# GhostTalk: Interactive Attack on Smartphone Voice System Through Power Line

Yuanda Wang, Hanqing Guo, Qiben Yan
SEIT Lab, Computer Science & Engineering, Michigan State University
{wangy208, guohanqi, qyan}@msu.edu

***Abstract*—Inaudible voice command injection is one of the most threatening attacks towards voice assistants. Existing attacks aim at injecting the attack signals over the air, but they require the access to the authorized user's voice for activating the voice assistants. Moreover, the effectiveness of the attacks can be greatly deteriorated in a noisy environment. In this paper, we explore a new type of channel, the power line side-channel, to launch the inaudible voice command injection. By injecting the audio signals over the power line through a modified c harging cable, the attack becomes more resilient against various environmental factors and liveness detection models. Meanwhile, the smartphone audio output can be eavesdropped through the modified cable, enabling a highly-interactive attack.**

**To exploit the power line side-channel, we present *GhostTalk*, a new hidden voice attack that is capable of injecting and eavesdropping simultaneously. Via a quick modification o f the power bank cables, the attackers could launch interactive attacks by remotely making a phone call or capturing private information from the voice assistants. *GhostTalk* overcomes the challenge of bypassing the speaker verification system b y s tealthily triggering a switch component to simulate the press button on the headphone. In case when the smartphones are charged by an unaltered standard cable, we discover that it is possible to recover the audio signal from smartphone loudspeakers by monitoring the charging current on the power line. To demonstrate the feasibility, we design *GhostTalk-SC*, an adaptive eavesdropper system targeting smartphones charged in the public USB ports. To correctly recognize the private information in the audio, *GhostTalk-SC* carefully extracts audio spectra and integrates a neural network model to classify spoken digits in the speech.**

**We launch *GhostTalk* and *GhostTalk-SC* attacks towards 9 main-stream commodity smartphones. The experimental results prove that *GhostTalk* can inject unauthorized voice commands to different smartphones with 100% success rate, and the injected audios can fool human ears and multiple liveness detection models. Moreover, *GhostTalk-SC* achieves 92% accuracy on average for recognizing spoken digits on different smartphones, which makes it an easily-deployable but highly-effective attack that could infiltrate sensitive information such as passwords and verification codes. For defense, we provide countermeasure recommendations to defend against this new threat.**

Network and Distributed Systems Security (NDSS) Symposium 2022
24-28 April 2022, San Diego, CA, USA
ISBN 1-891562-74-6
https://dx.doi.org/10.14722/ndss.2022.24254
www.ndss-symposium.org

## I. Introduction

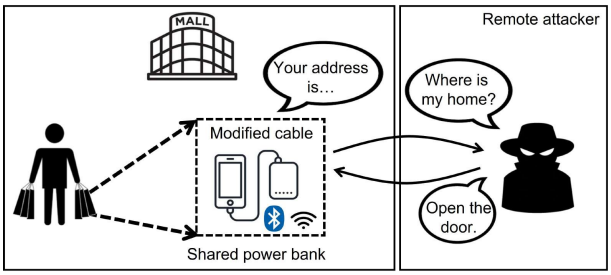
Smartphone has become an indispensable communication and entertainment tool in everyone's daily life. Many users nowadays spend a substantial amount of time on smartphone apps such as social networks, mobile games, and live streaming platforms. As the technology evolves from text messaging to image and video streaming, the energy consumption of smartphones increases dramatically, creating a pressing demand for large-capacity batteries and fast chargers. In recent years, public charger has become a popular utility for travellers in need of charging services, which has grown into a massive billion-dollar market [1].

Hundreds of millions of users have been using charging stations and power banks all over the world [2]. The mainstream charging stations can generally be classified into two different types: shared power bank and public charging port. A shared power bank usually offers different charging cables for different smartphones. The users typically scan a QR code using their smartphones before renting these power banks, such that they can pay the bill based on the usage time [3]. Meanwhile, the public charging port, e.g., a USB port, allows the users to charge their smartphones through the port. These public charging ports are widely deployed in public spaces, such as shopping malls, hotels, and airports [4].

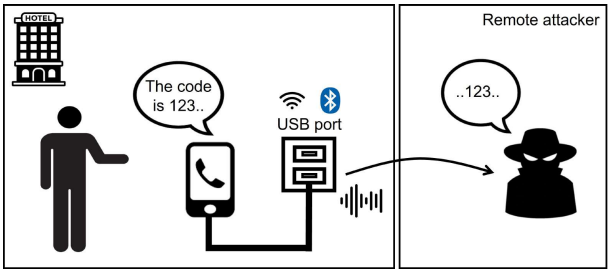
However, even though these charging stations bring convenience to the smartphone users, the ensued security threats have been rapidly escalating. For instance, security researchers have exposed numerous attacks that can sniff data transmission through the charging cable [5], or disclose sensitive app usage from the power consumption profiles [6]. A recent research demonstrates that, by monitoring the input voltage of the charger, an attacker can even recover the smartphone password [7].

From the attacker's point of view, these charging power sources, including the cables and power supply devices, can be modified [8], and this further exacerbates the threats. Moreover, the new generation of smartphones have discarded 3.5mm headphone jack and integrated the headphone audio functions into the charging port [9]. This innovation revamps the outlook of smartphones, but it imposes new threats to the smartphone audio system when users are charging in public spaces. In this research, we find that, after a quick modification of the charging cable, the attackers could have the capability to remotely compromise the smartphone and take control of its voice assistant.

A number of recent studies have demonstrated attacks




(a) In a shopping mall, the victim rents a hacked power bank to charge the smartphone. *GhostTalk* can remotely compromise the smartphone voice assistant through a modified charging cable.




(b) The victim answers a phone call when charging his/her phone on a public USB charging port. *GhostTalk-SC* eavesdrops private information by analyzing the charging power patterns.

Fig. 1: *GhostTalk* attacks the smartphones charged by the shared power banks, and *GhostTalk-SC* is able to spy private phone conversations when the phone is being charged by the public chargers.

that compromise voice assistants on smartphones. An early study shows that the attackers can compromise speaker verification systems and inject malicious commands into victim smartphones via a replay attack [10]. Yet, the replay of an audible voice command can be easily detected by the nearby victims. DolphinAttack [11] and SurfingAttack [12] both achieve inaudible voice command injection by leveraging the non-linearity of smartphone microphones. However, these existing inaudible voice command attacks cannot simultaneously achieve two attack goals, i.e., voice injection and eavesdropping.

In other words, the attackers have no way of accessing the responses from the voice assistants. As a result, they are generally incapable of measuring the attack outcome and realizing more complicated attacks, such as ghost phone calls or private information theft.

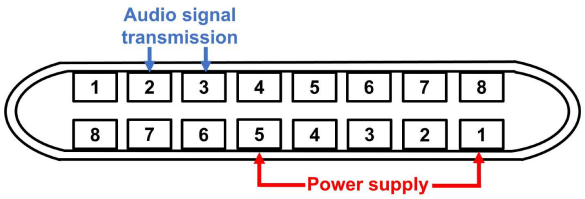
Moreover, the existing voice injection attacks are susceptible to the environmental noise. In order for the attack to succeed, the victim device should reside in a quiet environment and the attacker must stay close to it. Note that all these attacks suffer from a major shortcoming, i.e., they require the victim's voice to generate specific utterances, like "Hey Siri" or "Hello, Google", in order to activate the voice assistant. In case when the attacker has no access to the victim's voice, the attack could not be executed. Additionally, since these inaudible voice commands are usually transmitted by a loudspeaker, they could be effectively detected by the liveness detection modules [13].

To further extend the attack scenarios, we introduce *GhostTalk*, a new attack that attempts to compromise the smartphone voice assistants through a power line side-channel. By modifying the power bank charging cable and manipulating the electric signals in the modified cable, *GhostTalk successfully closes the gap between injection and eavesdropping*, i.e., it not only remotely injects malicious voice commands to the victim smartphone, but it also eavesdrops private information from the voice assistant. Notably, *GhostTalk* triggers a switch component to activate the *press* button operation that effectively activates the voice assistant without the requirement of an authorized speaker's voice. Fig. 1(a) illustrates an attack scenario of *GhostTalk*: when a user is charging his/her phone with a shared power bank, the attacker can remotely query the user's home address and then unlock the door by interacting with the smartphone's voice assistant. Compared with the existing work, *GhostTalk* is the first interactive attack that simultaneously achieves stealthy audio injection and eavesdropping, while remaining resilient against environmental noises and liveness detection systems.

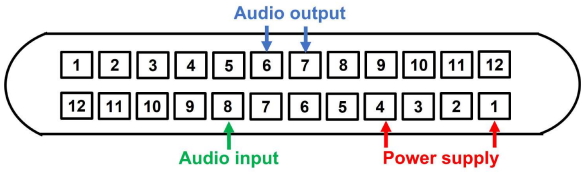
In another attack scenario when the users are charging their phones on the public charging ports, they typically insert their own standard charging cables. We experimentally observe that the power usage patterns of the phone's loudspeaker could be used as a side-channel for extracting private audio signals. Specifically, when the battery is over 95% charged, the attackers could extract the audio signal by passively monitoring the charging current. Based on this observation, we design *GhostTalk-SC* (i.e., *GhostTalk* with Standard Cable) to eavesdrop sensitive information from the smartphones charged by standard cables. During the attack, whenever the victim is playing audio through the smartphone loudspeaker, the adversary could recognize the leaked audio by measuring and analyzing the varying charging power. However, the background noise introduced by other smartphone applications has a substantial impact on the perceptibility of the captured audio. To overcome this challenge, *GhostTalk-SC* denoises the audio by signal processing and leverages deep neural networks (DNNs) to recognize sensitive digits within the conversation. A website is set up (**https://ghosttalkattack.github.io/**) to demonstrate the attacks.

In summary, this paper makes the following contributions:

- *GhostTalk* is the first interactive attack towards smartphone voice assistants over the charging cables. After slight modifications on the charging cable of the shared power banks, *GhostTalk* can achieve interactive attacks by inaudible audio injection and eavesdropping. In addition, *GhostTalk* attack requires no prior knowledge about the victim's voice and preserves the resilience against noisy environments and liveness detection models.
- We propose *GhostTalk-SC*, an eavesdropping attack that captures the audio signals from the power line side-channel. *GhostTalk-SC* can successfully extract audio signals from 8 out of 9 tested smartphones through standard charging cables, and can accurately recognize sensitive digit information using a DNN model.
- We evaluate *GhostTalk* and *GhostTalk-SC* attack performance with extensive real-world experiments using 9 popular commodity smartphones. The results prove




(a) Lightning port circuit.




(b) USB-C port circuit.

Fig. 2: Two mainstream charging port architectures of smartphones.

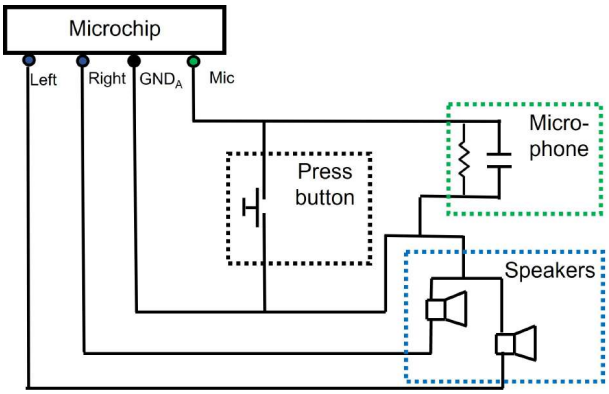



Fig. 3: A typical circuit of a wired headphone with microphone component and press button.

that *GhostTalk* achieves inaudible interactive voice injection and eavesdropping attacks on all the victim smartphones. Moreover, *GhostTalk-SC* can correctly classify more than 90% of spoken digits in the leaked audio when the smartphone plays the audio at its highest volume.

## II. Background

### A. Smartphone Charging Ports

Traditional charging ports on the smartphone possess two main functions: charging and data transmission. On the new generation of smartphones, the manufacturers are trending towards the complete removal of the headphone jacks, while supporting the audio signal transmission directly over the charging ports. Correspondingly, two mainstream charging ports, Lightning port and USB-C port, both support the audio transmission over the charging port.

Fig. 2(a) illustrates the circuit of Lightning charging ports equipped on iPhones. Generally, Lightning port can work under four modes: USB host, USB device, accessories, and power supply. The accessories mode supports the concurrent battery charging and audio transmission. Specifically, pin 2 and pin 3 transceive the audio signals, while pin 1 and pin 5 are responsible for charging the battery.

The USB-C port widely deployed in Android smartphones is shown in Fig. 2(b). When a headphone is plugged in, pin 6 and pin 7 will send the audio signals to the headphone, and pin 8 will receive the input audio signal from the microphone. Meanwhile, pin 1 and pin 4 connect to the DC power for charging. Therefore, USB-C also simultaneously supports charging and audio signal transmission. Mainly due to the integrated and versatile features of these ports, the smartphones are threatened by the unauthorized audio injection and eavesdropping attacks as shown in this work.

### B. Headphone Circuit

Fig. 3 displays the circuit of a typical wired headphone with Lightning or USB-C jack. In the headphone, 4 wires transceive audio signal from/to the smartphone: left speaker, right speaker, microphone (Mic) and audio ground (GND$_A$). When the headphone is playing audio, the smartphone outputs digital signals to the charging port, where the microchip digital to analog converter (DAC) converts them into analog voltage signals. After that, the voltage signals will trigger the change in the current of the headphone speaker coil. Such a changing current in turn stimulates the vibration of the speaker membranes to generate the audible sound wave.

Conversely, the sound waves cause membrane vibrations that modify the microphone capacity. As the voltage on the capacitor is constant, the changing capacity translates into the changing current, which produces the analog signals corresponding to the input audio. The microchip analog to digital converter (ADC) will then convert the analog audio signals into digital data, and transmit the data over to the smartphone.

Most of the smartphone headphones have a “press” button to allow smartphone operations such as making phone calls or controlling music players. When the button is clicked, the microphone and audio ground are shorted and the smartphone detects a current impulse from the microphone. It is noteworthy that the press button can also activate the voice assistants, and this function is exploited by *GhostTalk* to enable the hidden activation, as shown in Section V.

### C. Power Line Side-channel

Li-ion battery is widely deployed on smartphones. Generally, the charging process of a Li-ion battery can be divided into three stages [14]: (i) with a low battery status, the charger will offer a constant current to boost the battery voltage; (ii) during the charging process, the charging current adjusts to keep the charging voltage constant; (iii) once the battery is fully charged, the charging power is consumed to balance the smartphone power usage. At the final stage, the charging power is determined by both the smartphone hardware components and the running apps.

Recent work demonstrates that the charging power has a strong correlation with the smartphone apps when the battery state is over 95% [15]. As a result, the charging power pattern can reflect the smartphone’s working status, thereby opening up a side-channel for the attackers. For example, the attackers can fingerprint specific websites and apps by

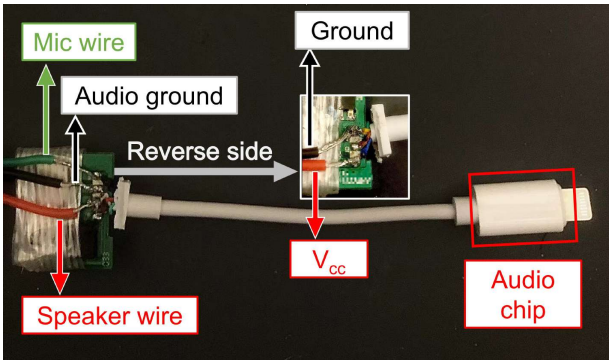



Fig. 4: The modified charging cable for *GhostTalk* attack.

recognizing different charging power patterns [6], [16], or even steal the lock-screen password by measuring the charging voltage fluctuation [7]. Our work develops new attacks to extract the audio signals from the power-line side channel.

## III. Attack Motivation

### A. Cable Modification

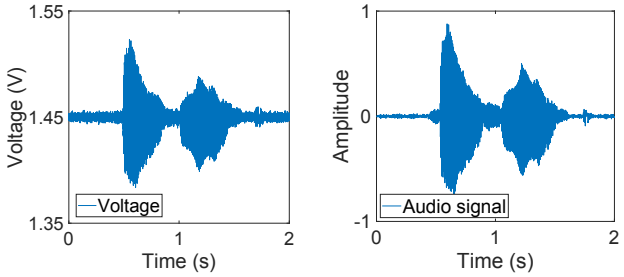
To implement the attack, the attacker has to modify the standard charging cables in order to support audio signal transmission. However, it will be extremely difficult for attackers to add audio functions in a standard cable. Fortunately, we can use the headphone adapter, whose cable allows concurrent audio signal transmission and charging, which is very popular on the market with a fair price (∼$10).

Fig. 4 shows a specially designed Lightning adapter cable that enables audio functions and charging. By integrating audio functions in the microchip, the cable can encode and decode audio signals. Two charging wires, as shown in the middle box of Fig. 4, charge the smartphone, while the four extra audio wires are used for audio signal transmission (see the left boxes in Fig. 4). Similar adapter cable exists for USB-C. The attackers can then replace the standard cables of the shared power bank with such specially designed cables and launch attacks towards the smartphones being charged.

### B. Inaudible Audio Injection through Charging Cable

As illustrated in Section II, the audio signals over the charging port are essentially represented by the changing current. Therefore, if the attackers can manipulate the current through the audio wires, they can inject the inaudible audio signals into the smartphone.

To verify the feasibility of audio signal injection through a charging cable, we add modulated voltage signals between the microphone and audio ground to change the current in the microphone. Specifically, we add an extra DC offset (∼1.45V) to the modulated signal, and apply it on the microphone wire. Then, the victim smartphone, i.e., iPhone X, records the injected audio signal from the Lightning port. Meanwhile, we use an ADC board to measure the input voltage signal. Fig. 5(a) shows the injected voltage waveform of a voice command "Hey Siri", and the resulting audio waveform is presented in Fig. 5(b). Apparently, the shapes of the voltage



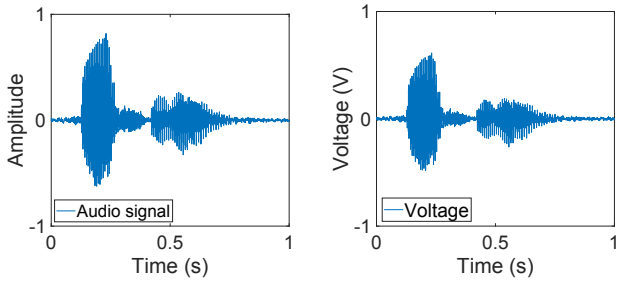

(a) The voltage measurement result between microphone wire and audio ground.

(b) Injected audio waveform recorded by the victim smartphone.

Fig. 5: The relationship between the voltage on microphone wire and the signal strength of the corresponding recorded audio.

and audio waveforms resemble each other. Hence, the results prove the feasibility of inaudible voice command injection by controlling the voltage input on the microphone wire of a charging cable. This phenomenon demonstrates the existence of a charging port backdoor that can be exploited to stealthily attack the voice assistants.

### C. Inaudible Audio Eavesdropping though Charging Cable




(a) The original audio waveform played by the victim smartphone.

(b) The voltage measurement result between speaker wire and audio ground.

Fig. 6: The relationship between the signal strength of the played audio and the voltage on the speaker wire.

Next, we evaluate the feasibility of eavesdropping by monitoring the voltage signal on the charging cable. First, we play a recorded word "password" on the same iPhone X, and monitor the voltage between the speaker and audio ground wires. The original audio waveform is shown in Fig. 6(a), while the measured voltage waveform is shown in Fig. 6(b). The voltage waveform almost perfectly resembles the audio waveform, which demonstrates that the audio signal can be accurately recovered by voltage measurement.

### D. Audio Eavesdropping through Standard Charging Cable

In case when the users plug their own standard charging cables in the public charging ports, the attackers could not access and modify these cables. However, we find that the charging power line side-channel could still leak the audio signal. This power side-channel may be caused by the high power profile of the loudspeaker, or the electromagnetic (EM)

field from the loudspeaker that alters the charging current due to the close proximity of the loudspeaker and charging port components. We design four experiments to find the root cause of this power line side-channel. Particularly, we measure the charging current of a fully-charged iPhone X via a shunt resistor. First, the phone plays a chirp audio (0∼2 kHz) through the left audio channel, originating from the bottom loudspeaker. Second, the phone plays the same chirp audio through the right audio channel, originating from the top loudspeaker. Third, the smartphone idles after playing the audio. Finally, we take additional measurement when the smartphone is off.

Fig. 7 presents the charging current spectrogram under different experiments. The results in ① and ② show that the signal strengths of charging current are exactly the same regardless of the positions of loudspeakers (i.e., top or bottom). Therefore, the power-line side channel is unlikely induced by EM interference, which varies notably across different positions. Fig. 7 also shows that the frequency of the charging current signal (0∼4 kHz) doubles that of the audio signal (0∼2 kHz). In fact, when the loudspeaker is playing a $k$ Hz audio signal, its power consumption $P_l$ can be expressed as:

$$P_l = I_l^2 R = (a\cos(2\pi kt))^2 R = \frac{a^2 R}{2}(1+\cos(4\pi kt)), \quad (1)$$

where $a$ is a constant and $R$ is the resistance of the loudspeaker. Eq. (1) illustrates that the frequency of power usage doubles that of the audio signal, which perfectly matches with our experimental result. Therefore, we can see that the audio signal patterns in the charging current is brought by the high power profile of the loudspeaker, which far exceeds the idling charging power.

However, as shown in Fig. 8, since the smartphone firmware and apps also draw power, the leaked audios would be too noisy to be recognized by human ears. Therefore, we use a convolutional neural network (CNN) to recognize sensitive information in the speech audio as shown in Section V. Also, given that the noise level is associated with the smartphone hardware design, the signal strengths of leaked audios vary significantly for different smartphones. The details are presented in Section VI.

## IV. Threat Model

### A. Threat Model of GhostTalk

*GhostTalk* works when the modification of charging cables of a shared power bank is a viable option.

**Power Bank Modification.** Since everyone has access to the shared power banks, it is reasonable to assume that the attacker can replace the charging cables of the shared power bank with specially designed cables, and hide extra hardware in the power banks. The victim rents a (hacked) power bank to charge the smartphone in a public space such as airport, hotel, or shopping mall.

**No Owner Interaction.** We assume that the victim users do not keep using their smartphones, when the phones are being charged by the power banks. This is a common assumption taken by almost all the voice command injection attacks [11]. For example, it is quite normal for people to put their phones in a handbag along with power banks.

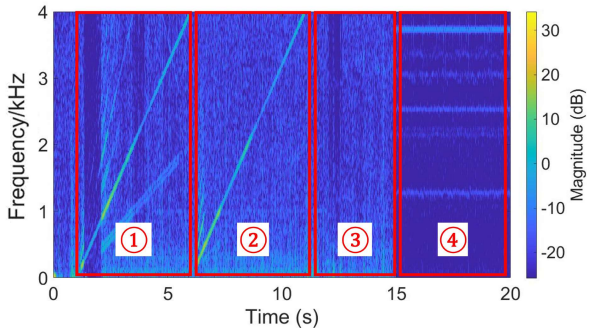



Fig. 7: When the smartphone plays a chirp audio signal via bottom speaker (①) and top speaker (②), the charging current spectrogram contains the chirp signal patterns. When the smartphone idles, the charging current spectrogram still carries the noise brought by the smartphone firmware and apps (③). The noise disappears once the smartphone is turned off (④).

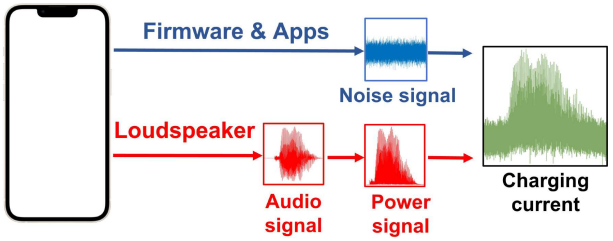


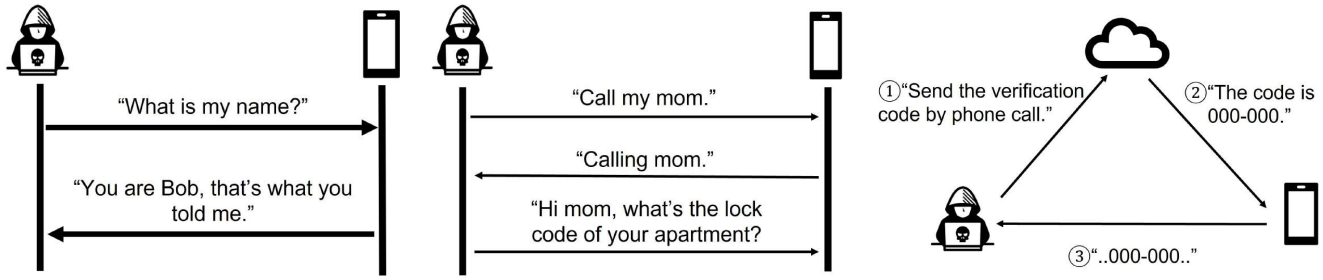

Fig. 8: Power side-channel leaks smartphone audio signals, but they are obfuscated by strong noise.

**Attack Scenarios.** For *GhostTalk* attack, the attacker does not need to be physically close to the victim device, as the attack device (i.e., modified power cable) contains a WiFi module. By connecting to public WiFi hotspots, the attacker could launch the attack remotely by sending/receiving audio signals from a remote site. Fig. 9 shows three specific attack scenarios of *GhostTalk*: (a) the attackers can query the voice assistant to steal private information, such as the user's name, home address, and phone number; (b) after the attackers retrieve the victim identity, they can collect or generate the victim's voice samples by crawling the social media or running speech synthesis, and search for the victim's family and job information on the Internet. Then, the attackers can launch a ghost phone call by injecting and eavesdropping voice signals as shown in Fig. 9(b); (c) the attackers can request a voice verification code to be sent to the smartphone, and stealthily eavesdrop it upon the reception. The verification code can be used to hack into the victim's social media or bank accounts.

### B. Threat Model of GhostTalk-SC

Although the *GhostTalk* attack brings a notable threat, it can only be implemented on smartphones being charged by modified power banks. To further extend the attack scenario, an alternative attack, *GhostTalk-SC*, works without the need of charging cable modification.

**Power Source Modification.** Recent work [17] shows the feasibility of hacking public USB charging ports by attaching




(a) Private information query. (b) Ghost phone call. (c) Hacking verification code.

Fig. 9: Specific attack scenarios of *GhostTalk*.

malicious hardware. For *GhostTalk-SC* attack, the attacker can also hide an ADC board in the USB ports of hotels and airports. The ADC board will keep monitoring the charging current and send the measurement results to the attacker.

**Victim's Behavior.** We assume the victim will not immediately stop charging after the battery state exceeds 95%. We also assume the victim will raise the loudspeaker volume when interacting with the phones in a hands-free mode, which is very common in our daily life.

**Attack Scenario.** If the victim keeps charging the smartphone after the battery state reaches 95%, the audio signal from the loudspeaker becomes extractable by the attacker. Despite the charging state assumption, the attack scenario is still quite realistic. As an example scenario, in a public space, a victim plugs the smartphone on a wall-mounted charging port. While charging, the victim retrieves verification codes or delivers private information such as credit card information and SSN number over a phone call. It is not unusual that the verification codes and conversations are played aloud by the smartphone speaker. *GhostTalk-SC* can then eavesdrop the voice verification codes or passwords by recognizing the charging current patterns.

## V. Attack System Design

### A. System Design of GhostTalk

Fig. 10 illustrates the system design of *GhostTalk*. After replacing the standard cables with specially designed cables described in Section III-A, the attacker is able to manipulate the voltage on the microphone wire and monitor the voltage on the speaker wire. The DC power in the power bank can charge the phone, and at the same time supply power for the hardware of *GhostTalk* system. We add two resistors $R_m$ and $R_s$ between the microphone, speaker, and $\text{GND}_A$ wires to emulate the existence of a headphone. The resistance of $R_m$ and $R_s$ are 2,000 and 20 ohms, respectively.

*1) Voice Assistant Activation:* The first challenge of the *GhostTalk* is to activate voice assistants without authorized user's voice. Previous inaudible voice command injection attacks [11], [12] require a collection of voice samples from the authorized user to generate specific commands such as "Hey Siri" or "Hello Google" to wake voice assistants. However, if there is a lack of access to available authorized voice samples, such attacks become infeasible.

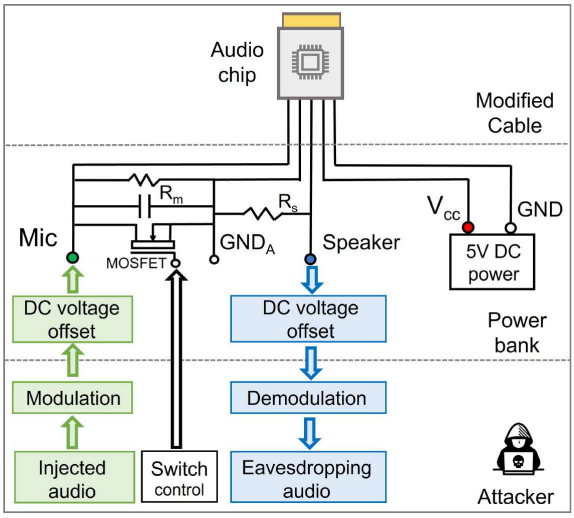
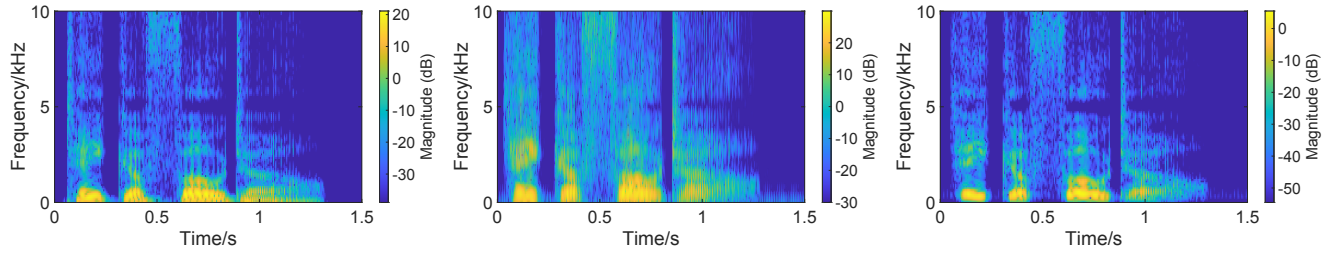



Fig. 10: The system design of *GhostTalk*.

To address this challenge, we simulate the headphone press button function by manipulating the voltage in the charging cable. Our idea comes from the following observation: when the user is using a wired headphone, the voice assistants can be activated by pressing the button even with a locked smartphone. Therefore, the attackers can leverage the *button pressing backdoor* to activate the voice assistant. To replicate the press button operation, we add a MOSFET between the microphone and $\text{GND}_A$ wires. Before injecting the malicious voice commands, the attacker activates the MOSFET to short the microphone and $\text{GND}_A$. This operation will let the phone mistakenly believe that the user is pressing the button, leading to the activation the voice assistant. Compared with other approaches on stealthy voice assistant activation, *GhostTalk* has two advantages: first, the *GhostTalk* attacker activates the voice assistant by electric signals, which are more resilient in noisy environments compared with voice command injection attacks; second, *GhostTalk* could bypass the speaker recognition system. After activating the voice assistant, the smartphone generally does not verify the speaker's voice again. Therefore, the attacker can command the voice assistant using any voice.

*2) Inaudible Audio Injection:* After activation, the attackers can inject inaudible voice commands to the smartphone. Eq. (2) illustrates the injected signal modulation process, where




(a) The original audio spectrogram. (b) The injected audio spectrogram (without using capacitor). (c) The injected audio spectrogram (with an additional capacitor).

Fig. 11: The capacitor in *GhostTalk* attack system can effectively reduce the noise in the injected audio.

$x_i(t)$ is the injected audio signal and $k$ is a factor to adjust the voltage range. Consider that the microphone capacitor has an initial voltage, we use an amplifier to add a DC offset $\Delta V_{in}$ ($\sim$ 1.5V) on the injected signal to compensate for the initial voltage of the microphone. The modulated voltage signal $V_i$ can be written as:

$$V_i(t) = kx_i(t) + \Delta V_{in}. \tag{2}$$

However, the direct injection of the modulated voltage will generate a noisy injected audio. Fig. 11(b) shows the spectrogram of an injected voice command "take a photo". Compared with the original audio spectrogram in Fig. 11(a), the injected audio has substantial background noise. Such noise could degrade the audio quality and allow the listeners to identify the injected audio.

Fortunately, in our experiments, we observe that the headphone microphone's capacitor is not only used for generating the changing current, but it also functions as a signal smoother that smooths discrete voltage signals. Therefore, we add an additional capacitor with similar capacitance to suppress the noise. Fig. 11(c) shows the injected audio spectrogram after adding the capacitor, which is almost indistinguishable with the original audio spectrogram.

*3) Inaudible Audio Eavesdropping:* When charged by the modified cable, the victim smartphone will play audio through a non-existent "headphone" rather than a loudspeaker. Therefore, the attackers are able to capture the audio signals without accessing the audible sound. Specifically, the attacker eavesdrops the audio signal by measuring the voltage of the speaker wire, as represented by blue boxes in Fig. 10. Note that the modified cable has two wires for left and right speakers respectively, and *GhostTalk* only needs to measure the voltage from one of them. Since the speaker wire carries analog signals, the attacker uses ADC to process and normalize the signals. Also, we use an amplifier to add an initial voltage offset $\Delta V_{out}$ ($\sim$ 1.5V) to obtain an absolutely positive voltage $V_o(t)$ for the audio input, since the attacker's ADC can only process the signals with positive voltage. Then, we can demodulate the audio signal $x_e(t)$ as follows:

$$x_e(t) = \frac{V_o(t) - \Delta V_{out}}{k}, \tag{3}$$

where $k = \max\{|V_{out} - \Delta V_{out}|\}$.

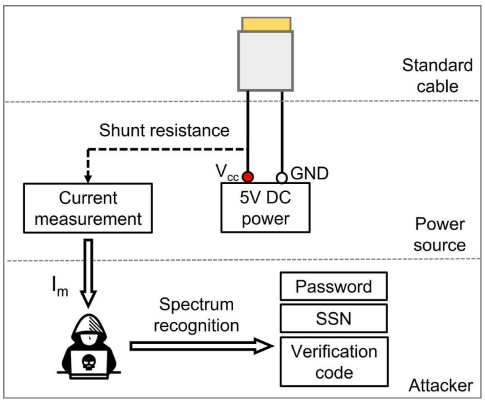



Fig. 12: The system design of *GhostTalk-SC*.

### B. System Design of GhostTalk-SC

In this case, the victims charge their phones using their own standard cables. By passively monitoring the charging current, the attackers can eavesdrop private information through the power line side-channel. Fig. 12 illustrates the system design of *GhostTalk-SC*. Compared with *GhostTalk*, *GhostTalk-SC* system only needs to measure the charging current in the standard cable. However, the demodulated audio only has a limited frequency band, which is distorted by strong background noise, making it incomprehensible for human ears. To resolve this challenge, we design a signal processing mechanism and apply a deep learning model to facilitate the recognition of the private information in the speech audio.

*1) Signal Processing:* After collecting the current measurement result $I_m(t)$, we use a high-pass filter to remove DC offset and low-frequency noise in the current signal, and recover a primitive audio $x_n(t)$. Following the technique in [18], we apply spectral-subtraction to enhance the speech audio signal in $x_n(t)$. First, we obtain $X_n(\omega)$, the frequency domain spectra of $x_n(t)$ by Fast Fourier Transformation (FFT). Meanwhile, by monitoring the idling smartphone charging current, we can estimate the signal strength of noise signal $N(\omega)$. Then, we denoise $X_n(\omega)$ by: $X_c(\omega) = X_n(\omega) - N(\omega)$, and transform the denoised frequency spectra $X_c(\omega)$ back to the time domain signal $x_c(t)$.

*2) Digit Classification:* Unfortunately, after removing the background noise, the recovered audio $x_c(t)$ is still unrecognizable for either human or AI models. This is because only low frequency components in the audio (lower than 2

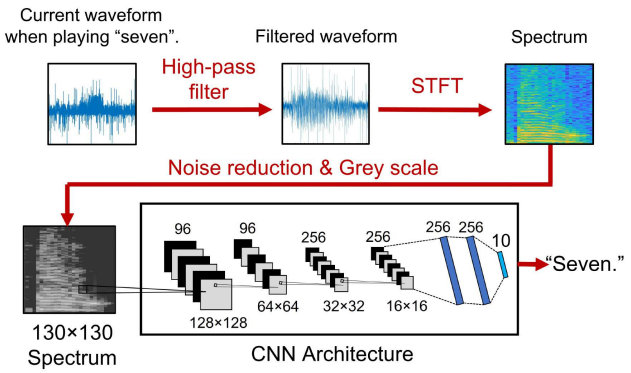



Fig. 13: The CNN architecture of *GhostTalk-SC*, which includes an input layer, two convolutional layers, two max-pooling layers, two dense layers, and an output layer.

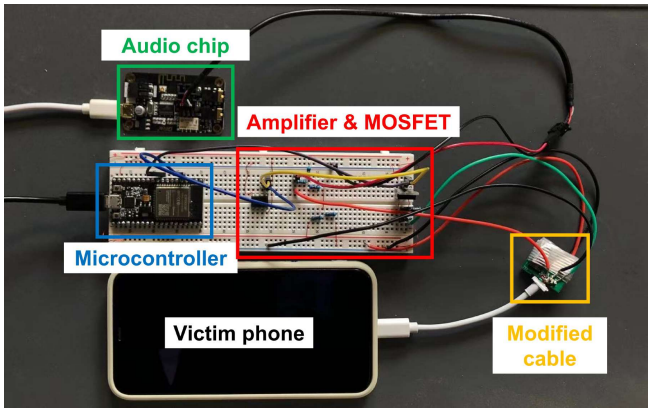



Fig. 14: Low-cost and portable experiment setup of *GhostTalk* attack. The hardware devices are small enough to be hidden in a power bank, and the modified cable has the same outlook as a standard cable.

kHz) have been recovered from the current signals due to the signal loss. Our intuition is that the deep learning models such as CNNs can extract more convoluted patterns from voice signals, which can help recognize the speech using the low-frequency audio. Similar to the existing attacks [19], *GhostTalk-SC* aims at realizing the digit recognition to extract sensitive information such as passwords, SSN numbers, and verification codes.

Fig. 13 shows the CNN architecture of *GhostTalk-SC*, which is used for classifying spoken digits from “zero” to “nine”. The input to the CNN is a $130 \times 130$ spectrogram matrix of denoised audio signals generated by Short-Time Fourier Transform (STFT). The CNN model consists of two convolutional layers with ReLU activation and two $2 \times 2$ max-pooling layers. Two dense layers with a dropout rate of 0.5 are used for improving the classification performance and preventing the overfitting [20]. Finally, a softmax layer outputs the probability distribution of ten digits. Using the trained model, the attacker can infer the spoken digits from the leaked speech audio. Compared with other speech recognition methods, the CNN architecture coherently learns from the time-domain and frequency-domain signals, which achieves high classification accuracy as shown in Section VI.

## VI. Evaluation

### A. GhostTalk Attack Evaluation

*1) Experiment Setup:* In the experiments, we evaluate the *GhostTalk* attack on 9 different smartphones from 5 mainstream manufacturers including Apple, Google, Samsung, Huawei, and Xiaomi. The experiment setup is shown in Fig. 14. An ESP-32 board with WiFi and Bluetooth modules is used to control the MOSFET and measure the voltage from the speaker wire. A Bluetooth audio chip injects the modulated audio commands to the victim phone, and an LM-358 dual-channel amplifier is used to apply the DC voltage offset.

*2) GhostTalk Injection Performance:* To evaluate the performance of *GhostTalk* injection attack, we use Google WaveNet API [21] to generate 20 voice commands, and each command contains $3 \sim 8$ words. After activating the voice assistant, we inject the voice commands to each victim smartphone, and repeat the experiment for 10 times. Then, we calculate the signal-to-noise ratio (SNR) of the voice command recordings.

The results are listed in Table I. For all the victim smartphones, the average injection audio SNR is higher than 15 dB, which can be clearly perceived by human ears [22]. Also, we notice that the injected audio SNR is related to the microphone sampling frequency $f_s$ of the victim smartphone. For example, for Samsung Note 10 ($f_s$ = 44.1 kHz), the injected recordings have higher average SNR value than Pixel 4XL ($f_s$ = 32.0 kHz). The attacker can further improve the injected audio volume by adjusting the amplification factor $k$ in Eq. (2). It is worth noting that a larger $k$ may degrade the performance of *GhostTalk* injection if the current in the microphone is beyond the smartphone sampling range. In our experiment, we set $k$ = 0.1 to balance the audio quality and SNR value.

Next, we repeat the experiment and test if these injected voice commands can be recognized by the voice assistants. we list the attack success rate (ASR) result of *GhostTalk* injection attack in the last column of Table I. Surprisingly, in spite of different hardware design and sampling frequency, all victim smartphones are vulnerable to *GhostTalk* injection attack. For all victim smartphones, *GhostTalk* injection attack can compromise their voice assistants with 100% success rate, which outperforms all state-of-the-art inaudible command injection attacks.

*3) GhostTalk Eavesdropping Performance:* To evaluate *GhostTalk* eavesdropping attack, we play 100 human speech samples from TIMIT dataset [23] with the maximum volume setting of the victim smartphone. Meanwhile, the ESP-32 board works as an ADC to measure the voltage output from the speaker cable with 10 kHz sampling frequency. For comparison, we play these speech samples with the loudest volume in a quiet environment (noise level $\leq$ 25 dB), and use an iPhone 8 to record the audio 30 cm away from the victim smartphone. Subsequently, we compare the eavesdropping performance of *GhostTalk* against a normal recording.

For normal recording, the smartphone loudspeaker's output power determines the SNR of eavesdropped audios. For *GhostTalk* attack, the recovered audio SNR is limited by the

| Num. | Manufacturer | Model | OS/Ver. | Assistants | $f_s$ (kHz) | *GhostTalk* | | | SNR (dB) | ASR |
|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | Act. | Inj. | Eav. | | |
| 1 | Apple | iPhone 5s | iOS 12.5 | Siri | 44.1 | ✓ | ✓ | ✓ | 19.7 | 100% |
| 2 | Apple | iPhone X | iOS 14.5 | Siri | 48.0 | ✓ | ✓ | ✓ | 21.3 | 100% |
| 3 | Huawei | Honor 10 | Android 9.0 | Google | 48.0 | ✓ | ✓ | ✓ | 20.4 | 100% |
| 4 | Xiaomi | MI 8 Lite | Android 9.0 | Google | 44.1 | ✓ | ✓ | ✓ | 18.9 | 100% |
| 5 | Xiaomi | Pocophone | Android 9.0 | Google | 48.0 | ✓ | ✓ | ✓ | 21.8 | 100% |
| 6 | Samsung | Note 10 | Android 10.0 | Google | 44.1 | ✓ | ✓ | ✓ | 21.2 | 100% |
| 7 | Samsung | Galaxy S9 | Android 10.0 | Google | 44.1 | ✓ | ✓ | ✓ | 20.1 | 100% |
| 8 | Google | Pixel 1 | Android 10.0 | Google | 44.1 | ✓ | ✓ | ✓ | 19.3 | 100% |
| 9 | Google | Pixel 4XL | Android 11.0 | Google | 32.0 | ✓ | ✓ | ✓ | 15.4 | 100% |

TABLE I: Experiment devices, operating systems, and microphone sampling frequencies. We test three components of *GhostTalk* attacks including voice assistant activation (Act.), inaudible voice command injection (Inj.), inaudible audio eavesdropping (Eav.). $f_s$: the sampling frequency of the smartphone microphone; SNR: Signal-to-Noise ratio of injected audio; ASR: injection attack success rate of *GhostTalk*.

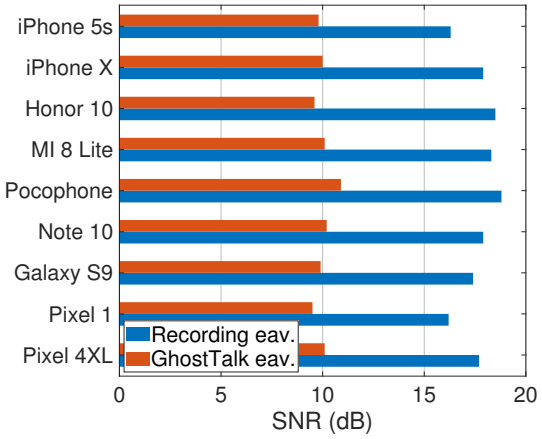



Fig. 15: SNR comparison of normal recording and *GhostTalk* eavesdropping.

voltage range of the speaker wire and the sampling frequency of ADC. Fig. 15 shows the audio SNR comparison of recording eavesdropping and *GhostTalk* eavesdropping. For all the victim smartphones, *GhostTalk* eavesdropping has lower average SNR values because of the low sampling frequency and restricted voltage amplitude. However, since most of human voice spectrogram is below 5 kHz [24], our eavesdropping attack can still recover clear human speech audios.

To further evaluate the eavesdropping audio quality and clarity, we use Google Speech-to-Text API [25] to recognize the speech contents in the eavesdropped audios. For both recording eavesdropping and *GhostTalk* eavesdropping, Google Speech-to-Text API can accurately recognize approximately 95% of words in the speech audios regardless of the SNR. The result demonstrates that *GhostTalk* could eavesdrop the audios through the power line, and attain the same audio clarity as an eavesdropping attack with normal audio recording in a quiet environment.

*4) Human Study:* As shown in Section IV, the attackers can launch ghost calls using *GhostTalk* attack, i.e., the attacker can initiate a phone call by injecting voice commands, and "speak" with the victim's voice. To deceive the human ears, the injected audios by *GhostTalk* are supposed to have the same quality as natural human speech. Therefore, to verify the feasibility of ghost call attack, we design a human study experiment to test if human ears can distinguish natural and injected human speech audios.

First, we use an iPhone X to record ten natural human speech samples from ten different speakers. Then, we launch the *GhostTalk* attack to inject these audio samples to the same smartphone, and obtain 10 corresponding injected human speech samples with the same speech contents. We then normalize all the benign and injected voice samples to eliminate the amplitude or length disparity.

In total, 20 volunteers (12 males and 8 females) participate in our human study. As a baseline, the volunteers are first requested to listen to two sets of voice samples as training examples. In each set, one sample is natural human speech, and the other one is an injected human speech sample from the circuit without the capacitor (see Fig. 11(b)). Since the injected samples present audible noise and high-frequency distortion, all the listeners can correctly recognize the injected samples. We also verbally explain to the volunteers that the injected samples may contain additional noise, and may be subject to frequency distortion in comparison with the natural audio samples.

Next, for each question set, we have one natural human speech sample and one injected sample from the *GhostTalk* injection (sample A & B). After listening to one set, the volunteers need to select whether the samples can be distinguishable or not. Then, the volunteers will select the likely injected sample.

In the end, we collected 200 answers. Table II summarizes the human study results, among which 150 answers depict the two samples as "indistinguishable", and the rest 50 answers

| | Type of answer | Number | Accuracy |
|---|---|---|---|
| Distinguishable | Random guess ✓ | 19 | 59.4% |
| | Random guess ✗ | 13 | |
| | Deterministic ✓ | 11 | 61.1% |
| | Deterministic ✗ | 7 | |
| Indistinguishable | N/A | 150 | N/A |
| Overall | N/A | 200 | 15% |

TABLE II: *GhostTalk* human study results. ✓and ✗ respectively indicate correct and wrong answers. Only 15% of answers correctly select the injected voice sample, and most samples are indistinguishable for the volunteers. In addition, the deterministic answers have a similar accuracy as the randomly guessed answers.

claim the opposite. Out of the 50 answers, only 30 of them successfully pinpoint the injected sample.

Along with the sample recognition study, we also conduct a survey of the volunteers. First, the volunteers are requested to mark their answers if they randomly guess the answers, and if not, they are asked to explain their selections. Out of the 50 "distinguishable" answers, 32 are derived by random guessing. Moreover, more than 70% of volunteers who provide deterministic answers claim that there is subtle audible noise in the injected samples, which is likely introduced by the attack circuit. Table II presents the accuracy of both random guess and deterministic answers. In fact, the deterministic answers, despite the responders' confidence, achieve very similar accuracy as the random guessing. Meanwhile, 17 out of 20 volunteers indicate that they will be incapable of recognizing the injected voice during a real phone call.

*5) Liveness Detection Robustness:* To defend against replay attack and inaudible voice command injection, the voice assistants can apply liveness detection models to recognize the maliciously injected voice commands. To evaluate the attack robustness of *GhostTalk* against liveness detection models, we inject 100 human speech samples from TIMIT dataset to an iPhone X, and input the injected recordings to three liveness detection models. The first model is ASVSpoof [26], the baseline liveness detection model of ASVSpoof 2017 challenge. ASVSpoof mainly considers the constant-Q cepstral coefficients (CQCC) features in the voice and leverages Gaussian Mixture Model (GMM) to separate the natural and replayed human speech. The second model, STC [27], is the best model in ASVSpoof 2017 challenge, which incorporates a Light Convolutional Neural Network (LCNN) to detect replay attacks. The third model, Void [13], is a state-of-the-art liveness detection system that detects replayed samples and inaudible voice commands using spectrogram delay patterns, peak patterns, and Linear Prediction Cepstrum coefficient (LPCC) features.

We use the reported results for replay and inaudible voice attacks in the respective defense studies, and evaluate the robustness of *GhostTalk* against the liveness detection models. The evaluation results are presented in Table III. For the replay attack using a loudspeaker, only a few samples can bypass the liveness detection systems. Moreover, Void could

| Attacks | ASVSpoof [26] | STC [27] | Void [13] |
|---|---|---|---|
| Replay attack | 24.77% | 6.73% | 8.7% |
| Inaudible attack | N/A | N/A | 0% |
| ***GhostTalk* (48kHz)** | **100%** | **100%** | **40.0%** |
| ***GhostTalk* (16kHz)** | **100%** | **63.0%** | **81.0%** |

TABLE III: The error rate of liveness detection systems against replay attack, inaudible voice command injection, and *GhostTalk* injection attack.

| Model | Charging port | Loudspeaker | SNR (dB) | Accuracy |
|---|---|---|---|---|
| iPhone 5s | Lightning | Single | 5.41 | 93.0% |
| iPhone X | Lightning | Dual | 4.75 | 92.7% |
| Honor 10 | USB-C | Single | 5.75 | 93.3% |
| MI 8 Lite | USB-C | Single | 4.93 | 92.7% |
| Note 10 | USB-C | Dual | 4.46 | 91.0% |
| Galaxy S9 | USB-C | Dual | 4.21 | 90.7% |
| Pixel 1 | USB-C | Single | 3.83 | 89.7% |
| Pixel 4XL | USB-C | Dual | 3.72 | 90.0% |
| **Pocophone** | **USB-C** | **Dual** | **1.51** | **36.0%** |

TABLE IV: Smartphone hardware information, leaked audio signal SNR, and digit classification accuracy of *GhostTalk-SC*.

accurately recognize all audio samples from inaudible voice command injection. As for *GhostTalk* injection attack, when we inject audio samples with 48 kHz sampling frequency, all of the samples can bypass ASVSpoof and STC models, which is understandable since *GhostTalk* injection occurs from the power line rather than the loudspeaker. It is worth noting that only 40% of the injected samples could successfully fool the Void system, which is likely due to the low-frequency patterns of the injected samples captured by the Void model. In response, we decrease the injected audio sampling rate from 48 kHz to 16 kHz, which effectively distorts the low-frequency patterns of the injected audio. As a result, 81% of injected samples can pass the Void model. On the other hand, the downsampling process also distorts or discards some high-frequency components, which degrades the error rate of the STC model to 63.0%. In summary, by tuning the injected audio sampling frequency, *GhostTalk* injection attack can successfully bypass different liveness detection models.

### B. GhostTalk-SC Attack Evaluation

*1) Experiment Setup:* We evaluate *GhostTalk-SC* attack on the smartphones listed in Table IV. The victim smartphones are charged by a 5V/1A DC power source, and the charging cables are standard Lightning or USB-C cables. An ESP-32 board is used to measure the charging current fluctuation with 8 kHz sampling frequency.

*2) Data Collection:* We train the CNN classifier with the Free Spoken Digit Dataset (FSDD) [28] consisting of 3,000 utterances from "zero" to "nine". We also collect 300 utterances from 15 speakers (8 males and 7 females, and each of them speaks 10 digits twice) and extract the leaked speech audio as the test dataset. Then, we classify the denoised voice samples by recognizing their spectrogram patterns under 2 kHz.

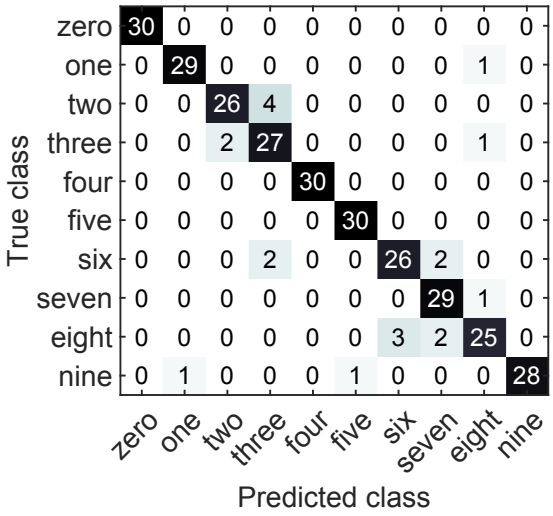



Fig. 16: Spoken digit classification confusion matrix (with Honor 10 smartphone).

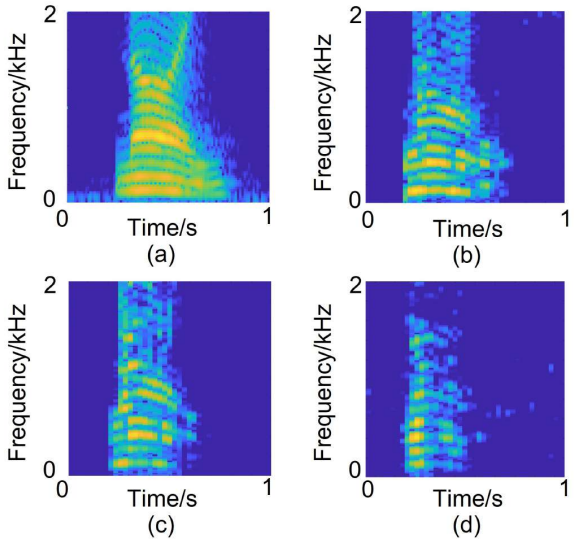



Fig. 17: spectrogram comparison of original audio and leaked audios under different volume settings. (a) is the spectrogram of original utterance "nine", and (b), (c) and (d) are leaked audio spectra when the volume level is 100%, 75% and 50%, respectively.

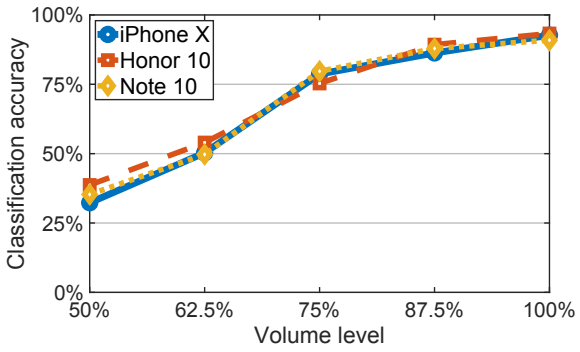



Fig. 18: Classification accuracy under different volume settings.

*3) Digit Classification Performance.:* Table IV lists the average leaked audio SNR of different smartphones and the spoken digits classification accuracy. The average SNR values vary substantially across different phone models due to the firmware and system difference.

The results show that *GhostTalk-SC* achieves satisfactory classification performance on 8 out of 9 victim smartphones. For the smartphones with stronger leaked audio signals, such as Honor 10, iPhone 5s, and iPhone X, *GhostTalk-SC* can achieve 92% or higher classification accuracy, which is significantly higher than random guessing (10%). For smartphones with weaker audio leakage, like Pixel 4XL, the accuracy descends. A special case is that *GhostTalk-SC* fails to classify most of the spoken digits from Pocophone as most of leaked audio signals are overwhelmed by the ambient noise. This exception may be attributed to the weaker loudspeaker power. Another possible explanation is that Pocophone's operating system (OS) or firmware runs with a higher power consumption that introduces excessive noise into the charging current.

Fig. 16 shows the digit classification confusion matrix (with counts of cases) from Honor 10 smartphone. From the confusion matrix, we notice that there are false predicted samples between "two" and "three", and "eight" is frequently misclassified as "six" or "seven", because these utterances have similar patterns in the low-frequency band. Due to the low SNR of leaked audio signal and frequency band limitation, the extracted digit utterances have lower distinguishability compared with the original utterances, which impacts the classification accuracy.

*4) GhostTalk-SC Performance under Different Volume Settings:* As we illustrate in Section III, the leaked audio signal in charging current is originated from the power side-channel. When the user turns down the volume, the loudspeaker power consumption decreases which in turn leads to the drop of the SNR of leaked audio signals. Fig. 17 shows the original utterance and leaked audio spectra under different volume settings of Huawei Honor 10. When the audio is played with the maximum volume, most of spectrogram patterns can be well recovered after denoising. If the volume is reduced to 75%, a portion of the patterns get lost or distorted after denoising. With 50% volume level, only the strongest frequency components remain in the spectrogram with most of patterns disappeared.

To evaluate the digit classification performance under different volume settings, we play the testing spoken digit audios on three victim smartphones including iPhone X, Honor 10, and Note 10. Each smartphone has 16 volume levels. We start with volume 100% (level 16) and repeat the experiment after tuning the volume.

The digit classification accuracy under different volume settings is presented in Fig. 18. For all the three victim phones, the classification accuracy drops when the volume is down. When the volume is 75% (level 12), the classification performance slightly degrades since most of utterances are still distinguishable. However, when the volume is set as 50% (level 8), the classification accuracy declines more drastically, i.e.,

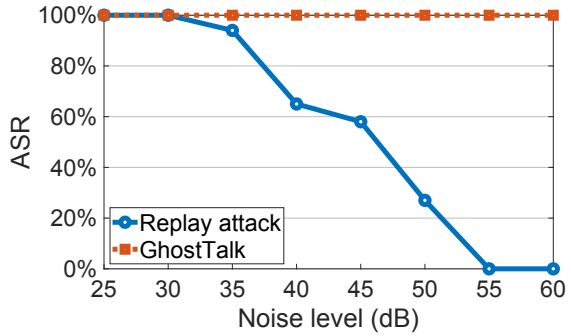



Fig. 19: ASR comparison of replay attack and *GhostTalk* injection attack in different noisy environments.

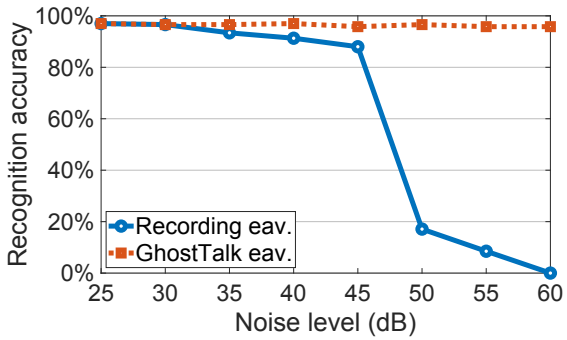



Fig. 20: Recognizability comparison of recording eavesdropping and *GhostTalk* eavesdropping attacks in different noisy environments.

only 35% spoken digits can be correctly classified. In a lower volume setting, the classification results approximate that of random guessing.

### C. Robustness Evaluation

*1) GhostTalk Injection Robustness:* Most of the existing voice attacks are susceptible to other acoustic interference, such as environmental noise, human conversation, and loud music. In the extreme, strong background noise will jam the microphone and block the voice command injection and audio eavesdropping.

To verify *GhostTalk* attack performance in noisy environments, we use a loudspeaker to play causal human conversations as background noise, and compare the robustness of *GhostTalk* with replay attack and recording eavesdropping attack. For the replay attack, we place the attacker (iPhone 8) 30cm away from the victim iPhone X, and replay voice commands with its maximum volume. For *GhostTalk* attack, we use the same experiment setup in Section VI-A2. The average noise level in quiet environment is 25 dB, and we repeat all the experiments with different background noise levels. Fig. 19 shows the robustness comparison of replay and *GhostTalk* injection attacks. When the noise level is below 30 dB, both replay attack and *GhostTalk* achieve 100% ASR. However, when the noise level is increased above 45 dB, the ASR of replay attack drops significantly. In the environments where noise level is above 55 dB, the replay attack cannot succeed. In contrast, *GhostTalk* leverages electric signals rather than acoustic signals to inject voice commands. As a result, the external noise will not affect the received audio signals. In all the noisy environments, *GhostTalk* injection attack can always achieve 100% ASR, which demonstrates the robustness of *GhostTalk* injection attack.

*2) GhostTalk Eavesdropping Robustness:* Moreover, we evaluate the robustness of *GhostTalk* eavesdropping attack on the iPhone X. Similar to the setup in Section VI-A3, we use an iPhone 8 as the recording device and compare the audio recognition with *GhostTalk* eavesdropping attack.

Fig. 20 illustrates the recognition accuracy comparison of recording eavesdropping and *GhostTalk* eavesdropping attack. Unsurprisingly, for normal recording eavesdropping, the recognition rate degrades when the environment noise becomes stronger. Specifically, when the background noise level is

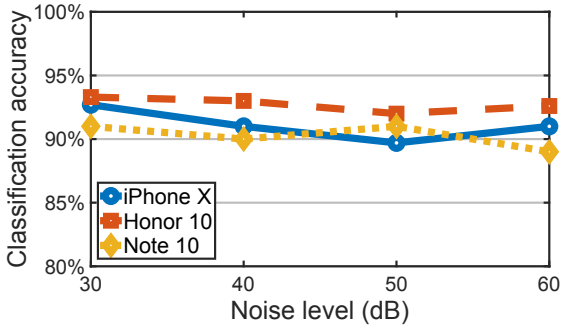



Fig. 21: *GhostTalk-SC* eavesdropping attack performance in different noisy environments.

higher than 50 dB, Google Speech-to-Text API can hardly recognize the speech contents. On the contrary, for the audios recovered by *GhostTalk* eavesdropping, their perceptibility remains at a constant level. Since the external noise has no impact on the electric signals, *GhostTalk* eavesdropping can still recover clear speech audios in noisy environments.

*3) GhostTalk-SC Eavesdropping Robustness:* To evaluate the robustness of *GhostTalk-SC* eavesdropping through standard cables, we repeat the experiment in Section VI-B3 in the environments with different noise levels. We test the robustness on 3 smartphones including iPhone X, Honor 10, and Note 10. All the phones play the testing speech audios with their maximum volume.

Fig. 21 presents the robustness evaluation results of *GhostTalk-SC* eavesdropping attack. We notice that even though the noise becomes increasingly stronger, the digit classification accuracy stays intact. The slight difference in accuracy is mainly caused by noise in the current measurement. The results prove that *GhostTalk-SC* eavesdropping attack is robust in noisy environments. It implies that *GhostTalk-SC* enables a much wider variety of attack scenarios compared with the existing eavesdropping attacks.

## VII. Discussion

**Word Eavesdropping Capability.** In the experiment, we evaluate the digital recognition performance of *GhostTalk-SC*. Here, we discuss its potential of word recognition. Fig. 22

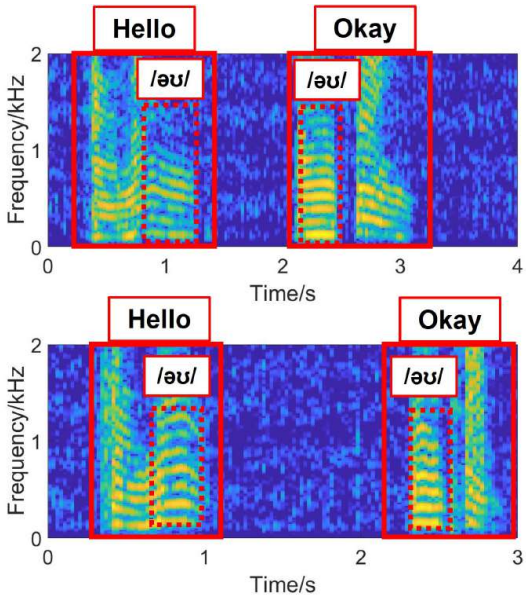



Fig. 22: The spectra of "hello" and "okay" from two volunteers with different speaking speeds.

shows the spectra of the words "hello" and "okay" from two male volunteers with different speaking speeds. Note that the denoised spectra of the same word pronounced by different volunteers present similar patterns. Therefore, it is potentially feasible to perform word recognition using GhostTask-SC. However, the CNN model requires a large dataset for the model training. Due to the lack of a large dataset containing speech samples of individual words, we cannot verify the word eavesdropping performance.

Based on the results in Fig. 22, we consider two potential approaches that can be used for eavesdropping words. First, similar to the FSDD dataset for the digit recognition, we can build a common word dataset, which stores spoken word audios from different human speakers. The dataset will be used to train a CNN model for word classification. This approach could potentially guarantee a high recognition accuracy, but it requires a large dataset of spoken words. Second, Fig. 22 also shows that the spectrogram components of the same phoneme /əʊ/ resemble each other regardless of the speaker identities. Therefore, we can build a phonetic symbol classification model to recognize phonemes, and the attackers can then recognize the words by annexing the phonemes. However, the accuracy of this approach could be degraded due to the difficulty of classifying short phonemes (i.e., they look alike). Moreover, the segmentation of phonemes poses a challenge in low-resolution spectra.

**Touching Screen Interference.** A recent attack, Charger-Surfing [7], demonstrates that the screen touching could produce notable disturbance in the charging current. Fig. 23 shows a charging current spectrogram when the user is touching the screen, and at the same time playing audio with the loudspeaker. The current noise caused by screen touching is much higher than the idling noise. Therefore, when the user touches the smartphone screen, the leaked audio signal will be overwhelmed in the strong interference and become unrecognizable. But *GhostTalk-SC* can still recognize the

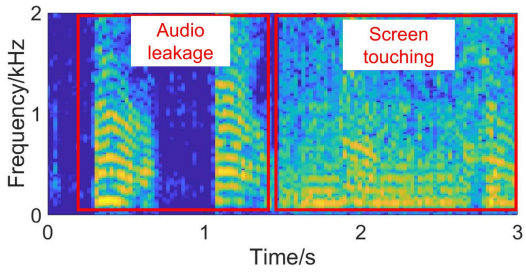



Fig. 23: The charging current spectrogram while the user is touching screen and playing audio.

digits between each screen touching.

**Attack Stealthiness.** The modified cable of *GhostTalk* in Fig. 4 has the same outlook as a standard cable. Moreover, our attack system prototype (4.5cm×2.5cm×1cm) is much smaller than a typical 20,000 mAh power bank (16.5cm×7.5cm×2.3cm). We can further shrink the size of the attack system by a better Printed Circuit Board (PCB) design. Therefore, it is realistic to hide the attack system board inside a power bank. Without opening up the power bank, the victims will not be able to notice the presence of the attack device. As for *GhostTalk-SC* attack, it is also quite feasible to hide the attack device behind the USB ports to make the attack stealthy as shown in other studies [17], [29].

## VIII. Countermeasure Recommendations

**Disable the Voice Assistant Activation by Headphones.** The key component of the *GhostTalk* attack is the activation of the voice assistant via shorting the microphone and audio ground wires. If the user disables this option, the attacker will be unable to activate the voice assistant to launch the attack.

**Headphone Notification.** Some Android smartphones, such as Huawei Honor 10, will display a "headphone detected" notification when a headphone is plugged in. If the attacker implements the *GhostTalk* attack on such types of smartphones, the users may be alerted by the notification and realize the underlying threat in the shared power bank.

**Stop Charging After Reaching High Percentage Battery Level.** Note that only the smartphones with their battery level exceeding 95% can be eavesdropped by *GhostTalk-SC* attack. If the victim stops charging before the battery state reaches that high level, *GhostTalk-SC* attack can be effectively avoided.

## IX. Related Work

### A. Inaudible Voice Command Injection

Inaudible voice command has been a serious threat for voice control systems. Backdoor [30] first illustrates the nonlinearity of microphones and transmits audio signal through inaudible ultrasound band. DolphinAttack [11] further leverages this nonlinearity to implement a hidden voice command attack to compromise voice control systems. LipRead [31] enables a long-range inaudible voice command attack using a speaker array. SurfingAttack [12] uses the guided ultrasound wave to inject inaudible voice commands though solid medium. However, these attacks must have prior-knowledge about the

authorized user's voice, and the attack success rate drops in noisy environments. Recently, Sugawara et al. [32] propose a long-range inaudible voice injection attack by projecting light signals to influence smart device microphones. Light commands directly modulate the voice commands on the light signals, making it resilient against noises. However, this attack only works in a line of sight scenario. Compared with existing voice command injection attacks, *GhostTalk* explores a new backdoor in the smartphone charging port to inject inaudible voice signals.

### B. Side-channel Eavesdropping Attacks

The loudspeaker of a smartphone vibrates when playing audio signals. The vibration side-channel can be exploited to implement the eavesdropping attacks by leveraging the smartphone motion sensors. GyroPhone [33] and Speechless [34] successfully recognize speaker identity and recover speech contents from motion sensor data. AccelEve [19] uses a motion sensor with a higher sampling frequency to further improve the attack performance, and it incorporates a DNN model to recognize the spoken digits. However, the attack performance is limited by the sampling frequency. Moreover, if the system requires the permissions for the access of motion sensor data, the attacks will no longer succeed. Recent research has also discovered that it is possible to eavesdrop audio signals by sensing the object vibration. ART [35] can eavesdrop loudspeakers by measuring the reflected wireless signal strengths and phase differences. Lamphone [36] can remotely eavesdrop audio signals by monitoring the slight changes in the brightness of vibrating bulbs. LidarPhone [18] exploits Lidar sensors on robot vacuum cleaners to measure the vibration of objects, which can effectively recover the private human speech. However, other audio sources may also interfere with the sensing of object vibration, which leads to the degraded attack performance in noisy environments. Compared with existing work, *GhostTalk* eavesdropping directly recovers pure audios played by the smartphone, and *GhostTalk-SC* can spy private information by extracting leaked audio signals from the power side-channel.

### C. Attacks via Charging Cables and Power-Line Channels

Malicious charging cables have been developed to compromise the smartphones. Lau et al. [17] successfully inject malware into iOS devices via malicious chargers. Shiroma et al. [37] successfully spy the victim device screen using a malicious USB cable. Spolaor et al. [38] further launch an attack to eavesdrop the sensitive information of Android smartphones from USB cables without requiring any permissions. In comparison, *GhostTalk* leverages the audio function backdoor in charging ports and hacks the voice system by deploying malicious charging cables.

In addition, the smartphone charging power is influenced by the smartphone apps when the battery state reaches a high level. Yang et al. [16] present an attack that fingerprints user's webpage history by monitoring the power usage, when the user is charging from a public power source. Cour et al. [15] further extend the fingerprinting attack towards wireless charging devices. POWERFUL [6] infers sensitive app usage through smartphone's power consumption profiles. Charger-Surfing [7] can recover the smartphone's lock screen password by monitoring the charging voltage. In this work, *GhostTalk-SC* utilizes the power side-channel to eavesdrop audios from the charging smartphones.

## X. Conclusion

In this paper, we explore the feasibility of voice injection and eavesdropping attacks via the power line. With a modified cable, *GhostTalk* can remotely inject and eavesdrop audio signals through the charging cable, enabling new interactive attack scenarios. *GhostTalk* does not need any authorized voice information and can work in a noisy environment. Meanwhile, with a standard cable, we design the *GhostTalk-SC* attack to launch an effective audio eavesdropping attack by measuring the charging current through the power line side-channel. By leveraging a DNN model, *GhostTalk-SC* can achieve higher than 92% accuracy in identifying spoken digits on various smartphones including iPhone 5s, Honor 10, and MI 8 Lite.

## Acknowledgement

The authors are grateful to the anonymous reviewers for their constructive comments and suggestions. This work is supported in part by the National Science Foundation grants CNS-1950171 and CNS-1949753.

Any opinions, findings, conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of United States Government or any agency thereof.

## References

[1] "Phone battery statistics across major US cities," https://veloxity.us/phone-battery-statistics/, Accessed on January 3, 2022.

[2] "Alibaba-backed battery-sharing startup energy monster files for U.S. IPO," https://supchina.com/2021/03/17/alibaba-backed-battery-sharing-startup-energy-monster-files-for-u-s-ipo/, Accessed on January 3, 2022.

[3] "Shared power bank in China," https://www.enmonster.com/, Accessed on January 3, 2022.

[4] "Public charging stations," https://www.teleadapt.com/, Accessed on January 3, 2022.

[5] M. Neugschwandtner, A. Beitler, and A. Kurmus, "A transparent defense against USB eavesdropping attacks," in *Proceedings of the 9th European Workshop on System Security*, 2016, pp. 1–6.

[6] Y. Chen, X. Jin, J. Sun, R. Zhang, and Y. Zhang, "POWERFUL: Mobile app fingerprinting via power analysis," in *IEEE INFOCOM 2017-IEEE Conference on Computer Communications*. IEEE, 2017, pp. 1–9.

[7] P. Cronin, X. Gao, C. Yang, and H. Wang, "Charger-surfing: Exploiting a power line side-channel for smartphone information leakage," in *30th USENIX Security Symposium (USENIX Security 21)*. USENIX Association, Aug. 2021.

[8] "O.MG cable can hack computers," https://shop.hak5.org/collections/mischief-gadgets/products/o-mg-cable-usb-a, Accessed on January 3, 2022.

[9] "Why Apple was right to remove the iPhone 7 headphone jack," https://www.forbes.com/sites/jvchamary/2016/09/16/apple-iphone-headphone-jack/, Accessed on January 3, 2022.

[10] Z. Wu, N. Evans, T. Kinnunen, J. Yamagishi, F. Alegre, and H. Li, "Spoofing and countermeasures for speaker verification: A survey," *speech communication*, vol. 66, pp. 130–153, 2015.

[11] 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*, 2017, pp. 103–117.

[12] Q. Yan, K. Liu, Q. Zhou, H. Guo, and N. Zhang, "Surfingattack: Interactive hidden attack on voice assistants using ultrasonic guided wave," in *Network and Distributed Systems Security (NDSS) Symposium*, 2020.

[13] M. E. Ahmed, I.-Y. Kwak, J. H. Huh, I. Kim, T. Oh, and H. Kim, "Void: A fast and light voice liveness detection system," in *29th USENIX Security Symposium (USENIX Security 20)*, 2020, pp. 2685–2702.

[14] "Charging Lithium-ion batteries: Not all charging systems are created equal," https://www.microchip.com/stellent/groups/designcenter_sg/documents/market_communication/en028061.pdf, Accessed on January 3, 2022.

[15] A. S. La Cour, K. K. Afridi, and G. E. Suh, "Wireless charging power side-channel attacks," in *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security*, ser. CCS '21, New York, NY, USA, 2021, p. 651–665. [Online]. Available: https://doi.org/10.1145/3460120.3484733

[16] Q. Yang, P. Gasti, G. Zhou, A. Farajidavar, and K. S. Balagani, "On inferring browsing activity on smartphones via USB power analysis side-channel," *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 5, pp. 1056–1066, 2016.

[17] B. Lau, Y. Jang, C. Song, T. Wang, P. H. Chung, and P. Royal, "Mactans: Injecting malware into ios devices via malicious chargers," *Black Hat USA*, vol. 92, 2013.

[18] S. Sami, Y. Dai, S. R. X. Tan, N. Roy, and J. Han, "Spying with your robot vacuum cleaner: eavesdropping via lidar sensors," in *Proceedings of the 18th Conference on Embedded Networked Sensor Systems*, 2020, pp. 354–367.

[19] Z. Ba, T. Zheng, X. Zhang, Z. Qin, B. Li, X. Liu, and K. Ren, "Learning-based practical smartphone eavesdropping with built-in accelerometer." in *NDSS*, 2020.

[20] L. Hertel, H. Phan, and A. Mertins, "Comparing time and frequency domain for audio event recognition using deep learning," in *2016 International Joint Conference on Neural Networks (Ijcnn)*. IEEE, 2016, pp. 3407–3411.

[21] A. v. d. Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, and K. Kavukcuoglu, "Wavenet: A generative model for raw audio," *arXiv preprint arXiv:1609.03499*, 2016.

[22] J. R. Stuart, "Noise: methods for estimating detectability and threshold," *Journal of the audio engineering society*, vol. 42, no. 3, pp. 124–140, 1994.

[23] V. Zue, S. Seneff, and J. Glass, "Speech database development at mit: Timit and beyond," *Speech communication*, vol. 9, no. 4, pp. 351–356, 1990.

[24] K. N. Stevens, *Acoustic phonetics*. MIT press, 2000, vol. 30.

[25] "Google speech-to-text API," https://cloud.google.com/speech-to-text/, Accessed in July 2021.

[26] T. Kinnunen, M. Sahidullah, H. Delgado, M. Todisco, N. Evans, J. Yamagishi, and K. A. Lee, "The asvspoof 2017 challenge: Assessing the limits of replay spoofing attack detection," 2017.

[27] G. Lavrentyeva, S. Novoselov, E. Malykh, A. Kozlov, O. Kudashev, and V. Shchemelinin, "Audio replay attack detection with deep learning frameworks." in *Interspeech*, 2017, pp. 82–86.

[28] Z. Jackson, C. Souza, J. Flaks, Y. Pan, H. Nicolas, and A. Thite, "Jakobovski/free-spoken-digit-dataset: v1. 0.8," 2018.

[29] "KeySweeper attack," https://samy.pl/keysweeper/, Accessed on January 3, 2022.

[30] N. Roy, H. Hassanieh, and R. Roy Choudhury, "Backdoor: Making microphones hear inaudible sounds," in *Proceedings of the 15th Annual International Conference on Mobile Systems, Applications, and Services*, 2017, pp. 2–14.

[31] N. Roy, S. Shen, H. Hassanieh, and R. R. Choudhury, "Inaudible voice commands: The long-range attack and defense," in *15th USENIX Symposium on Networked Systems Design and Implementation (NSDI 18)*, 2018, pp. 547–560.

[32] T. Sugawara, B. Cyr, S. Rampazzi, D. Genkin, and K. Fu, "Light commands: laser-based audio injection attacks on voice-controllable systems," in *29th USENIX Security Symposium (USENIX Security 20)*, 2020, pp. 2631–2648.

[33] Y. Michalevsky, D. Boneh, and G. Nakibly, "Gyrophone: Recognizing speech from gyroscope signals," in *23rd USENIX Security Symposium (USENIX Security 14)*, 2014, pp. 1053–1067.

[34] S. A. Anand and N. Saxena, "Speechless: Analyzing the threat to speech privacy from smartphone motion sensors," in *2018 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2018, pp. 1000–1017.

[35] T. Wei, S. Wang, A. Zhou, and X. Zhang, "Acoustic eavesdropping through wireless vibrometry," in *Proceedings of the 21st Annual International Conference on Mobile Computing and Networking*, 2015, pp. 130–141.

[36] B. Nassi, Y. Pirutin, A. Shamir, Y. Elovici, and B. Zadov, "Lamphone: Real-time passive sound recovery from light bulb vibrations." *IACR Cryptol. ePrint Arch.*, vol. 2020, p. 708, 2020.

[37] T. Shiroma, Y. Nishio, and H. Inoue, "A threat to mobile devices from spoofing public USB charging stations," in *2017 IEEE International Conference on Consumer Electronics (ICCE)*. IEEE, 2017, pp. 88–89.

[38] R. Spolaor, L. Abudahi, V. Moonsamy, M. Conti, and R. Poovendran, "No free charge theorem: A covert channel via usb charging cable on mobile devices," in *International Conference on Applied Cryptography and Network Security*. Springer, 2017, pp. 83–102.