perturbations at any position in the input stream to improve the robustness of AEs [21].

However, since these attacks rely on a strong assumption that the attacker has complete knowledge of the details of the target system, researchers have increasingly focused their attention on black-box attacks. VMask [12] uses a local model to guide the optimization of AEs against target SV systems. This approach shares some similarities with VoxCloak. However, owing to the architecture of FEs being significantly simpler than that of SRSs, attackers using VoxCloak have smaller differences between the results obtained locally and those in the target system, thereby achieving higher ASRs.

Abdullah *et al.* [22] proposed the first zero-query black-box attack. They leverage the common feature extraction process that converts the captured audio into model features and modifies the recorded audio by signal decomposition and reconstruction. This approach is highly effective for untargeted attacks, but it encounters limitations in implementing effective targeted attacks, as well as in attacking SV tasks. FakeBob [7] also achieves targeted black-box attacks against SV, CSI, and OSI tasks as VoxCloak does. However, it cannot attack most commercial SR systems because of the need for predicted probabilities/scores. Occam [6] has overcome the limitation of FakeBob, achieving a 100% ASR with an SNR of 14.23 dB, even if the target system does not disclose the scores. In comparison, VoxCloak achieves a similar ASR and higher SNR as Occam with the additional advantage of the one-time and extremely minimal number of queries required. Some recent works try to improve the transferability of the attack by stabilizing the direction of the gradient to avoid overfitting [23] or finding better loss functions for transferability [43]. However, they still only achieve a limited ASR of about 50%. In summary, all prior AE attacks against SRSs exhibit several limitations, such as the capability for only untargeted attacks, reliance on confidence scores, excessive queries, and unsatisfactory performance.

Table 8: Experimental results in the presence of commonly-used defenses.

SRS	Dataset	Downsampling			Low-pass Filtering			Quantization			MP3C
		8000	11025	12000	2Khz	4Khz	8Khz	256	512	1024	
iFytyke	Cmu	1/10	4/10	7/10	0/10	4/10	10/10	7/10	5/10	1/10	10/10
	Vox	1/10	6/10	9/10	2/10	6/10	10/10	6/10	3/10	2/10	10/10
Unisound	Cmu	0/10	5/10	7/10	4/10	7 /10	10/10	5/10	3/10	0/10	10/10
	Vox	1/10	6/10	7/10	6/10	8/10	10/10	7/10	6/10	2 /10	10/10
Talentedsoft	Cmu	3/10	5/10	5/10	7/10	8/10	10/10	6/10	4/10	1/10	8/10
	Vox	5/10	8/10	8/10	10/10	10/10	10/10	8/10	5/10	3/10	9/10
Benign Input	Cmu	7/10	9/10	10/10	7/10	10/10	10/10	10/10	8/10	6/10	10/10
	Vox	8/10	10/10	10/10	7/10	10/10	10/10	10/10	9/10	6/10	10/10

The results of "Benign Input" are the average accuracy from three online SRSs with respect to benign inputs.

6.2 Other Types of Attacks

Sensor spoofing attacks are another category of attacks that can deceive SRSs. Wu *et al.* [62] conducted a comprehensive study of various sensor spoofing attacks in SV tasks, including replay, impersonation, voice synthesis, and voice conversion attacks. These attacks aim to generate or mimic the voices of the target speakers. However, these attacks require playing voices not belonging to the victim, they are more readily detected by the victim, thereby offering lower stealth capability than AE attacks. Besides, some studies [13, 63] focus on hidden voice attacks. These attacks modify a voice in a way that, while still recognizable by the targeted SRSs, becomes incomprehensible to human listeners (usually sounds like noise). In contrast to AE attacks, hidden voice attacks are relatively easier to execute. Yet, they are also more susceptible to defense and detection, as humans are likely to notice and identify noise-like sounds as anomalies.

7 Discussion

7.1 Defenses

We evaluate the effectiveness of four common defenses against VoxCloak. The results are presented in Table 8.

Downsampling. Downsampling, a method used to lower the sampling rate of audio signals, can remove high-frequency components in adversarial perturbations, thereby reducing the effectiveness of AEs [6, 21]. Experimental results indicate that when AEs generated by VoxCloak are downsampled to 8 kHz and then upsampled back to 16 kHz, the ASR decreases to 18%, accompanied by about a 25% decrease in model accuracy. If the downsampling rate is set to 12 kHz (the minimum rate that does not affect the model accuracy), VoxCloak still maintains a 71% ASR. Moreover, if attackers are aware of the down/upsampling rates of the target system, they can add corresponding constraints in the optimization process to invalidate the defense.

Low-pass Filtering. Low-pass filtering is another common defense that removes the high-frequency components of audio AEs [6, 21]. However, results show that VoxCloak can still maintain a high ASR after employing low-pass filtering. For example, setting the filter frequency to 8 kHz results in an ASR of 100%. When the filter frequency is adjusted to 4 kHz, the ASR remains at 71%. However, reducing the filtering frequency to 2 kHz leads to a drop in ASR to 48%. These results indicate that the frequency range between 2 kHz and 4 kHz in the adversarial perturbations is important for the success of VoxCloak.

Quantization. It represents continuous analog audio signals with a discrete set of values. It can defend against AE attacks because it may delete some of the information within the perturbations [64]. In our experiments, we set the quantization interval to 256, resulting in an average ASR of 65%. When increasing the quantization interval to 1024, the ASR of our attack reduces to 15%. In addition, the accuracy of the SR model for clean examples also reduces to 60%. This reduction is mainly due to the removal of important information from both benign and AEs during the quantization process.

MP3 Copmression (MP3C). MP3 is a lossy compression format for audio files. Intuitively, the information loss caused by compression might make the AE attack ineffective [7, 21]. However, experimental results indicate that VoxCloak still maintains a high ASR, up to 95%, even after undergoing a 10-to-1 compression ratio, which reduces the audio file to one-tenth of its original size. This result implies that the information lost during MP3 compression is not the major component of the adversarial perturbations.

7.2 Limitations

Compared to previous works, VoxCloak marks significant advancements in terms of ASR, required background knowledge, and overall cost of the attack. Yet, VoxCloak does have some limitations. The first is its robustness in real-world environments. Although VoxCloak is designed to minimize the impact of environmental noise, excessive environmental noise, can still render the attack ineffective, as shown in Table 7. The second is the size of the candidate FE set. While our approach is effective against the common FEs used in many commercial APIs, it may fall short when SRSs use unknown or proprietary FEs. This is observed in our unsuccessful attempts to compromise systems like OPPO Breeno and Samsung Bixby. In these cases, we can only rely on the transferability of the attack. Note that due to the potential similarities in the feature spaces of different FEs, the AEs generated may also exhibit a certain degree of transferability. An ASR of nearly 50% for the two SRSs further illustrates this point.

8 Conclusion

In this paper, we proposed VoxCloak, a new black-box AE attack against commercial SRSs. VoxCloak can locally generate an arbitrary number of AEs after using a minimal number of queries to the target system to infer the used FE. Our extensive experiments across various popular commercial SRSs and tasks have shown the effectiveness of VoxCloak. Compared to existing query-based attacks, VoxCloak requires 30 times fewer queries to achieve a comparable ASR and a higher SNR, and it also surpasses the performance of existing transfer-based attacks in both metrics. Additionally, VoxCloak is available in the physical world, successfully misleading voice assistants at distances up to 4 meters. These results reveal the security concerns related to the reuse of FEs.

Acknowledgments

We thank the anonymous reviewers and our shepherd for their helpful and valuable feedback. This work was partially supported by the NSFC under Grants U20B2049, U21B2018, 62302344, 62132011, 62161160337, 62376210, 62006181, and U20A20177, HK RGC under Grants R6021-20F and N_CityU139/21, Shaanxi Province Key Industry Innovation Program 2023-ZDLGY-38 and 2021ZDLGY01-02, and Fundamental Research Funds for the Central Universities under grant xtr052023004 and xtr022019002.

References

[1] Sargur N Srihari, Chen Huang, Harish Srinivasan, and Vivek Shah. Biometric and forensic aspects of digital document processing. *Digital Document Processing: Major Directions and Recent Advances*, pages 379–405, 2007.

[2] Google Speaker ID. https://cloud.google.com/speaker-id?hl=zh-cn#section-5.

[3] Satish T Bhosale and BS Sawant. Security in e-banking via cardless biometric atms. *International Journal of Advanced Technology & Engineering Research*, pages 457–462, 2012.

[4] Yun Lei, Nicolas Scheffer, Luciana Ferrer, and Mitchell McLaren. A novel scheme for speaker recognition using a phonetically-aware deep neural network. In *Proc. of IEEE ICASSP*, pages 1695–1699, 2014.

[5] David Snyder, Daniel Garcia-Romero, Daniel Povey, and Sanjeev Khudanpur. Deep neural network embeddings for text-independent speaker verification. In *Proc. of Interspeech*, pages 999–1003, 2017.

[6] Baolin Zheng, Peipei Jiang, Qian Wang, Qi Li, Chao Shen, Cong Wang, Yunjie Ge, Qingyang Teng, and Shenyi Zhang. Black-box adversarial attacks on commercial speech platforms with minimal information. In *Proc. of ACM CCS*, pages 86–107, 2021.

[7] Guangke Chen, Sen Chen, Lingling Fan, Xiaoning Du, Zhe Zhao, Fu Song, and Yang Liu. Who is real bob? Adversarial attacks on speaker recognition systems. In *Proc. of IEEE S&P*, pages 694–711, 2021.

[8] Felix Kreuk, Yossi Adi, Moustapha Cisse, and Joseph Keshet. Fooling end-to-end speaker verification with adversarial examples. In *Proc. of IEEE ICASSP*, pages 1962–1966, 2018.

[9] Yi Xie, Cong Shi, Zhuohang Li, Jian Liu, Yingying Chen, and Bo Yuan. Real-time, universal, and robust adversarial attacks against speaker recognition systems. In *Proc. of IEEE ICASSP*, pages 1738–1742, 2020.

[10] Yuan Gong and Christian Poellabauer. Crafting adversarial examples for speech paralinguistics applications. *ArXiv Preprint*, 2017.

[11] Tianyu Du, Shouling Ji, Jinfeng Li, Qinchen Gu, Ting Wang, and Raheem Beyah. Sirenattack: Generating adversarial audio for end-to-end acoustic systems. In *Proc. of ACM CCS*, pages 357–369, 2020.

[12] Lei Zhang, Yan Meng, Jiahao Yu, Chong Xiang, Brandon Falk, and Haojin Zhu. Voiceprint mimicry attack towards speaker verification system in smart home. In *Proc. of IEEE INFOCOM*, pages 377–386, 2020.

[13] Hadi Abdullah, Washington Garcia, Christian Peeters, Patrick Traynor, Kevin RB Butler, and Joseph Wilson. Practical hidden voice attacks against speech and speaker recognition systems. In *Proc. of NDSS*, 2019.

[14] Steven Davis and Paul Mermelstein. Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, pages 357–366, 1980.

[15] Li Wan, Quan Wang, Alan Papir, and Ignacio Lopez Moreno. Generalized end-to-end loss for speaker verification. In *Proc. of IEEE ICASSP*, pages 4879–4883, 2018.

[16] David Snyder, Daniel Garcia-Romero, Gregory Sell, Daniel Povey, and Sanjeev Khudanpur. X-vectors: Robust dnn embeddings for speaker recognition. In *Proc. of IEEE ICASSP*, pages 5329–5333, 2018.

[17] Sreenivas Sremath Tirumala, Seyed Reza Shahamiri, Abhimanyu Singh Garhwal, and Ruili Wang. Speaker identification features extraction methods: A systematic review. *Expert Systems with Applications*, pages 250–271, 2017.

[18] Binghui Wang and Neil Zhenqiang Gong. Stealing hyperparameters in machine learning. In *Proc. of IEEE S&P*, pages 36–52, 2018.

[19] Florian Tramèr, Fan Zhang, Ari Juels, Michael K Reiter, and Thomas Ristenpart. Stealing machine learning models via prediction apis. In *Proc. of USENIX Security*, pages 601–618, 2016.

[20] Xu Li, Jinghua Zhong, Xixin Wu, Jianwei Yu, Xunying Liu, and Helen Meng. Adversarial attacks on gmm i-vector based speaker verification systems. In *Proc. of IEEE ICASSP*, pages 6579–6583, 2020.

[21] Zhuohang Li, Yi Wu, Jian Liu, Yingying Chen, and Bo Yuan. Advpulse: Universal, synchronization-free, and targeted audio adversarial attacks via subsecond perturbations. In *Proc. of ACM CCS*, pages 1121–1134, 2020.

[22] Hadi Abdullah, Muhammad Sajidur Rahman, Washington Garcia, Kevin Warren, Anurag Swarnim Yadav, Tom Shrimpton, and Patrick Traynor. Hear " no evil", see " kenansville"*: Efficient and transferable black-box attacks on speech recognition and voice identification systems. In *Proc. of IEEE S&P*, pages 712–729, 2021.

[23] Hao Tan, Junjian Zhang, Huan Zhang, Le Wang, Yaguan Qian, and Zhaoquan Gu. Nri-fgsm: An efficient transferable adversarial attack method for speaker recognition system. In *Proc. of Interspeech*, pages 18–22, 2022.

[24] Zhongxin Bai and Xiao-Lei Zhang. Speaker recognition based on deep learning: An overview. *Neural Networks*, pages 65–99, 2021.

[25] Stanley Smith Stevens, John Volkmann, and Edwin Broomell Newman. A scale for the measurement of the psychological magnitude pitch. *The Journal of The Acoustical Society of America*, pages 185–190, 1937.

[26] Douglas A Reynolds. Speaker identification and verification using gaussian mixture speaker models. *Speech Communication*, pages 91–108, 1995.

[27] Frédéric Bimbot, Jean-François Bonastre, Corinne Fredouille, Guillaume Gravier, Ivan Magrin-Chagnolleau, Sylvain Meignier, Teva Merlin, Javier Ortega-García, Dijana Petrovska-Delacrétaz, and Douglas A Reynolds. A tutorial on text-independent speaker verification. *EURASIP Journal on Advances in Signal Processing*, pages 1–22, 2004.

[28] AM Ariyaeeinia, J Fortuna, P Sivakumaran, and Aa Malegaonkar. Verification effectiveness in open-set speaker identification. *IEE Proceedings-Vision, Image and Signal Processing*, pages 618–624, 2006.

[29] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In *Proc. of ICLR*, 2015.

[30] Chaowei Xiao, Bo Li, Jun-Yan Zhu, Warren He, Mingyan Liu, and Dawn Song. Generating adversarial examples with adversarial networks. In *Proc. of IJCAI*, pages 3905–3911, 2018.

[31] Dan Hendrycks, Kevin Zhao, Steven Basart, Jacob Steinhardt, and Dawn Song. Natural adversarial examples. In *Proc. of IEEE CVPR*, pages 15262–15271, 2021.

[32] Ekin D Cubuk, Barret Zoph, Samuel S Schoenholz, and Quoc V Le. Intriguing properties of adversarial examples. In *Proc. of ICLR Workshop*, 2018.

[33] Wieland Brendel, Jonas Rauber, and Matthias Bethge. Decision-based adversarial attacks: Reliable attacks against black-box machine learning models. In *Proc. of ICLR*, 2018.

[34] Huichen Li, Xiaojun Xu, Xiaolu Zhang, Shuang Yang, and Bo Li. Qeba: Query-efficient boundary-based black-box attack. In *Proc. of IEEE CVPR*, pages 1221–1230, 2020.

[35] Francesco Croce and Matthias Hein. Minimally distorted adversarial examples with a fast adaptive boundary attack. In *Proc. of ICML*, pages 2196–2205, 2020.

[36] Ningping Mou, Baolin Zheng, Qian Wang, Yunjie Ge, and Binqing Guo. A few seconds can change everything: Fast decision-based attacks against dnns. In *Proc. of IJCAI*, pages 3342–3350, 2022.

[37] Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks. In *Proc. of IEEE S&P*, pages 39–57, 2017.

[38] Jacob M. Springer, Melanie Mitchell, and Garrett T. Kenyon. A little robustness goes a long way: Leveraging robust features for targeted transfer attacks. In *Proc. of NeurIPS*, pages 9759–9773.

[39] Seong Joon Oh, Bernt Schiele, and Mario Fritz. Towards reverse-engineering black-box neural networks. *Explainable AI: Interpreting, Explaining and Visualizing Deep Learning*, pages 121–144, 2019.

[40] Yuxuan Chen, Xuejing Yuan, Jiangshan Zhang, Yue Zhao, Shengzhi Zhang, Kai Chen, and XiaoFeng Wang. Devil's whisper: A general approach for physical adversarial attacks against commercial black-box speech recognition devices. In *Proc. of USENIX Security*, pages 2667–2684, 2020.

[41] Zhizheng Wu, Sheng Gao, Eng Siong Cling, and Haizhou Li. A study on replay attack and anti-spoofing for text-dependent speaker verification. In *Proc.of APSIPA*, pages 1–5, 2014.

[42] Dibya Mukhopadhyay, Maliheh Shirvanian, and Nitesh Saxena. All your voices are belong to us: Stealing voices to fool humans and machines. In *Proc. of ESORICS*, pages 599–621, 2015.

[43] Guangke Chen, Yedi Zhang, Zhe Zhao, and Fu Song. QFA2SR: Query-free adversarial transfer attacks to speaker recognition systems. In *Proc. of USENIX Security*, pages 2437–2454, 2023.

[44] Eberhard Zwicker and Hugo Fastl. *Psychoacoustics: Facts and models*. Springer Science & Business Media, 2013.

[45] Yao Qin, Nicholas Carlini, Garrison Cottrell, Ian Goodfellow, and Colin Raffel. Imperceptible, robust, and targeted adversarial examples for automatic speech recognition. In *Proc. of ICML*, pages 5231–5240, 2019.

[46] Yiqing Lin, Waleed H Abdulla, Yiqing Lin, and Waleed H Abdulla. Principles of psychoacoustics. *Audio Watermark: A Comprehensive Foundation Using MATLAB*, pages 15–49, 2015.

[47] Chinese AI Unicorn Unisound. https://www.unisound.com/.

[48] iFlytyke. https://global.iflytek.com/.

[49] TalantedSoft. http://www.talentedsoft.com/.

[50] Apple Siri. https://www.apple.com.cn/siri/.

[51] Tmall Genie. https://tmallgenie.com/pages/pcHome.html.

[52] Millet Xiaoai. https://www.mi.com/index.html.

[53] Google Assistant. https://developers.google.cn/assistant/surfaces/.

[54] Samsung Bixby. https://www.samsung.com.cn/apps/bixby/.

[55] Huawei Xiaoyi. https://www.huaweiupdate.com/huawei-xiaoyi-vs-iphone-siri/.

[56] OPPO Breeno. https://en.oppotr.com/how-to-use-oppo-breeno/.

[57] Anton Ratnarajah, Shi-Xiong Zhang, Meng Yu, Zhenyu Tang, Dinesh Manocha, and Dong Yu. Fast-rir: Fast neural diffuse room impulse response generator. In *Proc. of IEEE ICASSP*, pages 571–575, 2022.

[58] Arsha Nagrani, Joon Son Chung, Weidi Xie, and Andrew Zisserman. Voxceleb: Large-scale speaker verification in the wild. *Computer Speech & Language*, page 101027, 2020.

[59] John Kominek and A Black. The cmu arctic speech databases for speech synthesis research. *Language Technologies Institute*, 2003.

[60] Steven Chen, Nicholas Carlini, and David Wagner. Stateful detection of black-box adversarial attacks. In *Proc. of ACM AISec*, page 30–39, 2020.

[61] Antony W Rix, John G Beerends, Michael P Hollier, and Andries P Hekstra. Perceptual evaluation of speech quality (PESQ)-a new method for speech quality assessment of telephone networks and codecs. In *Proc. of IEEE ICASSP*, pages 749–752, 2001.

[62] Zhizheng Wu, Nicholas Evans, Tomi Kinnunen, Junichi Yamagishi, Federico Alegre, and Haizhou Li. Spoofing and countermeasures for speaker verification: A survey. *Speech Communication*, pages 130–153, 2015.

[63] Nicholas Carlini, Pratyush Mishra, Tavish Vaidya, Yuankai Zhang, Micah Sherr, Clay Shields, David A. Wagner, and Wenchao Zhou. Hidden voice commands. In *Proc. of USENIX Security*, pages 513–530, 2016.

[64] Zhuolin Yang, Bo Li, Pin-Yu Chen, and Dawn Song. Characterizing audio adversarial examples using temporal dependency. In *Proc. of ICLR*, 2019.

Appendix

A Detailed Experiment Setting

A.1 Dataset

In this paper, we use two popular audio datasets to evaluate the performance of VoxCloak against SR APIs, and collect

Table 9: Details of the commercial voice assistants.

Voice Assistant	**OS Version**	**Device**	**Wakeup Words**
Apple Siri	iOS 14.3	iPhone12	Hey, Siri
Samsung Bixby	Android11	Galaxy Note10+ 5G	Hi, Bixby
Google	Android11	G8 ThinQ	Hey, Google
Huwei Xiaoyi	HarmonyOS 3.0	P30 Pro	Xiaoyi,Xiaoyi
Tmall Genie	-	Tmall Gennie X2	Tianmaojingling, Woshishei
Millet Xiaoai	-	Millet Xiaoai Pro LOS	Xiaoaitongxue, Woshishei
Oppo Breeno	ColorOS 13.1	OPPO Reno5K+5G	Xiaobu, Xiaobu

Table 10: Survey of well-known companies about SR Service.

Company	**SR Service**	**Target Customer**	**Price**
Amazon	Yes	Authorized Person and Company	\$0.018*
Google	Yes	Authorized Company	\$0.01#
Microsoft	Yes	Authorized Company	\$0.005#
Apple	Yes	Authorized Devices	-
IBM	Yes	Authorized Person and Company	\$0.02*
NVIDIA	Yes	Open Source	-
Sensory	Yes	Only Company	-
OpenAI	-	-	-
Oracle	-	-	-

Note that, ∗ represents the price per minute of the service, and # represents the price per request.

some voice data (named the WakeUp dataset) from several volunteers for attacking voice assistants.

Voxceleb V1. The Voxceleb V1 dataset [58] is a large-scale dataset of speech recordings from celebrities. It contains over 100,000 utterances from 1,251 speakers of different accents, genders, and ages. The dataset is designed for speaker recognition and verification tasks, as well as other speech-related applications.

CMU_arctic. The CMU_arctic dataset [59] is a collection of speech recordings from 16 speakers of American English. The dataset is created by the Carnegie Mellon University. It contains about 1,132 utterances per speaker, covering a variety of topics and linguistic phenomena.

WakeUp. This dataset comprises 420 audio recordings collected from the voices of twenty volunteers, consisting of 11 males and 9 females. The volunteers are native Chinese speakers who are also proficient in English. Among the wake-up words used in the recordings, three are in English, while the remaining wake-up words are in Chinese. Each volunteer is asked to pronounce the corresponding wake-up words, including the designated Chinese and English wake-up words, within a quiet office environment with an SNR lower than 40 dB. Volunteers are instructed to complete the utterance within a 3-second time clip. The recordings are captured using an iPhone 12, and three separate voice recordings are obtained from each volunteer. To ensure consistency and standardization, the recordings are manually edited to remove any empty fragments at the beginning or end. As a result, all recordings in the dataset have a precise duration of 3 seconds.

Table 11: The transferability of adversarial examples targeting one source FE to other target FEs.

Source\Target	X-Vector	D-Vector	MFCC	Mel
X-Vector	100%	34%	37%	38%
D-Vector	15%	100%	22%	5%
MFCC	3%	45%	100%	51%
Mel	4%	18%	45%	100%

A.2 Candidate Feature Extractors

We select four common feature-extracting algorithms as candidates: Mel spectrum, MFCC, X-Vector, and D-Vector.

The Mel spectrum is based on the Mel scale, which simulates the sensitivity of the human ear to different sound frequencies [25]. In calculating the Mel spectrum, the audio signal is first transformed using the Fast Fourier Transform (FFT) to obtain the spectrum, which is then converted into the Mel spectrum using a set of Mel filter banks.

The Mel Frequency Cepstral Coefficients (MFCC) [14] are features extracted from the Mel spectrum, capturing the primary characteristics of the short-term power spectrum of the audio signals. These coefficients can reflect the ability of the human auditory system to perceive and distinguish different sound frequencies. The extraction process of MFCC includes steps such as preprocessing, FFT, Mel filtering, logarithmic operations, discrete cosine transform, and extraction of dynamic features.

D-Vector [15] and X-Vector [16] are speaker-embedding methods that embed speaker characteristics into a vector space using DNNs. Specifically, the D-Vector utilizes the hidden layers of a DNN to differentiate the features of speakers. These representation vectors can capture key information related to the voiceprint of speakers. X-Vector is an improvement upon D-Vector, offering better performance in processing more complex and varied speech data. Its core technology is time-delay neural networks, which are more adept at handling variable-length input sequences. The capability allows the X-Vector to process complete speech segments, thereby more effectively capturing the audio features.

A.3 Implementation Details

For the genetic algorithm, we configure a population size of 100 and a mutation probability of 0.1 and set the termination generation for the genetic algorithm to 30. We set the parameter α to 0.05 and use the Adam optimizer to solve Eqs. (6) and (14) with a learning rate of 3. VoxCloak was deployed on a server equipped with six GeForce RTX 2080 Ti GPUs, a 32-core Intel Xeon Gold 5117 CPU at 2.00 GHz, and 119 gigabytes of RAM. Detailed information about the target device

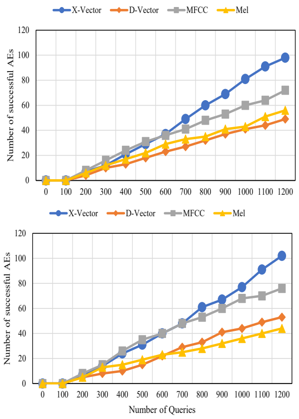


Figure 6: Four attempts to infer the FE using grid search, where AEs means adversarial examples.

is shown in Table 9. Furthermore, we survey more commercial SRSs as targets, as detailed in Table 10. Unfortunately, some of them are unavailable. For example, Amazon provides a speaker verification service in their connection application. As it is unavailable in our region, we are unable to use it in our experiments. We are also unable to utilize the service provided by IBM as our registration was denied.

B Supplementary Evaluation of VoxCloak

B.1 Grid Search-based Inference Method

Grid search is an exhaustive search method. It can accurately identify the optimal result by traversing all possible outcomes within the predefined search space. We have also attempted to use grid search to infer the target FE. The results in Figure 6 demonstrate that, in our four attempts, grid search successfully determined the target FE. However, it required a minimum of 600 queries (sometimes 900 queries), approximately two (or three) times more than VoxCloak. Additionally, an increase in the search space also leads to higher local computational costs for the attacker. Therefore, in terms of efficiency, our proposed genetic algorithm outperforms the grid search method.

B.2 Transferability of Feature Extractors

The success of VoxCloak primarily relies on the distinct feature spaces created by different FEs. We confirm this by evaluating the transferability of AEs. Transferability is measured by the proportion of AEs generated from one FE that can successfully attack a target system using a different FE. For each FE, we generated 100 AEs, with the results presented in Table 11. We observed that the ASR significantly drops when attacking an SRS based on a different FE, indicating the differences in the feature spaces of these FEs. We noticed that AEs generated using the X-Vector exhibit better transferability. Therefore, in scenarios where querying the target systems is unavailable (e.g., the physical attacks), using X-Vector to generate AEs tends to yield better performance.

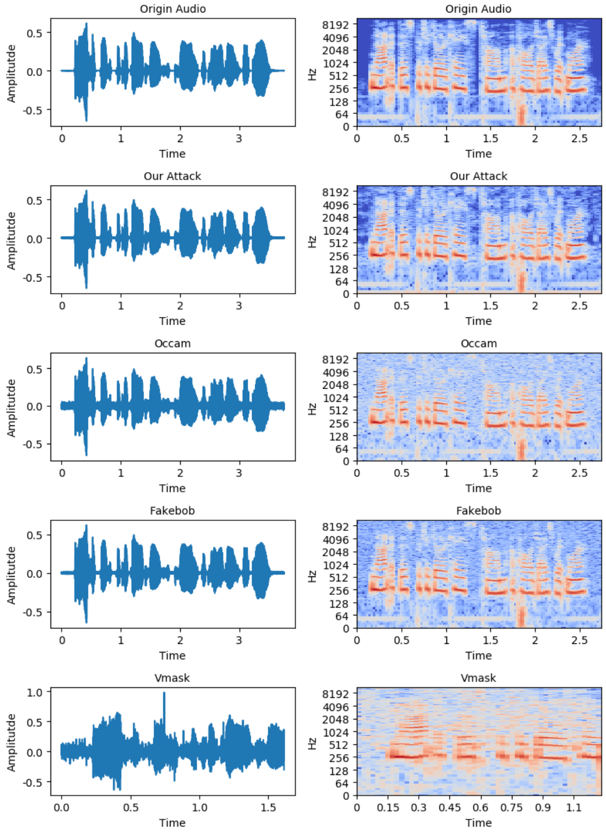


Figure 7: Waveforms and spectrograms of the original audio and adversarial audios generated by VoxCloak and baselines.

B.3 Visualization Comparison

To facilitate a more intuitive comparison with baselines, we visualized the waveforms and spectrograms of AEs generated by the attacks and the original audio, as shown in Figure 7. We can see that the waveform of the AE generated by VoxCloak is close to the original audio, whereas those produced by Occam, Fakebob, and Vmask show significant differences. Thus, the latter are more easily perceptible by the human ear. Furthermore, as Vmask requires audio inputs and generated AEs of a fixed length, its effectiveness diminishes for longer-duration authentication processes.