Your Microphone Array Retains Your Identity: A Robust Voice Liveness Detection System for Smart Speakers

Yan Meng and Jiachun Li, Shanghai Jiao Tong University; Matthew Pillari, Arjun Deopujari, Liam Brennan, and Hafsah Shamsie, University of Virginia; Haojin Zhu, Shanghai Jiao Tong University; Yuan Tian, University of Virginia

https://www.usenix.org/conference/usenixsecurity22/presentation/meng

This paper is included in the Proceedings of the 31st USENIX Security Symposium.
August 10–12, 2022 • Boston, MA, USA

978-1-939133-31-1

Open access to the Proceedings of the 31st USENIX Security Symposium is sponsored by USENIX.
Your Microphone Array Retains Your Identity:  
A Robust Voice Liveness Detection System for Smart Speakers  

Yan Meng\textsuperscript{1}, Jiachun Li\textsuperscript{1}, Matthew Pillari\textsuperscript{2}, Arjun Deopujari\textsuperscript{2}  
Liam Brennan\textsuperscript{2}, Hafsa Shamsie\textsuperscript{2}, Haojin Zhu\textsuperscript{1}, and Yuan Tian\textsuperscript{2}  
\textsuperscript{1}Shanghai Jiao Tong University, \{yan meng, jiachunli, zhu-hj\}@sjtu.edu.cn  
\textsuperscript{2}University of Virginia, \{pillari, ajd4mq, lfb6ek, his3uh, yuant\}@virginia.edu  

Abstract  
Though playing an essential role in smart home systems, smart speakers are vulnerable to voice spoofing attacks. Passive liveness detection, which utilizes only the collected audio rather than the deployed sensors to distinguish between live-human and replayed voices, has drawn increasing attention. However, it faces the challenge of performance degradation under the different environmental factors as well as the strict requirement of the fixed user gestures.  
In this study, we propose a novel liveness feature, array fingerprint, which utilizes the microphone array inherently adopted by the smart speaker to determine the identity of collected audios. Our theoretical analysis demonstrates that by leveraging the circular layout of microphones, compared with existing schemes, array fingerprint achieves a more robust performance under the environmental change and user’s movement. Then, to leverage such a fingerprint, we propose ARRAYID, a lightweight passive detection scheme, and elaborate a series of features working together with array fingerprint. Our evaluation on the dataset containing 32,780 audio samples and 14 spoofing devices shows that ARRAYID achieves an accuracy of 99.84\%, which is superior to existing passive liveness detection schemes.  

1 Introduction  
Nowadays, voice assistance-enabled smart speakers serve as the hub of popular smart home platforms (e.g., Amazon Alexa, Google Home) and allow the user to remotely control home appliances (e.g., smart lighter, locker, thermostat) or query information (e.g., weather, news) as long as it can hear the user. However, the inherent broadcast nature of voice unlocks a door for adversaries to inject malicious commands (i.e., spoofing attack). Besides the classical replay attack \cite{12,48}, emerging attacks leveraging flaws in smart speakers are also proposed by researchers. On the hardware side, the non-linearity of the microphone’s frequency response provides a door for inaudible ultrasound-based attacks (e.g., Dolphin attack \cite{52} and \textit{BackDoor} attack \cite{35}). For the software aspect, the deep learning models employed by both speech recognition and speaker verification are proved to be vulnerable to emerging adversarial attacks such as hidden voice \cite{7}, Commander-Song \cite{50}, and user impersonation \cite{53}. Spoofing attacks impose emerging safety issues (e.g., deliberately turn on the smart thermostat \cite{13}) and privacy risks (e.g., querying user’s schedule information) on the smart speaker and therefore cause great concern.  
To defend against spoofing attacks, researchers have proposed a variety of countermeasures. Almost all countermeasures leverage the fact that voices in the spoofing attack are played by electrical devices (e.g., high-quality loudspeaker \cite{48}, ultrasonic dynamic speaker \cite{52}). Thus, the physical characteristics, which are different between humans and machines, could be used as the “liveness” factors. The existing countermeasures (aka., liveness detection) could be divided into multi-factor authentication and passive scheme. The former combines the collected audio and additional physical quantity (e.g., acceleration \cite{15}, electromagnetic field \cite{9}, ultrasound \cite{55}, Wi-Fi \cite{32}, mm-Wave \cite{14}) to distinguish between the human voice and the machine-generated one. By contrast, the passive scheme only considers the audio data collected by the smart speaker. Its key insight is that the difference of articulatory manners between real humans (i.e., vocal vibration and mouth movement) and electrical machines (i.e., diaphragm vibration) will result in the subtle but significant differences in the collected audios’ spectrograms. Passive schemes based on mono audio \cite{3,6} and two-channel audio \cite{5,49} have already been proposed and could be directly incorporated in the smart speaker’s software level.  
However, the existing liveness detection schemes face a series of challenges in the aspects of usability and efficiency, which seriously hinder their real-world deployment in practice. On the one hand, to capture the liveness factor of a real human, multiple-factor authentication either requires the user to carry specialized sensors (e.g., accelerometer, magnetometer) or actively emits probe signals (e.g., ultrasounds, wireless signals), which adds additional burdens for users.
On the other hand, passive schemes leveraging sub-bass low-frequency area (20~300 Hz in [6]) or voice area (below 10 kHz in [3]) of mono audio’s spectrum as liveness factor are vulnerable to sound propagation channel’s change and the spectrum modulated-based attack [48]. Another scheme [49] aiming to extract audio’s fieldprint from two-channel audio requires the user to keep a fixed manner to ensure the robustness of such fingerprints. As a result, the scheme is difficult to be deployed in many practical scenarios (e.g., users walking or having gesture changes). Therefore, it is desirable to propose a novel passive liveness detection scheme with the following merits: (i) Device-free: performing passive detection only relying on the collected audio; (ii) Resilient to environment change: being robust to dynamic sound propagation channel and user’s movement, (iii) High accuracy: achieving high accuracy compared to existing works.

**Motivations.** To achieve a device-free, robust passive liveness detection, in this study, we propose ArrayID, a microphone array-based liveness detection system, to effectively defend against spoofing attacks. ArrayID is motivated from the basic observation that the microphone array has been widely adopted by the mainstream smart speakers (e.g., both of Amazon Echo 3rd Gen [30] and Google Home Max [45] having 6 microphones), which is expected to significantly enhance the diversity of the collected audios thanks to the different locations and mutual distances of the microphones in this array. By exploiting the audio diversity, ArrayID can extract more useful information related to the target user, which is expected to significantly improve the robustness and accuracy of the liveness detection.

**Challenges.** To implement this basic idea, this study tries to address the following three key challenges: (i) Theoretically, what is the advantage of adopting a microphone array compared with a single microphone? (ii) Considering the dynamic audio propagation channel, how can we eliminate the distortions caused by environment factors (e.g., dynamic air channel and user’s position changes) by leveraging the microphone array? (iii) Considering that our work is the first one to leverage microphone array for liveness detection and there is no large-scale microphone array-based indoor audio dataset available so far, how can we demonstrate the effectiveness and accuracy of the proposed scheme?

To solve the above three problems, we first build a sound propagation model based on the wave propagation theory and then leverage it to theoretically assess the impact of environment factors (e.g., articulatory gesture, sound decay pattern, propagation path) on the final collected audio’s spectrum. Secondly, after collecting multi-channel audio, we give a formal definition of array fingerprint and discuss the theoretic performance gain of adopting microphone array, which can leverage the relationship among different channels’ data to eliminate the distortions caused by factors including air channel and user’s position changes. Thirdly, we collect and build the first array fingerprint-based open dataset containing multi-channel voices from 38,720 voice commands. To evaluate the effectiveness of ArrayID, we compare ArrayID with previous passive schemes (i.e., CAFIELD [49], and VOID [3]) on both our dataset and a third-party dataset called ReMasc Core dataset [18]. ArrayID achieves the authentication accuracy of 99.84% and 97.78% on our dataset and ReMasc Core dataset, respectively, while the best performance of existing schemes [3,49] on these two datasets are 98.81% and 84.37% respectively. The experimental results well demonstrate the effectiveness and robustness of ArrayID.

To the best of our knowledge, our work is the first to exploit the circular microphone array of the smart speaker to perform passive liveness detection in a smart home environment. The contributions of this study are summarized as follows:

- **Novel system.** We design, implement and evaluate ArrayID for thwarting voice spoofing attacks. By only using audio collected from a smart speaker, ArrayID does not require the user to carry any device or conduct additional action.

- **Effective feature.** We give a theoretical analysis of principles behind passive detection and propose a robust liveness feature: the array fingerprint. This novel feature both enhances effectiveness and broadens the application scenarios of passive liveness detection.

- **Robust performance.** Experimental results on both our dataset and a third-party dataset show that ArrayID outperforms existing schemes. Besides, we evaluate multiple factors (e.g., distance, direction, spoofing devices, noise) and demonstrate the robustness of ArrayID.

- **New large-scale dataset.** A dataset containing 14 different spoofing devices collected by microphone array will be available to researchers, vendors, and developers for evaluating further liveness detection schemes.

The rest of this paper is organized as follows. In Section 2, we introduce the preliminaries of this study. In Section 3, we propose the concept of the array fingerprint and proof its advantages by both theoretical analysis and experiments. We elaborate on the detailed design of ArrayID in Section 4, which is followed by evaluation, discussion, and related work in Sections 5, 6, and 7, respectively. Finally, we conclude this paper in Section 8.

## 2 Preliminaries

### 2.1 Threat Model

In this study, we focus on the voice spoofing attack, which can be categorized into the following two types.

- **Classical replay attacks.** To fool the voice assistance, the attacker collects the legitimate user’s audio samples and then
plays it back with a high-quality loudspeaker [12]. The victim’s voice audio can be recorded or captured in many manners, which is not limited to websites, daily life talking, phone calls, etc. The replay attack is the most effective among various spoofing approaches since it preserves the most comprehensive voiceprint of the victim and requires no cumbersome hardware configurations and software parameter fine-tuning.

**Advanced adversarial attacks.** Even if attackers can only collect a limited number of the target user’s voice samples, by adopting the latest voice synthesized technique [8], it is still feasible to attack existing speech recognition and speaker verification systems. For instance, the adversary can craft subtle noises into the audio (e.g., hidden voice [7], music [50] or a broadcaster’s voice [53]) or inaudible ultrasounds [35, 52] to launch a replay attack without raising the victim’s concern. Moreover, by carefully modifying the spectrum of spoofing audio, the modulated attack [48] proposed by Wang et al., demonstrates the feasibility of bypassing existing mono audio-based liveness detection schemes [6].

Similar to the previous works [3, 6, 49], in this study, the adversary is assumed to already obtain the victim’s audio samples and can remotely control the victim’s audio device (e.g., smart TV, smartphone) to launch the voice spoofing attack. In this study, we mainly investigate how to leverage passive liveness detection to thwart replay attacks since most of the existing voice biometric-based authentication (human speaker verification) systems are vulnerable to this kind of replay attack. We also study ARRAYID’s performance on thwarting advanced attacks including modulated attack [48], hidden voice [7], and VMask [53] in Section 5.4.3.

## 2.2 Sound Generation and Propagation

Before reviewing existing passive liveness detection schemes, it is important to describe the sound generation and propagation process.

**Sound generation.** As shown in Figure 1(a), voice commands are generated by a human or electrical device (i.e., loudspeaker). For the loudspeaker, given an original voice command signal $x(f, t)$, where $f$ represents the frequency and $t$ is time, the loudspeaker utilizes the electromagnetic field change to vibrate the diaphragm. The movement of the diaphragm suspends and pushes air to generate the sound wave $s(f, t) = h_{dev}(f, t) \cdot x(f, t)$, where $h_{dev}(f, t)$ represents the channel gain in the sound signal modulation by the device as shown in Figure 1(b). Similarly, when a user speaks voice commands, their mouth and lips also modulate the air and we can use $h_{user}(f, t)$ to represent the modulation gain, where the generated sound is $s(f, t) = h_{user}(f, t) \cdot x(f, t)$. Finally, the generated sound $s(f, t)$ is spread through the air and captured by the smart speaker.

**Sound transmission.** Currently, smart speakers usually have a microphone array (e.g., Amazon Echo 3rd Gen [30] and Google Home Max [45] both have 6 microphones). For a given microphone, when sound is transmitted to it, the air pressure at the microphone’s location can be represented as $y(f, t) = h_{air}(d, f, t) \cdot s(f, t)$, where $d$ is the distance of the transmission path between the audio source and the microphone and $h_{air}(d, f, t)$ is the channel gain in the air propagation of the sound signal.

**Sound processing within the smart speaker.** Finally, $y(f, t)$ is converted to an electrical signal by the microphone. Since the microphones employed by mainstream smart speakers usually have a flat frequency response curve in the frequency area of the human voice, we assume smart speakers save original sensed data $y(f, t)$ which is also adopted by existing studies [49]. Finally, the collected audio signal is uploaded to the smart home cloud to further influence the actions of smart devices.

## 2.3 Passive Liveness Detection

The recently proposed liveness detection schemes could be divided into two categories: mono channel-based detection (e.g., Sub-bass [6] and VOID [3]) and fieldprint-based detection (i.e., CAFIELD [49]).

### 2.3.1 Mono Channel-based Detection

**Principles.** As shown in Figure 1(a), the different sound generation principles between real human and electrical spoofing devices could be characterized as two different filters: $h_{user}(f, t)$ and $h_{dev}(f, t)$. If ignoring the distortion in the sound signal transmission, $h_{air}(d, f, t)$ could be considered as a constant value $A$. Thus, the received audio samples in authentic and spoofing attack scenarios are $\mathcal{D}_{auth}(d, f, t) = A \cdot h_{user}(f, t) \cdot x(f, t)$ and $\mathcal{D}_{spoof}(d, f, t) = A \cdot h_{dev}(f, t) \cdot x(f, t)$.

1In the real-world scenario, there is no such $x(f, t)$ during human voice generation process. However, the concepts of $x(f, t)$ and $h_{user}(f, t)$ are widely used [6] and will help us understand features in Section 4.3.
respective. Since $A$ and $x(f,t)$ are the same, it means that the spectograms of the received audio samples already contain the identity of the audio source (the real user $h_{user}(f,t)$ or the spoofing one $h_{dev}(f,t)$). Figure 2(a) shows the spectrums of the voice command “OK Google” and its spoofing counterpart. It’s observed that the sub-bass spectrum (20-300 Hz) between two audio samples are quite different even if they are deemed similar, and this phenomenon is utilized by mono channel-based schemes such as Sub-base [6].

**Limitations.** However, in a real-world environment, $h_{air}(d,f,t)$ cannot always be regarded as a constant. The surrounding object’s shape and materials, the sound transmission path, and the absorption coefficient of air all affect the value of $h_{air}(d,f,t)$. As shown in Figure 2(a) and Figure 2(b), the spectograms of authentic and spoof audio samples change drastically when putting the smart speaker in different rooms. The experimental result from Section 5.2 and [3] demonstrates the performance of liveness detection undergoes degradation when handling datasets in which audios are collected from complicated environments (e.g., ASV Spoofing 2017 Challenge [23], ReMasc Core [18]).

### 2.3.2 Fieldprint-based Detection

**Principles.** The concept of fieldprint [49] is based on the assumption that audio sources with different articulatory behaviors will cause a unique “sound field” around them. By measuring the field characteristics around the audio source, it is feasible to induce the audio’s identity. CAFIELD is the typical scheme which deploys two microphones to receive two audios $y_1(f,t)$ and $y_2(f,t)$, and defines the fieldprint as:

$$Field = \log \left( \frac{y_1(f,t)}{y_2(f,t)} \right).$$

**Limitations.** Measuring stable and accurate fieldprint requires the position between the audio source and the print measure sensors must be relatively stable. For instance, CAFIELD only performs well when the user holds a smartphone equipped with two microphones close to the face in a fixed manner. The fieldprint struggles in far distances (e.g., greater than 40 cm in [49]), making it unsuitable for a home environment, in which users want to communicate with a speaker across the room. The goal of this study is to propose a novel and robust feature for passive liveness detection.

### 3 Array Fingerprint

In this section, we propose a novel and robust liveness feature **array fingerprint** and elaborate the rationale behind ARRAYID by answering the following critical questions:

**RQ1:** How can we model the sound propagation in smart speaker scenarios and answer why existing features (e.g., fieldprint) cannot be effective in such scenarios?

**RQ2:** How can we extract a useful feature from multi-channel voice samples that is robust regarding a user’s location and microphone array’s layout?

**RQ3:** What are the benefits of the array fingerprint? Is it effective and robust to the distortions caused by environmental factors?

#### 3.1 Theoretical Analysis on Sound Propagation for Smart Speakers

To answer question RQ1, we give a theoretical analysis of sound propagation in a smart speaker scenario by following the model proposed in Section 2.2 and discuss the limitations of the previous works.

**Sound propagation model for smart speakers.** Figure 3 illustrates the scenario when audio signals are transmitted from source to microphone array. The audio source is regarded as a point with coordinate $(L, 0)$ and the microphones are evenly distributed on a circle. Given the $k$-th microphone $M_k$, the collected audio data is $y_k(f,t) = h_{air}(d_k,f,t) \cdot s(f,t)$, where $d_k$ is the path distance from the audio source to $M_k$. In the theoretical analysis, to simplify the description of the channel gain $h_{air}(d_k,f,t)$, we apply the classic spherical sound wave transmission model in air [19].

$$h_{air}(d_k,f,t) = Ce^{-\alpha_k d_k} = Ce^{-\alpha(s(f,t))d_k},$$

where $C$ is the attenuation coefficient, and $\alpha_k$ is the absorption coefficient which varies with the signal frequency $f$. Therefore, we replace $\alpha_k$ with $\alpha(s(f,t))$. Then, from Section 2.2, the collected audio in $M_k$ can be represented as:

$$y_k(f,t) = h_{air}(d_k,f,t) \cdot s(f,t) = Ce^{-\alpha(s(f,t))d_k} \cdot s(f,t).$$

Existing passive liveness detection schemes are vulnerable to environmental changes. From equation 3, it is observed that changing the relative distance between the microphone and audio source will cause non-linear distortion on the microphone’s collected signal. Such distortion is related to the original $s(f,t)$ and thus is hard to be eliminated. This is the reason why mono channel-based detection schemes are fragile to the change of propagation path.

---

2 In real-world scenarios, sound decay in the air is correlated with many factors such as temperature, medium, and surrounding objects. Using the classical model simplifies the question and the effectiveness of ARRAYID will be demonstrated by experiments in Section 3.3.
For the fieldprint-based solution, from equation 1, the extracted feature can be represented as $\log(y_{i}/y_j) = −\alpha(s(f, t)) \cdot \log(e) \cdot (d_i − d_j)$. When the positions of the microphone pair are fixed (i.e., $d_i − d_j$ can be regarded as a constant), the above feature is a function of originally generated $s(f, t)$ containing liveness factor as described in Section 2.3. However, when the microphone’s position changes, the $d_i − d_j$ will no longer be a stable value, and leveraging such a feature becomes infeasible.

### 3.2 Advantage of Array Fingerprint: Definition and Simulation-based Demonstration

In this subsection, we answer RQ2 by defining the array fingerprint and mathematically demonstrating its effectiveness.

From the theoretical analysis in Section 3.1, to achieve robust liveness detection, the extracted channel feature has to minimize the effects of the propagation factors such as $C$ and $d_k$. Inspired by the circular layout of microphones in smart speaker as shown in Figure 3, we define the array fingerprint $A_F$ as below:

$$A_F = \text{std}(\log[y_1, y_2, \ldots, y_N])$$
$$= \text{std}(C − \alpha(s(f, t)) \cdot \log(e) \cdot [d_1, d_2, \ldots, d_N])$$
$$= −\alpha(s(f, t)) \cdot \log(e) \cdot \text{std}([d_1, d_2, \ldots, d_k])$$
$$= A_F(s(f, t), \sigma_d).$$

(4)

From equation 4, we know that the array fingerprint is mainly dominated by source audio $s(f, t)$ and standard deviation of propagation distances $\sigma_d = \text{std}([d_1, d_2, \ldots, d_N])$. However, to effectively capture the audio’s identity, which can be derived from $s(f, t)$, the hypothesis that $\sigma_d$ could be regarded as a constant parameter must be proved.

To demonstrate the above hypothesis, the propagation distance between audio source $S_0$ and each microphone should be precisely determined. To achieve this goal, as shown in Figure 3, we denote the center of the microphone array of the smart speaker and the audio source (e.g., human or electrical machine) as origin $O$ and $S_0(L, 0)$ respectively. For the $k$-th microphone $M_k$, its coordinate can be represented as:

$$\overline{OM}_k = \left(r \cdot \cos(\theta + \frac{2\pi(k-1)}{N}), \ r \cdot \sin(\theta + \frac{2\pi(k-1)}{N})\right),$$

(5)

where $r$ is the radius of the microphone array, $N$ is the number of microphones, and $\theta$ is the angle between $M_1$ and $M_N$.

### 3.3 Validation of Array Fingerprint

Besides theoretical analysis, to answer RQ3, we further validate the effectiveness of the proposed array fingerprint via a series of real-world case studies.

In the experiment, the participant is required to speak the command “Ok Google” at distances of 0.6 m and 1.2 m, respectively. Figure 5(a) shows the audio signal clips collected by a microphone array with six microphones, and the audio difference between different channels is obvious. When employing the concept of fieldprint, it is observed from Figure 5(b) that the fieldprints extracted from microphone pair...
(c) Dynamic power differences in different microphone pairs.

Figure 5: Illustration of stability of array fingerprint under two locations.

Figure 6: Differentiating human voice from two spoofing devices via array fingerprints under different propagation paths.

$(M_1, M_2)$ and $(M_1, M_3)$ are quite different. Among different distances, the fieldprints are also quite different. However, from Figure 5(c) we can see that the array fingerprints for different distances are very similar.

To show the distinctiveness of array-print, we also conducted replay attacks via smartphones and iPad (i.e., device #8 and # 3 in Table 6 of Appendix B). The normalized array fingerprints (i.e., $F_{SAP}$ in Section 4.3.1) are shown in Figure 6. It is observed that the array fingerprints for the same audio sources are quite similar, while array fingerprints for different audio sources are quite different. Our theoretical analysis and experimental results demonstrate the array fingerprint can serve as a better passive liveness detection feature. This motivates us to design a novel, lightweight and robust system which will be presented in the next section.

4 The Design of ARRAYID

As shown in Figure 7, we propose ARRAYID, a robust liveness detection system based on the proposed array fingerprint with other auxiliary features. ARRAYID consists of the following modules: Data Collection Module, Pre-processing Module, Feature Extraction Module, and Attack Detection Module. We will elaborate on the details of each module in this section.

4.1 Multi-channel Data Collection

Currently, most popular smart speakers, such as Amazon Echo and Google, employ a built-in microphone array to collect voice audio. However, due to privacy and commercial concerns, the user of the smart speaker cannot access the original audio data, only the transcribed text. To solve this problem, we utilize open modular development boards with voice interfaces (i.e., the Matrix Creator [31] and Seeed Respeaker [42]) to collect the data. Since these development boards have similar sizes to commercial smart speakers, ARRAYID evaluations on the above devices can be applied to a smart speaker without any notable alterations. Generally speaking, given a smart speaker with $N$ microphones, a sampling rate of $F_s$, and data collection time $T$, the collected voice sample is denoted as $V_{M \times N}$, where $M = F_s \times T$ and we let $V_i$ be the $i$-th channel’s audio $V(:, i)$. Then, the collected $V$ is sent to the next module.

4.2 Data Pre-processing

As shown in equation 4, the identity (i.e., real human or spoofing device) is hidden in the audio’s spectrogram. Therefore, before feature extraction, we conduct the frequency analysis on each channel’s signal and detect the audio’s direction.

Frequency analysis on multi-channel audio data. As described in Section 3.2, the audio spectrogram in the time-frequency domain contains crucial features for further liveness detection. ARRAYID performs Short-Time Fourier Transform (STFT) to obtain two-dimensional spectrograms of each channel’s audio signal. For the $i$-th channel’s audio $V_i$, which contains $M$ samples, ARRAYID applies a Hanning window to divide the signals into small chunks with lengths of 1024 points and overlapping sizes of 728 points. Finally, a 4096-sized Fast Fourier Transform (FFT) is performed for each chunk.

Figure 7: System overview.
and a spectrogram $S_i$ is obtained as shown in Figure 8(a).

**Direction detection.** Given a collected audio $V_{M \times N}$, to determine the microphone which is closest to the audio source, ARRAYID firstly applies a high pass filter with a cutoff frequency of 100 Hz to $V_{M \times N}$. Then, for the $i$-th microphone $M_i$, ARRAYID calculates the alignment errors $E_i = \text{mean}((V(:, i)-V(:,i))^2)$ [39]. Finally, from the calculated $E_i$, ARRAYID chooses the microphone with minimum alignment error as the corresponding microphone.

**4.3 Feature Extraction**

From equations 3 and 4, we observe that both audio spectrograms themselves and the microphone array’s difference contain the liveness features of collected audio. In this module, the following three features are selected by ARRAYID: **Spectrogram Array Fingerprint $F_{\text{SAP}}$, Spectrogram Distribution Fingerprint $F_{\text{SDP}}$, and Channel LPCC Features $F_{\text{LPC}}$.**

**4.3.1 Spectrogram Array Feature**

After obtaining the spectrogram $S = [S_1, S_2, \ldots, S_N]$ from $N$ channels’ audio data $V = [V_1, V_2, \ldots, V_N]$, ARRAYID firstly exploits the array fingerprint which is proposed in Section 3.2 to extract the identity of the audio source. To reduce the computation overhead, for $S_i$ with size $M_i \times N$, we only preserve the components in which frequency is less than the cutoff frequency $f_{\text{cutoff}}$. In this study, we empirically set $f_{\text{cutoff}}$ as 5 kHz. The resized spectrograms are denoted as $\text{Spec} = [\text{Spec}_1, \text{Spec}_2, \ldots, \text{Spec}_k]$, where $\text{Spec}_k = S_k(\cdot; M_{\text{spec}}, \cdot)$. In this study, with sampling rate $F_s = 48$ kHz and FFT points $N_{FFT} = 4096$, the $M_{\text{spec}}$ is $\left\lfloor \frac{F_s \times N_{FFT}}{f_{\text{cutoff}}} \right\rfloor = 426$.

Figure 8(a) illustrates $\text{Spec}$ of three channels of the command "OK Google." It is observed that different channels’ spectrograms are slightly different. However, directly using such subtle differences would cause an inaccurate feature. Thus, ARRAYID transforms $\text{Spec}_k$ into a grid matrix $G_k$ with size $M_G \times N_G$ by dividing $\text{Spec}_k$ into $M_G \times N_G$ chunks and calculates the sum of magnitudes within each chunk. The element of $G_k$ could be represented as:

$$G_k(i,j) = \text{sum}(\text{Spec}_k(1+(i-1) \cdot S_M ; i \cdot S_M , 1+(j-1) \cdot S_N ; j \cdot S_N)$$

where $S_M = \left\lfloor \frac{M_{\text{spec}}}{M_G} \right\rfloor$ and $S_N = \left\lfloor \frac{N_{\text{spec}}}{N_G} \right\rfloor$ are the width and length of each chunk. Note that some elements of $\text{Spec}_k$ may be discarded, however, it does not affect the feature generation, since ARRAYID focuses on the differences between spectrograms according to equation 4. In this study, $M_G$ and $N_G$ are set to 100 and 20 respectively, and Figure 8(b) shows the spectrogram grids from the first, third and fifth microphone. The difference among elements in $G = [G_1, G_2, \ldots, G_N]$ is now very obvious. For instance, the grid values in the red rectangles of Figure 8(b) are quite different.

Then, based on equation 4, ARRAYID calculates the array fingerprint $F_G$ from the spectrogram $G$. $F_G$ has the same size as $G_k$, and the elements of $F_G$ can be represented as:

$$F_G(i,j) = \text{std}(\{G_1(i,j), G_2(i,j), \ldots, G_N(i,j)\}).$$

Figure 8(c) illustrates the $F_G$ containing $N_G$ chunks calculated from spectrogram grids as shown in Figure 8(b). However, we find that in different time chunks, the $F_G(\cdot,i)$ varies. The reason is that different phonemes are pronounced by different articulatory gestures, which can be mapped to a different $h_{\text{phon}}(f,t)$ function in Section 2.2. To solve this problem, we leverage the idea that even though different phonemes contain different gestures, there are common components over a long duration of time. Therefore, ARRAYID averages the $F_G$ across the time axis, and Figure 8(c) shows the average result $\bar{F}_G$. ARRAYID performs a 5-point moving average and normalization on $\bar{F}_G$ to remove noise and generate the spectrogram array fingerprint $F_{\text{SAP}}$.

Figure 8(d) gives a simple demonstration about the effectiveness of the $F_{\text{SAP}}$ feature generation process. We test three
voice commands “OK Google”, “Turn on Bluetooth” and “Record a video”, while the distances between the speaker and microphone array are set as 0.6 m and 1.2 m in the first two commands and the last command, respectively. In Figure 8(d), it is observed that the different commands result in a similar array fingerprint, and the feature difference between authentic audio and spoofing audio is clear. Finally, since ARRAYID requires a fast response time, the feature should be lightweight. So, the $F_{\text{SDP}}$ is re-sampled to the length of $N_{\text{SDP}}$ points. In this study, we empirically choose $N_{\text{SDP}}$ as 40.

### 4.3.2 Spectrogram Distribution Feature

Besides $F_{\text{SDP}}$, as mentioned in equation 3, the spectrogram distribution also provides useful information related to the identity of the audio source. Thus we also extract spectrogram distribution fingerprint $F_{\text{SDP}}$ for liveness detection. Given a spectrogram $S_k$ from the $k$-th channel, ARRAYID calculates a $N_G$-dimension vector $Ch_k$ in which $Ch_k(i) = \sum_{j=1}^{M_{\text{spec}}} S_k(j,i)$, where $M_{\text{spec}}$ and $N_G$ are set as 85 and 20 respectively in this study.\footnote{When calculating $F_{\text{SDP}}$, we set the cutoff frequency as 1 kHz since most human voice frequency components are located in the 0–1 kHz range and the corresponding $M_{\text{spec}}$ is 85 under the the parameters in Section 4.3.1.} For the audio with $N$ channels, the channel frequency strength $Ch = [Ch_1, Ch_2, ..., Ch_N]$ is obtained.

Figure 9(a) and 9(b) show channel frequency strength $Ch_1$ and $Ch_4$ of first and fourth channels from authentic and spoofing audios. It is observed that $Ch$ from real human and spoofing device are quite different. Therefore, we utilize the average of channel frequency strengths $\bar{Ch}$ and re-sample its length to $N_{Ch}$ as the first component of $F_{\text{SDP}}$. In this study, $\bar{Ch}(i) = \text{mean}(\{Ch_1(i), Ch_2(i), ..., Ch_N(i)\})$ and $N_{Ch}$ is set to 20. We can also find that for the same audio, $Ch$ from different channels have slightly different magnitudes and distributions. To characterize the distribution of $Ch$, for $Ch_k$ from the $k$-th channel, ARRAYID first calculates the cumulative distribution function $Cum_k$ and then determines the indices $\mu$ which can split $Cum_k$ uniformly. As shown in Figure 9(a) and 9(b), the $Ch_k$ are segmented into 6 bands. ARRAYID sets the $Thr_1 = \{0.1, 0.3, 0.5, 0.7, 0.9\}$, and the index $\mu(k,i)$ of the $i$-th $Thr$ for $Ch_k$ satisfies the following condition:

$$Cum_k(\mu(k,i)) \leq Thr_1 \leq Cum_k(\mu(k,i) + 1).$$

After obtaining the $N \times 5$ indices $\mu$, we utilize the mean value $D_{\text{mean}}$ and standard deviation $D_{\text{std}}$ among different channels as a part of the spectrogram feature. Both $D_{\text{mean}}$ and $D_{\text{std}}$ are vectors with length of 5, where $D_{\text{mean}}(i) = \text{mean}(\mu(:,i))$ and $D_{\text{std}}(i) = \text{std}(\mu(:,i))$. Finally, ARRAYID obtains the spectrogram distribution fingerprint $F_{\text{SDP}} = \left[\bar{Ch}, D_{\text{mean}}, D_{\text{std}}\right]$. Figure 9(c) illustrates the $F_{\text{SDP}}$ from authentic and spoofing audios and demonstrates the robustness of $F_{\text{SDP}}$.

### 4.3.3 Channel LPCC Features

The final feature of ARRAYID is the Linear Prediction Cepstrum Coefficients (LPCC). Since each channel has unique physical properties, retaining the LPCC which characterizes a given audio signal could further improve the detection performance. For audio signal $y_k(t)$ collected by microphone $M_k$, ARRAYID calculates the LPCC with the order $p = 15$. Due to page limit, the details of LPCC extraction is introduced in Figure 16 and Appendix A respectively. To reduce the time overhead spent on LPCC extraction, we only preserve the LPCCs from aduio in these two channels ($M_1, M_{\text{modi}(+N/2, N)}$), where $M_i$ is the closet microphone derived from Section 4.2. Finally, we generate the final feature vector $X = [F_{\text{SDP}}, F_{\text{SDP}}, F_{\text{LPCC}}]$.

### 4.4 Classification Model

After generating the feature vector from the audio input, we choose a lightweight feed-forward back-propagation neural network to perform liveness detection. The neural network only contains three hidden layers with rectified-linear activation (layer sizes: 64, 32, 16). We adopt a lightweight neural network because it can achieve a quick response to the decision, which is essential for the devices in the smart home environment. We also discuss other possible classification models in Appendix C.

## 5 Evaluations

### 5.1 Experiment Setup

**Hardware setup.** Since it is hard for users to obtain audio files from popular smart speakers such as Google Home and Amazon Echo, in this study, to collect multi-channel audios, as shown in Figure 17 of Appendix B, we employ two open modular development boards (i.e., Matrix Creator, and Seeed ReSpeaker Core v2) with the sampling rate of 48 kHz to serve as smart speakers. The number of microphones in the Matrix and ReSpeaker are 8 and 6, respectively, and their radiuses are 5.4 cm and 4.7 cm respectively. For the spoofing device,
we employ 14 different electrical devices with various sizes and audio qualities whose detailed parameters are shown in Table 6 of Appendix B.

**Data collection procedure.** In this study, 20 participants are recruited to provide the multi-channel audio data. The data collection procedure consists of two phases: (i) **Authentic audio collection:** in this phase, the participant speaks 20 different voice commands as listed in Appendix B and the experimental session can be repeated multiple times by this participant. We pre-define four distances (i.e., 0.6 m, 1.2 m, 1.8 m, 2.4 m) between the microphone array and the participant can choose any of them in each session. For the speaking behavior, we ask the participant to speak command as she/he likes and did not specify any fixed speed/tone. (ii) **Spoofing audio collection:** in this phase, the manners adopted by the previous works [3, 49, 54], after collecting the authentic voice samples, we utilize the spoofing devices as listed in Table 6 to automatically replay the samples without the participant’s involvement. When replaying a voice command, the electrical device is placed at the same location as the corresponding participant.

**Dataset description.** After finishing experiments, we utilize PyAudioAnalysis tool to split the collected audio into multiple voice command samples.6 After removing incorrectly recognized samples, we get a dataset containing 32,780 audio samples. We refer to this dataset as MALD dataset and utilize it to assess ARRAY ID.7 The details of MALD dataset are shown in Table 7 of Appendix B. For instance, user #7 provides 600 authentic samples at three different positions (i.e., the distance of 0.6 m, 1.2 m and 1.8 m) and we utilize these collected samples with three spoofing devices (i.e., SoundLink, iPad, iPhone) to generate 1,800 spoofing samples.

**Training procedure.** As mentioned in Section 4.4, ARRAY ID needs to be trained with audio samples before detecting spoofing attacks. When evaluating the overall performance of ARRAY ID on the collected MALD dataset in Section 5.2, we perform the two-fold cross-validation. In each fold (i.e., training procedure), half samples are chosen to generate a classifier and the validation dataset proportion is set as 30%. When evaluating the impact of other factors as shown in Section 5.3 and Section 5.4, the training procedure depends on the specific experiment, and we show the training dataset before presenting the evaluation results.

**Evaluation metrics.** Similar to previous works [3, 32, 49], in this study, we choose accuracy, false acceptance rate (FAR), false rejection rate (FRR), true rejection rate (TRR), and equal error rate (ERR) as metrics to evaluate ARRAY ID. The accuracy means the percentage of the correctly recognized samples among all samples. FAR represents the rate at which a spoofing sample is wrongly accepted by ARRAY ID, and FRR characterizes the rate at which an authentic sample is falsely rejected. EER provides a balanced view of FAR and FRR and it is the rate at which the FAR is equal to FRR.

**Ethics consideration.** The experiments are under the approval of the institutional review board (IRB) of our institutions. During the experiments, we explicitly inform the participants about the experimental purpose. Since only the voice data are collected and stored in an encrypted dataset, there is no health or privacy risk for the participant.

### 5.2 Performance of ARRAY ID

**Overall accuracy.** When evaluating ARRAY ID on our own MALD dataset, we choose two-fold cross-validation, which means the training and testing datasets are divided equally. ARRAY ID achieves the detection accuracy of 99.84% and the EER of 0.17%. More specifically, for all 32,780 samples, the overall FAR and FRR are only 0.05% (i.e., 13 out of 22,539 spoofing samples are wrongly accepted) and 0.39% (i.e., 40 out of 10,241 authentic samples are wrongly rejected) respectively. The results show that ARRAY ID is highly effective in thwarting spoofing attacks.

**Per-user breakdown analysis.** To evaluate the performance of ARRAY ID on different users, we show the FAR and FRR of each user in Figure 10. Note that, for six users (i.e., users #11, #12, #15, #16, #17, #18) which are not shown in this figure, there is no detection error. When considering FAR, it is observed that the false acceptance cases only exist in 6 users. Even in the worst cases (i.e., user #20), the false acceptance rate is still less than 0.51%. When considering FRR, the false rejection cases are distributed among 14 users. It’s observed that only the FRRs of users #3 and #20 are above 1%. Although the performance of ARRAY ID on different users is different, even for the worst-case (i.e., user #20), the detection accuracy is still at 99.0%, which demonstrates the effectiveness of ARRAY ID.

**Time overhead.** For a desktop with Intel i7-7700T CPU and 16 GB RAM, the average time overhead on 6-channel and 8-channel audios are 0.12 second and 0.38 seconds, respectively. Note that it is easy for the existing smart home systems (e.g., Amazon Alexa) to incorporate ARRAY ID to their current industrial level solutions in the near future. In that case, both the speech recognition and liveness detection can be done in the cloud [30]. Therefore, by leveraging the hardware configuration of the smart speaker’s cloud (e.g., Amazon Cloud [16]),
which is much better than our existing one (CPU processor), we believe that the time overhead can be further reduced and will not incur notable delays.

**Comparison with previous works.** We further compare the performance of ARRAYID with existing works to demonstrate the superiority of the proposed array fingerprints. To eliminate the potential bias in our collected MALD dataset, we also exploit a third-party dataset named ReMasc Core which contains 12,023 voice samples from 40 different users.

We re-implement mono audio-based scheme VOID [3] and two-channel audio-based scheme CAFIELD [49]. For a fair comparison, we replicate their parameters and classification models as shown in Appendix C.

As shown in Table 1, since MALD dataset is collected in the indoor smart home environment and ReMasc is collected in both indoor, outdoor, and vehicle environments, the detection accuracy varies among these two datasets. ARRAYID is superior to previous works in both datasets. Especially for the ReMasc Core dataset in which only half of the audio samples are collected in the indoor environment, ARRAYID is the only scheme that achieves an accuracy above 98.25%. The two-channel-based scheme CAFIELD, gets relatively low performance on both the MALD dataset and ReMasc dataset. It is quite natural since CAFIELD claimed it needs the user to hold the device with fixed gestures and short distances. In summary, these results demonstrate that compared with mono audio-based or two-channel-based scheme, exploiting microphone array-based feature achieves superior performance in the liveness detection task.

### 5.3 Impact of Various Factors on ARRAYID

In this subsection, we evaluate the impact of various factors (e.g., distance, direction, user movement, spoofing device, microphone array type) on ARRAYID.

#### Impact of changing distance.

To evaluate the performance of ARRAYID on a totally new distance, we recruit four participants to attend experiments at three different locations (i.e., 1.2 m, 1.8 m, 2.4 m). We totally collect 2,410 authentic and 2,379 spoofing audio samples. For a given distance, the classifier is trained with audios at this distance and tested on audios at other distances. As shown in Table 2, compared with the performance in Section 5.2, ARRAYID’s performance undergoes degradation when the audio source (i.e., the human or the spoofing device) changes its location. However, in all cases, ARRAYID achieves an accuracy above 99.4%, which demonstrates ARRAYID is robust to the training distance. This result is also conform to the theoretical analysis in Section 3.2

#### Impact of changing direction.

In Section 5.1, when collecting audio samples, most participants face the smart speaker while generating voice commands. To explore the impact of the angles between the user’s face direction and the microphone array, we recruit 10 participants to additionally collect authentic voice samples in four different directions (i.e., front, left, right, back) and then the spoofing device #8 in Table 6 is utilized to generate spoofing audios. As shown in Table 3, we totally collect 4,219 authentic samples and 3,830 spoofing samples. Then, we use the classification model trained in Section 5.2 to conduct liveness detection. It is observed from Table 3 that in all scenarios, ARRAYID achieves an accuracy above 99.3%, which means ARRAYID is robust to the change of direction.

#### Impact of user movement.

As similar to the above paragraphs, we recruit 10 participants to speak while walking. Then, the participant walks while holding a spoofing device (i.e., Amazon Echo) and plays spoofing audios. We collect 1,999 authentic and 1,799 spoofing samples, and the classifier is the same as that in Section 5.2. The detection accuracy is 98.2% which demonstrates that ARRAYID and the array fingerprint are robust even with the user’s movement.

#### Impact of microphone numbers in the smart speaker.

Studying the relationship between ARRAYID’s performance and the number of microphones could help the smart speaker vendors to determine microphone configurations. Note that the data in MALD dataset can be divided into six-channel (collected by ReSpeaker) and eight-channel (collected by Ma-

### Table 1: The detection accuracy on both datasets.

<table>
<thead>
<tr>
<th>Liveness feature</th>
<th>Dataset</th>
<th>MALD dataset</th>
<th>ReMasc dataset</th>
</tr>
</thead>
<tbody>
<tr>
<td>Microphone array</td>
<td>99.84%</td>
<td>97.78%</td>
<td></td>
</tr>
<tr>
<td>Mono feature</td>
<td>98.81%</td>
<td>84.37%</td>
<td></td>
</tr>
<tr>
<td>Two-channel</td>
<td>77.99%</td>
<td>82.44%</td>
<td></td>
</tr>
</tbody>
</table>

### Table 2: Performance when changing the distance.

<table>
<thead>
<tr>
<th>Training position (m)</th>
<th>1.2</th>
<th>1.8</th>
<th>2.4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accuracy (%)</td>
<td>99.41</td>
<td>99.53</td>
<td>99.66</td>
</tr>
<tr>
<td>EER (%)</td>
<td>1.11</td>
<td>0.93</td>
<td>0.69</td>
</tr>
</tbody>
</table>

### Table 3: Performance under different directions.

<table>
<thead>
<tr>
<th>Direction</th>
<th>Front</th>
<th>Left</th>
<th>Right</th>
<th>Back</th>
</tr>
</thead>
<tbody>
<tr>
<td># authentic samples</td>
<td>1020</td>
<td>1004</td>
<td>1195</td>
<td>1000</td>
</tr>
<tr>
<td># spoofing samples</td>
<td>980</td>
<td>947</td>
<td>971</td>
<td>932</td>
</tr>
<tr>
<td>Accuracy (%)</td>
<td>100</td>
<td>99.69</td>
<td>99.51</td>
<td>99.74</td>
</tr>
<tr>
<td>EER (%)</td>
<td>0</td>
<td>0.59</td>
<td>1.08</td>
<td>0.43</td>
</tr>
</tbody>
</table>
trix) audios. Then, we generate four-channel audio data from the data collected by Matrix device by extracting data from microphones \( (M_1, M_3, M_5, M_7) \).

For three audio groups with 4, 6, and 8 channels respectively, after conducting two-fold cross-validation on each group, the detection accuracies of ARRAYID are 99.78\%, 99.82\%, and 99.90\% respectively. That means changing the number of channels doesn’t cause a significant effect on ARRAYID’s performance. From the theoretical analysis in Section 3.2 and Figure 4, the standard deviation of paths from source to each microphone could be regarded as a constant in a smart speaker scenario. Therefore, as long as the microphone array has a circular layout, ARRAYID could provide robust protection on thwarting voice spoofing.

**Impact of Spoofing Devices.** It is well known that different devices have different frequency-amplitude response properties, and thus may have different attacker power. To evaluate ARRAYID’s performance on thwarting different spoofing devices, we conduct an experiment based on the MALD dataset containing 14 spoofing devices as listed in Table 6 of Appendix B. As discussed in Section 6.1, to reduce the user’s enrollment burden, we set the training proportion as 10\%.

Table 4 illustrates the FAR of ARRAYID on each device in this case. It is observed that among 14 devices, the overall FAR is 0.58\% (i.e., 117 out of 20,290 spoofing samples are wrongly accepted). Besides, ARRAYID achieves overall 100\% detection accuracy on 5 devices (i.e., devices #2, #3, #5, #6, #7). Even in the worst case (i.e., device #12 Megaboom), the true rejection rate is still at 95.86\%. Furthermore, as shown in Section 5.2, when increasing the training proportion to 50\%, the false accept rate (FAR) of ARRAYID is only 0.05\%. In summary, ARRAYID is robust to various spoofing devices.

### 5.4 Robustness of ARRAYID

#### 5.4.1 Handling the Incomplete Enrollment Procedure

Similar to previous works [3, 49, 54], in Section 5.2, ARRAYID requires the user to participate in the enrollment procedures (i.e., providing both authentic and spoofing voice samples). Considering that participating in enrollment is not always feasible, we explore the robustness of ARRAYID in handling the case that users who did not participate in the complete enrollment procedures.

**Case 1: handling users who did not participate in any enrollment procedure.** In this case, we add an experiment to evaluate the performance of ARRAYID on participants that did not participate in the enrollment (i.e., unseen users). In the experiment, for each user in the MALD dataset, we train the classifier using other 19 users’ legitimate and spoofing voice samples and regard the user’s samples as the testing dataset. The detection accuracy of each user is shown in Figure 11. We also show the results described in Section 5.2 when users participate in the enrollment as a comparison.

From Figure 11, it is observed that the overall detection accuracy decreases from 99.84\% to 92.97\%. In the worst case (i.e., user #12), the detection accuracy decreases from 99.87\% to 74.53\%. The results demonstrate that ability of ARRAYID on addressing unseen users varies with different users. However, for 11 users, ARRAYID can still achieve detection accuracies higher than 95\%. The overall results demonstrate that ARRAYID is still effective when addressing unseen users.

The performance degradation when addressing unseen users remains an open problem in the area of liveness detection [3, 6, 32, 54]. To partially mitigate this issue, a practical solution is requiring the unseen users to provide only authentic voice samples to enhance the classifier (i.e., case 2 discussed below).

**Case 2: handling a user with only authentic samples (without spoofing samples).** In this case, we consider another situation that the user partially participates in the enrollment and provides only authentic voice samples. We add an experiment by leveraging the MALD dataset. Note that, we assume the attacker only utilizes existing devices in the smart home to conduct spoofing. Thus a total of 18 users are selected (i.e., users #19 and #20 are excluded because their spoofing devices are never used by others in MALD dataset), whose spoofing devices are listed in Table 6 of Appendix B. During the experiment, for each selected user, ARRAYID is trained with this user’s authentic voice samples, and generic spoofing samples provided by other 17 users. Then, in the evaluation phase, we test the ability of ARRAYID to detect attack samples of this user and calculate the corresponding detection accuracy (i.e., TRR).

Figure 12 illustrates the detection accuracy under two different enrollment configurations. For all 18 users, the overall accuracy (i.e., TRR) decreases from 99.96\% in the classical enrollment scenario described in Section 5.2 to 99.68\% in this partial enrollment scenario. For 11 users (i.e., user #4, #5, #8, #9, #10, #11, #13, #14, #15, #16, and #17), the overall detection accuracies are higher than 95\%.
Only authentic samples

Figure 12: Detection performance under partial enrollment.

9, #11, #12, #14, #15, #16, #17, #18), the accuracy remains 100% in both scenarios. For the other 7 users, the accuracy decreases slightly due to a lack of knowledge of the user’s attack samples in the classifier, but all of them achieve the accuracy of above 96%, which demonstrates the effectiveness of ARRAYID in the partial enrollment scenario.

5.4.2 Liveness Detection on Noisy Environments

We add an experiment to evaluate the impact of background noise. As shown in Figure 13(a), to ensure the noise level is consistent when the user is speaking a voice command, we place a noise generator to play white noise during the data collection. We utilize an advanced sound level meter (i.e., Smart Sensor AR824) with an A-weighted decibel to measure the background noise level. The strengths of noise level at the microphone array are set as 45 dB, 50 dB, 55 dB, 60 dB, and 65 dB respectively, and a total of 4,528 audio samples are collected from 10 participants and the spoofing device #13 (i.e., Amazon Echo plus).

We utilize the classifier in Section 5.2 where the noise level is 30 dB to conduct liveness detection. As shown in Figure 13(b), when increasing the noise level from 45 dB to 65 dB, the accuracy decreases from 98.8 % to 86.3 %. It is observed that ARRAYID can still work well when the background noise is less than 50 dB, which also explains why ARRAYID can handle the audio samples of the ReMasc Core dataset collected in an outdoor environment. However, when there exists strong noise, since the feature of ARRAYID is only based on the collected audios, the performance of ARRAYID degrades sharply. We discuss this limitation in Section 6.3 and leave it for future work.

5.4.3 Defending against Advanced Spoofing Attacks

Thwarting modulated attacks. In this subsection, we first study the performance of ARRAYID under the emerging modulated attack [48]. By modulating the spectrum of replayed audio, the modulated attack [48] identifies an important threat to existing liveness detection schemes. To achieve this goal, in the attack model, the adversary first needs to use a microphone of the target device to collect the target user’s authentic voice samples. Then, the adversary physically approaches the spoofing device to measure its frequency amplitude curve and the corresponding inverse filter using the target microphone. Finally, by applying the inverse filter on the authentic audio and playback it via the spoofing device, for the target microphone, the spectrum of the collected modulated audio is similar to the collected authentic audio as shown in Figure 14(a). However, since the array fingerprint characterizes the difference among the multiple microphones, it is feasible for ARRAYID to thwart modulated attacks.

We conduct a case study to demonstrate the robustness of the array fingerprint. We select an Amazon Echo and a ReSpeaker microphone array as the spoofing and target device, respectively, and follow the steps in [48] to re-implement modulated attack. We recruit a volunteer to provide an authentic voice command and then collect its corresponding classic replay and modulated audios generated by the Echo device.

Figure 14 shows spectrums and array fingerprints of authentic audio and its corresponding replay and modulated samples. It is observed from Figure 14(a) that, for a given channel, the spectrum of modulated audio (i.e., FFT Amplitude of the first channel audio $V_1$) is similar to that in the authentic audio, which means it can bypass many existing liveness detection schemes. However, since the human vocal organs and spoofing devices cannot be regarded as a point sound source, the sounds received in multiple microphones show the obvious
differences.\textsuperscript{10} And the difference between multiple channel audios (\textit{i.e.}, six channels in this experiment) characterized by array fingerprints still retains the audio's identity. As shown in Figure 14(b), the array fingerprint of the modulated sample is still similar to that of classic replay audio, which shows it is feasible for ARRAYID to thwart the modulated attack.

Then, we evaluate the effectiveness of ARRAYID on thwarting the modulated attack. In the experiment, we choose three different spoofing loudspeakers #3, #13, and #14 (\textit{i.e.}, Echo Plus, iPad 9, and Mi 9). We recruit 10 participants to provide authentic samples and follow the steps described in [48] to generate 1,990, 1,791, and 1,994 modulated attack samples for Echo, iPad, and Mi respectively. Due to the page limit, the details of modulated attacks are shown in Appendix D.

When employing the classifier in Section 5.2, the accuracy of ARRAYID on detecting the modulated samples among Echo, iPad, and Mi are 100\%, 92.74\%, and 97.29\% respectively. In summary, ARRAYID can successfully defend against the modulated attack, but the performance varies with different spoofing devices. Considering combining ARRAYID with the dual-domain detection proposed in [48] can further improve the security of smart speakers.

**Other adversarial example attacks.** To validate ARRAYID’s robustness under adversarial attacks, we re-implement hidden voice attacks [7] and VMask [53] which breach speech recognition and speaker verification schemes, respectively. For each type of attack, we conduct voice spoofing 100 times, and the experimental results show that ARRAYID detects 100\% of attack audios for both attacks. The reason why ARRAYID could detect these attacks is that these attacks only aim to add subtle noises into source audio to manipulate the features (\textit{e.g.}, MFCC) interested by speech/speaker recognition schemes but the array fingerprint cannot be fully converted to that of the target victim.

6 Discussions

6.1 User Enrollment Time in Training

**Impact of training dataset size.** To reduce the user’s registration burden, we explore the impact of training data size on the performance of ARRAYID. For our collected MALD dataset, we set the training dataset proportion as 10\%, 20\%, 30\%, and 50\% respectively. The results are shown in Table 5. It is observed that the detection performance increases from 99.14\% to 99.84\% when involving more training samples. Note that, even if we only choose 10\% samples for training, ARRAYID still achieves the accuracy of 99.14\% and EER of 0.96\%, which is superior to previous works [54].

**Time overhead of user’s enrollment.** As mentioned in Section 5.1, the participant does not need to provide spoofing audio samples. Besides, as shown in Table 5, when setting the training proportion as 10\%, among 10,241 authentic samples from 20 users, the average number of audio samples provided by each user during the enrollment is only 51. Since the average time length of the voice command is smaller than 3 seconds, the enrollment can be done in less than 3 minutes. Compared with the time overhead on deploying an Alexa skill which is up to 5 minutes [20], requiring 3 minutes for enrollment is acceptable in real-world scenarios.

![Figure 15: Feature separation of 5 different users.](image)

<table>
<thead>
<tr>
<th>Training proportion</th>
<th>Authentic samples</th>
<th>Time (mm:ss)</th>
<th>Accuracy (%)</th>
<th>EER (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>10%</td>
<td>51</td>
<td>02:33</td>
<td>99.14</td>
<td>0.96</td>
</tr>
<tr>
<td>20%</td>
<td>103</td>
<td>05:09</td>
<td>99.47</td>
<td>0.55</td>
</tr>
<tr>
<td>30%</td>
<td>155</td>
<td>07:45</td>
<td>99.63</td>
<td>0.43</td>
</tr>
<tr>
<td>50%</td>
<td>263</td>
<td>13:09</td>
<td>99.84</td>
<td>0.17</td>
</tr>
</tbody>
</table>

6.2 Distinguish between Different Users

Since ARRAYID is designed for liveness detection, we mainly consider the voice command generated by the electrical loudspeaker as a spoofing sample in this study. This subsection explores the feasibility of user classification. We randomly select 250 authentic samples from 5 different users and then utilize t-Distributed Stochastic Neighbor Embedding (t-SNE) to reduce the dimension of their corresponding features. As shown in Figure 15, the feature vectors from different users are visually clustered after dimension reduction, which shows the feasibility of user classification. For all 10,241 authentic samples from 20 users, by leveraging two-fold cross-validation, ARRAYID achieves an overall speaker recognition accuracy of 99.88\%. Besides, the accuracy among different users ranges from 98.5\% to 100\%, which validates the effectiveness of ARRAYID on user authentication.

6.3 Limitations and Countermeasures

We discuss some limitations of ARRAYID in this subsection. The user’s burden on the enrollment. We can incorporate the enrollment into daily use to reduce the user’s time overhead on training ARRAYID. Firstly, the evaluation results from Section 5.3 show that ARRAYID is robust to the

\footnote{In the theoretical analysis of Section 3.1, to simplify the analysis of the classic replay attack, we regard the human and loudspeaker as points.}
change of user’s position, direction, and movement. That means the user can participate in the enrollment anytime. Then, to achieve this goal, we divide ARRAYID into working and idle phases. In the working phase, when a user generates a voice command, ARRAYID collects the audio and saves the extracted features. During the idle phase, ARRAYID can automatically update the classifier based on these new generated features. These steps can be done automatically without human involvement, which means ARRAYID can continuously improve its performance along with daily use. However, we admit allowing the automatically continuous retraining process may involve other potential risks. For instance, attackers can launch poisoning attacks to reduce the performance of speech recognition and speaker verification [1, 2, 11].

Impact of noise and other speakers. During the user’s enrollment, we assume the environment is silent and there is no user who is talking. As shown in Section 5.4.2, since ARRAYID is a passive liveness detection that only depends on audios, the strong noise or other speaker’s voice existing in the collected audios will inevitably degrade its performance. Therefore, the existence of noise and other users who are talking will increase the enrollment time. Fortunately, since ARRAYID is designed for the smart home or office environment, asking the users to keep a silent environment during enrollment is a reasonable assumption. We leave this issue as future work.

Temporal stability of array fingerprint. To evaluate the timeliness of ARRAYID, we recruit a participant to provide 100 authentic voice commands and launch voice spoofing per 24 hours. When using the classification model as described in Section 5.2 and the audio dataset collected by 24 hours and 48 hours later, ARRAYID still achieves over 98% accuracy. We admit that the generated feature may be variant when the participant changes her/his speaking manner or suffers from mood swings. As mentioned in Section 6.3, a feasible solution to address this issue is incorporating the enrollment into the user’s daily use to ensure the freshness of the classification model of ARRAYID.

7 Related Works

Attacks on smart speakers. The voice assistant is more vulnerable to the replay attack [4, 12, 21, 33]. Apart from the classic replay attack, other advanced attacks are proposed. Firstly, the attacker can leverage medias including ultrasonic and laser to spoof voice assistance without incurring the user’s perception [36, 41, 43, 52]. Secondly, the subtle noises can be employed to generate the adversarial examples attacks [7, 24, 26, 38, 46, 50, 59]. Thirdly, several attacking methods can activate the malicious app to threaten the security of our smart home system [17, 25, 57, 58]. Finally, Wang et al. [48] propose modulated attack, which is the latest advanced voice spoofing method, and we evaluate it in Section 5.4.3.

Multi-factor based defenses. As for the detecting method, some researches [15, 28, 29] are based on wearable devices. Besides, several works utilize the Doppler effect [34, 37], gestures according to sound [44], or other biometry characteristics to deal with the security issue. Lei et al. [28] and Meng et al. [32] proposed a wireless signal based method to thwart voice spoofing. Lee et al. [27] proposed a sonar-based solution to determine the user’s AoA (angle of arrival) to do liveness detection. Zhang et al. [55, 56] and Chen et al. [9] utilize the Doppler effect of ultrasonic and magnetic fields from loudspeakers as the essential characteristic for detecting attacks, respectively. However, these methods either require the user to wear some specialized devices or utilize other devices (e.g., wireless sensors) to measure the environmental change caused by humans.

Defenses relying on the collected audios. Shiota et al. [40] and Wang et al. [47] utilized the Pop noise when the human speaks to differentiate the voice commands generated by real humans and devices. Yan et al. [49] proposed the concept of using a fieldprint to detect spoofing attacks. Furthermore, Blue et al. [6] and Ahmed et al. [3] utilized spectral power patterns to identify spoofing attacks alongside a single classification model to achieve lightness in authentication. Besides, in terms of feature selection, Defraene et al. [10] and Kamble et al. [22] propose novel spectrum-based features respectively. We analyze these passive liveness detection schemes in Section 3.1. Recently, Zhang et al. [51] propose EarArray to defend against ultrasonic-based attacks (e.g., dolphin attacks [52]), but it is not designed to detect spoofing audios with human voice frequency.

8 Conclusion

In this study, we propose a novel liveness detection system ARRAYID for thwarting voice spoofing attacks without any extra devices. We give a theoretical analysis of existing popular passive liveness detection schemes and propose a robust liveness feature array fingerprint. This novel feature both enhances effectiveness and broadens the application scenarios of passive liveness detection. ARRAYID is tested on both our MALD dataset and another public dataset, and the experimental results demonstrate ARRAYID is superior to existing passive liveness detection schemes. Besides, we evaluate multiple factors and demonstrate the robustness of ARRAYID.

Acknowledgments

We thank the shepherd, Rahul Chatterjee, and other anonymous reviewers for their insightful comments. The authors affiliated with Shanghai Jiao Tong University were, in part, supported by the National Natural Science Foundation of China under Grant 62132013, and 61972453. Yuan Tian is partially supported by NSF award #1943100. Haojin Zhu is the corresponding author.
References


A LPCC Generation Process

For audio signal $y_k(t)$ collected by microphone $M_k$, to calculate the LPCC with the order $p = 15$, we firstly calculate the Linear Prediction Coding (LPC) as $a$:

$$a = \text{LPC}(y_k(t), p),$$

(10)

where $p$ is the order of LPC, and the collected LPC can be represented as $a = [a_0, a_1, \ldots, a_p]$. Then, for the LPCC coefficient $c = [c_0, c_1, \cdots, c_p]$, we have $c_0 = \ln(p)$, and for other elements could be calculated as:

$$c_i = -a_i - \sum_{k=1}^{i} \left(1 - \frac{k}{i}\right)a_k c_{i-k}.$$  

(11)

In this study, the order $p$ is set to 15, and the LPCCs on each channel are shown in Figure 16. In this figure, when $M_1$ is the closest microphone, for a microphone array with six channels, the opposite microphone is $M_6$. The LPCCs from these two channels are selected as $F_{LPC}$ in Section 4.3.3.

B Dataset Descriptions

First, the spoofing devices’ information including manufacturing, model, and size is shown in Table 6. Second, for each user, the data collection conditions including spoofing devices, distances, audio samples are summarized in Table 7. The dataset is collected by Matrix Creator and Seeed ReSpeaker core V2, which are shown in Figure 17. Finally, we list the 20 voice commands used in our experiments as below:

1. OK Google.
2. Turn on Bluetooth.
3. Record a video.
4. Take a photo.
5. Open music player.
6. Set an alarm for 6:30 am.
7. What is my schedule for tomorrow?
8. Square root of 2105?
10. Decrease volume.
11. Turn on flashlight.
12. Set the volume to full.
13. Mute the volume.
14. What’s the definition of transmit?
15. Call Pizza Hut.
16. Call the nearest computer shop.
17. Show me my messages.
18. Translate please give me directions to Chinese.
19. How do you say good night in Japanese?
20. Square root of 2105?

C Experimental Details of Comparison with Existing Schemes

When comparing ARRAYID with prior works, we strictly follow the steps described in VOID [3] and CAFIELD [49]. In this section, we take VOID as an example to show that ARRAYID is superior to existing schemes under various conditions. More specifically, we add an experiment to explore the impact of different classifier models on the liveness detection performance of ARRAYID and VOID.

We choose four different classification models: neural network, support vector machine with radial basis function kernel (SVM-RBF), k-Nearest Neighbor (kNN), decision tree. We fine-tune the parameters of each model. The results are shown in Table 8. It observed that VOID achieves the best accuracy of 98.81% when selecting SVM-RBF, which is the same as the paper [3]. These results prove VOID is effective in detecting spoofing samples on ARRAYID dataset. However, it is observed that the performance of ARRAYID is better than that of VOID under every classifier model. Besides, when applying these schemes on the third-party ReMasc Core dataset [18], the performance of ARRAYID (i.e., the accuracy of 97.78%) is still better than that of VOID (i.e., the accuracy of 84.37%). In summary, compared with the mono channel-based scheme, exploiting multi-channel features achieves superior performance in the liveness detection task.

Figure 16: Microphone array: Matrix Creator and Seeed ReSpeaker core V2.

Figure 17: LPCC in each channel.

(17) Call the nearest computer shop.
(18) Show me my messages.
(19) Translate please give me directions to Chinese.
(20) How do you say good night in Japanese?
Table 6: Loudspeaker used for generating spoofing attacks.

<table>
<thead>
<tr>
<th>No.</th>
<th>Type</th>
<th>Manufacture</th>
<th>Model</th>
<th>Size (L<em>W</em>H in cm)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Loudspeaker</td>
<td>Bose</td>
<td>SoundLink Mini</td>
<td>5.6 x 18.0 x 5.1</td>
</tr>
<tr>
<td>2</td>
<td>Tablet</td>
<td>Apple</td>
<td>iPad 6</td>
<td>24.0 x 16.9 x 0.7</td>
</tr>
<tr>
<td>3</td>
<td>Tablet</td>
<td>Apple</td>
<td>iPad 9</td>
<td>24.0 x 16.9 x 0.7</td>
</tr>
<tr>
<td>4</td>
<td>Loudspeaker</td>
<td>GGMM</td>
<td>Ture 360</td>
<td>17.5 x 10.9 x 10.9</td>
</tr>
<tr>
<td>5</td>
<td>Smartphone</td>
<td>Apple</td>
<td>iPhone 8 Plus</td>
<td>15.8 x 7.8 x 0.7</td>
</tr>
<tr>
<td>6</td>
<td>Smartphone</td>
<td>Apple</td>
<td>iPhone 8</td>
<td>13.8 x 6.7 x 0.7</td>
</tr>
<tr>
<td>7</td>
<td>Smartphone</td>
<td>Apple</td>
<td>iPhone 6s</td>
<td>13.8 x 6.7 x 0.7</td>
</tr>
<tr>
<td>8</td>
<td>Smartphone</td>
<td>Xiaomi</td>
<td>MIX2</td>
<td>15.2 x 7.6 x 0.8</td>
</tr>
<tr>
<td>9</td>
<td>Loudspeaker</td>
<td>Amazon</td>
<td>Echo Dot (2nd Gen)</td>
<td>8.4 x 3.2 x 8.4</td>
</tr>
<tr>
<td>10</td>
<td>Laptop</td>
<td>Apple</td>
<td>MacBook Pro (2017)</td>
<td>30.4 x 21.2 x 1.5</td>
</tr>
<tr>
<td>11</td>
<td>Loudspeaker</td>
<td>VicTsing</td>
<td>SoundHot</td>
<td>12.7 x 12.2 x 5.6</td>
</tr>
<tr>
<td>12</td>
<td>Loudspeaker</td>
<td>Ultimate Ears</td>
<td>Megaboom</td>
<td>8.3 x 8.3 x 22.6</td>
</tr>
<tr>
<td>13</td>
<td>Loudspeaker</td>
<td>Amazon</td>
<td>Echo Plus (1st Gen)</td>
<td>23.4 x 8.4 x 8.4</td>
</tr>
<tr>
<td>14</td>
<td>Smartphone</td>
<td>Xiaomi</td>
<td>Mi 9</td>
<td>15.8 x 7.5 x 0.8</td>
</tr>
</tbody>
</table>

Table 7: Detailed information of MALD dataset.

<table>
<thead>
<tr>
<th>User #</th>
<th># Authentic Samples</th>
<th># Spoofing Samples</th>
<th>Distance (cm)</th>
<th>Spoofing Devices</th>
</tr>
</thead>
<tbody>
<tr>
<td>1, 7</td>
<td>1200</td>
<td>3600</td>
<td>60,120,180</td>
<td>SoundLink Mini, iPad 6, iPhone 8 Plus</td>
</tr>
<tr>
<td>2</td>
<td>600</td>
<td>1079</td>
<td>60,120,180</td>
<td>Ture360, iPhone 6s</td>
</tr>
<tr>
<td>3</td>
<td>533</td>
<td>904</td>
<td>60, 120, 180</td>
<td>Ture360, iPad9</td>
</tr>
<tr>
<td>4–6, 8</td>
<td>2305</td>
<td>6415</td>
<td>60, 120, 180</td>
<td>iPad9, Ture360, MIX2</td>
</tr>
<tr>
<td>9–12</td>
<td>3211</td>
<td>3198</td>
<td>60, 120, 180, 240</td>
<td>Echo Plus (1st Gen)</td>
</tr>
<tr>
<td>13–18</td>
<td>1191</td>
<td>4577</td>
<td>180</td>
<td>iPad9, Mi 9, Echo Plus (1st Gen)</td>
</tr>
<tr>
<td>19</td>
<td>591</td>
<td>1767</td>
<td>60,120,180</td>
<td>iPhone 8, Echo Dot (2nd Gen), MacBook Pro (2017)</td>
</tr>
<tr>
<td>20</td>
<td>610</td>
<td>998</td>
<td>60, 120, 180</td>
<td>SoundHot, Megaboom</td>
</tr>
</tbody>
</table>

Table 8: Liveness detection performance under different classification models on the MALD dataset.

<table>
<thead>
<tr>
<th>Classifier type</th>
<th>Accuracy / EER (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Accuracy / EER (%)</td>
</tr>
<tr>
<td>Neural network</td>
<td>99.84 / 0.17</td>
</tr>
<tr>
<td>SVM-RBF</td>
<td>99.48 / 1.07</td>
</tr>
<tr>
<td>kNN</td>
<td>99.62 / 0.48</td>
</tr>
<tr>
<td>Decision tree</td>
<td>96.35 / 5.97</td>
</tr>
<tr>
<td></td>
<td>94.84 / 7.34</td>
</tr>
</tbody>
</table>

D Details of Modulated Attacks

When re-implementing the modulated attack and calculating the detection accuracy of ARRAYID, we choose three spoofing devices #3, #13 and #14 (i.e., iPad 9, Mi phone 9, and Amazon Echo Plus) as spoofing devices and Respeaker microphone array as the target device. To calculate the inverse filter for each device, we follow the steps described in the modulated attacks [48]. The frequency responses and their inverse filters of three spoofing devices are shown in Figure 18.

Then, after applying calculated inverse filters into the audios collected by the target device, we generate 1,990, 1,791, and 1,994 modulated attack samples for Echo, iPad, and Mi respectively.