Voiceprint Mimicry Attack Towards Speaker Verification System in Smart Home

Lei Zhang*, Yan Meng*, Jiahao Yu*, Chong Xiang*, Brandon Falk*, Haojin Zhu*†
* *Shanghai Jiao Tong University*
†*Shanghai Institute for Advanced Communication and Data Science*
{zhanglei1949, yan_meng, yujiahao, danco2015, bfalk95, zhu-hj}@sjtu.edu.cn

***Abstract*—The advancement of voice controllable systems (VCSes) has dramatically affected our daily lifestyle and catalyzed the smart home's deployment. Currently, most VCSes exploit automatic speaker verification (ASV) to prevent various voice attacks (*e.g.*, replay attack). In this study, we present VMask, a novel and practical voiceprint mimicry attack that could fool ASV in smart home and inject the malicious voice command disguised as a legitimate user. The key observation behind VMask is that the deep learning models utilized by ASV are vulnerable to the subtle perturbations in the voice input space. To generate these subtle perturbations, VMask leverages the idea of adversarial examples. Then by adding the subtle perturbations to the recordings from an arbitrary speaker, VMask can mislead the ASV into classifying the crafted speech samples, which mirror the former speaker for human, as the targeted victim. Moreover, psychoacoustic masking is employed to manipulate the adversarial perturbations under human perception threshold, thus making victim unaware of ongoing attacks. We validate the effectiveness of VMask by performing comprehensive experiments on both grey box (VGGVox) and black box (Microsoft Azure Speaker Verification) ASVs. Additionally, a real-world case study on Apple HomeKit proves the VMask's practicability on smart home platforms.**

***Index Terms*—speaker verification, adversarial examples, smart home, voiceprint mimicry**

I. Introduction

With the widespread deployment of smart home environment, our daily lives are becoming more convenient and intelligent through various home appliances (*e.g.*, heaters, doors, windows) functioning automatically [1], [2]. Among the diverse user interfaces (*e.g.*, the image, the voice, motion sensors) provided by the smart home, the voice interface plays a key role to facilitate the users to have the control of the smart devices and services without physical interaction. Currently, voice interfaces are widely integrated in most popular smart home platforms (*e.g.*, Apple Homekit [3], Amazon Alexa [4], Microsoft Cortana [5]) and the market revenue of these voice controllable systems (VCSes) is predicted to achieve $31.8 billion by 2025 according to Grand View Research's report [6].

Despite the convenience brought by the voice interface, it also faces an ever increasing threat of security and privacy [7]. For example, replay attack can fool the VCS via replaying the pre-collected legitimate user's voice samples [8]. Researchers also tried to exploit the non-linearity of the microphone circuit and the defect of deep learning algorithms to propose ultrasonic based [9], [10] or adversarial example based [11], [12] attacks respectively. In these attacks, the malicious audio samples are imperceptible and could be even injected in audio played by commodity devices.

To thwart these attacks, automatic speaker verification (ASV) or voiceprint recognition techniques are widely adopted by popular VCSes for user authentication. With ASV, voice can be used as a unique biometric signature to reflect a person's identity. As per "Fundamentals of Biometric Technology" published by the United States National Biosignature Test Center, voiceprint is a type of biometric signature allowing for ease of use, high accuracy, and low cost [13]. The industry has widely accepted ASV as an important bio-identification technology that extracts phonetic features from the speaker's voice signals to validate the speaker's identity. For example, Apple devices require "Hey Siri" as the activation command before any actions are taken. Wechat[1] and Alipay[2] have also supported voiceprint as an important alternative solution for user authentication.

While there exists research that are working on exploring the vulnerabilities in speech recognition components of VCS [10], [11], [14], less attention has been given to the security of ASV. If the adversary can mimic the voiceprint of the victim, he can impersonate the victim to log into his account and perform the subsequent attacks such as malicious bank transfer [15] and ordering [10], posing a great threat on the legitimate user's security. However, to launch a practical attack towards ASV, it faces the following research challenges. First, the model setup, including the architecture establishment and the parameters selection of most ASVs are kept private and the system works in a black box manner. Second, in practice, many ASVs add random challenges in authentication process, which makes the replay attack or synthesizing attack less practical in smart home environment. Therefore, how to generate an arbitrary voice command which could mimic the voiceprint of the target user remains a big challenge.

In this paper, we present the first practical attack, coined as VMask, towards ASV systems in smart home environment. The basic goal of VMask is allowing a source speaker (or attacker) to mimic the voice of the target speaker (or victim). By adding some carefully crafted adversarial perturbations to one benign speech sample from the source speaker, VMask is

[1] https://blog.wechat.com/2015/05/21/voiceprint-the-new-wechat-password/
[2] https://www.alipay.com/

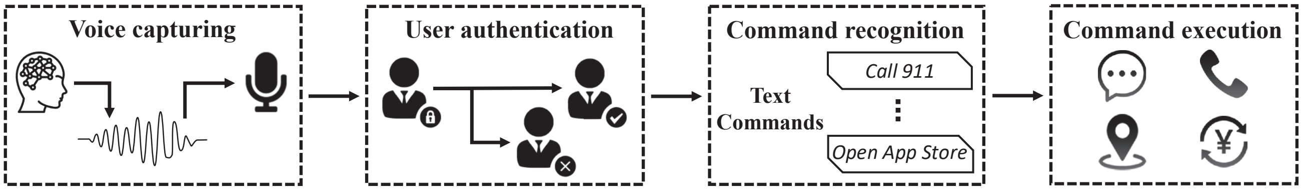


Fig. 1: The architecture of typical voice interface of smart home.

able to generate attack audios that still sound like the source speaker but would be recognized as the target speaker by the ASV. To launch such a successful voiceprint attack, the adversary only needs a few recordings from the target speaker containing voiceprint from arbitrary content.

To overcome the black box property of popular ASVs, we first conduct a tentative attack towards a grey box system. Based on the success of grey box attack, we then exploit the transferability of adversarial examples and launch black box attack against real-world ASV systems. To improve the practicability of VMask, psychoacoustic masking is deployed during the adversarial audio generation to keep the adversarial perturbations under the human perception threshold and avoid incurring human's suspicion. The evaluation results show that VMask can successfully breach the popular ASVs including grey box VGGVox [16] and black box Microsoft Azure Speaker Recognition API [17]. A real-world case study on Apple HomeKit also proves the effectiveness of VMask in smart home environment. In summary, the contributions of this paper are listed as follows.

- We present VMask, an adversarial example based voiceprint mimicry attack in smart home environment. Different from the previous works, VMask could be conducted using commercial-off-the-shelf devices without cumbersome data pre-collection and is practical in smart home platforms.
- We propose a novel adversarial audio generation method to fool the target ASV in which the neural network setup is totally unknown to the adversary. Specifically, we propose two different adversarial audio generation mechanisms targeting for grey box and black box ASV systems respectively.
- We implement VMask on popular ASVs such as VGGVox and Microsoft Azure Speaker Verification (MS-ASV). The evaluation results show that VMask can achieve success rates of near 100% and 70% in grey box and black-box scenarios respectively.
- We perform a case study on the Apple HomeKit, a popular smart home platform. We enable our VMask to attack the Siri speaker verification system, and the experimental results validate the robustness of VMask in real-world environment.

To the best of our knowledge, this is the first work to perform real-world adversarial attacks towards speaker verification on smart home platform. The remainder of this paper is organized as follows. Section II provides some preliminary knowledge; In Sec. III, we illustrate our threat model and attack assumptions; Sec. IV and V present the formulation of both grey box attack and black box attack. The evaluation results are given in Sec. VI, followed by a case study on Apple HomeKit in Sec.VII. Discussion and related work are presented in Sec.VIII and IX respectively. Finally we conclude this study in Sec. VIII.

II. Preliminaries

In this section, we introduce the prerequisite knowledge for this paper.

A. Voice Interface of Smart Home

The voice interface has become the primary user interface along with the widespread deployment of smart home. As illustrated in Fig. 1, a typical VCS works in four phases: *voice capturing*, *user authentication*, *command recognition* and *command execution*. First, in *voice capturing*, the user's speech samples are recorded and preprocessed. After that, the *user authentication* is conducted based on the voice biometrics extracted from processed speech samples. The *command recognition* translates the speech into text only if the *user authentication* is successful. Finally, VCS executes actions according to the recognized commands (*e.g.*, "open the door"). Usually, the *user authentication* and *command recognition* parts are separated (*e.g.*, Apple Siri and Amazon Alexa require the user to say the activation words "Hey Siri" and "Alexa" respectively before initiating any voice commands).

B. Automatic Speaker Verification Techniques

As shown in Fig. 1, the *user authentication* phase plays a key role in VCS to secure sensitive operations such as texting and financial transaction. Recently, automatic speaker verification (ASV) technique is widely adopted by popular VCSes, as it can verify the speaker's identity only using the speaker's speech samples. Most of ASVs require an utterance of a predefined phrase (*e.g.*, "Hey Siri"), and the authentication would succeed only in the case of both voiceprint match and audio content match.

ASV works in three phases: *developing*, *enrollment* and *verification*. In *developing* phase, large-scale corpora are utilized to train a background model shaping the speaker manifold. Then, in the *enrollment* phase, new speakers are enrolled by deriving speaker specific information from the enrollment utterances with the help of background model. Finally, the user's speech samples are taken as the input of *verification* phase, and the similarity between the features of input and

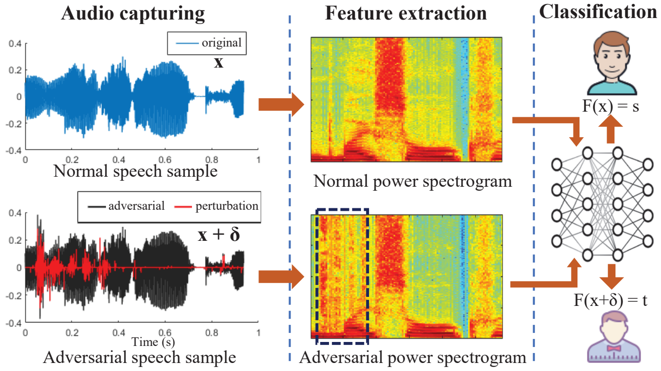


Fig. 2: An illustration of adversarial examples in ASV.

legitimate user's speech samples are calculated. The verification is passed only if the similarity score is larger than the pre-defined threshold [18].

The recent progress in deep learning has inspired the development of neural network based ASVs [19], [20]. They are also known as speaker embedding systems, as they extract embedding vectors from the speaker's utterances. With the help of these embedding systems, speaker verification can be done by measuring the distance of embedding vectors. In this paper, we only study the deep learning based ASV since deep learning has become the prominent trend in ASV developments [16], [19]. However, we show our attack is not restricted to deep learning based models in Sec. VI by presenting VMask against black box models without knowing the model architecture.

C. Adversarial Examples

Adversarial examples are carefully crafted data records which are similar to the original ones but can lead to the misclassification of machine learning model [21], [22]. As shown in Fig. 2, given a machine learning based classifier F which predicts the class label $F(x)$ for an input instance x, the adversarial example generation algorithm tries to find a perturbation δ which satisfy : (1) x is similar to $x + \delta$ in a given distance calculation criterion (*e.g.*, two audio clips sound similar by human ears); (2) $F(x + \delta) \neq F(x)$. In the case of ASV, the attacker's goal is to manipulate the verification results using their speech sample x together with the subtle adversarial perturbation δ.

The fundamental cause of adversarial examples lies in the neural network's sensitivity to the perturbations in the input space. In this paper, we formulate the generation of adversarial audios as an optimization problem, and the optimal permutation δ could be obtained by using gradient based method [23]. As shown in Sec. IV and Sec. V, we design two different adversarial example generation schemes for popular grey box and black box ASVs.

III. Threat Model

In this section, we give a picture of VMask's working scenario and elaborate the capability of the adversary.

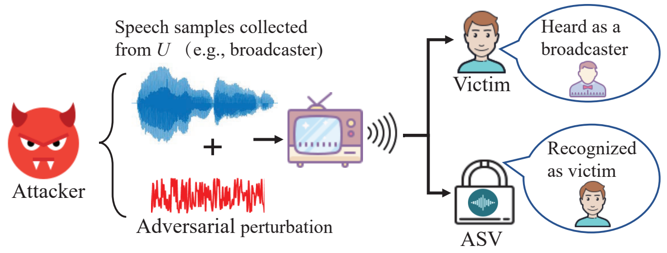


Fig. 3: A typical attack scenario of VMask.

A. Attack Scenario

Fig. 3 illustrates a typical attack scenario of VMask. There is a smart home platform which belongs to the victim, and the ASV authenticates the input voice command by checking both the semantic content and voiceprint. U denotes a speaker whose voice would extract no suspicion from the victim. For example, U can be a broadcaster the victim familiar with. To fool the ASV without raising victim's suspicion (*i.e.*, the replay attack cannot be deployed), VMask takes three steps. First, to generate the audio with semantic context required by ASV, VMask concatenates speech segments which are pre-collected from U. Second, from the victim's voice segment obtained in public media, VMask extracts the victim's voiceprint and crafts the adversarial subtle perturbations, and then adds the perturbations to the audios concatenated from U. Finally, the generated audio is embedded on a video clip and played using a loudspeaker close to the victim. The victim would be unaware of this ongoing attack, since this adversarial audio sounds like speaker U in the video being watched without associating themselves to the voice.

B. Adversary's Capability in This Study

In this study, we assume VMask has zero knowledge of the neural network model (*e.g.*, architecture, parameters and training data) used for ASV. We also assume VMask is unable to make any modifications on commodity speakers and ASV's microphone. However, the VMask can hijack the loudspeaker placed in close vicinity to the target ASV to conduct attack.

We assume two different access schemes to the victim ASV models: grey box and black box. The ***grey box*** ASV returns the verification result (accept or reject) along with the numerical confidence value, while the ***black box*** scheme only returns the verification result [24]. Note that both grey box and black box reveal no model setup information to VMask.

IV. Grey Box Attack

As a prologue before delving into black box attack, we first demonstrate the feasibility of our attack on grey box ASVs. Since a grey box ASV returns both the verification result and the confidence score, the intuition of our grey box attack is to estimate the gradient based on the difference of similarity scores from multiple queries, and then dynamically update the generated adversarial perturbations. Specifically, we utilize zeroth order optimization [25] to uplift the matching score while maintaining the audio content unchanged simultaneously. A visual illustration of our grey box attack is shown in Fig. 4.

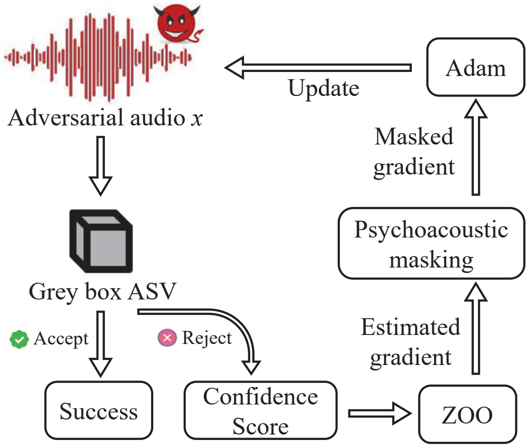


Fig. 4: An illustration of our grey box attack.

A. Attack Formulation

A grey box ASV system V is composed of two parts: $V(x,P,t) = V_p(x,P) \cdot V_i(x,t)$, where x is the audio to be verified, P is the semantic content required by ASV and t is the claimed identity (*i.e.*, the victim's identity in attack scenario). V_p and V_i can be defined as

$$\begin{aligned} V_p(x,P) &= \begin{cases} 1, & \text{if } Trans(x) = P \\ 0, & \text{otherwise} \end{cases} \\ V_i(x,t) &= \begin{cases} 1, & \text{if } f(x,t) \geq TH \\ 0, & \text{otherwise} \end{cases} \\ f(x,t) &= \textit{cos-sim}(F(x), E_t) \end{aligned} \tag{1}$$

where $Trans(x)$ and $F(x)$ are the semantic content and embedding extracted from x respectively, E_t is the victim speaker embedding, $\textit{cos-sim}(\cdot,\cdot)$ calculates the cosine similarity, and TH is the pre-defined threshold. $V_p(x,P)$ checks whether the transcription of x matches pre-defined phrase P while $V_i(x,t)$ checks whether the voiceprint of x matches the target speaker t. Then, once provided an audio clip $x \in (-1,1)^n$ uttered by the source speaker s, the adversary seeks a minimum perturbation Δx, subjecting to the following constraints:

1) x is verified as an utterance from victim speaker t : $V_i(x,t) = 1$, i.e. $f(x,t) \geq TH$.
2) The semantic content of generated audio $x+\Delta x$ remains unchanged: $V_p(x+\Delta x, P) = 1 = V_p(x,P)$.
3) The perturbation Δx is too subtle for human to notice.

We formulate the above problem as an optimization problem.

$$\begin{aligned} \min \quad & h(x+\Delta x, t) + C \cdot d(x, x+\Delta x) \\ s.t. \quad & V_i(x+\Delta x, t) = 1 \\ & V_p(x+\Delta x, P) = 1 \\ & x + \Delta x \in (-1,1)^n \end{aligned} \tag{2}$$

where $d(x, x+\Delta x)$ measures the distance between x and $x+\Delta x$, $h(x+\Delta x, t)$ evaluates the impacts of our attack on $f(x,t)$, and C is the adjusting factor balancing between $d(\cdot,\cdot)$ and $h(\cdot,\cdot)$. In this study, $h(\cdot)$ is defined as:

$$h(x+\Delta x, t) = \max\{\log f(x,t) - \log f(x+\Delta x, t), -\tau\} \tag{3}$$

where $\tau > 0$ is a constant to specify the upper bound of the optimization, as the loss function in Eqn. 2 converges when $\log f(x+\Delta x, t) - \log f(x,t) > \tau$. For the distance metric $d(x, x+\Delta x)$, we choose L_2 distance metric for $d(\cdot,\cdot)$.

Note that $V_p(x+\Delta x, P) = 1$ is not included in our objective function to improve the efficiency of our attack. Instead, we check the audio transcription after obtaining the optimal adversarial audio. This simplification is based on our observation as mentioned in Sec. VI that audio perturbations generated with a proper L_2 constraint is unlikely to change the audio content.

B. Zeroth Order Optimization (ZOO)

VMask utilizes zeroth order optimization [25] to solve the above problem. Formally, we estimate the partial derivative $\frac{\partial h(x,t)}{\partial x_i}$ with the technique of symmetric difference quotient through two queries : $h(x+\epsilon z_i, t)$ and $h(x-\epsilon z_i, t)$

$$\widehat{\frac{\partial h(x,t)}{\partial x_i}} = \left\{ \frac{h(x+\epsilon z_i, t) - h(x-\epsilon z_i, t)}{2\epsilon} \right\} \tag{4}$$

where ϵ is a small constant (*e.g.*, $\epsilon = 0.0001$ in this study), and $z_i \in \{0,1\}^n$ is a unit vector with only $z_i[i] = 1$. With $2n$ queries to the victim ASV, we can compute the partial derivatives for all n coordinates. Based on the estimated gradient, we can perform various gradient based methods to minimize $h(x+\Delta x, t)$. However, $2n$ times queries for one batch update is too expensive and unpractical for the adversary, Therefore, a *weighted stochastic coordinate gradient descent* method is leveraged to reduce the cost in each update.

In *weighted stochastic coordinate gradient descent*, only some coordinates are updated in each step. Considering the remarkable influence of coordinates selection strategy on the optimization efficiency, we use a weighted sampling strategy rather than a random strategy to select more important coordinates. The weight vector is computed from STFT as follows.

$$W_{i\in[1,n]} = \begin{cases} \sum_{k\in K} STFT(\lfloor \frac{i}{w_1} \rfloor, k), & i \leq n - w_0 \\ \sum_{k\in K} STFT(T-1, k), & \text{otherwise} \end{cases} \tag{5}$$

where w_0 and w_1 denote the hop length and window length of FFT sliding window, $STFT(T,K)$ is the STFT of x. Applying L_2 normalization on W we obtain W^* as our sampling weight vector.

C. Psychoacoustic Masking

In the above formulation, L_2 distance is used as a regularizer to restrain the adversarial perturbations. In this study, we introduce psychoacoustic masking to improve this naive regularization. Psychoacoustics studies the relationship between sound and the hearing it caused [26]. By exploiting psychoacoustic model, VMask can compute the hearing threshold which indicates the masking threshold between different frequencies. Then hearing threshold is leveraged to restrain the adversarial perturbations under human perception threshold.

Specifically, we first derive a scaling factor [12] in each iteration, which is then multiplied to the back propagated gradient resulting the final gradient. The scaling factor functions like a mask to repress the frequencies with sound level over the threshold while allowing more perturbations on the frequencies with sound level lower than the threshold. We rewrite the gradient estimation with scaling matrix $S(x)$ introduced as:

$$\frac{\widehat{\partial h(x,t)}}{\partial x_i} = \lim_{\epsilon \to \vec{0}} \left(S(x) \cdot \frac{\Delta h(x,t)}{\Delta m(x)} \right) \cdot \frac{\Delta m(x)}{2\epsilon} \tag{6}$$

where $\Delta h(x,t) = h(x+\epsilon z_i, t) - h(x-\epsilon z_i, t)$, $\Delta m(x) = m(x+\epsilon z_i) - m(x-\epsilon z_i)$, and $m(x)$ is the power spectrogram matrix computed locally. Note that in order to match the shape of $S(x)$, we write $\frac{\Delta h(x,t)}{2\epsilon}$ into $\frac{\Delta h(x,t)}{\Delta m(x)} \frac{\Delta m(x)}{2\epsilon}$. We follow [12] to compute the scaling matrix as:

$$S(x) = \Phi^*(x, x_0) \cdot H^*(x_0) \tag{7}$$

where $H^*(x_0)$ is the normalized hearing threshold computed for the original audio signal x_0, and $\Phi^*(x, x_0)$ is a normalization of $\Phi(x, x_0)$ computed as the following equation:

$$\begin{aligned} \Phi(x, x_0) &= H(x_0) - D(x, x_0) + \lambda \\ &= H(x_0) - 20 \ \log_{10} \frac{|m(x_0) - m(x)|}{\max(|m(x_0)|)} + \lambda \end{aligned} \tag{8}$$

where λ is a constant added to allowing the noises cross the threshold slightly. We let $\lambda = 10$ in all of our experiments conducted. The full attack algorithm is presented in Algorithm 1, where $Adam$ denotes the Adam optimizer [27] with default parameters, R is the number of steps and B is the batch size.

V. Black box attack

Inspired by the recent success of adversarial attacks in image recognition [28], [29], we leverage the transferability of adversarial examples to attack black box ASV systems. The basic idea of our black box attack is that despite the difference of deep learning models used in ASV systems, they all project the high dimensional audio space into a similar low dimensional speaker space. As a result, by separately training a local deep learning based ASV, we may be able to imitate the victim black box system. The first step of our black box attack is to extract the victim speaker embedding out of several victim recordings with arbitrary content. Then starting from an audio containing the required audio content uttered by an arbitrary person, we carefully add noises to it under the guidance of the victim speaker embedding. Finally, we are able to generate attack audios containing both the required content and victim voiceprint. The whole procedure is visualized in Fig. 5.

Algorithm 1 Weighted batch stochastic coordinate gradient descent with Psychoacoustic masking

```
Input:
    Source audio $x_0 \in (-1,1)^n$;
    Target speaker $t$;
Output:
    The adversarial audio $x$;
$x \leftarrow x_0$;
$W^* \leftarrow CalcSamplingWeight(x_0)$;
$H \leftarrow CalcThreshold(x_0)$;
for i $\leftarrow 1$ to $R$ do
    $Coordinates\ C \leftarrow \{1, \cdots, m\}.sample(W^*, B)$;
    $D \leftarrow CalcSpecDiff(x, x_0)$
    $\Phi = H - D + \lambda$;
    $S \leftarrow \Phi^* \cdot H^*$;
    for j $\leftarrow 1$ to $B$ do
        $\Delta h_j \leftarrow f(x + \epsilon z_{C_j}, t) - f(x - \epsilon z_{C_j}, t)$
        $\Delta m_j \leftarrow m(x + \epsilon z_{C_j}) - m(x - \epsilon z_{C_j})$
        $\hat{g}_j \leftarrow S \cdot \frac{\Delta h_j}{\Delta m_j} \frac{\Delta m_j}{2\epsilon}$ ;
    end for
    $\Delta x \leftarrow Adam(\hat{g})$;
    $x \leftarrow x + \Delta x$
end for
return $x$
```

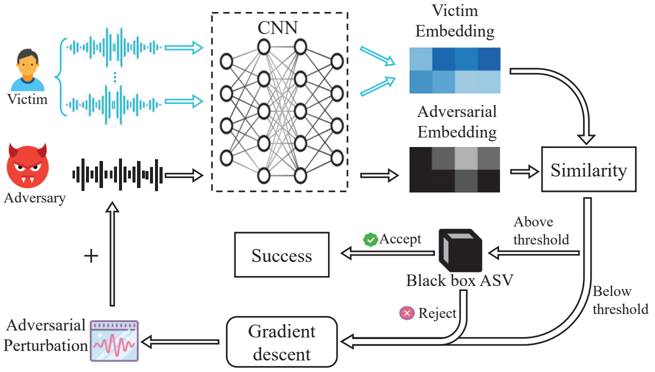


Fig. 5: A visualization of our black box attack.

A. Obtaining Victim Speaker Representation

We train a ResNet-style [30] speaker embedding system following Li *et al.* [20]. This CNN takes fixed-length audio as input. Temporal averaging is applied to the flattened output of the last Residual block to compute a utterance-level activation. Besides, triplet loss, softmax pre-training and hard-negative mining [20] are also exploited. With the trained speaker embedding system, we can obtain a 512-dimensional victim speaker model by averaging the embedding vectors extracted from victim recordings with arbitrary content.

B. Attack Audio Generation

A successful attack audio should contain both the required content and the victim voiceprint. To satisfy the first constraint, we start the attack from an arbitrary speaker's utterance for the verification phrase. We assume that with only subtle noise added, the content of the adversarial audio would change discreetly. To satisfy the second constraint, we first extract the victim embedding E_t from a few recordings with arbitrary content, and use E_t as a guideline to generate some subtle perturbations such that the embedding extracted from the adversarial audio is similar to the victim embedding E_t in the sense of some distance measure. Our formulation of black box attack is the same as Eqn. 2, except for a different $h(\cdot)$:

$$h(x + \Delta x, t) = 1 - cos\text{-}sim(F'(x + \Delta x), E_t), \qquad (9)$$

where $F'(x + \Delta x)$ denotes the embedding vector extracted for the adversarial audio and $cos\text{-}sim(\cdot, \cdot)$ denotes the cosine similarity between two vectors. Following [31], we use a L_2 distance metric in our work, so $d(x, x + \Delta x) = (\Delta x)^2$.

As common speech models take the acoustic feature as input, previous attack generates attack audios by reversing the modified MFCC features [14], which introduces a large overhead and information loss. We implement our attack in an end-to-end manner, allowing direct modifications on raw audio signal. Specifically, we implement a differentiable mel-spectrogram extraction layer in front of the former speaker embedding system. A loss layer is added after the speaker embedding extraction layer to calculate the distance between the adversarial embedding and victim speaker embedding. This end-to-end neural network allows us to apply backpropagation to approximate the optimal adversarial perturbations.

C. Psychoacoustic Masking

Similar to what we do in Sec.IV, psychoacoustic model is used to restrict adversarial perturbations under human hearing threshold. Different from grey box attack, we do not have to estimate the partial derivatives, as the local speaker embedding system remains a white box to us. So according to the chain rule, the gradient of the loss function with respect to the raw input can be calculated according to following equation:

$$\frac{\partial h(x,t)}{\partial x_i} = \left(S(x) \cdot \frac{\partial h(x,t)}{\partial m(x)} \right) \cdot \frac{\partial m(x)}{\partial x_i} \qquad (10)$$

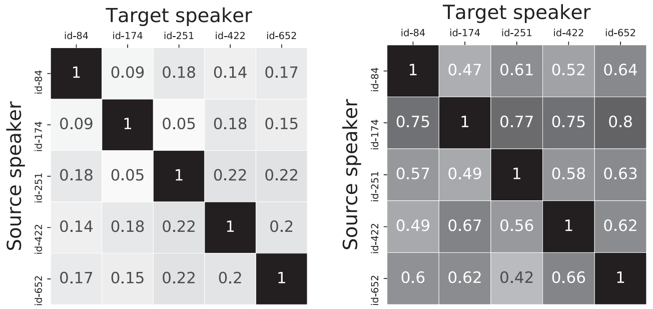
where $m(x)$ denotes the output of feature extraction layer, and $S(x)$ is derived from Eqn. 7. λ remains as a value of 10. With the masked gradient, we apply Adam [27] updating rule to craft adversarial perturbations in each iteration.

VI. Evaluation

In this section, we present the evaluation results of grey box attack and black box attack.

A. Evaluation Setup

NVIDIA GeForce GTX 1070 GPU and Intel i7-8700K CPU are used to generate the attack speech samples. To verify whether the attack audios preserve the desired audio content, we use Baidu speech recognition API [32] to transcribe the audios. We utilize a free text-independent speech corpus LibriSpeech [33] as our evaluation set. *train-clean-100* contains 28539 utterances from 251 speakers and is used for pre-training the local speaker embedding system for black box, *train-clean-360* contains 104014 utterances from 921 speakers and is used for fine-tuning, *dev-clean*, with 2703 utterances from 40 speakers, is used as the test set for local ASV system and source audios for generating adversarial audios for towards both grey box and black box ASVs.



(a) The matching scores of attack trials before adversarial manipulation.

(b) The matching scores of attack trials after adversarial manipulation.

Fig. 6: The columns represent the source speakers, the rows represent the target speakers. A darker color indicates a higher score.

B. Grey Box Attack

We evaluate our grey box attack against one of the state-of-the-art speaker embeddings system, VGGVox[3] which is developed on Voxceleb2 [16], a large-scale real-world corpus. The open sourced ASV system can achieves a best EER of 3.95% on Voxceleb2's test set according to the author.

We implement the our ZOO based attack in python, Matlab engine API[4] is used to access the grey box matlab model. For the parameter choice, C is set to 0.01, and τ is set to 2 to allow a substantial perturbation. Since VGGVox does not provide a threshold, we determine a threshold of 0.45 based on the evaluation results of VGGVox on *dev-clean*. As for the STFT computation, we utilize a FFT window with length 0.25s and hop length 0.01s.

Our attack trials are constructed from a test set consisting of 5 randomly selected speakers from *dev-clean*. For each speaker, we craft adversarial perturbations towards other 4 speakers resulting in 20 attack trials. For each attack trial, 500 iterations are performed, and in each iteration, we randomly select a batch of 352 coordinates out of 25840 audio sampling points, updating these coordinate with an Adam optimizer of default parameters having the learning rate set to a value of 0.01. A success is reported if the grey box returns an Accept. Finally, a success rate of 95% is achieved for the 20 trials.

Fig. 6 visualizes the matching score before and after adversarial perturbations in two confusion matrix. The value in each block represents the confidence score of the corresponding source speaker's utterance verified as the corresponding target

[3] https://github.com/a-nagrani/VGGVox

[4] https://ww2.mathworks.cn/help/matlab/matlab_external/install-the-matlab-engine-for-python.html?lang=en

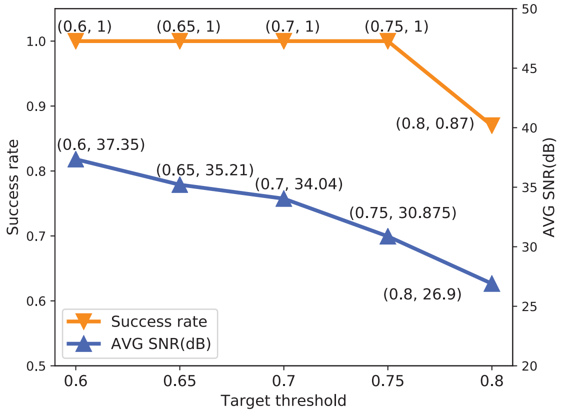


Fig. 7: The success rate and average SNR vary when we set different target thresholds.

speaker. The case that the source speaker and target speaker are the same person is not considered, thus all diagonal elements are all set as 1 in Fig. 6b. By comparing Fig. 6a with 6b, we can see that the confidence scores are substantially improved after subtle adversarial manipulation is applied, with an average improvement of 306%. Meanwhile, the adversarial perturbation is acceptable, as the average SNR is 13.13dB.

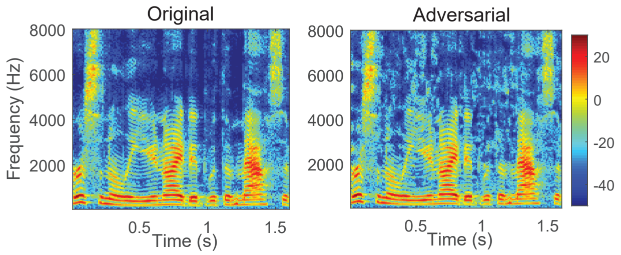
The transcription results are also checked to see whether the contents are preserved in the adversarial audios. To evaluate to what extent the contents are twisted, three metrics commonly used in speech recognition are chosen, namely WER (Word Error Rate), WRR (Word Recognition Rate), and SER (Sentance Error Rate). Evaluating on the 20 trials we obtain WER of 9.804%, WRR of 90.196% and SER of 31.250%, which means around 70% of the adversarial audios have exactly the same transcription as the original audios, and the probability of a word in the adversarial audio having the same transcription compared to the original audio is greater than 90%.

Moreover we evaluate the influence of λ on the perturbation. We try different λ values, 0, 10, 20 and 40 in our experiments and finally let $\lambda = 10$ in both grey box attack and black box attack to obtain a earlier converge with a relatively larger perturbation.

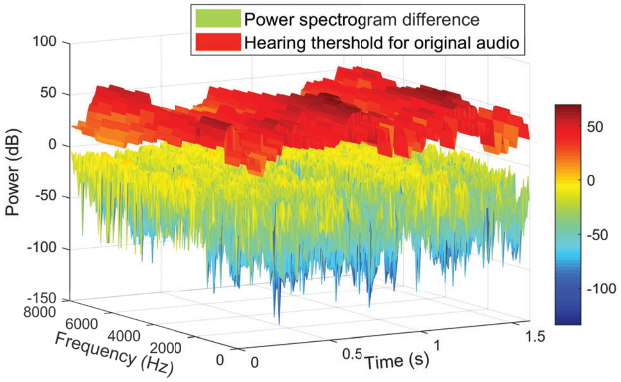
C. Black Box Attack

1) Speaker embedding system: VAD (Voice Activity Detection) is applied to eliminate the non-vocal signal for speech samples. Each audio is then partitioned into fixed length audio clips for which mel-spectrogram feature vectors of shape (160, 64) are extracted by leveraging a python package *python speech feature*[5].

For the softmax pre-training, we can reach an 83.12% training accuracy after 10 epochs training with an Adam optimizer of a 0.001 learning rate. Then with 6000 steps of fine-tuning with triplet loss, we can achieve a verification accuracy of around 99.2% and an EER (equal error rate) of around 3.5% on the test set. In the following experiments, we utilize this speaker embedding system for speaker model extraction.



(a) A comparison between the original audio and the adversarial audio in power spectrogram.



(b) A comparison between the power spectrogram difference and the human hearing threshold.

Fig. 8: An Illustration of psychoacoustic model.

2) Crafting adversarial examples for local ASV system: We implement our attack on Tensorflow [34] platform, and totally 400 attack trials are constructed. Each trial contains two different speakers randomly sampled from *dev-clean*, one as the source speaker and the other as the victim speaker. Before launching the attack, each victim speaker's embedding is extracted from 5 randomly selected utterances. An Adam optimizer with a 0.01 learning rate is applied to minimize the loss. Observing an insignificant decrease of loss after 500 iterations, we limit the number of iterations to 500 in the following experiments.

The threshold is varied to observe the performance of our attack under different settings. As shown in Fig. 7, we can achieve a 100% success rate when the target threshold is set to 0.6∼0.75. Even for a high threshold 0.8, we can still achieve a success rate of 87%. Meanwhile the average SNR decreases as the pre-defined threshold increases, which indicates that larger perturbation is needed to make the score cross a higher threshold. However, the average SNR of attack audios is always larger than 26dB, which means the perturbation is too subtle to raise victim's suspicion.

For a better understanding of adversarial perturbation and psychoacousic masking, we visualize the power spectrogram of one original audio and the corresponding adversarial audio in Fig. 8a, while the difference of spectrogram and hearing threshold are shown in Fig. 8b. We can see that the adver-

[5] https://pypi.org/project/python_speech_features/

sarial perturbation is well-controlled and is below the human perception threshold.

The comparison between the transcription of original audios and adversarial audios is shown in Table I, In the worst case, our attack is still able to achieve a WER of 20.757%, and a SRR of 46.250%. We observe a sharp increase of WER and a sharp drop of SRR when move the target threshold from 0.75 to 0.8. This may be attributed to the distance metric which acts like a L_2 regularizer in the iteration.

TABLE I: Audio transcription checking.

Threshold	WER	WRR	SRR
0.6	10.359%	91.231%	74.5%
0.65	11.633%	88.980%	73.0%
0.7	13.026%	89.077%	72.0%
0.75	14.694%	87.143%	68.0%
0.8	20.757%	83.920%	46.250%

3) Transferring attack: Now we aim to show the effectiveness of VMask towards a black box ASV systems, Microsoft Azure Speaker Verification API (MS-ASV) [17]. For enrollment, the user has to choose one phrase from Table II, and repeat it for three times. Then for verification, the user has to utter the same phrase, with only the verification result (Accept/Reject) and a confidence value (Normal/High/Very High) returned.

We build up a real-world dataset containing utterances from 4 speakers. For each speaker, their speaker model is extracted from 3 speech samples of arbitrary content. We also collect 3 speech samples for each of the 10 verification phrases for each speaker, which are used as enrollments on MS-ASV and also as the source audio in our attack.

For each phrase, we start with one speaker and craft adversarial audios against the other 3 speakers. At the end, we evaluate the crafted audios on the black box MS-ASV API. Success is reported for each phrase if we can bypass MS-ASV API at least once. As shown in Table II, our attack reaches a 70% phrase-level success rate in fooling MS-ASV API, which demonstrate the effectiveness of VMask in black-box settings. It worth noticing that we can't break 3 phrases. We attribute the failure to two possible reasons: (1) MS-ASV API may use a different model architecture (even an i-vector based model). (2) the adversarial noise may be corrupted by the preprocessing techniques of the API.

VII. Case Study in Smart Home

A. System Setup

To demonstrate the practicability of VMask, we perform a real-world case study of voice impersonation on Apple HomeKit [3], a popular smart home platform. In Apple HomeKit environment, Siri serves as the voice interface and provides the speaker verification function. As shown in Fig. 9, the home appliance (Aqara smart LED bulb) is connected with the Siri via an Aqara smart hub which is compatible with HomeKit architecture allowing the user to turn on the bulb by saying "Hey Siri, turn on the light". In this experiment, we recruited 5 volunteers. First, the speaker model of each volunteer is built from utterances containing arbitrary content. Then in each round, one of volunteers is chosen as the victim. With the victim enrolled to Siri, we ask the other 4 volunteers to say "Hey Siri, turn on the light" for six times and use these samples as source audios in black box attack. The adversarial perturbations are added to the source audios as mentioned in Sec. V. The loudspeaker (*i.e.*, Amazon Echo) is used to play the adversarial audio and the bulb will become illuminated if the attack succeeds. We continued to select different victim volunteers and conduct this experiment on different five types of Apple devices (*i.e.*, iPhone SE, iPhone 7 plus, iPhone 8, iPhone X, iPad 6) to ensure the generality of VMask.

TABLE II: Black box attack results against MS-ASV API.

No.	Phrase content	Results
1	I am going to make him an offer he cannot refuse	✓
2	Houston we have had a problem	✓
3	My voice is my passport verify me	✓
4	Apple juice tastes funny after toothpaste	✓
5	You can get in without your password	✓
6	You can activate security system now	✓
7	My voice is stronger than passwords	×
8	My password is not your business	×
9	My name is unknown to you	×
10	Be yourself everyone else is already taken	✓

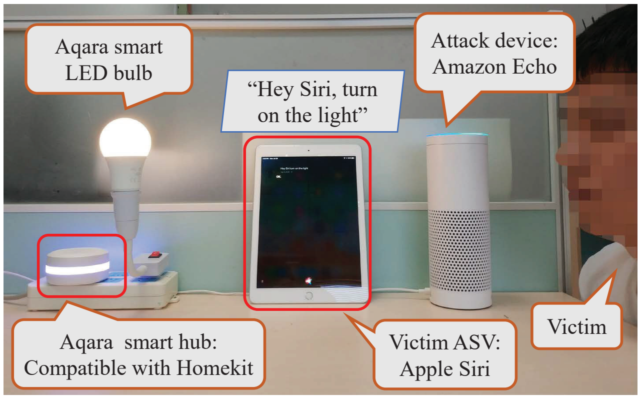
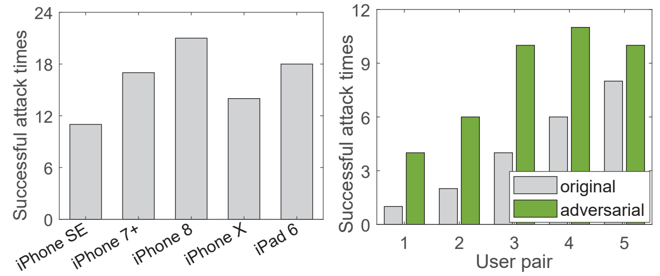

Aqara smart LED bulb
"Hey Siri, turn on the light"
Attack device: Amazon Echo
Aqara smart hub: Compatible with Homekit
Victim ASV: Apple Siri
Victim

Fig. 9: Case study testbed.

B. Experimental Results

The attack capability of VMask is first evaluated. Five Apple devices are assigned to five volunteers one by one. Then, for each volunteer holding the victim device, we play adversarial audios from other volunteers as described in Sec. VII-A. For each victim device, it suffers from attacks from VMask for $6 \times 4 = 24$ times, and the successful attack times are shown in Fig. 10a. It is observed that the average successful attack rate achieves $81/120 = 67.5\%$, meaning that all Siri in test devices are vulnerable to our VMask. We notice that the success rates among different Apple devices are quite different. Since Siri is an online ASV system, the reason behind this phenomenon



(a) The experiments on different Apple devices.

(b) Comparison between original and adversarial audios.

Fig. 10: Case study Results.

can be attributed to the difference of hardware condition and device holder's voice profile.

The high successful attack rates in Fig. 10a are caused by the fact that Siri opts for a relatively low verification threshold to guarantee user experience. Existing research shows Siri's capability of differentiating speakers is not ideal [35]. Therefore, to prove the effectiveness of adversarial audios, we utilize another volunteer's original and adversarial audios to attack Siri for 12 times respectively and then conducted these experiments among the 5 volunteer pairs on iPad 6 devices. As shown in Fig. 10b, after adding adversarial examples to original examples, the successful attack times (rates) raise from 20 (33.3%) to 41 (68.3%), which demonstrates VMask's attack capability. It's worth noticing that even for the 5th user pair of whom the voice profiles are similar, by adding adversarial noises, the attack success times (rate) also raised from 8 (66.7%) to 10 (83.3%). This further demonstrates the practicality of VMask in real-world smart home environment.

VIII. Discussion

A. Countermeasures

To prevent ASV from VMask's attack, an intuitive strategy is training a detector to distinguish adversarial audios from benign ones. However, a large amount of attack audios are needed. Moreover, this strategy may raise ASV's false alarm which reduces the user experience. Another possible defense mechanism may focus on destructing adversarial perturbations, down-sampling and noise reduction methods may be utilized. However, down-sampling and noise reduction may cause the degradation of speech recognition since they remove the legitimate user's speech information. Liveness detection like iris liveness detection can also be adapted as a defense method.

B. Limitations and Future Work

Although VMask achieves a good attack performance and reveals security vulnerabilities of popular ASVs, there still exists limitations in our study. The main limitation is the lack of modeling noise in real-world attack environment, which reduces the success rate of VMask. Building the noise model is difficult, because it needs to consider multiple factors (*e.g.*, circumstance, audio hardware) which are changed dramatically. Besides, in current stage, the perturbation generation speed in grey box attack (*i.e.*, 3 seconds to do one batch update) still needs to be improved. To address this issue, designing a algorithm better than *weighted stochastic gradient descent* is a feasible solution. In the current stage of our study, VMasks performance various on different sentences. Since the architecture and parameters of black-box ASVs are unknown to attacker, this is still an open issue. We leave these issues to future research.

IX. Related Work

Traditional attacks aiming at voice interface. Traditional voice interface attacks mainly focus on fooling automatic speech recognition (ASR), while little effort has been pushed to attack the ASVs. Carlini *et al.* [14] proposed a scheme to attack a HMM based ASR system. In this attack, the generated audios are heard as noises by human, but can be translated to malicious commands by ASR. Zhang *et al.* [10] leverage the hardware drawbacks of microphones to launch ultrasonic based attack which is inaudible to human. However, these attacks cannot manipulate victim's voiceprint, and applying them on ASV needs to pre-collected victim's speech samples which is unpractical in smart home environment.

Adversarial examples based attacks on voice interface. The success of adversarial examples in fooling image recognition models [36], [37] have inspired the researcher to explore the feasibility of applying adversarial examples on deep learning based ASR and ASV. Carlini *et al.* [31] utilize subtle perturbations to fool Deep Speech. Yuan *et al.* [11] propose a practical adversarial attacks against ASR by injecting malicious perturbation into songs. Furthermore, psychoacoustic models are applied to optimize the perturbation for attacks against DNN-based ASR system [12], [38]. However, these target ASV systems in these attacks are designed towards *white-box* and cannot be deployed in proprietary *black-box* ASVs. Kreuk *et al.* [39] utilizes transferability of adversarial examples to launch attack on a *black-box* ASV. However, this attack needs to know the model's basic architecture in advance and cannot be deployed in the ASVs studied in this paper.

X. Conclusion

In this paper, we propose VMask, a novel and practical voiceprint attack aiming at ASV in smart home. To mislead ASV's classification model, VMask enables arbitrary speech samples to have victim's voiceprint by adding carefully crafted noises. VMask is practical because the added noises are too subtle to raise the victim's suspicion. We propose adversarial audio generation algorithms for both grey box and black box ASVs, and implement VMask on both VGGVox and Microsoft Azure platforms. Finally, we conduct attacks on Apple HomeKit platform, and the experiment results demonstrate the feasibility of VMask in real-world circumstances.

Acknowledgement

This work is supported by the National Key Research and Development Program of China (2018YFE0216000) and the National Natural Science Foundation of China (No.61972453, No.61672350). The corresponding author is Haojin Zhu.

References

[1] W. Zhang, Y. Meng, Y. Liu, X. Zhang, Y. Zhang, and H. Zhu, "Homonit: Monitoring smart home apps from encrypted traffic," in *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*. New York, NY, USA: Association for Computing Machinery, Oct 2018, p. 10741088.

[2] M. Li, Y. Meng, J. Liu, H. Zhu, X. Liang, Y. Liu, and N. Ruan, "When csi meets public wifi: Inferring your mobile phone password via wifi signals," in *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*. New York, NY, USA: Association for Computing Machinery, Oct 2016, p. 10681079.

[3] Apple. (2019) Homekit - apple developer. [Online]. Available: https://developer.apple.com/homekit/

[4] Amazon. (2019) Alexa. [Online]. Available: https://www.alexa.com/

[5] Microsoft. (2019) Cortana home assistant. [Online]. Available: https://www.microsoft.com/en-us/cortana

[6] G. V. Research. (2018) Voice and speech recognition market size, share and trends analysis report. [Online]. Available: https://www.grandviewresearch.com/press-release/global-voice-recognition-industry

[7] Y. Meng, Z. Wang, W. Zhang, P. Wu, H. Zhu, X. Liang, and Y. Liu, "Wivo: Enhancing the security of voice control system via wireless signal in iot environment," in *Proceedings of the Eighteenth ACM International Symposium on Mobile Ad Hoc Networking and Computing*. New York, NY, USA: Association for Computing Machinery, June 2018, pp. 81–90.

[8] W. Diao, X. Liu, Z. Zhou, and K. Zhang, "Your voice assistant is mine: How to abuse speakers to steal information and control your phone," in *Proceedings of the 4th ACM Workshop on Security and Privacy in Smartphones & Mobile Devices (SPSM)*, Scottsdale, Arizona, USA, Nov. 2014, pp. 63–74.

[9] N. Roy, H. Hassanieh, and R. Roy Choudhury, "Backdoor: Making microphones hear inaudible sounds," in *Proceedings of the 15th ACM Annual International Conference on Mobile Systems, Applications, and Services (MobiSys)*, Niagara Falls, New York, USA, Jun. 2017, pp. 2–14.

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

[11] X. Yuan, Y. Chen, Y. Zhao, Y. Long, X. Liu, K. Chen, S. Zhang, H. Huang, X. Wang, and C. A. Gunter, "Commandersong: A systematic approach for practical adversarial voice recognition," in *27th USENIX Security Symposium (USENIX Security 18)*, Baltimore, MD, Aug. 2018, pp. 49–64.

[12] L. Schönherr, K. Kohls, S. Zeiler, T. Holz, and D. Kolossa, "Adversarial attacks against automatic speech recognition systems via psychoacoustic hiding," in *26th Annual Network and Distributed System Security Symposium, NDSS*, San Diego, California, USA, Feb. 2019, pp. 1–15.

[13] J. L. Wayman, "Fundamentals of biometric authentication technologies," *International Journal of Image and Graphics*, vol. 1, no. 01, pp. 93–113, 2001.

[14] N. Carlini, P. Mishra, T. Vaidya, Y. Zhang, M. Sherr, C. Shields, D. Wagner, and W. Zhou, "Hidden voice commands." in *Proceedings of USENIX Security Symposium (USENIX Security)*, Austin, TX, USA, Aug. 2016, pp. 513–530.

[15] Say-tec. (2018) Say tech. [Online]. Available: https://www.say-tec.com/

[16] J. S. Chung, A. Nagrani, and A. Zisserman, "Voxceleb2: Deep speaker recognition," in *Interspeech 2018, 19th Annual Conference of the International Speech Communication Association, 2-6 September 2018.*, Hyderabad, India, Sep. 2018, pp. 1086–1090.

[17] Microsoft. (2019) Speaker recognition api — microsoft azure. [Online]. Available: https://azure.microsoft.com/en-us/services/cognitive-services/speaker-recognition/

[18] G. Heigold, I. Moreno, S. Bengio, and N. Shazeer, "End-to-end text-dependent speaker verification," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Shanghai, China, Mar. 2016, pp. 5115–5119.

[19] D. Snyder, D. Garcia-Romero, G. Sell, D. Povey, and S. Khudanpur, "X-vectors: Robust dnn embeddings for speaker recognition," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Calgary, Alberta, Canada, Apr. 2018, pp. 5329–5333.

[20] C. Li, X. Ma, B. Jiang, X. Li, X. Zhang, X. Liu, Y. Cao, A. Kannan, and Z. Zhu, "Deep speaker: an end-to-end neural speaker embedding system," *arXiv*, vol. abs/1705.02304, pp. 1–8, May 2017.

[21] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. J. Goodfellow, and R. Fergus, "Intriguing properties of neural networks," in *2nd International Conference on Learning Representations, ICLR 2014, , Conference Track Proceedings*, Banff, AB, Canada, Apr 2014, pp. 1–10.

[22] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," in *3rd International Conference on Learning Representations, ICLR*, San Diego, CA, USA, May 2015, pp. 1–11.

[23] N. Carlini and D. Wagner, "Towards evaluating the robustness of neural networks," in *2017 IEEE Symposium on Security and Privacy (SP)*, San Jose, CA, May 2017, pp. 39–57.

[24] S. Fnu, J. Chi, D. Evans, and Y. Tian, "Hybrid batch attacks: Finding black-box adversarial examples with limited queries," in *29th USENIX Security Symposium*, Boston, MA, USA, Aug. 2020.

[25] P.-Y. Chen, H. Zhang, Y. Sharma, J. Yi, and C.-J. Hsieh, "Zoo: Zeroth order optimization based black-box attacks to deep neural networks without training substitute models," in *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security*, Dallas, Texas, USA, Nov. 2017, pp. 15–26.

[26] E. Zwicker and H. Fastl, *Psychoacoustics: Facts and models*. Berlin/Heidelberg, Germany: Springer Science & Business Media, 2013.

[27] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *3rd International Conference on Learning Representations, ICLR 2015, Conference Track Proceedings*, San Diego, CA, USA, May 2015, pp. 1–15.

[28] Y. Liu, X. Chen, C. Liu, and D. Song, "Delving into transferable adversarial examples and black-box attacks," in *5th International Conference on Learning Representations, ICLR 2017, Conference Track Proceedings*, Toulon, France, Apr 2017, pp. 1–14.

[29] N. Carlini and D. Wagner, "Adversarial examples are not easily detected: Bypassing ten detection methods," in *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security*, Dallas, Texas, USA, Nov. 2017, pp. 3–14.

[30] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition, (CVPR)*, Las Vegas, NV, USA, June 2016, pp. 770–778.

[31] N. Carlini and D. Wagner, "Audio adversarial examples: Targeted attacks on speech-to-text," in *2018 IEEE Security and Privacy Workshops (SPW)*, Regency, San Francisco, CA, May 2018, pp. 1–7.

[32] Baidu. (2019) Baidu ai speech recognition. [Online]. Available: https://ai.baidu.com/tech/speech/asr

[33] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: An asr corpus based on public domain audio books," in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, South Brisbane, Queensland, Australia, Apr. 2015, pp. 5206–5210.

[34] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard *et al.*, "Tensorflow: A system for large-scale machine learning," in *12th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 16)*, SAVANNAH, GA, USA, Nov 2016, pp. 265–283.

[35] Apple. (2019) Hey siri: An on-device dnn-powered voice trigger for apples personal assistant. [Online]. Available: https://machinelearning.apple.com/2017/10/01/hey-siri.html

[36] K. Eykholt, I. Evtimov, E. Fernandes, B. Li, A. Rahmati, C. Xiao, A. Prakash, T. Kohno, and D. Song, "Robust physical-world attacks on deep learning visual classification," in *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018*, Salt Lake City, UT, USA, June 2018, pp. 1625–1634.

[37] T. B. Brown, D. Mané, A. Roy, M. Abadi, and J. Gilmer, "Adversarial patch," *CoRR*, vol. abs/1712.09665, pp. 1–6, Dec. 2017.

[38] H. Abdullah, W. Garcia, C. Peeters, P. Traynor, K. R. B. Butler, and J. Wilson, "Practical hidden voice attacks against speech and speaker recognition systems," in *26th Annual Network and Distributed System Security Symposium, NDSS*, San Diego, California, USA, Feb 2019, pp. 1–15.

[39] F. Kreuk, Y. Adi, M. Cisse, and J. Keshet, "Fooling end-to-end speaker verification with adversarial examples," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Calgary, Alberta, Canada, April 2018, pp. 1962–1966.