A Survey on Voice Assistant Security: Attacks and Countermeasures

CHEN YAN, XIAOYU JI, KAI WANG, QINHONG JIANG, ZIZHI JIN, and WENYUAN XU, Zhejiang University

Voice assistants (VA) have become prevalent on a wide range of personal devices such as smartphones and smart speakers. As companies build voice assistants with extra functionalities, attacks that trick a voice assistant into performing malicious behaviors can pose a significant threat to a user's security, privacy, and even safety. However, the diverse attacks and stand-alone defenses in the literature often lack a systematic perspective, making it challenging for designers to properly identify, understand, and mitigate the security threats against voice assistants. To overcome this problem, this article provides a thorough survey of the attacks and countermeasures for voice assistants. We systematize a broad category of relevant but seemingly unrelated attacks by the vulnerable system components and attack methods, and categorize existing countermeasures based on the defensive strategies from a system designer's perspective. To assist designers in planning defense based on their demands, we provide a qualitative comparison of existing countermeasures by the implementation cost, usability, and security and propose practical suggestions. We envision this work can help build more reliability into voice assistants and promote research in this fast-evolving area.

CCS Concepts: • **Security and privacy → Systems security**; • **Human-centered computing → Human computer interaction (HCI);**

Additional Key Words and Phrases: Voice assistant, security, attack, defense, speech, voice interaction

ACM Reference format:
Chen Yan, Xiaoyu Ji, Kai Wang, Qinhong Jiang, Zizhi Jin, and Wenyuan Xu. 2022. A Survey on Voice Assistant Security: Attacks and Countermeasures. *ACM Comput. Surv.* 55, 4, Article 84 (November 2022), 36 pages.
https://doi.org/10.1145/3527153

1 INTRODUCTION

A **voice assistant** (**VA**) takes and executes voice commands from users, such as making phone calls, playing music, finding answers, and controlling home appliances. As a natural way to interact with machines, there is no surprise that voice assistants have been widely deployed on smartphones, laptops, smart speakers, vehicles, industrial applications [160], and even military warships [105]. As of 2019, an estimated 3.25 billion voice assistants have been used worldwide, and by 2023 the number is expected to reach around 8 billion—nearly one VA per person on average [146]. As companies rush to build voice assistants with more functionalities, attacks that trick a voice assistant into performing malicious behaviors can pose a significant threat to the owner's

This work is supported by China NSFC Grant 61925109, 62071428, and 61941120.
Author's address: C. Yan, X. Ji (corresponding author), K. Wang, Q. Jiang, Z. Jin, and W. Xu (corresponding author), Zhejiang University, 38 Zheda Road, Hangzhou, Zhejiang 310027, China; emails: {yanchen, xji, wyxu}@zju.edu.cn.

0360-0300/2022/11-ART84 $15.00
https://doi.org/10.1145/3527153

security, privacy, and even safety. For example, malicious voice commands may make voice assistants browse malicious websites, forward private e-mails, make payments, or unlock homes and vehicles.

As voice assistants become increasingly prevalent, it is essential to comprehensively understand their security risks and mitigate the threats before it is too late. However, based on the current literature, we identify the following gaps that hinder the progress of securing voice assistants.

Gap 1: diverse attacks. Voice assistants are complex systems built with various software and hardware components. As a result, relevant attacks involve diverse vulnerabilities, threat models, and methods that may seem utterly unrelated, making it challenging to understand security threats systematically. To name a few, an attacker can generate adversarial examples [20] that sound benign to humans but are transcribed to malicious commands by a VA's speech recognition software. She may also create a malicious third-party skill and wait for it to be invoked accidentally due to faulty natural language understanding [79]. She may even exploit vulnerable sensor hardware and use ultrasound [205], laser [148], or electromagnetic waves [80] to inject inaudible voice commands.

Gap 2: stand-alone defenses. Most of the existing defenses aim at mitigating only one type of attack in stand-alone setups. For example, liveness detection [30, 43, 100, 107, 109, 123, 131, 138–141, 144, 169, 179, 209, 210, 214] is proposed to detect voice spoofing attacks only, and adversarial training [12, 37, 150, 151, 172] is designed solely to resist adversarial example attacks. It is unknown how these defenses may apply to voice assistants in a complex adversarial environment, especially how they compare with each other in terms of the implementation cost, usability, and security. Despite the promising results presented by numerous articles, it is still challenging for VA designers to select and implement proper protection.

Gap 3: lack of systematic perspectives. Many relevant studies only target a VA component, e.g., speech recognition or speaker verification, rather than the entire system, sometimes even without a VA context at all. For example, most of the attack articles [20, 21, 26, 37, 81, 97, 124, 135, 136, 153, 155, 175, 184, 190, 199] and defense articles [6, 12, 19, 35, 37, 56, 68, 90, 134, 150, 151, 172] related to the security of speech recognition, i.e., the core component of a VA, focus on stand-alone models, many of which have not yet been used in commercial voice assistants. However, these studies, which may not be indexed with a VA keyword, may apply to voice assistants in the future and shall be considered equally.

These gaps have made it a big challenge to properly identify, understand, and mitigate the security threats against voice assistants. Besides, the massive number of publications holds researchers back from gaining a complete picture of the field's state of the art, development, key challenges, and future directions. Therefore, in this work, we aim at closing these gaps and providing a comprehensive investigation of voice assistant security for security researchers and VA designers.

A few surveys have shed light on relevant topics, including the security of speech recognition [4, 63, 171], speaker verification [4, 34, 41, 186], the speech interface [14], smart speakers [120], voice-controlled systems [50], and personal assistants in vehicle [214] and at home [39]. However, none of them has discussed the security of voice assistants to the breadth and depth we expect. It remains unclear mainly how vulnerable voice assistants are to various attacks, how the attacks work, and how to effectively protect voice assistants from a system perspective.

In this article, we conduct an extensive literature analysis on voice assistant security. In particular, we consider a study relevant if it can potentially be used to attack or defend voice assistants. To address the attacks' diversity, we stand from a system designer's perspective and organize the attacks based on the VA structure. We systematize all relevant attacks from two dimensions: (a) *the attack method*, and (b) *the vulnerable VA component*. Within each attack category, we extract the shared methodologies and analyze the differences in depth. To integrate stand-alone defenses

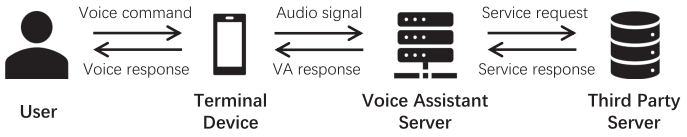

Fig. 1. A general workflow of VA service.

for future voice assistants, we categorize existing methods by the defensive strategies they share rather than the attacks they were designed to mitigate, enabling us to qualitatively compare and assess their applicability in a unified voice assistant context. We further discuss the potential directions for future research and propose practical suggestions to VA designers and users.

Contributions. We envision this work can help build more reliability into voice assistants and promote research in this fast-evolving area. We summarize our contributions as follows:

- To the best of our knowledge, our work is the first in-depth study on voice assistant security from a system perspective. Based on an extensive literature survey of relevant attacks and defenses, we present a complete picture of the state of the art, development, key challenges, and future directions for VA security research.
- We organize the attack literature based on the VA structure and systematize relevant attacks by the vulnerable VA components and attack methods. The organization helps bridge the gap between a broad category of seemingly unrelated attacks and vulnerabilities, enabling proper identification, understanding, and assessment of the security threats against voice assistants.
- We systematize the countermeasures based on the defensive strategies from a system designer's perspective. To assist designers in planning defense based on their demands, we provide a qualitative comparison of existing methods by the implementation cost, usability, and security and propose practical suggestions.

We organize the remainder of this article as follows: Section 2 gives a brief introduction to voice assistants. Section 3 overviews the attacker's goal, threat model, existing attack methods, and the idealism of a practical attack and introduces how this article organizes the attacks based on the vulnerable system components. Sections 4–7 respectively elaborate on the attacks that exploit the vulnerabilities of the sound-to-audio, audio-to-text, audio-to-identity, and text-to-intent subsystems of a voice assistant. In Section 8, we systematize the defense strategies that can detect or prevent the above-mentioned attacks. We discuss future research directions and give suggestions to voice assistant designers and users in Section 9. Section 10 concludes this article.

2 VOICE ASSISTANTS

2.1 Overview of Voice Assistants

Voice assistants are becoming a de facto standard for smart personal devices. Almost every smartphone, smart speaker, and smart car has been implemented with at least one voice assistant. The most well-known examples include Apple Siri, Amazon Alexa, Google Assistant, Samsung Bixby, Microsoft Cortana, and so on. Regardless of the manufacturer, voice assistants generally have a workflow similar to Figure 1. The user interacts with a terminal device, e.g., a smartphone or smart speaker, which records the user's voice and runs a local VA software that handles the interface, wake-word detection, and user authentication and streams the voice to the VA server. The server interprets the voice stream and requests the corresponding service, sometimes from third-party servers [31]. The service or response is sent back to the terminal device and provided to the user.

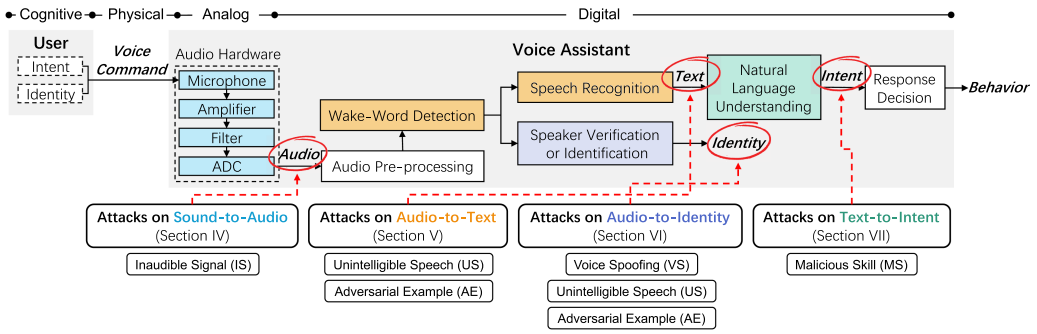

Fig. 2. A diagram of the functional components and signal flow in voice assistants. We categorize existing attacks based on the point where malicious output appears, e.g., as audio, text, identity, or intent.

To better understand voice assistants and their security exploits, we break down a voice assistant into several functional subsystems: *sound-to-audio*, *audio pre-processing*, *audio-to-text*, *audio-to-identity*, *text-to-intent*, and *intent-to-behavior*. Each subsystem is designed to perform a specific task, as the names suggest. For example, the sound-to-audio subsystem converts the physical sound of a human voice to a digital audio stream. We introduce these subsystems with a summarized diagram shown in the upper part of Figure 2.

2.2 Sound-to-Audio

When a user speaks to a VA, the VA's microphone acts as a transducer that converts the acoustic sounds into electrical signals. To obtain high-quality audio, the captured signals are usually processed in several steps. An amplifier first amplifies the raw electrical signals, and then a low-pass filter removes the ambient noise and irrelevant frequencies in the signal before sending it to an analog-to-digital converter, which converts the analog signal to digital audio.

2.3 Audio Pre-Processing

As voice assistants continuously listen for voice commands, the captured audio may not contain human speech. Thus, the audio first passes through a pre-processing phase, which mainly involves a **voice activity detection** (**VAD**) that identifies the presence of human speech in the audio. VAD discriminates speech from non-speech sections such as background noises, and it has become essential for both speech recognition and speaker verification/identification. VAD is traditionally based on feature engineering and statistical signal processing. More recently, **deep neural network** (**DNN**) based methods have also been proposed [130].

2.4 Audio-to-Text

The audio-to-text subsystem is designed to recognize the audio's content. Since a typical procedure of VA interaction is to first activate it with a wake-word (e.g., "Hey Siri") and then provide voice commands, audio-to-text generally involves two phases: wake-word detection and speech recognition. Once a wake-word is detected, the VA is activated and streams the following audio to the cloud for speech recognition.

2.4.1 Wake-Word Detection. A wake-word is a keyword that users can speak to activate a VA. While VAs are "always listening," they do not perform any action until they detect a pre-determined wake-word, such as "Alexa," "OK Google," "Hey Siri," and so on. VAs usually detect wake-words with a **keyword spotting system** (**KWS**) running locally on the terminal device, which typically

consists of a feature extractor and a neural network-based classifier. Traditional KWS uses **Hidden Markov Models** (**HMMs**) and Viterbi decoding, while recent techniques include discriminative models with a large-margin problem formulation and **recurrent neural networks** (**RNNs**) [44].

2.4.2 Speech Recognition. **Automatic speech recognition** (**ASR**) converts audio into its textual transcription, allowing voice assistants to recognize a voice command's content. A typical ASR system includes three procedures: pre-processing, feature extraction, and model-based decoding. We skip the introduction of pre-processing as it mainly involves VAD.

Feature Extraction. Feature extraction retrieves necessary information from the audio. The pre-processed audio is first split into short segments or frames, and then features are extracted from each of them. Various signal processing techniques are used for the feature extraction, including **Discrete Fourier Transforms** (**DFT**), **Mel Frequency Cepstral Coefficients** (**MFCC**), **Linear Predictive Coding** (**LPC**), and the **Perceptual Linear Prediction** (**PLP**) method [115].

Decoding. In this procedure, a trained machine learning model decodes the extracted features into a sequence of possible phonemes, characters, and words. Traditional decoder models are composed of an acoustic model and a language model [116]. The acoustic model matches features to phonemes, while the language model refines the results using grammar rules, commonly-used words, and so on. Traditional ASR systems employ models such as **Gaussian Mixture Models** (**GMMs**), HMMs, RNNs, and **Convolutional Neural Networks** (**CNNs**) [132]. State-of-the-art ASR systems include Kaldi (DNN-HMM), CMU Sphinx (GMM-HMM), Mozilla DeepSpeech (RNN with CTC loss function), Wav2Letter (CNNs-based), and so on.

2.5 Audio-to-Identity

The audio-to-identity subsystem recognizes or verifies the speaker's identity from the captured audio. There are two types of tasks on voice assistants: **speaker identification** (**SI**) and **speaker verification** (**SV**) [128]. The goal of speaker identification is to determine which speaker's voice in a known group best matches the audio, while speaker verification aims at determining whether the audio matches the voice of the user whom the speaker claims to be. The speaker recognition pipeline is mostly similar to ASR systems, which involves pre-processing, feature extraction, and decoding. Traditional speaker models can be divided into template models (**Vector Quantization** (**VQ**), **Dynamic Time Warping** (**DTW**)), and stochastic models (GMM, HMM, GMM-UBM). State-of-the-art speaker models mainly include i-vector-based (i-vector, PLDA), DNN-based (d-vector, j-vector, x-vector), improved algorithm (learning to rank, attention mechanism), and so on.

Audio-to-identity systems can be further divided into text-dependent and text-independent based on whether it depends on pre-determined audio content. Existing voice assistants follow a text-dependent way. The speaker model is trained based on the temporal dependencies between the feature vectors of the wake-word spoken by the user [77]. For text-independent systems, the speaker model is trained based on the feature distribution. Text-independent systems allow speakers to utter any audio content and may be used in the future for verifying voice commands.

2.6 Text-to-Intent

Intents are simply sets of utterances that exemplify the intention of the speaker to perform an action, convey or obtain information, and so on. The text of a user's command is sent to a user intention identifier, which labels the text with a list of intention hypotheses (a domain name or a service name) and their confidence scores. A central controller distributes the user's command with

its intention labels and confidence scores to all the domain experts and the service modules [180]. When receiving a query, the domain expert will use its **Natural Language Understanding** (**NLU**) module to parse the utterance and update its dialogue state in consideration of both the NLU output and the intention labels.

2.7 Intent-to-Behavior

If the dialogue state can be updated, the domain expert will return a natural language utterance realized by the **Natural Language Generation** (**NLG**) module or a set of data records obtained from its database [180]. Finally, the server will provide the voice assistant with appropriate information to the user **through text-to-speech** (**TTS**) translation, play the requested media, or complete other tasks with various connected services and devices.

3 ATTACK OVERVIEW

This section provides an overview of the attacker's goal, threat model, existing attack methods, and a summarized idealism of practical attacks. To help system designers better understand the threats, we systematize the attacks using a vulnerability-oriented approach based on the VA structure.

3.1 Attacker's Goal

The general goal of attacking a voice assistant is to make it perform malicious behaviors without the owner's authorization. Examples of the attack's impact include:

Security. The attacker may initiate the drive-by download of malware by having the VA visit a malicious website, steal money from the user by making the VA perform a bank transfer or online shopping, inject fake events into the user's calendar, or send fake messages in the user's name.

Privacy. The attacker may eavesdrop on the user's private conversation via a malicious third-party application or by letting the VA make a video/phone call to the attacker. The attacker may also expose the user's private daily schedule and travel information by inquiring about the VA.

Safety. The attacker may manipulate other devices by commanding the VA, e.g., unlocking the front door or controlling a connected car, which may put the user's safety at risk.

The security and privacy of voice assistants have been a public concern in recent years. For example, in 2017, Alexa owners watching a TV broadcast found it accidentally ordered dollhouses [117]. Burger King launched a TV commercial that intentionally prompts Google Home speakers to read a description of the Whopper from Wikipedia [183]. Though benign, these real-world cases show that voice assistants can be easily misused without the owner's authorization, and it is only a matter of time before malicious actors exploit them. In fact, a few researchers have demonstrated practical attacks that can open the garage gate or dial a spying phone call in real-world setups [59].

3.2 Threat Model

We consider attacks that conform to the following assumptions:

Unaltered Voice Assistant. We assume that the target voice assistant system remains unaltered throughout the attack. As voice assistants are, in most cases, highly enclosed systems implemented on properly secured servers and personal devices, it is difficult for the attacker to modify any software or hardware component of the voice assistant.

Trusted Platform and Network. We assume that the voice assistant is implemented on a trusted platform (operating system and computer hardware), and the network traffic is secured with state-of-the-art protocols. Though it may be feasible for the attacker to compromise the operating system or manipulate the network traffic, e.g., via **Distributed Denial of Service** (**DDoS**) attacks, such attacks are not unique to voice assistants and thus are not our focus in this article.

Attack Triggered by Voice Commands. As voice assistants perform tasks only after receiving voice commands, we assume all malicious behaviors of a voice assistant are triggered by the voice command it receives physically, which can be either a malicious command from the attacker or a benign command from the user. Therefore, we consider network attacks out of scope.

Limited Knowledge of the Voice Assistant. The attacker may have the following levels of information of the target voice assistant.

- Black-box: the attacker has zero knowledge, e.g., she can only query the model in a black-box manner for the label or confidence score output but has no access to the model's type, structure, or parameters. Black-box is the most common knowledge regarding commercial voice assistants.
- Grey-box: the attacker has partial knowledge, e.g., she may know the model type or feature representations, but she has no access to the model structure or parameters. Grey-box knowledge usually applies when the technical information of a voice assistant is made public.

3.3 Attack Methods

According to the threat model, we identify and sort existing attack methods into six categories, which are separated mainly by human perception.

`Normal Speech (NS).` The easiest and most straightforward way of attack is to directly command the VA using a normal speech, which can be spoken by the attacker or generated by text-to-speech software. However, as the malicious intention is fully exposed, such attacks can be easily discovered.

`Voice Spoofing (VS).` Superior to normal speech, voice spoofing attacks mimic the VA owner's voice to bypass speaker verification. Though more powerful than normal speech, voice spoofing still fully exposes the malicious intention and can be easily discovered, especially by the owner.

`Unintelligible Speech (US).` To hide the malicious intention, attacks within this category use speech that humans cannot understand to deliver malicious commands. Nonetheless, unintelligible speech is odd in real life and may raise user suspicion.

`Adversarial Example (AE).` Audio adversarial examples are specifically designed sounds that appear normal and benign to users but are mispredicted by machine learning models. Such attacks are stealthy if the appearance of the sound is not suspicious.

`Inaudible Signal (IS).` Attackers may use physical signals inaudible to humans, such as ultrasound, light, and electromagnetic waves, as carriers to inject malicious commands. As users do not hear anything, these attacks are usually hard to notice, even for alert users.

`Malicious Skill (MS).` Besides actively giving a malicious command to the VA with the above methods, an attacker may wait for the user to say a command that unintentionally triggers a malicious third-party skill (extended services supported by many VAs) and causes malicious outcomes. It is tough for users to notice that the provided service is not legitimate.

These attack methods, except for `MS`, are essentially different approaches to injecting malicious voice commands. For example, an attacker may make a VA unlock the front door with an "open the door" command formed as either normal speech, synthesized speech, unintelligible speech, adversarial example, or inaudible signals, which have the same attack outcome but appear differently for human perception. For `MS` attacks, since the VA's malicious behaviors are induced by the user's benign commands instead of the attacker's malicious commands, the attack's impact depends on the skill's ability. Most of the existing `MS` attacks focus on the privacy impact.

Table 1. A Comparison of Existing Attack Methods by their Human Perception and Satisfaction to a Practical Attack's Idealism

Attack Method	Human Perception			Idealism of a Practical Attack				Induced Erroneous Output				Sec.
	Audible	Intelligible	Malicious	Over-the-air	Black-box	Unsuspicious	Real-time	Audio	Text	Identity	Intent	
Normal Speech	✔	✔	✔	✔	✔	✖	✔	○	○	○	○	–
Voice Spoofing	✔	✔	✔	✔	✔	✖	✖	○	○	●	○	6.1
Unintelligible Speech	✔	✖	✖	✔	〓	✖	✖	○	●	○	○	5.1
								○	○	●	○	6.2
Adversarial Example	✔	✔	✖	〓	〓	〓	〓	○	●	○	○	5.2, 5.3
								○	○	●	○	6.3
Inaudible Signal	✖	✖	✖	〓	✔	✔	✔	●	○	○	○	4
Malicious Skill	✖	✖	✖	✖	✔	✔	✖	○	○	○	●	7

✔ Positive ✖ Negative 〓 Case by case ● Applicable ○ Not applicable – None.
In this Article, we Reorganize the Attacks by the Induced Erroneous Output (i.e., vulnerable component) in a Voice Assistant and List the Mapped Section Numbers in this Article.

In the rest of this article, we focus on attack methods except for normal speech. Though a few studies proposed to make normal speech attacks stealthier by playing the malicious command via a hijacked device nearby [198] or the VA terminal device itself [8, 36, 66, 212], such attacks can still be noticed easily with users nearby, therefore posing limited threats.

3.4 Idealism of a Practical Attack

We summarize four ideal properties for a practical attack in the real world. In general, attacks that possess more of these properties may pose more threats in practice.

(1) **Over-the-air.** The attack is launched over the air instead of over the line, e.g., via physical voice commands or signals rather than digital audio files.
(2) **Black-box Knowledge.** Most state-of-the-art voice assistants are commercial, meaning that the attacker may only have black-box knowledge.
(3) **Unsuspicious.** The attack does not produce anything odd that can be easily noticed by users.
(4) **Real-time.** The attack can be generated or adjusted in real-time in case of varying objectives and attack scenarios.

We compare the attack methods by how humans perceive them and how they apply to a practical attack's idealism in Table 1. It shows that different attacks pose various levels of threats and may even vary from case to case.

3.5 Systematizing Attacks based on Vulnerabilities

The research community has widely accepted the attack categorization based on human perception. However, different attack methods may seem utterly unrelated to each other as they involve varying signal modalities and vulnerabilities. To help designers better understand the threats from a system perspective, we reorganize the attacks in a vulnerability-oriented approach based on the VA's system components, as shown at the bottom of Figure 2.

We note that all attacks exploit at least one VA component's vulnerabilities to make it generate erroneous outputs. For example, inaudible signal attacks exploit vulnerable microphone hardware to transform non-acoustic signals into audio. In this case, the audio is an erroneous output of the sound-to-audio hardware because the microphone is supposed to receive only audible sounds. Therefore, we describe the attack against voice assistants as: **an attack that makes a voice assistant's components generate *erroneous outputs* that lead to malicious system behaviors.**

It now becomes necessary to investigate which and how VA components are prone to produce erroneous outputs under attack. We sort the attacks into four classes based on the VA structure introduced in Section 2, namely the attacks on sound-to-audio, audio-to-text, audio-to-identity,

and text-to-intent components. The attacks within each class exploit the corresponding component's vulnerabilities to generate erroneous audio, text, identity, or intent. Table 1 shows a mapping of the attack methods to the induced erroneous output, i.e., vulnerable components. The following sections will elaborate on the attacks by the vulnerable components in a voice assistant they exploit.

4 ATTACKING SOUND-TO-AUDIO

The sound-to-audio subsystem, i.e., the microphone hardware, is designed to convert audible voice commands to digital audio signals. However, due to the microphone's vulnerabilities, such as the sensitivity to ultrasound and light, **electromagnetic** (**EM**) susceptibility, and nonlinearity, an attacker may induce malicious audio using physical signals that are inaudible to humans, such as ultrasound, light, EM waves, and electrical signals. Since the signals are usually imperceptible, such attacks can be very stealthy. In this section, we introduce `Inaudible Signal` attacks by the signal type.

Ultrasound. Ultrasounds are sounds with a frequency above 20 kHz and are inaudible to humans. However, most microphones by design can receive both audible sounds and ultrasounds, enabling inaudible voice commands. An attacker can modulate a malicious command on an ultrasound carrier and exploit the inherent nonlinearity of microphones and amplifiers to demodulate the signal in a VA's hardware. Several studies have demonstrated practical attacks on voice assistants with ultrasounds transmitted over the air [64, 70, 129, 145, 193, 205] and through a solid surface [194].

Light. Microphones are also sensitive to light pressure. Sugawara et al. [148] found that a focused laser beam can vibrate a microphone's membrane and generate audio signals. By modulating the laser's intensity with a malicious command as the baseband, they managed to invoke and control a voice assistant from as far as 110 meters away due to the laser's high power and directivity.

EM Waves. The connection wire in a microphone circuit can act as an antenna that unintentionally picks up electromagnetic interference. Attackers may exploit this effect to inject modulated EM waves into the hardware. Similar to ultrasound attacks, the nonlinearity of amplifiers is exploited to demodulate the malicious command. Foo Kune et al. [80] first demonstrated such attacks on voice recorders. Kasmi and Esteves [72] later showed it is feasible to attack voice assistants on smartphones through the EM coupling on a headphone cable.

Electrical Signal. Attackers may directly inject electrical audio signals to a voice assistant by connecting to the device's audio input port [197] or via conducted interference through the charging cable [40]. This enables the attacker to inject malicious audio inaudibly but requires physical contact with the VA device, which may be infeasible in some scenarios.

Inaudible signal attacks are also known as transduction attacks [45] and are well-studied in analog sensor security. We refer readers to [192] for a systematization of knowledge on this topic.

5 ATTACKING AUDIO-TO-TEXT

The audio-to-text subsystem recognizes the speech content of digital audio signals. By exploiting the discrepancy of speech perception between humans and machines, attackers can generate audio signals that do not sound malicious to users but are converted to malicious texts by the audio-to-text subsystem. We introduce two types of attack in this section: `Unintelligible Speech` and `Adversarial Examples`. A summary and comparison of existing studies are shown in Table 2.

5.1 Unintelligible Speech

Unintelligible speech attacks aim at generating unintelligible sounds such as noise [2, 19, 163] and nonsensical sounds [15] that can be misclassified as targeted malicious commands. Most attacks

exploit the lossy process of signal processing and feature extraction in ASR to generate unintelligible speech that shares similar features with a targeted intelligible command. Since speech recognition models perform the prediction based on the extracted features, two audio signals that share similar features may be recognized as similar results, no matter how different they sound. For example, Vaidya and Carlini et al. [19, 163] proposed to generate unintelligible speech by calculating the MFCC features of a targeted command and conducting an inverse-MFCC process to recover new audio. Abdullah et al. [2] used signal processing techniques, such as time-domain inversion and random phase generation, to perturb an intelligible audio signal until it becomes unintelligible while preserving its acoustic features. Another track of method exploits the ability of voice assistants to match nonsense syllables to meaningful words. Bispham et al. [15] proposed a speech mangling process that replaces consonant phonemes in a targeted command to generate nonsensical sounds. Due to the language model in ASR that corrects errors in the context of the preceding words, the nonsensical sounds may be recognized as the targeted command. Compared with normal speech attacks, unintelligible speech attacks are stealthier because the attacker's intention is not exposed to the user. Nonetheless, such attacks may be suspicious as unintelligible speech is odd in daily life.

5.2 Adversarial Examples

Audio **adversarial examples** (**AEs**) refer to specifically designed sounds that appear normal and benign to users but are mispredicted by machine learning models. Adversarial examples are first proposed and studied in the image domain [152] and have recently also become active in the audio domain. In this section, we focus on adversarial examples against audio-to-text systems.

An attacker may cause two types of erroneous system outputs:

- **Untargeted Output.** The model's prediction is erroneous but cannot be specified. Untargeted attacks can disrupt the model's functionality, e.g., for attackers to prevent users from using voice assistants [47, 58, 87] and for users to avoid being automatically wiretapped [3, 21].
- **Targeted Output.** The model's prediction can be specified by the attacker. Targeted attacks are more threatening than untargeted ones, but they are also much more challenging [32], especially for longer target phrases [20]. So far, most studies on audio AE have focused on targeted attacks.

We elaborate on the methods of generating audio adversarial examples in the following.

5.2.1 The Basic Idea of Adversarial Example. The basic idea behind most work is to optimize for an adversarial perturbation that, when added to benign audio, can maximize the likelihood of an intended erroneous prediction while having the perturbation being constrained to a small value such that it may be imperceptible to humans [87]. During the optimization, the benign audio and the model are kept unchanged, and only the perturbation is updated. This idea is conceptually similar to training a neural network and in line with generating adversarial examples in other domains [200]. However, existing methods, e.g., in the image domain, are not directly applicable to audio adversarial examples [9, 20, 161] because audio models initially operate at a higher level than the "pixel level" of their image counterparts [3]. The primary audio characteristics include heavy pre-processing [135], time-dependency [190], and high sampling rates [37], which pose unique challenges. In the following, we introduce the key considerations in formulating and solving the optimization problem.

5.2.2 Formulating the Optimization Problem. The formulation of an optimization problem depends on many factors, such as the attacker's goal, the model, and the attacker's knowledge about

the model. In general, an optimization problem requires an objective function to be minimized or maximized and some constraints that limit the space for variables.

Objective Function. The objective function is usually a function of the perturbation δ, a benign audio x, and a model output t. δ is the variable to optimize for, and x and t are invariants that are known or specified by the attacker. We can describe a general objective function $f(\cdot)$ [97] as

$$f(x, \delta, t) = \ell_{\text{model}}(F(x+\delta), t) + c \cdot \ell_{\text{metric}}(x, x+\delta), \tag{1}$$

where ℓ_{model} is a model-related loss function that measures the difference between a predetermined model output t and the actual output $F(x+\delta)$, suppose the model's function is $F(\cdot)$. ℓ_{metric} is a loss function used to measure the distortion that the perturbation introduces to the original audio, and c is a coefficient that trades off attack success and audio quality.

Take a targeted attack as an example, we can describe the optimization goal as given a benign audio x, a targeted model output t, and a trained model $F(\cdot)$, find the optimal δ that makes $F(x+\delta)$ as close to t ($F(x) \neq t$) as possible while $x+\delta$ and x are as similar as possible. If such a δ exists, then we can consider $x' = x+\delta$ as an adversarial example. In untargeted attacks, the goal is to find a minimal δ that makes $F(x+\delta)$ as different from t ($F(x) = t$) as possible.

Common ℓ_{model} includes (a) loss functions of the original model, such as CTC loss [20, 21, 26, 37, 81, 97, 112, 153, 155] and cross-entropy loss [124]; (b) self-defined loss based on the intermediate model output, such as the distance or cross-entropy between pdf-id (the DNN's output before the decoding step) [28, 135, 184, 199]; and (c) self-defined loss based on the final model output, such as fitness/confidence scores [1, 9, 22, 37, 58] and edit distance between texts [74]. Common ℓ_{metric} includes the p-norm of δ [20–22, 26, 37, 112], **sound pressure level** (**SPL**) [1], **decibel** (**dB**) distance [26, 81], frequency masking [21, 87, 124], **total variation denoising** (**TVD**) [97], and the Euclidean distance between MFCC [58, 74].

In formulating the loss functions, the perturbation δ may be in various forms depending on the model and attack method:

- Audio waveform: Optimization over raw audio is known as the "end-to-end" method [20] and has been adopted by most work. Operating directly over the audio can make AE more imperceptible, but it is more challenging because optimization through the audio pre-processing and feature extraction stages is proved to be difficult [9, 20].
- Spectrogram: A few works [22, 32] generate adversarial perturbation over the audio spectrogram and acquire adversarial audio by inverse transformation, but they face the challenge of optimizing through feature extraction.
- MFCC: Some works [65, 190] optimize the MFCC features as a straightforward way to avoid the above challenges. However, since the Mel-frequency cepstrum transformation is a lossy process, rebuilding adversarial audio by inverse MFCC will significantly reduce the audio quality [37].

Constraints. The constraints are distortion metrics that restrict the perturbation δ within a range during the optimization to ensure a basic imperceptibility level. Attackers may apply constraints to the perturbation's amplitude and frequency. We will elaborate on the constraints in Section 5.3.3.

5.2.3 Solving the Optimization Problem. By solving the formulated optimization problems, attackers can generate audio adversarial examples that are theoretically effective in the digital domain. We divide state-of-the-art solutions into gradient descent, gradient-free optimization methods (particle swarm optimization and genetic algorithm), and machine learning-based methods.

Gradient Descent. Gradient descent (**GD**) is a common optimization approach that has been widely applied for training differentiable models such as neural networks, which are currently

the state-of-the-art architecture for ASR. Similar methods can be used against these models to optimize for an adversarial input—attackers may calculate the gradient of the objective function to the perturbation δ, and adjust δ in the direction that minimizes the objective function. Most existing works adopt gradient descent as their solution.

Gradient calculation requires white-box knowledge of the model function F. Both the objective function and model function need to be differentiable, which is challenging for end-to-end methods because the **Mel-Frequency Cepstrum** (**MFC**) and spectrogram transformations are non-differentiable. There is no efficient way to compute the gradient through them [9, 20, 161]. To overcome this challenge, Schonherr et al. [135, 136] integrated the preprocessing, feature extraction, and the original DNN of Kaldi into one joint DNN and calculated the gradients of the new DNN. Other studies avoid this challenge by directly targeting sequence-to-sequence systems that already incorporate pre-processing and feature extraction as differentiable network layers, such as DeepSpeech and Lingvo.

Gradient descent generally requires three iterative steps [135]:

(1) Measure the loss with the objective function and the ith iterated perturbation δ_i.
(2) Calculate the gradient $\nabla\delta_i$ by partial derivatives and the chain rule. The derivative of F is derived by back-propagation.
(3) Update the input according to the gradient and a learning rate α as $\delta_{i+1} = \delta_i - \alpha\nabla\delta_i$.

These steps are repeated until the loss converges or a pre-defined number of iterations is reached. A higher number of iterations may increase the success rate, but it can also increase the amount of noise [135]. Since gradient descent does not solve problems analytically, it finds local minimums instead of the global minimum. Nevertheless, the solution generally can produce an adversarial example that is adequately effective. Standard gradient descent algorithms that have been used include Adam [76], stochastic gradient descent [37], **fast gradient sign method** (**FGSM**) [54], **projected gradient descent** (**PGD**) [101], and DeepFool [111].

Gradient-free Optimization. Methods in this category do not require gradient information. Existing studies have adopted two methods, particle swarm optimization, and genetic algorithm.

Particle swarm optimization (**PSO**) is a heuristic and stochastic algorithm that solve optimization problems by imitating the behavior of a swarm of birds, and it can search a vast space of candidate solutions without gradient information. Du et al. [37] exploited PSO as an initial phase to efficiently generate coarse-grained adversarial examples and then combined gradient descent to generate fine-grained AE if the model is white-box.

Genetic algorithm (**GA**) is another gradient-free optimization method that can search a large amount of space efficiently by mimicking natural selection, i.e., improving on each iteration through evolutionary methods such as crossover and mutation. Within each generation, candidate adversarial examples with higher fitness scores are more likely to evolve and become part of the next generation. GA is instrumental in solving a variety of optimization problems that are not well suited for standard optimization algorithms, including problems where the objective function is discontinuous, non-differentiable, stochastic, or highly nonlinear [184]. Since the genetic algorithm only requires the model's prediction results and confidence scores, it is model-agnostic and has been widely used for black-box attacks [9, 58, 74, 155, 184].

Machine Learning-based Methods. The above methods iteratively optimize the objective function and maybe computational heavy for real-time attacks. Another line of research adopts machine learning to imitate the generation of adversarial examples, i.e., use learning to substitute optimization. Chang et al. [22] trained a RNN to mimic the perturbations generated by the iterative FGSM and generate targeted adversarial examples in real-time. Gong et al. [48] used RNN to imitate the behavior of other non-real-time AE crafting techniques and combined reinforcement

Table 2. A Comparison of Unintelligible Speech and Adversarial Example Attacks on Audio-to-text Systems

Type	Year	Article	P. Idealism I	P. Idealism II	P. Idealism III	P. Idealism IV	Goal U	Goal T	Attack Target (Model Knowledge)	Attack Method	Airborne (Distance)	Performance Success	Performance Time	Open Res.
Unintelligible Speech	2015	Cocaine Noodles [163]	○	○	○	○	○	●	Google Now (**B**)	Inv-MFCC	✔(0.3 m)	–	–	–
	2016	Carlini et al. [19]	○	○	○	○	○	●	Google Now (**B**)	Inv-MFCC	✔(3 m)	60%	–	🔊
									CMU Sphinx (**W**)	GD	✔(–)	82%	30 h	
	2018	Bispham et al. [15]	○	○	○	○	○	●	Google Assist. (**B**)	Mangling	✔(–)	–	–	–
	2019	Abdullah et al. [2]	○	○	○	○	○	●	12 models (**B**)	Sig. proc.	✔(0.3 m)	80%	seconds	🔊
Audio Adversarial Examples	2017	Houdini [32]	○	○	○	○	●	○	DeepSpeech2 (**W**)	GD	✔(–)	–	–	–
									Google Voice (**B**)					
		Alzantot et al. [9]	○	●	○	○	○	●	CNN-based KWS (**B**)	GA	✖	87%	37s	</>
		Iter et al. [65]	○	○	○	○	○	●	WaveNet (**W**)	GD	✖	–	–	</>
	2018	Carlini et al. [20]	○	○	○	○	○	●	DeepSpeech (**W**)	GD	✖	100%	1 h	</> 🔊
		CommanderSong [199]	●	●	○	○	○	●	Kaldi (**W**)	GD	✔(1.5 m)	100%	2 h	🔊
									iFLYTEK (**B**)	Transfer	✔(–)	66.7%	–	
									DeepSpeech (**B**)		✖	0%		
	2019	Schonherr et al. [135]	○	○	●	○	○	●	Kaldi (**W**)	GD	✖	98%	2 min	</> 🔊
		Yakura et al. [190]	●	○	○	○	○	●	DeepSpeech (**W**)	GD	✔(0.5 m)	50-100%	–	</> 🔊
		Qin et al. [124]	●	○	●	○	○	●	Lingvo (**W**)	GD	✖	100%	–	</> 🔊
		Taori et al. [155]	○	●	○	○	○	●	DeepSpeech (**G**)	GA+GD	✖	35%	–	</> 🔊
		SirenAttack [37]	○	●	○	○	○	●	DeepSpeech (**W**)	PSO+GD	✖	100%	28.8 min	–
									7 models (**B**)			88.6%	100.7 s	
		Neekhara et al. [112]	○	○	○	●	●	○	2 models (**W**)	GD	✖	89.6%	–	🔊
		Kwon et al. [81]	○	○	○	○	○	●	DeepSpeech (**W**)	GD	✖	91.7%	1 h	–
		Li et al. [87]	●	●	●	○	●	○	Amazon Alexa (**G**)	GD	✔(2.3 m)	–	–	🔊
		Vadillo et al. [161]	○	○	○	●	●	○	CNN-based KWS (**W**)	GD	✖	–	–	</> 🔊
		Gong et al. [48]	○	○	○	●	●	○	CNN-based KWS (**G**)	RL+RNN	✖	43.5%	0.1 s	</>
		Liu et al. [97]	○	○	●	●	○	●	DeepSpeech (**W**)	GD	✖	100%	4–5 min	🔊
		Abdullah et al. [3]	○	●	○	○	●	○	7 models (**B**)	Reconstruct	✖	–	–	🔊
		Imperio [136]	●	○	●	○	○	●	Kaldi (**W**)	GD	✔(3 m)	–	80 min	🔊
		Szurley et al. [153]	●	○	●	○	○	●	DeepSpeech (**W**)	GD	✔(0.15 m)	–	–	–
		Nickel to Lego [58]	○	●	○	●	●	○	Google STT API (**B**)	GA	✖	86%	–	–
		Khare et al. [74]	○	●	○	○	●	●	2 models (**B**)	GA	✖	–	–	🔊
		Wu et al. [184]	○	●	○	○	○	●	Kaldi (**G**)	GA	✖	20-90%	–	–
		AudiDoS [47]	○	○	○	●	●	○	2 models (**W**)	GD	✔(0.3 m)	–	–	–
		Chai et al. [21]	○	○	○	●	●	○	DeepSpeech (**W**)	GD	✖	–	–	–
	2020	Devil's Whisper [28]	○	●	○	○	○	●	4 commercial (**B**)	GD	✔(0.3 m)	100%	–	</> 🔊
		Chang et al. [22]	●	○	○	●	○	●	KWS (**W**)	RNN	✔(4 m)	84.3%	0.096 s	–
		Metamorph [26]	●	○	●	○	○	●	DeepSpeech (**W**)	GD	✔(6 m)	90%	5–7 h	🔊
		Wang et al. [170]	○	○	○	●	○	●	2 models (**W**)	GAN	✖	92.3%	0.009 s	</>
		SEGA [175]	○	●	○	○	○	●	DeepSpeech (**G**)	GA	✖	98%	–	–
		AdvPulse [93]	●	○	●	●	○	●	CNN-based KWS (**W**)	GD	✔(3 m)	89.9%	–	🔊

● Applicable ○ Not applicable **B** Black-box **W** White-box **G** Grey-box ✔ Positive ✖ Negative – None </> Code
🔊 Audio U: Untargeted T: Targeted.
The Idealism of a Practical Attack Includes: (I) Over-the-air Attacks, (II) Attacking Black-box Systems, (III) Imperceptible Adversarial Perturbations, and (IV) Real-time Attacks.

learning for real-time untargeted attacks. Wang et al. [170] used a GAN to create AE faster than the optimization-based schemes.

5.3 Practical Audio Adversarial Examples

Audio adversarial examples generated with the above general methodologies are effective when digitally fed to the speech recognition system. However, as we mentioned in Section 3.4, practical attacks in the real world need to satisfy the following idealism: over-the-air, black-box knowledge, unsuspicious, and real-time. Most studies since 2019 have focused on achieving this idealism, which generally involves modifications to the above methodologies. We compare the attacks in Table 2 and elaborate on the idealism of practical adversarial examples in the following.

5.3.1 Over-the-Air Attacks. Adversarial examples generated in Section 5.2 generally fail when they are played over-the-air by loudspeakers [9, 20], therefore posing a limited real-world threat. The small perturbations in AE are sensitive to distortions introduced by the over-the-air process,

including device distortion, channel effect, and ambient noise [26, 28, 190, 199]. Therefore in over-the-air attacks, AEs are required to be robust against unknown environments and devices. Common enhancing approaches involve proactive methods that improve AE robustness to these distortion factors.

Robust to Device Distortion. Loudspeakers and recording devices distort audio due to their inherent frequency selectivity and electrical noises. Yuan et al. [199] measured their devices' distortion model and integrated it into the loss function. Since the measured model is device-dependent, it may not apply well to unknown devices. Chen et al. [26] used public datasets that contain measurements from heterogeneous sender-receiver pairs to generate generic AEs that work on unknown devices.

Robust to Channel Effect. The airborne channel introduces distortions due to reverberation, which is caused by multi-path transmission. Several studies [22, 26, 87, 124, 136, 153, 190] have proposed to mitigate the channel effect by modeling it with impulse responses and adding it to the AE generation process. Attackers may obtain the impulse responses by measuring in the targeted environment [190], using **room impulse response** (**RIR**) simulators [22, 87, 124, 136, 153], or leveraging public datasets [26, 93]. Since attackers may not have physical access to the environment in advance, they can collect or simulate impulse responses of diverse environments, e.g., rooms of varying size and surface reflection coefficients, and use them to generate generic AEs that work in varying environments [22, 26, 124, 136, 190].

Robust to Ambient Noise. Ambient noise widely exists in the air and can further distort AEs. Several studies [22, 93, 190, 199] introduced Gaussian white noise in the generation process to make the AE more noise-resistant.

Attackers may increase the AE's robustness to all three distortion factors. However, a few researchers [26, 199] suggested that the device distortion and channel effect may significantly impact over-the-air attacks more than ambient noise. So far, Chen et al. [26] achieved the longest over-the-air attack of 6 meters. They used a domain discriminator to exclude the device- and environment-specific features and capture the core impacts, making AEs more generic and robust when played over-the-air.

5.3.2 Attacking Black-box Systems. Most state-of-the-art voice assistants deploy their audio-to-text subsystems on the cloud, making it difficult to obtain full knowledge of the model and perform a white-box attack. Therefore, attackers may only query the model in a black-box manner and obtain the model's final output, e.g., transcriptions or confidence scores.

Query-based Optimization. Attackers may optimize an AE directly based on the model queries. Since gradient information is not available, studies have proposed using gradient-free methods such as genetic algorithm [9, 58, 74, 155, 184] and particle swarm optimization [37] introduced in Section 5.2. Compared with the efficiency and effectiveness of white-box attacks, these methods are generally inferior because they rely on intensively querying the model and the generated AEs are not theoretically optimal [37]. Taori et al. [155] proposed to improve the effectiveness of GA with gradient estimation [11], but the computation can be costly [87, 155], and it requires tens of thousands of queries. To increase the efficiency and success rate, Wang et al. [175] proposed a **Selective Gradient Estimation Attack** (**SGEA**) that can reduce the number of queries by 66%.

Transferability. Several studies [3, 28, 32, 199] have shown that audio adversarial examples can transfer between models, i.e., AEs generated for a white-box model (e.g., using gradient descent) may be effective against other black-box models. Studies in the image domain have shown that AEs can transfer between models even if they have different architectures and are trained on different datasets [42, 152]. However, Chen et al. [28] suggested that transferability's success depends on the similarity between the internal structure and parameters of the white-box and black-box

models. Transferability-based attacks generally have poor performance if the two models are very different [28, 199].

Substitute Model. Attackers may improve transferability by generating AEs against a substitute white-box model similar to the black-box model. A substitute model can be developed from disclosed model information [87] or by query-based model stealing [28, 118, 119, 159, 168]. To reduce the number of queries and address the complexity of the target model, Chen et al. [28] proposed to train a substitute model that partially approximates the black-box model on the most common and interesting phrases and further enhanced it with a well-developed ASR model.

5.3.3 Imperceptible Adversarial Perturbations. Attackers may use the following methods to increase the imperceptibility of adversarial perturbations.

Amplitude Optimization and Constraints. Amplitude directly affects the human perception of loudness. Thus perturbations with lower amplitudes are less likely to be noticed. As introduced earlier in Section 5.2.2, attacker can reduce the perturbation amplitude by incorporating amplitude loss as ℓ_{metric} and setting amplitude constraints, such as the perturbation's 0-norm [48], 1-norm [9, 199], 2-norm [20, 153], p-norm [32, 161], and max-norm [21, 47, 81, 112, 124].

Frequency Optimization and Constraints. Human ears do not perceive sounds at different frequencies equally. Taori et al. [155] exploited this phenomenon and limited the perturbations to only being in the high-frequency range, which is less audible. However, perturbations at high frequencies may be distorted by loudspeakers and microphones and become ineffective [190]. Researchers [87, 124, 135, 136, 153] achieved a more significant improvement by limiting the perturbations within frequency masking thresholds on the spectrum. They exploited the psychoacoustic hiding effect of human auditory systems, which refers to the phenomenon that a louder signal can make other signals at nearby frequencies imperceptible [94]. In this case, dominant frequencies in the original audio may mask perturbations at nearby frequencies. Attackers can incorporate frequency masking thresholds into ℓ_{metric} as imperceptibility loss [87, 124, 153] or apply input transformation [135] to make perturbations better fall into these frequency ranges.

Optimal Temporal Alignment. Perturbations are less likely to be noticed if they occasionally appear in the adversarial audio. Schonherr et al. [135] used the forced alignment algorithm to find and move the target transcription into parts of the original audio sample where human users are less likely to notice. Du et al. [37] used VAD [75] to find the active part of the original audio and only added noise to this region, which can increase the SNR of the adversarial audio. Chen et al. [26] applied an amplitude threshold to ignore small values perturbations and reduce the perturbation coverage.

Noise Feature Removal. The perturbations are less suspicious if they do not sound like noise. Liu et al. [97] used TVD as ℓ_{metric} to remove most of the impulses in an AE and make it sound more like the original audio. Chen et al. [26] and Li et al. [93] proposed to optimize the perturbation to make it sound similar to familiar background sounds, such as soft music, traffic sound, birds singing, or HVAC noise.

Choice of the Original Audio. A few works [28, 135, 184] suggested that the original audio may significantly influence AE's quality. They proposed to use non-speech sounds such as music as the original audio to improve both effectiveness and stealthiness.

5.3.4 Real-time Attacks. A real-time attack requires the attacker to generate or adjust the attack on-site. However, most of the work that reported time efficiency [20, 26, 81, 136, 199] requires more than an hour to generate an AE of only a few seconds. Researchers have proposed two types of approaches to achieve real-time attacks: (a) by generating universal perturbations that can be added on any benign audio, and (b) by reducing the AE generation time.

Universal Adversarial Perturbations. Universal perturbations make it easier to deploy adversary examples in the real world because attackers do not need to change the perturbation when the benign audio changes. So far, most studies [21, 47, 112, 161] have focused on untargeted attacks and involved various benign audio in the optimization process for the generation of universal perturbations. Li et al. [93] proposed the first targeted universal attack with sub-second adversarial perturbations generated by incorporating the varying time delay into the optimization process.

Reducing the Generation Time. Studies have shown that it is possible to reduce the AE generation time to several minutes by utilizing the parallel nature of GPUs [20, 135] or with more efficient optimization technologies such as weighted perturbation [97] and efficient GA [58]. Moreover, a few works [22, 48, 170] achieved near-real-time (<0.1s) attacks with machine learning models that are trained in advance to imitate the behavior of expert optimization methods.

5.4 Real-World Attacks on Voice Assistants

Despite the above research efforts, launching powerful AE attacks against state-of-the-art voice assistants in the real world remains an open question. Though several studies [3, 28, 32, 37, 58, 87, 199] have demonstrated AE attacks on commercial voice assistants or ASR services, none of them represents a sufficiently powerful attack. A powerful attack requires the attacker to achieve the four practical idealism simultaneously, i.e., generating imperceptible audio adversarial examples that can be played over the air to attack black-box systems in real-time. We believe it is worthwhile to comprehensively investigate the real-world threat of adversarial examples on voice assistants.

6 ATTACKING AUDIO-TO-IDENTITY

The audio-to-identity subsystem verifies or recognizes the speaker's identity. To have the malicious command accepted by the voice assistant, attackers need to bypass the speaker verification. This section introduces three categories of attacks against audio-to-identity: `Voice Spoofing`, `Unintelligible Speech`, and `Adversarial Examples`. The last two categories are similar to the methods introduced in Section 5, and we will mainly focus on their differences.

6.1 Voice Spoofing

Voice spoofing attacks bypass speaker verification by mimicking the VA owner's voice. Existing techniques mainly include replay, speech synthesis, voice conversion, and human impersonation.

6.1.1 Replay. Replay attacks spoof speaker verification by replaying speech samples recorded from a genuine target speaker in the form of either continuous speech recordings or concatenated speech samples extracted from several speech recordings [164]. Replay is so far the most viable voice spoofing attack because it is technically easy to perform and can effectively spoof existing speaker verification systems [95, 164, 165, 187]. Replay attack's success and applicability depend on the quality and content of speech recordings. Recently, Yoon et al. [196] proposed a method to improve the quality of replay attacks by recording with the same microphone as the speaker verification system and replaying the speech samples with a high-quality loudspeaker.

6.1.2 Speech Synthesis. Speech synthesis is a technique for generating artificial speech that sounds like a target speaker for any specific text. Unlike TTS synthesis, the speech synthesis attack needs to generate speaker-specific speech to fool speaker verification. There are two major approaches: unit selection and **statistical parametric speech synthesis** (**SPSS**). Unit-selection's basic idea is to synthesize speech by selecting appropriate sub-word units from a target speaker's utterance database. The quality of unit selection is directly affected by the quality of the database. Using an extensive database may seem like a solution, but it will cause other problems such as long waiting times [53]. By contrast, SPSS has grown more popular in recent years. A typical SPSS

system consists of a training part and a synthesis part. In the training part, parametric representations of speech are extracted from a speech database and then modeled by a set of generative models [202]. In the synthesis part, the attacker first generates speech parameters for a target word sequence using the trained generative model, e.g., HMM [158, 201], and then synthesizes a speech according to these parameters. Newer models have been proposed in recent years, such as Wavenet [113, 114], which can generate a specific speaker's speech with subjective naturalness. GAN are also used to enhance the quality of synthesized speech [13, 121, 133]. Lorenzo et al. [98] demonstrated synthesizing Obama's voice using GAN, WaveNet, and low-quality data.

6.1.3 Voice Conversion. VC techniques convert a source speaker's speech to a target speaker's. A conversion model is trained for this task with a set of utterances recorded from the source and target speakers. Voice conversion still faces a few technical challenges. One of them is non-parallel VC. Most of the VC techniques in the literature use parallel corpora to train the model, which contains speech of the same content spoken by both source and target speakers. Instead of recording a parallel corpus, using non-parallel corpora will be more labor-saving. In addition, cross-language [167, 215], small-corpus [46, 67], and many-to-one (converting several source speakers to a target speaker) [149, 156] voice conversion are all active research directions.

6.1.4 Human Impersonation. Human impersonation refers to the attacker mimicking a target speaker by mouth without computer-aided technologies [41]. Although human impersonation is easier to perform compared to other spoofing methods, the impact of such attacks mainly depends on the impersonator's ability and remains undetermined. Several studies [83, 84] show that non-professional impersonators can spoof GMM-UBM ASV systems if the impersonator's natural voice is similar to that of the target speaker. Nevertheless, the results in [60, 103] suggest that even a professional impersonator may not be able to significantly degrade the performance of GMM-UBM and i-vector ASV systems if his or her natural voice is very different from that of the target speaker.

We refer readers to [41, 110, 186] for more comprehensive surveys on voice spoofing attacks.

6.2 Unintelligible Speech

Similar to the attacks in Section 5.1, unintelligible sounds may be misclassified by audio-to-identity systems as a registered speaker label. Abdullah et al. [2] proposed to generate unintelligible speech by perturbing a human voice with signal processing techniques in time, phase, and frequency domains and presented successful attacks on Microsoft Azure.

6.3 Adversarial Examples

An AE attack on audio-to-identity systems aims at crafting an audio sample that sounds like speaker A to a human listener but is misclassified by the system as uttered by other random speakers (untargeted attack) or a specific speaker B (targeted attack). Compared with spoofing attacks, AE attacks raise less suspicion even with the victim's presence, and in most cases, they do not require the adversary to collect any audio clips from the victim user [92]. Existing studies have demonstrated untargeted and targeted attacks on a wide variety of speaker verification [78, 91, 104, 172, 208] and speaker identification [24, 37, 49, 88, 89, 92, 173, 189] systems. We summarize and compare existing studies in Table 3.

6.3.1 Adversarial Example Generation. Similar to the adversarial examples for audio-to-text, adversarial examples for audio-to-identity systems are normally generated by optimizing an adversarial perturbation and adding it to the original speaker samples. Likewise, adversarial perturbations can be generated for audio waveforms [49, 89], acoustic features [78, 91], or spectrograms [104, 173] using optimization-based methods such as FGSM [49, 78, 91, 92, 172], **Stochastic**

Table 3. A Comparison of Adversarial Example Attacks on Audio-to-Identity Systems

Type	Year	Article	P. Idealism I	P. Idealism II	P. Idealism III	P. Idealism IV	Goal U	Goal T	Attack Target (Model Knowledge)	Attack Method	Airborne (Distance)	Performance Success	Performance Time	Open Res.
Audio Adversarial Examples	2018	Kreuk et al. [78]	○	○	○	○	●	○	DNN E2E (**W, B**)	FGSM	✖	–	–	–
		Gong et al. [49]	○	○	○	○	●	○	WaveRNN (**W**)	FGSM	✖	–	–	–
	2019	SirenAttack [37]	○	●	○	○	○	●	6 models (**B**)	PSO	✖	99.45%	376.4 s	–
		FakeBob [24]	●	●	○	○	●	●	3 models (**B**)	NES+IGS	✔(0.5 m)	100%/95%	3.8 min	</> 🔊
									Talentedsoft (**B**)		✖	–/100%	–	
									Microsoft Azure (**B**)	Transferability	✔(0.5 m)	77%/9%		
		Wang et al. [172]	○	○	○	○	●	○	DNN E2E (**W**)	FGSM	✖	–	–	–
		Marras et al. [104]	○	○	○	●	○	❓	VGGVox (**W**)	SGD	✖	10-80%	–	–
		Abdullah et al. [3]	○	○	○	○	●	○	Microsoft Azure (**B**)	Audio reconstruct	✖	–	–	–
	2020	Xie et al. [189]	●	○	○	●	○	●	DNN x-vector (**W**)	–	✖	90%	0.015 s	–
		Li et al. [91]	○	○	○	○	●	○	GMM i-vector (**W, B**)	FGSM	✖	–	–	</> 🔊
		Li et al. [92]	●	○	○	○	●	●	DNN x-vector (**W**)	FGSM	✔(1 m)	50%	–	–
		Li et al. [88]	○	○	○	●	●	●	SincNet (**G**)	Generative network	✖	–	–	–
		VMask [208]	○	○	●	○	○	●	VGGVox (**G**)	SGD	✖	95%	–	–
									Microsoft Azure (**B**)		✖	70%		
									Apple Siri (**B**)		✔(–)	67.5%		
		Li et al. [89]	○	○	○	●	●	○	SincNet (**W**)	ATN	✖	–	0.042 RTF	</>🔊
		Wang et al. [173]	○	○	●	○	○	●	DNN x-vector (**W**)	GD	✖	98.5%	–	🔊
		AdvPulse [93]	●	○	●	●	○	●	DNN x-vector (**W**)	GD	✔(3 m)	89.3%	–	🔊
		FOOLHD [137]	○	○	●	○	●	●	DNN x-vector (**W**)	GCA	✖	99.6%/99.2%	–	</>🔊

● Applicable ❓ Probably applicable ○ Not applicable **B** Black-box **W** White-box **G** Grey-box ✔ Positive ✖ Negative – None </> Code 🔊 Audio U: Untargeted T: Targeted RTF: the ratio of the processing time to the input duration.

The Idealism of a Practical Attack Includes: (I) Over-the-air Attacks, (II) Attacking Black-box Systems, (III) Imperceptible Adversarial Perturbations, and (IV) Real-Time Attacks.

Gradient Descent (SGD) [104, 208], **Iterative Gradient Sign (IGS)** [24], **Particle Swarm Optimization (PSO)** [37], and so on. The methodologies to generate adversarial examples for audio-to-text and audio-to-identity systems are similar. For example, Du et al. [37] and Li et al. [93] used the same method to attack speech recognition and speaker recognition systems. Due to the similarity, we mainly introduce the unique issues and practical idealism of attacking audio-to-identity systems.

6.3.2 Unique Issues of Attacking Audio-to-Identity Systems.

— **Score Threshold.** Most audio-to-identity systems generate a score for a speaker sample and apply a threshold to make decisions. A successful attack needs to score higher than the threshold, while the score is unavailable from a black-box system that only outputs decisions.
— **Gender.** Inter-gender attacks are more complicated than intra-gender due to the significant difference between male and female voices.

Therefore, targeted, inter-gender, over-the-air, and decision-only black-box adversarial examples are the most practical yet the most challenging attacks against audio-to-identity systems [24].

6.3.3 Practical Audio Adversarial Examples. Similarly, we discuss the idealism of a practical attack.

Over-the-air Attacks. To mitigate over-the-air distortions, an adversary may measure or simulate the RIR and incorporate it into the AE generation [92, 189].

Attacking Black-box Systems. Traditional methods cannot be directly used to attack black-box systems due to the absence of necessary information, e.g., gradient and score threshold. To overcome this problem, researchers have proposed to approximate the information via NES-based gradient estimation and threshold estimation [24] or use non-gradient-based optimization methods such as PSO [37]. An attacker may also generate AEs for a white-box system and then use them for black-box attacks. Several studies [24, 78, 91, 208] have demonstrated the success of transferability in cross-dataset, cross-feature, cross-parameter, and cross-architecture circumstances.

However, when the gap between source and target systems is large, the transferability rate may be limited [24].

Imperceptible Adversarial Perturbations. Restricting the max-norm is usually sufficient to craft imperceptible adversarial perturbations [24]. However, an attacker may still employ psychoacoustic masking to make the perturbations more imperceptible [173, 208]. She may also generate AEs based on non-speech sounds (e.g., music) for stealthier attacks [173]. Recently, Shamsabadi et al. [137] borrowed the idea of speech steganography and generated imperceptible perturbations using a frequency-domain **Gated Convolutional Autoencoder** (**GCA**).

Real-time Attack & Universal Adversarial Perturbations. Several studies [88, 93, 189] have demonstrated universal adversarial perturbations that can be directly added to an arbitrary speaker's utterance and make the system identify the voice as any target speaker label. Such universal perturbations can save the considerable time of training perturbations for each voice input, making real-time attacks possible. Another line of research seeks to reduce the generation time by training a neural network, e.g., **adversarial transformation networks** (**ATNs**) [89], that can transform an input into an adversarial example in real-time.

Master Voice. Master voices are adversarial utterances optimized to match against a large number of users by pure chance. Such attacks allow targeting a large speaker population without having specific knowledge of individuals or their speech models, which is a unique threat for audio-to-identity systems. Marras et al. [104] have demonstrated the existence of master voices that can match approximately 20% of females and 10% of males without any knowledge about the population.

7 ATTACKING TEXT-TO-INTENT

In this category, attacks exploit audio inputs that text-to-intent systems interpret as having a different meaning than that understood by humans. Due to the gap of speech interpretation between humans and machines, the user's benign speech may unintentionally invoke a `Malicious Skill` developed by the attacker. An attacker can also generate a benign command that can covertly cause a benign third-party skill's sensitive behaviors. We categorize existing attacks into two categories: (a) invocation of malicious applications and (b) initiation of sensitive behaviors.

7.1 Invocation of Malicious Applications

For extended functionalities, voice assistants such as Amazon Alexa and Google Home allow the development of third-party applications (called skill or action) that users can access through the VA service. Before a new skill is published, it must pass a vetting process that verifies that it meets the necessary content and privacy policies. However, several studies showed that many policy-violating skills already exist on the market [57, 147], yet current skill vetting mechanisms fail to detect and suspend these skills in practice [29]. An attacker may exploit such weakness and develop a malicious skill that phishes sensitive information and eavesdrops on users [18]. The key to such an attack is how to make users unintentionally invoke malicious skills.

Squatting Attacks. Kumar et al. [79] and Zhang et al. [211] found that systematic errors appear consistently in speech interpretation, which attackers may exploit to design specific invocation names that route a user to a malicious application without their knowledge. This type of attack is known as skill squatting or voice squatting attacks. There are two ways to cause misinterpretations. (a) By designing an invocation name that has similar pronunciation but different spelling (i.e., homophones) to the target skill [79, 211]. For example, a user intending to invoke a benign skill called "capital one" may unintentionally open a malicious skill named "capital won." An attacker may target a specific group of individuals based on their dialect regions or genders as their pronunciations are more predictable [79]. (b) By designing an invocation name as a variation in

how a target skill is spoken [211]. For example, a user may invoke the skill "capital one" by saying "open capital one please," but it may trigger a malicious skill if its name is "capital one please." Such attacks are feasible because the NLU of VA favors the longest matching skill name when processing voice commands [211]. In practice, an attacker may combine both methods to craft a malicious skill that bypasses potential name regulations. A recent study [213] suggests that the intent classifier in NLU is the root cause of the misinterpretation behind squatting attacks because it determines users' semantic intents and fixes the ASR's potential transcription errors.

7.2 Initiation of Sensitive Behaviors

A study [143] has identified that 5.55% of all third-party applications perform sensitive behaviors, either via controlling the system to perform a sensitive action or obtaining sensitive information. For example, "unlock my front door" and "how much money do I have" are sensitive behaviors that may put a user's safety and privacy at risk. An attacker may exploit the intrinsic gap of understanding between humans and machines to initiate an application's sensitive behaviors using seemingly benign voice commands.

Missense Attacks. Bispham et al. [15] presented missense attacks on the natural language understanding used by third-party skills. NLU takes account of both the individual words and the syntactic structure of the command to determine a user's intent. However, they found that either of these two aspects alone may be sufficient to trigger a sensitive behavior in some instances, thereby hiding the malicious intent from humans. For example, "how much ice-cream do I have" and "what is a current" may both trigger a bank application to reveal the user's current account balance.

8 DEFENSE

We classify the defensive strategies into two categories: attack detection and attack prevention. Detection methods aim at detecting an attack's occurrence, while prevention methods ensure proper voice assistant behaviors even in the presence of an attack. We systematize existing defenses within the two categories and compare them by the attacks they mitigate and their properties in Table 4.

8.1 Detection Methods

Existing detection methods mainly identify attacks by three types of features: lack of live speakers, false speaker identity, and unique attack properties. The first two types are shared by most attacks, while the last one depends on the specific attack method. Once an attack is detected, the voice assistant may reject the command and alert the user.

8.1.1 Detection by Speaker's Liveness. Liveness detection aims at verifying whether a live speaker generates the command. These methods are based on the idea that legitimate voice commands should only come from human users. However, the majority of attacks leverage loudspeakers or other transducers to generate malicious commands. Though liveness detection is mainly proposed for detecting voice spoofing attacks, it can potentially be used against other attacks such as inaudible signal attacks and adversarial examples [24], making it one of the most effective and all-purpose defense strategies.

Challenge-Response. Challenge-response is a scheme that asks the user a question (challenge) and requires a response within a limited time. The voice assistant will not execute the received command unless it receives the correct response. Similar to CAPTCHA, audio challenge-response can determine whether or not the speaker is a human, and it has been suggested as a defense to voice assistants [19, 23, 148, 198]. However, the extra layer of interaction often comes at the cost of usability. It may also be circumvented if the attacker can respond to the challenge.

By Human Features. Speaking is a complex process involving various parts of the human body, e.g., the mouth, vocal tract, vocal cord, lung, and so on. The generation of the human voice

relies on these organs' movement, which inevitably introduces measurable characteristics other than sounds, such as the airflow of breath, movement of the mouth, and vibration of the body. A defender could measure these speaking-related characteristics and use them to verify whether a live human speaker generates the sound. The measurement may require extra devices or user cooperation.

(1) *Breath:* humans speak and breathe simultaneously, causing the correlation between the sound and breath-related characteristics, such as the pop noise [109, 144, 214], breathing rate [123], and airflow pressure [179].
(2) *Mouth motion:* the motion of the mouth, tongue, and vocal tract are the main factors that create different voices. Existing studies have suggested matching the received voice with mouth motion, which can be directly measured with wireless signals [107], an ultrasound Doppler radar [209], or a video camera [30, 100]. The motion of these vocal organs also changes the location of phonemes (the smallest units of speech) in the vocal tract [210] and the ear canal pressure [141], which can be measured and matched with the received voice.
(3) *Body vibration:* besides airborne transmission, the human voice also propagates through the body via vibrations. A defender may exploit the correlation between audio measured in the air and on the body. Existing studies have proposed to measure the body vibration with the motion sensors in smartphones [138] and wearables [43, 169] or using contact microphones as peripherals [131] and in augmented reality headsets [139, 140].

By Loudspeaker Features. A defender can also detect attacks by verifying that a loudspeaker generates the sound through the following characteristics.

(1) *Signal distortion:* loudspeakers introduce distortions to the generated sound due to their non-uniform frequency response and circuit noise. A defender can train a detector model to detect such distortions in acoustic features, which human speakers do not possess [5, 17, 51, 123, 181]. Though an attacker may try to escape this detection by compensating for the device distortions, a recent work [176] suggests that loudspeakers will always cause ringing artifacts in the time domain or spectrum distortions in the frequency domain.
(2) *Magnetic field:* the electromagnets in most loudspeakers emit magnetic fields when generating sounds, which can be measured with the magnetometer in smartphones [25]. However, this method requires the user to move the smartphone and keep it close to the speaker.

8.1.2 Detection by Speaker's Identity. This type of defense verifies whether a speaker of legitimate identity generates the command. The system will reject voice commands that do not correspond to the owner's identity. However, it generally requires training a user profile beforehand.

By Voiceprint Features. Speaker verification with voiceprint [61] was originally designed to protect voice assistants from being misused, and it has been suggested by many as a straightforward defense against various types of attacks [19, 36, 72, 212]. However, as discussed in Section 6, traditional **automatic speaker verification** (**ASV**) systems are vulnerable to voice spoofing, unintelligible speech, and adversarial example attacks, rendering them insufficient. Nonetheless, traditional speaker verification can still provide an essential layer of protection by increasing the difficulty for other attacks at a low cost.

By Non-voiceprint Features. Due to the incompetence of traditional ASV systems, researchers seek to verify speakers with alternative biometrics available during speaking.

(1) *Throat:* due to the differing conduction properties of the human tissue, throat vibrations vary from human to human, which can be measured by a throat microphone [131] or a mmWave radar [86] and used for speaker recognition.

(2) *Vocal tract:* the vocal tract, including the static shape and dynamic movements, exhibits individual uniqueness during speaking. Lu et al. [99] proposed a user authentication system on smartphones by sensing the vocal tract with acoustic radars.
(3) *Sound field:* the sound field created around a speaker is affected by the unique appearance of the human mouth, face, and head. Yan et al. [191] showed that the acoustic biometrics embedded in sound fields could be used to verify speakers using two microphones on a smartphone.
(4) *Pop noise:* Wang et al. [174] showed that the relationship between phonemes and pop noise is unique and can be utilized for user authentication. The pop noise is measured using the built-in microphone, which needs to stay close to the mouth (2–6 cm).

Localization. In some scenarios, a legitimate command should only come from "legitimate" locations, regardless of who the speaker is. For example, on a vehicle, only the person sitting on the driver's seat is allowed to give safety-sensitive commands; in a home, only legitimate users can access the physical space, while the attacker needs to exploit remote-controllable devices. A defender may verify the speaker's legitimacy by localizing the command with multiple microphones and checking whether it conforms to the location rules [16, 85, 177].

8.1.3 Detection by Unique Attack Features. Apart from the speaker's liveness and identity, attacks also possess unique patterns that can be used as discriminative traces to detect them.

Inaudible Signal Features. Attacks in this category emit physical signals other than sound and normally modulate the signals. A defender may detect such attacks by looking for these physical signals and modulation patterns.

(1) *Signal activities:* a straightforward way to detect the attacks is to monitor if unusual signals other than audible sounds appear in the environment, such as EM waves [72] and ultrasound [102]. However, this method generally requires extra hardware and may easily report false alarms.
(2) *Signal behaviors:* though attackers can use ultrasound, EM waves, and light to inject audio into microphones, these signals behave differently than audible sound. For example, a device with multiple microphones can measure an audible sound nearly equally on all channels, while it is difficult for an attacker to inject equal audio into multiple microphones. Sugawara et al. [148] proposed to detect laser attacks by fusing multiple microphones. Zhang et al. [204] managed to detect ultrasound attacks by the sound field difference at multiple microphones.
(3) *Modulation:* inaudible signals are generally modulated before emission and demodulated by vulnerable sensor hardware. A defender may distinguish inaudible commands from normal ones based on the signal patterns caused by such a process. Studies have shown that demodulated signals differ from normal signals in the frequency range below 50 Hz range [129], between 500 and 1,000 Hz [193, 205], and between 5 kHz–20 kHz [194], depending on the attack method.

Voice Spoofing Features. Voice spoofing techniques, such as replay, synthesis, and conversion, may generate slight distortions in the audio signal that make it different from normal ones. For example, the cut-and-paste process of speech concatenation changes the audio's pitch and MFCC contours [166]; voice conversion does not keep the original phase information [185, 188] and reduces the pair-wise distance between consecutive feature vectors [7]. A defender may distinguish genuine and spoofing samples leveraging the differences of temporal, spectral, and spatial features with machine learning models [52, 69, 71, 122, 178, 182]. Several studies proposed to detect replay attacks by other patterns, such as the similarity to previous recordings [142] and far-field patterns when the attack happens from a distance to the voice assistant [165, 166]. The ASVspoof

challenges [157] were introduced to push forward the state-of-the-art in detecting voice spoofing attacks. As the spoofing countermeasures evolve, a study [96] suggested that the detection models are vulnerable to adversarial examples as well.

Adversarial Example Features. AEs differ from benign audio in the following ways:

(1) *Model transferability:* due to the low transferability, adversarial examples that successfully attack one model may be predicted as entirely different results on other models, while normal commands will always be recognized similarly. Thus, a defender may detect adversarial examples by comparing the predictions of different models [203].
(2) *Temporal dependency:* natural audio sequences have an explicit temporal dependency, i.e., correlations in consecutive waveform segments. Yang et al. [195] found that adversarial examples, however, fail to preserve the original sequence's temporal information. A defender may detect AEs by clipping the audio into sections and measuring if their transcriptions are consistent with their counterparts in the entire audio. However, a recent work [206] proposed a method to preserve the temporal dependency in AEs, which may circumvent this detection.
(3) *Classification:* the defender may treat the detection of adversarial examples and unintelligible speech as a classification problem [6, 19, 35, 56, 68, 90, 134]. By training a detector model with benign audio and attack samples, a defender may detect known attacks with high accuracy, but it requires a large amount of attack audio, and the performance may degrade dramatically for unknown attacks [134].

Malicious Skill Features. A malicious skill may have a similar invocation name with a benign skill. Based on this pattern, Zhang et al. [211] developed a skill-name scanner to detect malicious skills by identifying suspicious variations of benign invocation names. They also proposed a context-sensitive detector that alerts the user of suspicious responses from a malicious skill.

8.2 Prevention Methods

Prevention methods ensure proper behaviors of voice assistants even in the presence of an attack. We classify them into four types: hardware enhancement, software enhancement, audio transformation, and signal injection.

8.2.1 Hardware Enhancement. Inaudible signal attacks exploit hardware vulnerabilities to inject malicious commands. Therefore, a defender may prevent such attacks by enhancing the hardware.

Microphone Redesign. The microphones of voice assistants are designed to receive audible commands. However, due to their miniature structure, most microphones can also receive ultrasounds and enable ultrasound-based attacks. A defender can redesign the microphone's layout to suppress the sensitivity to any acoustic vibration whose frequencies are in the ultrasound range [193, 194, 205]. Researchers [193, 205] also suggest careful microphone circuit designs that reduce nonlinearity.

Physical Barriers. A defender may also prevent malicious signals from getting into the microphone with external physical barriers. For example, a light-blocking barrier [148], shielding of the headphone cable [72, 80], and a soft woven fabric [194] can prevent microphones from receiving light-based, EM-based, and conductive ultrasound-based malicious signals, respectively. As the physical barriers are device-level, they may apply to off-the-shelf microphones.

8.2.2 Software Enhancement. Similarly, a defender may mitigate software vulnerabilities and make the software components more robust to attacks.

Adversarial Training. A primary way to improve a model's robustness is to retrain it with more representative samples. Adversarial training includes the attack samples into the training set, and it has been proved effective in resisting adversarial examples on speech recognition [12, 37, 150, 151, 172]. However, a limitation of adversarial training is that it needs prior knowledge of the attack and adequate adversarial examples for training, which means that it is weak in preventing unknown attacks [50]. Besides, the retraining process involves extra overhead.

Strict Skill Certification. Several studies suggested mitigating malicious third-party skills with strict skill certification [29, 57]. Third-party skills should be automatically reviewed via voice interaction or back-end code analysis before being put in the store, and they should be periodically checked and removed if broken. For example, the review should carefully examine privacy-related contents and ensure the description is consistent with the real functionality. Kumar et al. [79] suggested VA platform providers perform a word or phoneme-based analysis of a new skill's invocation name to determine whether it may be confused with skills that are already registered.

8.2.3 Audio Transformation. A defender may perform extra transformation processes to the audio measured by the hardware before sending it to the software components. This defense is based on the idea that adversarial examples and unintelligible speech attacks are sensitive to audio transformations, while benign audio is only slightly affected. We identify three types of audio transformation: sampling rate conversion, amplitude conversion, and audio compression.

Sampling Rate Conversion. Several studies proposed to change the audio's sampling rate, e.g., by downsampling [19, 28, 37, 195, 199] or setting a dynamic sampling rate [154]. If the sampling rate that the model accepts is fixed, a defender needs to recover the signal after sampling rate conversion, e.g., by upsampling after downsampling [28, 195]. According to the Nyquist-Shannon sampling theorem, the sampling rate needs to stay above twice the highest frequency of the original audio to avoid distortion. A sampling rate conversion may change the added adversarial perturbations and make an adversarial example fail. However, if the attacker knows the up/downsampling rates of the defense, she could train an AE robust against it [28].

Amplitude Conversion. By slightly changing the audio's amplitude, the adversarial perturbation can be disrupted as its amplitude is usually small in the input space. Methods of amplitude conversion include local signal smoothing (using moving average or median filter, etc.) [28, 37, 195], quantization [195], and other noise reduction methods [55, 154]. However, Chen et al. [24] suggested that these methods are ineffective in mitigating adversarial examples against speaker verification.

Audio Compression. A defender may also utilize audio compression techniques, which reduce the amount of data in the recorded waveform, as a defense against adversarial examples. Studies have proposed to use audio codecs such as **Adaptive Multi-Rate** (**AMR**), MP3, **Advanced Audio Coding** (**AAC**), G.729, Speex, and so on. [10, 33, 125, 135, 207], or exploit the ensemble of multiple codecs [127].

Despite the efficiency of audio transformation in defeating adversarial examples, attackers aware of the transformation parameters can optimize the AEs to circumvent the defense [20, 195]. As such, the use of any transformation method alone may be insufficient to defend against more advanced attacks. Rajaratnam et al. [127] suggested a combined deployment of transformation methods to provide a more robust defense.

8.2.4 Signal Injection. Besides transforming the audio, a defender may disrupt an attack by injecting new signals into the audio.

Adding Noise. Compared to normal voices, adversarial examples are more sensitive to noise. A defender may intentionally add random noises to the recorded audio while ensuring that the

noise barely affects the original system's performance [82, 106, 126, 199]. However, random noise may not work well against over-the-air AEs designed to be robust against environmental noise. To overcome this problem, Du et al. [38] recently proposed a multi-fragment noise padding method to destroy the continuity of adversarial examples.

Reactive Cancellation. Inaudible signal attacks that exploit hardware nonlinearity will leave predictable traces in the recorded audio or the environment. A defender may detect such traces, predict the attack signal hidden in the audio, and generate an opposite signal that can cancel out the attack signal in the audio and therefore neutralize the attack's effect [62, 80, 193, 205].

8.3 Comparison of Defensive Methods

To help VA designers select proper countermeasures, we compare the above-mentioned methods by the attacks they are designed to (or potentially can) mitigate and their properties in Table 4. As a quantitative performance comparison of individual articles can be biased due to the absence of unified metrics, datasets, and setups, we provide a high-level qualitative comparison of the systematized defense methodologies shared by groups of articles.

8.3.1 Mitigated Attacks. The defenses are first compared by their ability to detect or prevent the six types of attacks against voice assistants, i.e., **normal speech** (**NS**), **voice spoofing** (**VS**), **unintelligible speech** (**US**), AE, **inaudible signal** (**IS**), and **malicious skill** (**MS**). We note that though a method may be proposed against one type of attack, it can potentially mitigate other attacks as well. For example, liveness and identity-based methods were mainly designed to detect voice spoofing attacks. However, they may also detect unintelligible speech, adversarial examples, and inaudible signal attacks as they do not employ live speakers or legitimate identities. We mark our assumption of such cases as "probably applicable" in Table 4 if it has not been confirmed by existing studies. The comparison suggests that liveness and identity-based detection can (potentially) mitigate nearly all types of attacks except malicious skills, while other methods can only work against one or two types of attacks.

8.3.2 Properties. From a system designer's perspective, a proper defense strategy should provide maximal usability and reliability at a minimal implementation cost. We compare the defensive methods by the following properties that may affect the design choice.

- **Type of Implementation:** a defense may involve modifications to a VA's software, hardware, or both. In general, a software update is implemented faster than a hardware redesign, especially on off-the-shelf devices.
- **Need for Extra Devices:** some mitigations require extra devices other than the VA terminal device, which introduces extra cost and may be inconvenient for the user to carry.
- **Need for User Cooperation:** a few methods require the user to cooperate by performing specific behaviors that are not necessarily required during regular VA interactions.
- **Requirement on User-to-device Distance:** some methods require a user to stay within a specific distance to the device, e.g., within 10 cm (close) or hand-held distances (medium).
- **Vulnerability to Attacks:** defenses may be circumvented by attacks that are aware of the mitigation strategy.

Our comparison shows that while liveness and identity-based detections can mitigate a broad category of attacks, most of them require extra devices, user cooperation, or limited user-to-device distance that reduce their usability in practice. Though there is no ideal solution, a designer can assess and combine several methods with a tradeoff to meet their own VA applications' demands.

Table 4. A Qualitative Comparison of Existing Defensive Methods by the Attacks they Mitigate (NS: normal speech, VS: voice spoofing, US: unintelligible speech, AE: adversarial example, IS: inaudible signal, MS: malicious skill) and their Properties (the type of implementation, the need for extra devices and user cooperation, the requirement on user-to-device distance, and vulnerability to attacks that can circumvent the defense)

Categories		Defensive Methods		Mitigated Attacks						Properties					Relevant Articles
				NS	VS	US	AE	IS	MS	Type	Dev.	Coop.	Dist.	Attack	
Detection	Liveness	Challenge-response		◐	◐	◐	◐	◐	○	SW	✖	✔	Any	✔	[19, 23, 148, 198]
		Human features	Breath	◐	◐	●	●	●	○	SW	☰	✔	Close	✖	[109, 123, 144, 179, 214]
			Mouth motion	◐	◐	●	●	●	○	SW	☰	✔	Medium	✖	[30, 100, 107, 141, 209, 210]
			Body vibration	◐	◐	●	❓	●	○	SW	☰	☰	Close	✖	[43, 131, 138–140, 169]
		Loudspeaker features	Sig. distortion	◐	◐	❓	❓	❓	○	SW	✖	✖	Any	✔	[5, 17, 51, 123, 176, 181]
			Magnetic field	◐	◐	●	●	○	○	SW	✖	✔	Medium	✖	[25]
	Identity	Voiceprint features		●	○	◐	◐	◐	○	SW	✖	✖	Any	✔	[19, 36, 72, 73, 212]
		Non-voiceprint features	Throat	●	●	❓	❓	❓	○	SW	✔	✖	Close	✖	[86, 131]
			Vocal tract	●	●	❓	❓	❓	○	SW	✖	✔	Medium	✖	[99]
			Sound field	●	●	❓	❓	❓	○	SW	✖	✔	Medium	✖	[191]
			Pop noise	●	●	❓	❓	❓	○	SW	✖	✔	Close	✖	[174]
		Localization		●	●	❓	❓	❓	○	SW	☰	✖	☰	✖	[16, 85, 177]
	Unique Attack Features	Inaudible features	Sig. activity	○	○	○	○	●	○	SW	✔	✖	Any	✖	[72, 102]
			Sig. behavior	○	○	○	○	●	○	SW	✖	✖	Any	✖	[148]
			Modulation	○	○	○	○	●	○	SW	✖	✖	Any	✖	[129, 193, 194, 205]
		Spoofing features		○	●	○	○	○	○	SW	✖	✖	Any	✖	[7, 142, 165, 166, 185, 188]
		AE features	Transferability	○	○	❓	●	○	○	SW	✖	✖	Any	✖	[203]
			Temporal dep.	○	○	○	●	○	○	SW	✖	✖	Any	✖	[195, 206]
			Classification	○	○	●	●	○	○	SW	✖	✖	Any	✖	[6, 19, 35, 56, 68, 90, 134]
		Malicious skill features		○	○	○	○	○	●	SW	✖	✖	Any	✖	[211]
Prevention	Hardware Enhance	Microphone redesign		○	○	○	○	●	○	HW	✖	✖	Any	✖	[193, 194, 205]
		Physical barriers		○	○	○	○	●	○	HW	✖	✖	Any	✖	[72, 80, 148, 194]
	Software Enhance	Adversarial training		○	○	❓	●	○	○	SW	✖	✖	Any	✔	[12, 37, 150, 151, 172]
		Strict skill certification		○	○	○	○	○	●	SW	✖	✖	Any	✖	[29, 57, 79]
	Audio Transform	Sampling rate conversion		○	○	●	●	○	○	SW	✖	✖	Any	✔	[19, 28, 37, 154, 195, 199]
		Amplitude conversion		○	○	❓	●	○	○	SW	✖	✖	Any	✔	[28, 37, 55, 154, 195]
		Audio compression		○	○	❓	●	○	○	SW	✖	✖	Any	✔	[10, 33, 125, 127, 135, 207]
	Signal Injection	Adding noise		○	○	○	◐	○	○	SW	✖	✖	Any	✔	[38, 82, 106, 126, 199]
		Reactive cancellation		○	○	○	○	◐	○	Both	✔	✖	Any	✖	[62, 80, 193, 205]

● Applicable ◐ Partially applicable ❓ Probably applicable ○ Not applicable ✔ Positive ✖ Negative ☰ Case by case.

9 DISCUSSION

9.1 Directions for Future Research

As voice assistants become increasingly powerful, future research should seek to more extensively analyze and enhance VA's security with strong attacks and defenses in practical settings. In particular, we identify the following trends.

Adversarial Inaudible Signal Attacks. Inaudible signal attacks generally suffer from low audio quality due to the distortions inevitably introduced by the hardware and signal transmission. The distortions have decreased the attack success rate, especially in attacking speaker verification systems, as they require high-quality audio. An attacker may overcome this problem by combining inaudible signal attacks and over-the-air adversarial examples, i.e., optimizing the inaudible signals against a target machine learning model to achieve a higher success rate. However, modeling the physical signals will be a major challenge for such attacks.

Adversarial Examples against both Audio-to-Text and Audio-to-Identity. Existing adversarial examples are generated either for audio-to-text or audio-to-identity systems. However, a practical attack may require attacking both systems simultaneously. For example, to attack the wake-word detection, the attacker must generate an adversarial example that neither sounds like the wake-word nor spoken by the victim user. Future voice assistants may also perform stricter user authentication, e.g., with text-independent speaker verification that checks the speaker's identity behind every received command. In this case, an adversarial example needs to cause misprediction of both speech recognition and speaker verification systems.

Universal Adversarial Perturbations. Though several works [88, 93, 189] have shown the existence of universal adversarial perturbations, the effectiveness of such an attack remains questionable because universality and imperceptibility are essentially contradictory. Existing studies

achieved "universal" perturbations only on a small set of benign audio. However, if an adversarial perturbation can truly cause a targeted misprediction, *whatever* benign audio is added, intuitively, the perturbation will resemble the targeted misprediction and become perceptible. We call for future research to investigate the boundary of universal adversarial perturbations.

Combining Existing Attack Vectors. It is possible to build more powerful attacks by combining existing methods. For example, Mitev et al. [108] used an inaudible signal attack to trigger a carefully crafted malicious skill stealthily, allowing an adversary to arbitrarily control and manipulate interactions between the user and other benign skills. Similarly, an attacker can combine adversarial examples and malicious skills, inaudible signals and adversarial examples, voice spoofing and inaudible signals, and so on, to achieve various new attack scenarios.

Robust Defense. As we showed in Section 8, there are seldom defenses that achieve high usability, reliability, and security at a low implementation cost. Though we remain suspicious of such an "ideal" defense, we call for researchers to seek robust defense strategies that can significantly increase the bar for attacking voice assistants at an acceptable cost. In particular, a robust defense should generalize well to varying application scenarios and attack modalities, and it should be able to survive and evolve in the non-stop arm race between attacks and defenses.

Unified Metrics and Security Standards. A gap that hinders the comparison of existing works is the lack of unified evaluation metrics. For example, studies adopted different metrics and methods to evaluate adversarial perturbations' human perception, many of which may not be a reliable measure [162]. Also, existing defenses are often biased in metrics, datasets, assumptions, and application scenarios that favor the presentation of the work. It is challenging for VA designers to decide the appropriate security mitigation. Our article serves as an initial effort to solve this gap. Nevertheless, we call for the research community to work with VA designers to establish unified metrics and security standards that can build more reliability into voice assistants in practice.

9.2 Suggestions to VA Designers

We believe a defense is preferable if it has higher effectiveness and lower cost, which is also a rationale used in many articles [4, 39, 191]. Based on an overall comparison of defense methods in Table 4, we classify existing defenses into three categories: specialized, all-purpose, and complementary. Specialized defenses can mitigate a specific type of attack and show a good balance of effectiveness and cost. All-purpose defenses can mitigate multiple types of attacks at the same time. Complementary defenses are the other methods that may require a higher cost for similar effectiveness. We recommend designers implement the specialized or all-purpose methods as an essential layer of defense and consider the complementary methods for additional protection.

Specialized Defense. We select the specialized defensive methods for each type of attack that are easy to implement and, at the same time, barely affect VA performance or usability.

- Voiceprint verification against `Normal Speech` attacks.
- Signal distortion-based detection against `Voice Spoofing` attacks.
- Audio transformation against `Unintelligible Speech` and `Adversarial Example` attacks.
- Hardware enhancement against `Inaudible Signal` attacks.
- Strict skill certification against `Malicious Skill` attacks.

Though these methods may not be the most effective solutions, they can greatly increase the bar of various attacks at relatively low costs.

All-purpose Defense. We sort liveness and identity-based detection methods into this level as a defender can use them to mitigate a broad category of attacks instead of implementing separate protections. Nonetheless, we suggest that designers adopt these methods after properly assessing their effectiveness, implementation cost, and usability based on the VA's application scenario. For

example, a user may not prefer to carry extra devices or accept a restricted user-to-device distance in a smart home environment. **Complementary Defense.** The remaining defenses generally involve a higher cost than the specialized methods and can only address one or two types of attack. We suggest that designers consider these methods for additional security enhancement when the above methods are insufficient.

9.3 Suggestions to Users

Despite that it is the users who actually suffer from the security consequences, there are relatively few they can do to prevent the attacks. Based on our systematization of attacks, we recommend all users, as an essential layer of protection, to turn on the voice assistant's speaker verification as default (if there is) and avoid enabling high-risk VA behaviors, such as operating a bank account and unlocking home, if unnecessary. Security-sensitive users should avoid leaving the terminal device unattended, and they can customize a secure wake-word [27], disallow the wake-word detection, or disable the VA when the device is locked if the voice assistant supports them. A better understanding of the voice assistant's security and privacy policies can also help users assess the risks before using a voice assistant.

10 CONCLUSION

This article provides a thorough survey on voice assistant security, focusing on the attacks that can trigger a voice assistant into performing malicious behaviors and the countermeasures to mitigate the attacks. We first introduce a voice assistant's general structure and then overview the attacks by the attacker's goal, threat model, attack methods, a practical attack's idealism, and how the attacks exploit a VA's various subsystems. We systematize and elaborate on the five types of attack methods, i.e., normal speech, voice spoofing, unintelligible speech, adversarial example, inaudible signal, and malicious skill, based on the vulnerable VA subsystem and extract the shared methodologies. We divide existing countermeasures into two categories, i.e., detection and prevention, and systematize them by the shared defensive strategies rather than the attacks they were designed to mitigate, enabling us to compare and assess their applicability by the implementation cost, usability, and security in a unified voice assistant context. We further discuss the potential directions for future research and propose practical suggestions to designers and users.

REFERENCES

[1] Sajjad Abdoli, Luiz G. Hafemann, Jerome Rony, Ismail Ben Ayed, Patrick Cardinal, and Alessandro L. Koerich. 2019. Universal adversarial audio perturbations. arXiv:1908.03173. Retrieved from https://arxiv.org/abs/1908.03173.

[2] Hadi Abdullah, Washington Garcia, Christian Peeters, Patrick Traynor, Kevin R. B. Butler, and Joseph Wilson. 2019. Practical hidden voice attacks against speech and speaker recognition systems. In *Proceedings of the NDSS 2019*.

[3] Hadi Abdullah, Muhammad Sajidur Rahman, Washington Garcia, Logan Blue, Kevin Warren, Anurag Swarnim Yadav, Tom Shrimpton, and Patrick Traynor. 2019. Hear "No Evil", See "Kenansville": Efficient and transferable black-box attacks on speech recognition and voice identification systems. arXiv:1910.05262. Retrieved from https://arxiv.org/abs/1910.05262.

[4] Hadi Abdullah, Kevin Warren, Vincent Bindschaedler, Nicolas Papernot, and Patrick Traynor. 2020. The faults in our ASRs: An overview of attacks against automatic speech recognition and speaker identification systems. arXiv:2007.06622. Retrieved from https://arxiv.org/abs/2007.06622.

[5] Muhammad Ejaz Ahmed, Il-Youp Kwak, Jun Ho Huh, Iljoo Kim, Taekkyung Oh, and Hyoungshick Kim. 2020. Void: A fast and light voice liveness detection system. In *Proceedings of the 29th USENIX Security Symposium*. 2685–2702.

[6] Victor Akinwande, Celia Cintas, Skyler Speakman, and Srihari Sridharan. 2020. Identifying audio adversarial examples via anomalous pattern detection. arXiv:2002.05463. Retrieved from https://arxiv.org/abs/2002.05463.

[7] Federico Alegre, Asmaa Amehraye, and Nicholas Evans. 2013. Spoofing countermeasures to protect automatic speaker verification from voice conversion. In *Proceedings of the IEEE ICASSP 2013*. IEEE, 3068–3072.

[8] Efthimios Alepis and Constantinos Patsakis. 2017. Monkey says, monkey does: Security and privacy on voice assistants. *IEEE Access* 5 (2017), 17841–17851.

[9] Moustafa Alzantot, Bharathan Balaji, and Mani Srivastava. 2017. Did you hear that? Adversarial examples against automatic speech recognition. In *Proceedings of the 31st Conference on Neural Information Processing Systems*. 6.
[10] Iustina Andronic, Ludwig Kürzinger, Edgar Ricardo Chavez Rosas, Gerhard Rigoll, and Bernhard U. Seeber. 2020. MP3 compression to diminish adversarial noise in end-to-end speech recognition. arXiv:2007.12892. Retrieved from https://arxiv.org/abs/2007.12892.
[11] Arjun Nitin Bhagoji, Warren He, Bo Li, and Dawn Song. 2017. Exploring the space of black-box attacks on deep neural networks. arXiv:1712.09491. Retrieved from https://arxiv.org/abs/1712.09491.
[12] Sourav Bhattacharya, Dionysis Manousakas, Alberto Gil C. P. Ramos, Stylianos I. Venieris, Nicholas D. Lane, and Cecilia Mascolo. 2020. Countering acoustic adversarial attacks in microphone-equipped smart home devices. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 4, 2 (2020), 1–24.
[13] Mikołaj Bińkowski, Jeff Donahue, Sander Dieleman, Aidan Clark, Erich Elsen, Norman Casagrande, Luis C. Cobo, and Karen Simonyan. 2019. High fidelity speech synthesis with adversarial networks. arXiv:1909.11646. Retrieved from https://arxiv.org/abs/1909.11646.
[14] M. Bispham, Ioannis Agrafiotis, and Michael Goldsmith. 2019. The speech interface as an attack surface: An overview. *International Journal On Advances in Security* 12, 1 and 2 (2019).
[15] Mary K. Bispham, Ioannis Agrafiotis, and Michael Goldsmith. 2019. Nonsense attacks on google assistant and missense attacks on amazon alexa. In *Proceedings of the 5th ICISSP*. SciTe Press.
[16] Logan Blue, Hadi Abdullah, Luis Vargas, and Patrick Traynor. 2018. 2ma: Verifying voice commands via two microphone authentication. In *Proceedings of the ACM AsiaCCS 2018*. 89–100.
[17] Logan Blue, Luis Vargas, and Patrick Traynor. 2018. Hello, is it me you're looking for? differentiating between human and electronic speakers for voice interface security. In *Proceedings of the ACM WiSec 2018*. 123–133.
[18] Fabian Braunlein and Luise Frerichs. 2020. Smart Spies: Alexa and Google Home Expose Users to Vishing and Eavesdropping. Retrieved from https://srlabs.de/bites/smart-spies/.
[19] Nicholas Carlini, Pratyush Mishra, Tavish Vaidya, Yuankai Zhang, Micah Sherr, Clay Shields, David Wagner, and Wenchao Zhou. 2016. Hidden voice commands. In *Proceedings of the 25th USENIX Security Symposium*. 513–530.
[20] Nicholas Carlini and David Wagner. 2018. Audio adversarial examples: Targeted attacks on speech-to-text. In *Proceedings of the 2018 IEEE Security and Privacy Workshops*. IEEE, 1–7.
[21] Lucy Chai, Thavishi Illandara, and Zhongxia Yan. 2019. Private speech adversaries. (2019).
[22] Kuei-Huan Chang, Po-Hao Huang, Honggang Yu, Yier Jin, and Ting-Chi Wang. 2020. Audio adversarial examples generation with recurrent neural networks. In *Proceedings of the 25th ASPDAC*.
[23] Yun-Tai Chang. 2018. *A Two-layer Authentication Using Voiceprint for Voice Assistants*. Ph.D. Dissertation.
[24] Guangke Chen, Sen Chen, Lingling Fan, Xiaoning Du, Zhe Zhao, Fu Song, and Yang Liu. 2019. Who is real bob? adversarial attacks on speaker recognition systems. arXiv:1911.01840. Retrieved from https://arxiv.org/abs/1911.01840.
[25] Si Chen, Kui Ren, Sixu Piao, Cong Wang, Qian Wang, Jian Weng, Lu Su, and Aziz Mohaisen. 2017. You can hear but you cannot steal: Defending against voice impersonation attacks on smartphones. In *Proceedings of the IEEE 37th ICDCS*. IEEE, 183–195.
[26] Tao Chen, Longfei Shangguan, Zhenjiang Li, and Kyle Jamieson. 2020. Metamorph: Injecting inaudible commands into over-the-air voice controlled systems. In *Proceedings of the NDSS 2020*. 17.
[27] Yanjiao Chen, Yijie Bai, Richard Mitev, Kaibo Wang, Ahmad-Reza Sadeghi, and Wenyuan Xu. 2021. FakeWake: Understanding and mitigating fake wake-up words of voice assistants. In *Proceedings of the ACM CCS 2021*. 1861–1883.
[28] Yuxuan Chen, Xuejing Yuan, Jiangshan Zhang, Yue Zhao, Shengzhi Zhang, Kai Chen, and XiaoFeng Wang. 2020. Devil's whisper: A general approach for physical adversarial attacks against commercial black-box speech recognition devices. In *Proceedings of the 29th USENIX Security Symposium*.
[29] Long Cheng, Christin Wilson, Jeffrey Alan Young, Daniel Dong, and Hongxin Hu. 2020. Dangerous skills got certified: Measuring the trustworthiness of amazon alexa platform. (2020).
[30] Girija Chetty and Michael Wagner. 2004. Liveness verification in audio-video speaker authentication. In *Proceedings of the 10th ASSTA Conference*. Macquarie University Press, 358–363.
[31] Geumhwan Cho, Jusop Choi, Hyoungshick Kim, Sangwon Hyun, and Jungwoo Ryoo. 2018. Threat modeling and analysis of voice assistant applications. In *Proceedings of the WISA 2018*. Springer, 197–209.
[32] Moustapha M. Cisse, Yossi Adi, Natalia Neverova, and Joseph Keshet. 2017. Houdini: Fooling deep structured visual and speech recognition models with adversarial examples. In *Proceedings of the 31st NeurIPS*. 6977–6987.
[33] Nilaksh Das, Madhuri Shanbhogue, Shang-Tse Chen, Li Chen, Michael E. Kounavis, and Duen Horng Chau. 2018. Adagio: Interactive experimentation with adversarial attack and defense for audio. In *Proceedings of the Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, 677–681.
[34] Rohan Kumar Das, Xiaohai Tian, Tomi Kinnunen, and Haizhou Li. 2020. The attacker's perspective on automatic speaker verification: An overview. arXiv:2004.08849. Retrieved from https://arxiv.org/abs/2004.08849.

[35] Sina Däubener, Lea Schönherr, Asja Fischer, and Dorothea Kolossa. 2020. Detecting adversarial examples for speech recognition via uncertainty quantification. arXiv:2005.14611. Retrieved from https://arxiv.org/abs/2005.14611.
[36] Wenrui Diao, Xiangyu Liu, Zhe Zhou, and Kehuan Zhang. 2014. Your voice assistant is mine: How to abuse speakers to steal information and control your phone. In *Proceedings of the 4th ACM SPSM*. 63–74.
[37] Tianyu Du, Shouling Ji, Jinfeng Li, Qinchen Gu, Ting Wang, and Raheem Beyah. 2019. Sirenattack: Generating adversarial audio for end-to-end acoustic systems. arXiv:1901.07846. Retrieved from https://arxiv.org/abs/1901.07846.
[38] Xia Du, Chi-Man Pun, and Zheng Zhang. 2020. A unified framework for detecting audio adversarial examples. In *Proceedings of the 28th ACM International Conference on Multimedia*. 3986–3994.
[39] Jide S. Edu, Jose M. Such, and Guillermo Suarez-Tangil. 2019. Smart home personal assistants: A security and privacy review. arXiv:1903.05593. Retrieved from https://arxiv.org/abs/1903.05593.
[40] J. Lopes Esteves and C. Kasmi. 2018. Remote and silent voice command injection on a smartphone through conducted IEMI: Threats of smart IEMI for information security. *Technical Report* 48 (2018).
[41] Nicholas W. D. Evans, Tomi Kinnunen, and Junichi Yamagishi. 2013. Spoofing and countermeasures for automatic speaker verification. In *Proceedings of the INTERSPEECH*. 925–929.
[42] Alhussein Fawzi, Hamza Fawzi, and Omar Fawzi. 2018. Adversarial vulnerability for any classifier. In *Proceedings of the Advances in Neural Information Processing Systems*. 1178–1187.
[43] Huan Feng, Kassem Fawaz, and Kang G. Shin. 2017. Continuous authentication for voice assistants. In *Proceedings of the ACM MobiCom 2017*. 343–355.
[44] Santiago Fernández, Alex Graves, and Jürgen Schmidhuber. 2007. An application of recurrent neural networks to discriminative keyword spotting. In *Proceedings of the ICANN 2007*. Springer, 220–229.
[45] Kevin Fu and Wenyuan Xu. 2018. Risks of trusting the physics of sensors. *Communications of the ACM* 61, 2 (2018), 20–23.
[46] Mostafa Ghorbandoost, Abolghasem Sayadiyan, Mohsen Ahangar, Hamid Sheikhzadeh, Abdoreza Sabzi Shahrebabaki, and Jamal Amini. 2015. Voice conversion based on feature combination with limited training data. *Speech Communication* 67 (2015), 113–128.
[47] Taesik Gong, Alberto Gil C. P. Ramos, Sourav Bhattacharya, Akhil Mathur, and Fahim Kawsar. 2019. AudiDoS: Real-time denial-of-service adversarial attacks on deep audio models. In *Proceedings of the IEEE ICMLA 2019*. 978–985.
[48] Yuan Gong, Boyang Li, Christian Poellabauer, and Yiyu Shi. 2019. Real-time adversarial attacks. In *Proceedings of the 28th IJCAI*. 4672–4680.
[49] Yuan Gong and Christian Poellabauer. 2018. Crafting adversarial examples for speech paralinguistics applications. In *Proceedings of the DYNAMICS 2018*. ACM, 1–8.
[50] Yuan Gong and Christian Poellabauer. 2018. An overview of vulnerabilities of voice controlled systems. arXiv:1803.09156. Retrieved from https://arxiv.org/abs/1803.09156.
[51] Yuan Gong and Christian Poellabauer. 2018. Protecting voice controlled systems using sound source identification based on acoustic cues. In *Proceedings of the 27th ICCCN*. IEEE, 1–9.
[52] Yuan Gong, Jian Yang, and Christian Poellabauer. 2020. Detecting replay attacks using multi-channel audio: A neural network-based method. *IEEE Signal Processing Letters* (2020).
[53] Xavi Gonzalvo, Siamak Tazari, Chun-an Chan, Markus Becker, Alexander Gutkin, and Hanna Silen. 2016. Recent advances in Google real-time HMM-driven unit selection synthesizer. (2016).
[54] Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. 2014. Explaining and harnessing adversarial examples. arXiv:1412.6572. Retrieved from https://arxiv.org/abs/1412.6572.
[55] Qingli Guo, Jing Ye, Yiran Chen, Yu Hu, Yazhu Lan, Guohe Zhang, and Xiaowei Li. 2020. INOR–an intelligent noise reduction method to defend against adversarial audio examples. *Neurocomputing* (2020).
[56] Qingli Guo, Jing Ye, Yu Hu, Guohe Zhang, Xiaowei Li, and Huawei Li. 2020. MultiPAD: A multivariant partition-based method for audio adversarial examples detection. *IEEE Access* 8 (2020), 63368–63380.
[57] Zhixiu Guo, Zijin Lin, Pan Li, and Kai Chen. 2020. SkillExplorer: Understanding the behavior of skills in large scale. In *Proceedings of the 29th USENIX Security Symposium*. 1–18.
[58] Joon Kuy Han, Hyoungshick Kim, and Simon S. Woo. 2019. Nickel to lego: Using foolgle to create adversarial examples to fool google cloud speech-to-text API. In *Proceedings of the ACM CCS 2019*. 2593–2595.
[59] Abeerah Hashim. 2019. New Attack Strategy Against Smart Assistants Dubbed LightCommands. Retrieved from https://latesthackingnews.com/2019/11/10/new-attack-strategy-against-smart-assistants-dubbed-lightcommands/.
[60] Rosa González Hautamäki, Tomi Kinnunen, Ville Hautamäki, Timo Leino, and Anne-Maria Laukkanen. 2013. I-vectors meet imitators: On vulnerability of speaker verification systems against voice mimicry. In *Proceedings of the INTERSPEECH*. 930–934.
[61] Ruiwen He, Xiaoyu Ji, Xinfeng Li, Yushi Cheng, and Wenyuan Xu. 2022. "OK, Siri" or "Hey, Google": Evaluating voiceprint distinctiveness via content-based PROLE score. In *Proceedings of the 31th USENIX Security Symposium*.

[62] Yitao He, Junyu Bian, Xinyu Tong, Zihui Qian, Wei Zhu, Xiaohua Tian, and Xinbing Wang. 2019. Canceling inaudible voice commands against voice control systems. In *Proceedings of the ACM MbiCom 2019*. 1–15.
[63] Shengshan Hu, Xingcan Shang, Zhan Qin, Minghui Li, Qian Wang, and Cong Wang. 2019. Adversarial examples for automatic speech recognition: Attacks and countermeasures. *IEEE Communications Magazine* 57, 10 (2019), 120–126.
[64] Ryo Iijima, Shota Minami, Zhou Yunao, Tatsuya Takehisa, Takeshi Takahashi, Yasuhiro Oikawa, and Tatsuya Mori. 2018. Audio hotspot attack: An attack on voice assistance systems using directional sound beams. In *Proceedings of the ACM CCS 2018*. 2222–2224.
[65] Dan Iter, Jade Huang, and Mike Jermann. 2017. Generating adversarial examples for speech recognition. *Stanford Technical Report* (2017).
[66] Yeongjin Jang, Chengyu Song, Simon P. Chung, Tielei Wang, and Wenke Lee. 2014. A11y attacks: Exploiting accessibility in operating systems. In *Proceedings of ACM CCS 2014*. 103–115.
[67] Mohammad Javad Jannati and Abolghasem Sayadiyan. 2018. Part-syllable transformation-based voice conversion with very limited training data. *Circuits, Systems, and Signal Processing* 37, 5 (2018), 1935–1957.
[68] Tejas Jayashankar, Jonathan Le Roux, and Pierre Moulin. 2020. Detecting audio attacks on ASR systems with dropout uncertainty. arXiv:2006.01906. Retrieved from https://arxiv.org/abs/2006.01906.
[69] Sarfaraz Jelil, Sishir Kalita, S. R. Mahadeva Prasanna, and Rohit Sinha. 2018. Exploration of compressed ILPR features for replay attack detection. In *Proceedings of the INTERSPEECH*. 631–635.
[70] Xiaoyu Ji, Juchuan Zhang, Shui Jiang, Jishen Li, and Wenyuan Xu. 2021. CapSpeaker: Injecting voices to microphones via capacitors. In *Proceedings of the ACM CCS 2021*. 1915–1929.
[71] Madhu R. Kamble, Hemlata Tak, and Hemant A. Patil. 2018. Effectiveness of speech demodulation-based features for replay detection. In *Proceedings of the INTERSPEECH*. 641–645.
[72] Chaouki Kasmi and Jose Lopes Esteves. 2015. IEMI threats for information security: Remote command injection on modern smartphones. *IEEE Transactions on Electromagnetic Compatibility* 57, 6 (2015), 1752–1755.
[73] Lawrence George Kersta. 1962. Voiceprint identification. *Nature* 196, 4861 (1962), 1253–1257.
[74] Shreya Khare, Rahul Aralikatte, and Senthil Mani. 2019. Adversarial black-box attacks on automatic speech recognition systems using multi-objective evolutionary optimization. In *Proceedings of the INTERSPEECH*.
[75] Juntae Kim and Minsoo Hahn. 2018. Voice activity detection using an adaptive context attention model. *IEEE Signal Processing Letters* 25, 8 (2018), 1181–1185.
[76] Diederik P. Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. arXiv:1412.6980. Retrieved from https://arxiv.org/abs/1412.6980.
[77] Tomi Kinnunen and Haizhou Li. 2010. An overview of text-independent speaker recognition: From features to supervectors. *Speech Communication* 52, 1 (2010), 12–40.
[78] Felix Kreuk, Yossi Adi, Moustapha Cisse, and Joseph Keshet. 2018. Fooling end-to-end speaker verification with adversarial examples. In *Proceedings of the 2018 ICASSP*. IEEE, 1962–1966.
[79] Deepak Kumar, Riccardo Paccagnella, Paul Murley, Eric Hennenfent, Joshua Mason, Adam Bates, and Michael Bailey. 2018. Skill squatting attacks on amazon alexa. In *Proceedings of the 27th USENIX Security Symposium*. 33–47.
[80] Denis Foo Kune, John Backes, Shane S. Clark, Daniel Kramer, Matthew Reynolds, Kevin Fu, Yongdae Kim, and Wenyuan Xu. 2013. Ghost talk: Mitigating EMI signal injection attacks against analog sensors. In *Proceedings of the IEEE Symposium on Security and Privacy*. IEEE, 145–159.
[81] Hyun Kwon, Yongchul Kim, Hyunsoo Yoon, and Daeseon Choi. 2019. Selective audio adversarial example in evasion attack on speech recognition system. *IEEE Transactions on Information Forensics and Security* 15 (2019), 526–538.
[82] Hyun Kwon, Hyunsoo Yoon, and Ki-Woong Park. 2019. POSTER: Detecting audio adversarial example through audio modification. In *Proceedings of the ACM CCS 2019*. 2521–2523.
[83] Yee W. Lau, Dat Tran, and Michael Wagner. 2005. Testing voice mimicry with the YOHO speaker verification corpus. In *Proceedings of the KES 2005*. Springer, 15–21.
[84] Yee Wah Lau, Michael Wagner, and Dat Tran. 2004. Vulnerability of speaker verification to voice mimicking. In *Proceedings of the ISIMP 2004*. IEEE, 145–148.
[85] Yeonjoon Lee, Yue Zhao, Jiutian Zeng, Kwangwuk Lee, Nan Zhang, Faysal Hossain Shezan, Yuan Tian, Kai Chen, and XiaoFeng Wang. 2020. Using sonar for liveness detection to protect smart speakers against remote attackers. *Proceedings of the ACM UbiComp 2020* 4, 1 (2020), 1–28.
[86] Huining Li, Chenhan Xu, Aditya Singh Rathore, Zhengxiong Li, Hanbin Zhang, Chen Song, Kun Wang, Lu Su, Feng Lin, Kui Ren, et al. 2020. VocalPrint: Exploring a resilient and secure voice authentication via mmWave biometric interrogation. In *Proceedings of the SenSys 2020*. 312–325.
[87] Juncheng Li, Shuhui Qu, Xinjian Li, Joseph Szurley, J. Zico Kolter, and Florian Metze. 2019. Adversarial music: Real world audio adversary against wake-word detection system. In *Proceedings of the 33rd NeurIPS*. 11908–11918.
[88] Jiguo Li, Xinfeng Zhang, Chuanmin Jia, Jizheng Xu, Li Zhang, Yue Wang, Siwei Ma, and Wen Gao. 2020. Universal adversarial perturbations generative network for speaker recognition. In *Proceedings of the ICME 2020*. IEEE, 1–6.

[89] Jiguo Li, Xinfeng Zhang, Jizheng Xu, Li Zhang, Yue Wang, Siwei Ma, and Wen Gao. 2020. Learning to fool the speaker recognition. In *Proceedings of the 2020 IEEE ICASSP*. IEEE, 2937–2941.
[90] Xu Li, Na Li, Jinghua Zhong, Xixin Wu, Xunying Liu, Dan Su, Dong Yu, and Helen Meng. 2020. Investigating robustness of adversarial samples detection for automatic speaker verification. arXiv:2006.06186. Retrieved from https://arxiv.org/abs/2006.06186.
[91] Xu Li, Jinghua Zhong, Xixin Wu, Jianwei Yu, Xunying Liu, and Helen Meng. 2020. Adversarial attacks on GMM i-vector based speaker verification systems. In *Proceedings of the 2020 IEEE ICASSP*. IEEE, 6579–6583.
[92] Zhuohang Li, Cong Shi, Yi Xie, Jian Liu, Bo Yuan, and Yingying Chen. 2020. Practical adversarial attacks against speaker recognition systems. In *Proceedings of the 21st HotMobile*. 9–14.
[93] Zhuohang Li, Yi Wu, Jian Liu, Yingying Chen, and Bo Yuan. 2020. AdvPulse: Universal, synchronization-free, and targeted audio adversarial attacks via subsecond perturbations. In *Proceedings of the ACM CCS 2020*.
[94] Yiqing Lin and Waleed H. Abdulla. 2015. Principles of psychoacoustics. In *Proceedings of the Audio Watermark*. Springer, 15–49.
[95] Johan Lindberg and Mats Blomberg. 1999. Vulnerability in speaker verification-a study of technical impostor techniques. In *Proceedings of the 6th EUROSPEECH*.
[96] Songxiang Liu, Haibin Wu, Hung-Yi Lee, and Helen Meng. 2019. Adversarial attacks on spoofing countermeasures of automatic speaker verification. In *Proceedings of the IEEE ASRU 2019*. IEEE, 312–319.
[97] Xiaolei Liu, Xiaosong Zhang, Kun Wan, Qingxin Zhu, and Yufei Ding. 2019. Towards weighted-sampling audio adversarial example attack. arXiv:1901.10300. Retrieved from https://arxiv.org/abs/1901.10300.
[98] Jaime Lorenzo-Trueba, Fuming Fang, Xin Wang, Isao Echizen, Junichi Yamagishi, and Tomi Kinnunen. 2018. Can we steal your vocal identity from the Internet?: Initial investigation of cloning Obama's voice using GAN, WaveNet and low-quality found data. arXiv:1803.00860. Retrieved from https://arxiv.org/abs/1803.00860.
[99] Li Lu, Jiadi Yu, Yingying Chen, and Yan Wang. 2020. VocalLock: Sensing vocal tract for passphrase-independent user authentication leveraging acoustic signals on smartphones. *Proceedings of the ACM UbiComp 2020* 4, 2 (2020), 1–24.
[100] Pingchuan Ma, Stavros Petridis, and Maja Pantic. 2019. Detecting adversarial attacks on audio-visual speech recognition. arXiv:1912.08639. Retrieved from https://arxiv.org/abs/1912.08639.
[101] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. 2017. Towards deep learning models resistant to adversarial attacks. arXiv:1706.06083. Retrieved from https://arxiv.org/abs/1706.06083.
[102] Jian Mao, Shishi Zhu, Dai Xuan, Qixiao Lin, and Jianwei Liu. 2020. Watchdog: Detecting ultrasonic-based inaudible voice attacks to smart home systems. *IEEE Internet of Things Journal* (2020).
[103] Johnny Mariéthoz and Samy Bengio. 2005. *Can a Professional Imitator Fool a GMM-based Speaker Verification System?* Technical Report. IDIAP.
[104] Mirko Marras, Pawel Korus, Nasir D. Memon, and Gianni Fenu. 2019. Adversarial optimization for dictionary attacks on speaker verification. In *Proceedings of the INTERSPEECH*. 2913–2917.
[105] Jenny Medeiros. 2019. Here's How The Military is Using Voice Technology. Retrieved from https://www.voicesummit.ai/blog/how-the-military-is-using-voice-technology.
[106] Ethan Mendes and Kyle Hogan. 2020. Defending against imperceptible audio adversarial examples using proportional additive gaussian noise. (2020).
[107] Yan Meng, Zichang Wang, Wei Zhang, Peilin Wu, Haojin Zhu, Xiaohui Liang, and Yao Liu. 2018. Wivo: Enhancing the security of voice control system via wireless signal in iot environment. In *Proceedings of the ACM Mobihoc 2018*. 81–90.
[108] Richard Mitev, Markus Miettinen, and Ahmad-Reza Sadeghi. 2019. Alexa lied to me: Skill-based man-in-the-middle attacks on virtual assistants. In *Proceedings of the ACM AsiaCCS 2019*. 465–478.
[109] Shihono Mochizuki, Sayaka Shiota, and Hitoshi Kiya. 2018. Voice liveness detection using phoneme-based pop-noise detector for speaker verifcation. In *Proceedings of the Odyssey 2018 Speaker and Language Recognition Workshop*.
[110] Seyed Hamidreza Mohammadi and Alexander Kain. 2017. An overview of voice conversion systems. *Speech Communication* 88 (2017), 65–82.
[111] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, and Pascal Frossard. 2016. Deepfool: A simple and accurate method to fool deep neural networks. In *Proceedings of the IEEE CVPR 2016*. 2574–2582.
[112] Paarth Neekhara, Shehzeen Hussain, Prakhar Pandey, Shlomo Dubnov, Julian McAuley, and Farinaz Koushanfar. 2019. Universal adversarial perturbations for speech recognition systems. arXiv:1905.03828. Retrieved from https://arxiv.org/abs/1905.03828.
[113] Aaron Oord, Yazhe Li, Igor Babuschkin, Karen Simonyan, Oriol Vinyals, Koray Kavukcuoglu, George Driessche, Edward Lockhart, Luis Cobo, Florian Stimberg, et al. 2018. Parallel wavenet: Fast high-fidelity speech synthesis. In *Proceedings of the ICML 2018*. 3918–3926.
[114] Aaron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew Senior, and Koray Kavukcuoglu. 2016. Wavenet: A generative model for raw audio. arXiv:1609.03499. Retrieved from https://arxiv.org/abs/1609.03499.

[115] E. Oppenheim and R. W. Schafer. 1980. Digital processing of speech signals. *Moscow: Mir.–323* (1980).
[116] Jayashree Padmanabhan and Melvin Jose Johnson Premkumar. 2015. Machine learning in automatic speech recognition: A survey. *IETE Technical Review* 32, 4 (2015), 240–251.
[117] Mary Papenfuss. 201. Amazon Voice Assistant Alexa Orders Herself Some Dollhouses. Retrieved from https://www.huffpost.com/entry/amazon-alexa-orders-dollhouses_n_587317bbe4b099cdb0fdff5b.
[118] Nicolas Papernot, Patrick McDaniel, and Ian Goodfellow. 2016. Transferability in machine learning: From phenomena to black-box attacks using adversarial samples. arXiv:1605.07277. Retrieved from https://arxiv.org/abs/1605.07277.
[119] Nicolas Papernot, Patrick McDaniel, Ian Goodfellow, Somesh Jha, Z. Berkay Celik, and Ananthram Swami. 2017. Practical black-box attacks against machine learning. In *Proceedings of the ACM AsiaCCS 2017*. 506–519.
[120] Youngseok Park, Hyunsang Choi, Sanghyun Cho, and Young-Gab Kim. 2019. Security analysis of smart speaker: Security attacks and mitigation. *CMC-COMPUTERS MATERIALS & CONTINUA* 61, 3 (2019), 1075–1090.
[121] Santiago Pascual, Antonio Bonafonte, and Joan Serra. 2017. SEGAN: Speech enhancement generative adversarial network. arXiv:1703.09452. Retrieved from https://arxiv.org/abs/1703.09452.
[122] Ankur T. Patil, Rajul Acharya, Pulikonda Krishna Aditya Sai, and Hemant A. Patil. 2019. Energy separation-based instantaneous frequency estimation for cochlear cepstral feature for replay spoof detection. In *Proceedings of the INTERSPEECH*. 2898–2902.
[123] Swadhin Pradhan, Wei Sun, Ghufran Baig, and Lili Qiu. 2019. Combating replay attacks against voice assistants. *Proceedings of the ACM UbiComp 2019* 3, 3 (2019), 1–26.
[124] Yao Qin, Nicholas Carlini, Ian Goodfellow, Garrison Cottrell, and Colin Raffel. 2019. Imperceptible, robust, and targeted adversarial examples for automatic speech recognition. In *Proceedings of the 36th ICML*.
[125] Krishan Rajaratnam, Basemah Alshemali, Kunal Shah, and Jugal Kalita. 2018. Speech Coding and Audio Preprocessing for Mitigating and Detecting Audio Adversarial Examples on Automatic Speech Recognition.
[126] Krishan Rajaratnam and Jugal Kalita. 2018. Noise flooding for detecting audio adversarial examples against automatic speech recognition. In *Proceedings of the IEEE ISSPIT 2018*. IEEE, 197–201.
[127] Krishan Rajaratnam, Kunal Shah, and Jugal Kalita. 2018. Isolated and ensemble audio preprocessing methods for detecting adversarial examples against automatic speech recognition. arXiv:1809.04397. Retrieved from https://arxiv.org/abs/1809.04397.
[128] Douglas A. Reynolds. 1995. Speaker identification and verification using gaussian mixture speaker models. *Speech Communication* 17, 1–2 (1995), 91–108.
[129] Nirupam Roy, Sheng Shen, Haitham Hassanieh, and Romit Roy Choudhury. 2018. Inaudible voice commands: The long-range attack and defense. In *Proceedings of the 15th NSDI*. 547–560.
[130] Neville Ryant, Mark Liberman, and Jiahong Yuan. 2013. Speech activity detection on youtube using deep neural networks. In *Proceedings of the INTERSPEECH*. Lyon, France, 728–731.
[131] Md Sahidullah, Dennis Alexander Lehmann Thomsen, Rosa Gonzalez Hautamäki, Tomi Kinnunen, Zheng-Hua Tan, Robert Parts, and Martti Pitkänen. 2017. Robust voice liveness detection and speaker verification using throat microphones. *IEEE/ACM TASLP* 26, 1 (2017), 44–56.
[132] Tara N. Sainath, Oriol Vinyals, Andrew Senior, and Haşim Sak. 2015. Convolutional, long short-term memory, fully connected deep neural networks. In *Proceedings of the IEEE ICASSP 2015*. IEEE, 4580–4584.
[133] Yuki Saito, Shinnosuke Takamichi, and Hiroshi Saruwatari. 2017. Statistical parametric speech synthesis incorporating generative adversarial networks. *IEEE/ACM TASLP* 26, 1 (2017), 84–96.
[134] Saeid Samizade, Zheng-Hua Tan, Chao Shen, and Xiaohong Guan. 2020. Adversarial example detection by classification for deep speech recognition. In *Proceedings of the IEEE ICASSP 2020*. IEEE, 3102–3106.
[135] Lea Schonherr, Katharina Kohls, Steffen Zeiler, Thorsten Holz, and Dorothea Kolossa. 2019. Adversarial attacks against automatic speech recognition systems via psychoacoustic hiding. In *Proceedings of the NDSS 2019*.
[136] Lea Schönherr, Steffen Zeiler, Thorsten Holz, and Dorothea Kolossa. 2019. Imperio: Robust over-the-air adversarial examples against automatic speech recognition systems. arXiv:1908.01551. Retrieved from https://arxiv.org/abs/1908.01551.
[137] Ali Shahin Shamsabadi, Francisco Sepúlveda Teixeira, Alberto Abad, Bhiksha Raj, Andrea Cavallaro, and Isabel Trancoso. 2020. FoolHD: Fooling speaker identification by highly imperceptible adversarial disturbances. arXiv:2011.08483. *arXiv preprint arXiv:2011.08483*.
[138] Jiacheng Shang, Si Chen, and Jie Wu. 2018. Defending against voice spoofing: A robust software-based liveness detection system. In *Proceedings of the IEEE MASS 2018*. IEEE, 28–36.
[139] Jiacheng Shang and Jie Wu. 2019. Enabling secure voice input on augmented reality headsets using internal body voice. In *Proceedings of the IEEE SECON 2019*. IEEE, 1–9.
[140] Jiacheng Shang and Jie Wu. 2020. Secure voice input on augmented reality headsets. *IEEE TMC* (2020).
[141] Jiacheng Shang and Jie Wu. 2020. Voice liveness detection for voice assistants using ear canal pressure. In *Proceedings of the IEEE MASS 2020*.

[142] W. Shang and M. Stevenson. 2010. Score normalization in playback attack detection. In *Proceedings of the IEEE ICASSP 2010*. 1678–1681.
[143] Faysal Hossain Shezan, Hang Hu, Jiamin Wang, Gang Wang, and Yuan Tian. 2020. Read between the lines: An empirical measurement of sensitive applications of voice personal assistant systems. In *Proceedings of the Web Conference 2020*. 1006–1017.
[144] Sayaka Shiota, Fernando Villavicencio, Junichi Yamagishi, Nobutaka Ono, Isao Echizen, and Tomoko Matsui. 2016. Voice liveness detection for speaker verification based on a tandem single/double-channel pop noise detector. In *Proceeding of the Odyssey 2016 Speaker and Language Recognition Workshop*. 259–263.
[145] Liwei Song and Prateek Mittal. 2017. Poster: Inaudible voice commands. In *Proceedings of the ACM CCS 2017*. 2583–2585.
[146] Statista. 2020. Number of Digital Voice Assistants in Use Worldwide from 2019 to 2023. Retrieved from https://www.statista.com/statistics/973815/worldwide-digital-voice-assistant-in-use/.
[147] Dan Su, Jiqiang Liu, Sencun Zhu, Xiaoyang Wang, and Wei Wang. 2020. "Are you home alone?" "Yes" disclosing security and privacy vulnerabilities in alexa skills. arXiv:2010.10788. Retrieved from https://arxiv.org/abs/2010.10788.
[148] Takeshi Sugawara, Benjamin Cyr, Sara Rampazzi, Daniel Genkin, and Kevin Fu. 2019. Light commands: Laser-based audio injection attacks on voice-controllable systems. (2019).
[149] Lifa Sun, Kun Li, Hao Wang, Shiyin Kang, and Helen Meng. 2016. Phonetic posteriorgrams for many-to-one voice conversion without parallel data training. In *Proceedings of the IEEE ICME 2016*. IEEE, 1–6.
[150] Sining Sun, Pengcheng Guo, Lei Xie, and Mei-Yuh Hwang. 2019. Adversarial regularization for attention based end-to-end robust speech recognition. *IEEE/ACM TASLP* 27, 11 (2019), 1826–1838.
[151] Sining Sun, Ching-Feng Yeh, Mari Ostendorf, Mei-Yuh Hwang, and Lei Xie. 2018. Training augmentation with adversarial examples for robust speech recognition. arXiv:1806.02782. Retrieved from https://arxiv.org/abs/1806.02782.
[152] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. 2013. Intriguing properties of neural networks. arXiv:1312.6199. Retrieved from https://arxiv.org/abs/1312.6199.
[153] Joseph Szurley and J. Zico Kolter. 2019. Perceptual based adversarial audio attacks. arXiv:1906.06355. Retrieved from https://arxiv.org/abs/1906.06355.
[154] Keiichi Tamura, Akitada Omagari, and Shuichi Hashida. 2019. Novel defense method against audio adversarial example for speech-to-text transcription neural networks. In *Proceedings of the IWCIA 2019*. IEEE, 115–120.
[155] Rohan Taori, Amog Kamsetty, Brenton Chu, and Nikita Vemuri. 2019. Targeted adversarial examples for black box audio systems. In *Proceedings of the 2019 IEEE Security and Privacy Workshops*. IEEE, 15–20.
[156] Tomoki Toda, Yamato Ohtani, and Kiyohiro Shikano. 2007. One-to-many and many-to-one voice conversion based on eigenvoices. In *Proceedings of the IEEE ICASSP 2007*. IEEE, IV–1249.
[157] Massimiliano Todisco, Xin Wang, Ville Vestman, Md Sahidullah, Héctor Delgado, Andreas Nautsch, Junichi Yamagishi, Nicholas Evans, Tomi Kinnunen, and Kong Aik Lee. 2019. Asvspoof 2019: Future horizons in spoofed and fake audio detection. arXiv:1904.05441. Retrieved from https://arxiv.org/abs/1904.05441.
[158] Keiichi Tokuda, Yoshihiko Nankaku, Tomoki Toda, Heiga Zen, Junichi Yamagishi, and Keiichiro Oura. 2013. Speech synthesis based on hidden Markov models. 101, 5 (2013), 1234–1252.
[159] Florian Tramèr, Fan Zhang, Ari Juels, Michael K. Reiter, and Thomas Ristenpart. 2016. Stealing machine learning models via prediction apis. In *Proceedings of the 25th USENIX Security Symposium*. 601–618.
[160] Kai Tubbesing. 2018. Alexa is Engaged as Voice Assistant in Industry. Retrieved from https://www.hannovermesse.de/en/news/news-articles/alexa-is-engaged-as-voice-assistant-in-industry.
[161] Jon Vadillo and Roberto Santana. 2019. Universal adversarial examples in speech command classification. arXiv:1911.10182. Retrieved from https://arxiv.org/abs/1911.10182.
[162] Jon Vadillo and Roberto Santana. 2020. On the human evaluation of audio adversarial examples. arXiv:2001.08444. Retrieved from https://arxiv.org/abs/2001.08444.
[163] Tavish Vaidya, Yuankai Zhang, Micah Sherr, and Clay Shields. 2015. Cocaine noodles: Exploiting the gap between human and machine speech recognition. In *Proceedings of the 9th USENIX Workshop on Offensive Technologies*.
[164] Jesús Villalba and Eduardo Lleida. 2010. Speaker verification performance degradation against spoofing and tampering attacks. In *Proceedings of the FALA Workshop*. 131–134.
[165] Jesús Villalba and Eduardo Lleida. 2011. Detecting replay attacks from far-field recordings on speaker verification systems. In *Proceedings of the European Workshop on Biometrics and Identity Management*. Springer, 274–285.
[166] J. Villalba and E. Lleida. 2011. Preventing replay attacks on speaker verification systems. In *Proceedings of the 2011 Carnahan Conference on Security Technology*. 1–8.
[167] Wolfgang Wahlster. 2013. *Verbmobil: Foundations of Speech-to-speech Translation*. Springer Science & Business Media.
[168] Binghui Wang and Neil Zhenqiang Gong. 2018. Stealing hyperparameters in machine learning. In *Proceedings of the 2018 IEEE Symposium on Security and Privacy*. IEEE, 36–52.

[169] Chen Wang, Cong Shi, Yingying Chen, Yan Wang, and Nitesh Saxena. 2020. WearID: Wearable-assisted low-effort authentication to voice assistants using cross-domain speech similarity. arXiv:2003.09083. Retrieved from https://arxiv.org/abs/2003.09083.
[170] Donghua Wang, Li Dong, Rangding Wang, Diqun Yan, and Jie Wang. 2020. Targeted speech adversarial example generation with generative adversarial network. *IEEE Access* (2020).
[171] Donghua Wang, Rangding Wang, Li Dong, Diqun Yan, Xueyuan Zhang, and Yongkang Gong. 2020. Adversarial examples attack and countermeasure for speech recognition system: A survey. In *Proceedings of the SPDE*. 443–468.
[172] Qing Wang, Pengcheng Guo, Sining Sun, Lei Xie, and John H. L. Hansen. 2019. Adversarial regularization for end-to-end robust speaker verification. In *Proceedings of the INTERSPEECH*. 4010–4014.
[173] Qing Wang, Pengcheng Guo, and Lei Xie. 2020. Inaudible adversarial perturbations for targeted attack in speaker recognition. arXiv:2005.10637. Retrieved from https://arxiv.org/abs/2005.10637.
[174] Qian Wang, Xiu Lin, Man Zhou, Yanjiao Chen, Cong Wang, Qi Li, and Xiangyang Luo. 2019. Voicepop: A pop noise based anti-spoofing system for voice authentication on smartphones. In *Proceedings of the INFOCOM*. IEEE, 2062–2070.
[175] Qian Wang, Baolin Zheng, Qi Li, Chao Shen, and Zhongjie Ba. 2020. Towards query-efficient adversarial attacks against automatic speech recognition systems. *IEEE Transactions on Information Forensics and Security* (2020).
[176] Shu Wang, Jiahao Cao, Xu He, Kun Sun, and Qi Li. 2020. When the differences in frequency domain are compensated: Understanding and defeating modulated replay attacks on automatic speech recognition. In *Proceedings of the ACM CCS 2020*.
[177] Shu Wang, Jiahao Cao, Kun Sun, and Qi Li. 2020. SIEVE: Secure in-vehicle automatic speech recognition systems. In *Proceedings of the RAID 2020*. 365–379.
[178] Xianliang Wang, Yanhong Xiao, and Xuan Zhu. 2017. Feature selection based on CQCCs for automatic speaker verification spoofing. In *Proceedings of the INTERSPEECH*. 32–36.
[179] Yao Wang, Wandong Cai, Tao Gu, Wei Shao, Yannan Li, and Yong Yu. 2019. Secure your voice: An oral airflow-based continuous liveness detection for voice assistants. 3, 4 (2019), 1–28.
[180] Zhuoran Wang, Hongliang Chen, Guanchun Wang, Hao Tian, Hua Wu, and Haifeng Wang. 2014. Policy learning for domain selection in an extensible multi-domain spoken dialogue system. In *Proceedings of the EMNLP 2014*. 57–67.
[181] Z. Wang, G. Wei, and Q. He. 2011. Channel pattern noise based playback attack detection algorithm for speaker recognition. In *Proceedings of the ICMLC 2011*. 1708–1713.
[182] Marcin Witkowski, Stanislaw Kacprzak, Piotr Zelasko, Konrad Kowalczyk, and Jakub Galka. 2017. Audio replay attack detection using high-frequency features. In *Proceedings of the INTERSPEECH*. 27–31.
[183] Venessa Wong. 2017. Burger King's New Ad Will Hijack Your Google Home. Retrieved from https://www.cnbc.com/2017/04/12/burger-kings-new-ad-will-hijack-your-google-home.html.
[184] Yi Wu, Jian Liu, Yingying Chen, and Jerry Cheng. 2019. Semi-black-box attacks against speech recognition systems using adversarial samples. In *Proceedings of the 2019 DySPAN*. IEEE, 1–5.
[185] Zhizheng Wu, Eng Siong Chng, and Haizhou Li. 2012. Detecting converted speech and natural speech for anti-spoofing attack in speaker recognition. In *Proceedings of the INTERSPEECH*.
[186] Zhizheng Wu, Nicholas Evans, Tomi Kinnunen, Junichi Yamagishi, Federico Alegre, and Haizhou Li. 2015. Spoofing and countermeasures for speaker verification: A survey. *Speech Communication* 66 (2015), 130–153.
[187] Zhizheng Wu, Sheng Gao, Eng Siong Cling, and Haizhou Li. 2014. A study on replay attack and anti-spoofing for text-dependent speaker verification. In *Proceedings of the ASPIPA ASC*. IEEE, 1–5.
[188] Zhizheng Wu, Tomi Kinnunen, Eng Siong Chng, Haizhou Li, and Eliathamby Ambikairajah. 2012. A study on spoofing attack in state-of-the-art speaker verification: The telephone speech case. In *Proceedings of the APSIPA ASC*. IEEE, 1–5.
[189] Yi Xie, Cong Shi, Zhuohang Li, Jian Liu, Yingying Chen, and Bo Yuan. 2020. Real-time, universal, and robust adversarial attacks against speaker recognition systems. In *Proceedings of the 2020 IEEE ICASSP*. IEEE, 1738–1742.
[190] Hiromu Yakura and Jun Sakuma. 2019. Robust audio adversarial example for a physical attack. In *Proceedings of the 28th IJCAI*. 5334–5341.
[191] Chen Yan, Yan Long, Xiaoyu Ji, and Wenyuan Xu. 2019. The catcher in the field: A fieldprint based spoofing detection for text-independent speaker verification. In *Proceedings of the ACM CCS 2019*. 1215–1229.
[192] Chen Yan, Hocheol Shin, Connor Bolton, Wenyuan Xu, Yongdae Kim, and Kevin Fu. 2020. SoK: A minimalist approach to formalizing analog sensor security. In *Proceedings of the 2020 IEEE Symposium on Security and Privacy*. 480–495.
[193] Chen Yan, Guoming Zhang, Xiaoyu Ji, Tianchen Zhang, Taimin Zhang, and Wenyuan Xu. 2019. The feasibility of injecting inaudible voice commands to voice assistants. *IEEE TDSC* (2019).
[194] Qiben Yan, Kehai Liu, Qin Zhou, Hanqing Guo, and Ning Zhang. 2020. SurfingAttack: Interactive hidden attack on voice assistants using ultrasonic guided waves. In *Proceedings of the NDSS 2020*. Internet Society.

[195] Zhuolin Yang, Bo Li, Pin-Yu Chen, and Dawn Song. 2018. Characterizing audio adversarial examples using temporal dependency. arXiv:1809.10875. Retrieved from https://arxiv.org/abs/1809.10875.
[196] Sung-Hyun Yoon, Min-Sung Koh, Jae-Han Park, and Ha-Jin Yu. 2020. A new replay attack against automatic speaker verification systems. *IEEE Access* 8 (2020), 36080–36088.
[197] Park Joon Young, Jo Hyo Jin, Samuel Woo, and Dong Hoon Lee. 2016. BadVoice: Soundless voice-control replay attack on modern smartphones. In *Proceedings of the 8th ICUFN*. IEEE, 882–887.
[198] Xuejing Yuan, Yuxuan Chen, Aohui Wang, Kai Chen, Shengzhi Zhang, Heqing Huang, and Ian M. Molloy. 2018. All your alexa are belong to us: A remote voice control attack against echo. In *Proceedings of the GLOBECOM*. IEEE, 1–6.
[199] Xuejing Yuan, Yuxuan Chen, Yue Zhao, Yunhui Long, Xiaokang Liu, Kai Chen, Shengzhi Zhang, Heqing Huang, XiaoFeng Wang, and Carl A. Gunter. 2018. Commandersong: A systematic approach for practical adversarial voice recognition. In *Proceedings of the 27th USENIX Security Symposium*. 49–64.
[200] Xiaoyong Yuan, Pan He, Qile Zhu, and Xiaolin Li. 2019. Adversarial examples: Attacks and defenses for deep learning. *IEEE Transactions on Neural Networks and Learning Systems* 30, 9 (2019), 2805–2824.
[201] Heiga Zen, Takashi Nose, Junichi Yamagishi, Shinji Sako, Takashi Masuko, Alan W. Black, and Keiichi Tokuda. 2007. The HMM-based speech synthesis system (HTS) version 2.0. In *Proceedings of the SSW 2007*. Citeseer, 294–299.
[202] Heiga Zen, Keiichi Tokuda, and Alan W. Black. 2009. Statistical parametric speech synthesis. *Speech Communication* 51, 11 (2009), 1039–1064.
[203] Qiang Zeng, Jianhai Su, Chenglong Fu, Golam Kayas, Lannan Luo, Xiaojiang Du, Chiu C. Tan, and Jie Wu. 2019. A multiversion programming inspired approach to detecting audio adversarial examples. In *Proceedings of the DSN 2019*. IEEE, 39–51.
[204] Guoming Zhang, Xiaoyu Ji, Xinfeng Li, Gang Qu, and Wenyuan Xu. 2021. EarArray: Defending against dolphinattack via acoustic attenuation. In *Proceedings of the NDSS 2021*.
[205] Guoming Zhang, Chen Yan, Xiaoyu Ji, Tianchen Zhang, Taimin Zhang, and Wenyuan Xu. 2017. Dolphinattack: Inaudible voice commands. In *Proceedings of the ACM CCS 2017*. 103–117.
[206] Hongting Zhang, Qiben Yan, Pan Zhou, and Xiao-Yang Liu. 2020. Generating robust audio adversarial examples with temporal dependency. In *Proceedings of the 29th IJCAI*. 3167–3173.
[207] Jiajie Zhang, Bingsheng Zhang, and Bincheng Zhang. 2019. Defending adversarial attacks on cloud-aided automatic speech recognition systems. In *Proceedings of the 7th International Workshop on Security in Cloud Computing*. 23–31.
[208] Lei Zhang, Yan Meng, Jiahao Yu, Chong Xiang, Brandon Falk, and Haojin Zhu. 2020. Voiceprint mimicry attack towards speaker verification system in smart home. In *Proceedings of the IEEE INFOCOM 2020*.
[209] Linghan Zhang, Sheng Tan, and Jie Yang. 2017. Hearing your voice is not enough: An articulatory gesture based liveness detection for voice authentication. In *Proceedings of the ACM CCS 2017*. 57–71.
[210] Linghan Zhang, Sheng Tan, Jie Yang, and Yingying Chen. 2016. Voicelive: A phoneme localization based liveness detection for voice authentication on smartphones. In *Proceedings of the ACM CCS 2016*. 1080–1091.
[211] Nan Zhang, Xianghang Mi, Xuan Feng, XiaoFeng Wang, Yuan Tian, and Feng Qian. 2019. Dangerous skills: Understanding and mitigating security risks of voice-controlled third-party functions on virtual personal assistant systems. In *Proceedings of the 2019 IEEE Symposium on Security and Privacy*. IEEE, 1381–1396.
[212] Rongjunchen Zhang, Xiao Chen, Sheng Wen, Xi Zheng, and Yong Ding. 2019. Using AI to attack VA: A stealthy spyware against voice assistances in smart phones. *IEEE Access* 7 (2019), 153542–153554.
[213] Yangyong Zhang, Lei Xu, Abner Mendoza, Guangliang Yang, Phakpoom Chinprutthiwong, and Guofei Gu. 2019. Life after speech recognition: fuzzing semantic misinterpretation for voice assistant applications. In *Proceedings of the NDSS 2019*.
[214] Man Zhou, Zhan Qin, Xiu Lin, Shengshan Hu, Qian Wang, and Kui Ren. 2019. Hidden voice commands: Attacks and defenses on the vcs of autonomous driving cars. *IEEE Wireless Communications* 26, 5 (2019), 128–133.
[215] Yi Zhou, Xiaohai Tian, Haihua Xu, Rohan Kumar Das, and Haizhou Li. 2019. Cross-lingual voice conversion with bilingual phonetic posteriorgram and average modeling. In *Proceedings of the IEEE ICASSP 2019*. IEEE, 6790–6794.

Received 15 December 2020; revised 9 February 2022; accepted 14 March 2022