XPorter: A Study of the Multi-Port Charger Security on Privacy Leakage and Voice Injection

Tao Ni
City University of Hong Kong
Hong Kong SAR
taoni2-c@my.cityu.edu.hk

Yongliang Chen
City University of Hong Kong
Hong Kong SAR
cs.ylchen@my.cityu.edu.hk

Weitao Xu
City University of Hong Kong
Hong Kong SAR
weitaoxu@cityu.edu.hk

Lei Xue
Sun Yat-Sen University
Shenzhen, China
xuelei3@mail.sysu.edu.cn

Qingchuan Zhao✉
City University of Hong Kong
Hong Kong SAR
qizhao@cityu.edu.hk

ABSTRACT

Multi-port chargers, capable of simultaneously charging multiple mobile devices such as smartphones, have gained immense popularity and sold millions of units in recent years. However, this charging-targeted feature can also pose security and privacy risks by allowing one of the simultaneously charging devices to communicate with another one if not properly designed and implemented as these devices are actually interconnected. Unfortunately, such risks have not been thoroughly investigated and we have identified a novel attack surface in the circuit design of multi-port chargers, which allows an adversary to exploit one port to *(i)* eavesdrop on the activities of other devices being charged and *(ii)* inaudibly inject malicious audio commands if the charging device supports voice assistants and USB-C interface.

In this paper, we design and implement a novel framework, XPorter, to analyze and demonstrate the uncovered security and privacy threats in multi-port chargers. Specifically, it leverages the changes in the voltage signals on one neighbor port to monitor the voltage changes of the charging port induced by various user activities, including recognizing the running apps and uncovering keystrokes. Moreover, XPorter can also achieve inaudible audio injection attacks from the neighbor port to the charging mobile device via the USB-C interface. We extensively evaluate the effectiveness of XPorter using five commodity multi-port chargers and five mobile devices. The evaluation results show its high effectiveness in recognizing the launching of 20 mobile apps (88.7%) and uncovering unlocking passcode (98.8%). Furthermore, XPorter achieves 100% success rates in inaudible audio injection attacks on three voice assistants. In addition, our study also shows that XPorter is resilient to various impact factors and presents the potential of attacking multiple victims.

CCS CONCEPTS

• **Security and privacy → Mobile and wireless security**; **Side-channel analysis and countermeasures**.

KEYWORDS

Multi-port charger, Privacy leakage, Voice injection, USB-C interface

ACM Reference Format:
Tao Ni, Yongliang Chen, Weitao Xu, Lei Xue, and Qingchuan Zhao. 2023. XPorter: A Study of the Multi-Port Charger Security on Privacy Leakage and Voice Injection. In *The 29th Annual International Conference on Mobile Computing and Networking (ACM MobiCom '23), October 2–6, 2023, Madrid, Spain.* ACM, New York, NY, USA, 15 pages. https://doi.org/10.1145/3570361.3613293

1 INTRODUCTION

The recent explosive growth of mobile devices, such as smartphones and tablets, has fostered various styles and capabilities of battery-charging accessories whose relevant market has been projected to reach approximately 1,580 million US dollars by 2022 [12]. Multi-port chargers are one of those representative accessories that provide multiple ports (*e.g.*, two or more USB-C/USB-A ports) for users to charge multiple mobile devices simultaneously. This type of charger is becoming extremely popular over the past five years because

✉The corresponding author.

ACM MobiCom '23, October 2–6, 2023, Madrid, Spain

ACM ISBN 978-1-4503-9990-6/23/10…$15.00
https://doi.org/10.1145/3570361.3613293

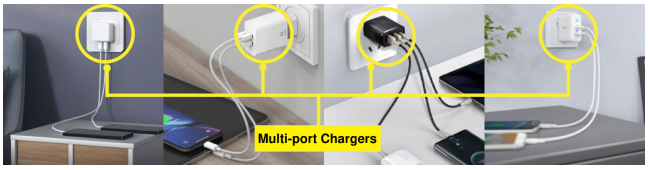


Figure 1: Illustration of multi-port chargers in real-life scenarios. A multi-port charger can support battery charging for multiple mobile devices simultaneously.

people have a growing demand for charging multiple devices with varying charging specifications [12]. Figure 1 shows four typical real-life scenarios that demonstrate the usage of commodity multi-port chargers.

However, this multi-device charging feature exposes an attack surface that could allow one device to conduct malicious actions on other devices when they are charging simultaneously. This vulnerability stems from the fundamental design of the multi-port charger, where all charging ports are connected in parallel and share the same voltage. Consequently, any voltage change in one port could affect all other parallel-connected ports, making it possible to launch attacks on one port to *eavesdrop or inject voice commands* into other connected devices. Previous research has revealed that the voltage changes of a charging mobile device can reveal sensitive information, such as button presses, keystrokes on the unlocking screen, and various running apps on smartphones [9, 17, 20, 39], and these voltage changes can also be exploited to manipulate the charging device's voice assistant, enabling the injection of malicious voice commands and potentially leading to the interpretation of incorrect information [37].

Unfortunately, these severe security and privacy concerns associated with multi-port chargers have been largely neglected. One possible reason for this oversight is that multi-port chargers appear to be immune to these security issues since they are not primarily designed for data transfer, which is an essential attack surface for eavesdropping and voice command injection attacks on other target devices (*e.g.*, USB hubs [13, 32, 35]). Therefore, we aim to *fill this knowledge gap* by analyzing two typical attacks, *i.e.*, *(i)* eavesdropping attacks and *(ii)* inaudible audio injection attacks, as the first step towards shedding light on these previously overlooked threats posed by multi-port chargers and contribute to enhancing their security measures.

We design and implement a novel framework, XPorter, to facilitate our study on eavesdropping and audio injection attacks stemming from the communication across (*X*) charging ports of a multi-*PORT* charg*ER*. Specifically, in regard to the eavesdropping attack, XPorter first detects the leakage of the voltage signals from one of the neighbor ports and then conducts signal processing to obtain the informative voltage clips to training models to recognize user activities to infer sensitive information on other charging devices. On the other hand, in respect of the inaudible audio injection attack, XPorter leverages the USB-C charging interface to activate the voice assistant of the charging device while bypassing the speech verification system and then injects malicious voice commands through a compromised multi-port charger.

We have implemented XPorter with a custom-built attacking device to demonstrate the feasibility of the aforementioned two attacks. As a proof of concept, first, XPorter aims to eavesdrop on three particular types of sensitive information, *i.e.*, unlocking passcode, launching apps, and sensitive keystrokes, from the charging device due to the fundamental design flaw existing in multi-port chargers. Specifically, we use the attacking device to collect the leaked voltage signals from 20 popular mobile apps and two soft keyboards (*i.e.*, unlocking numeric keyboard and full-size QWERTY keyboard) running on five mobile devices (*i.e.*, iPhone 13 Pro, iPhone 11, OnePlus 10 Pro, Google Pixel 4, and iPad Pro 2019) that are charging with five commodity multi-port chargers from different vendors (*i.e.*, UGREEN 40W dual USB-C charger, Anker 65W dual USB-C charger, Belkin 45W dual wall charger, Apple 35W dual USB-C charger, and ROMOSS 10W dual USB-A charger). Our evaluation results of the eavesdropping attacks show high effectiveness of XPorter where it achieves 98.8% in recognizing the unlocking passcode, 88.7% in fingerprinting the 20 mobile apps, and 83.0% in uncovering the alphabetic keystrokes of a QWERTY keyboard. Moreover, it also demonstrates that XPorter is resilient to various practical impact factors, including different multi-port chargers, mobile devices, and battery levels of the charging devices. In addition, we show the potential of eavesdropping on multiple victims' activities and provide efficient countermeasures to smooth out the voltage leakages to defend against XPorter.

In respect of demonstrating the inaudible audio injection attack, we evaluate it over three commodity voice assistants, including Apple Siri, Google Assistant, and OnePlus Breeno. Specifically, the attacking device can receive the voice commands remotely from the adversary by Bluetooth and then modulate them to injectable audio clips. Next, it leverages the audio pin of the USB-C interface to automatically activate the voice assistant of the charging smartphone while bypassing the speech verification mechanism that is widely deployed in commodity mobile devices. Finally, the modulated audio clips that contain malicious voice commands would be injected into the charging device to obtain more private information about the device's owner or manipulate the voice-controllable IoT devices (*e.g.*, Apple HomeKit). The extensive evaluation shows that XPorter achieves 100% success rate in activating the three voice assistants, injecting different voice commands, and 12 trials of end-to-end injection attacks. A demo video is available at https://youtu.be/X9HY9mIDTGw.

Contributions. We summarize the contributions as follows:

- **A novel attack.** We introduce a new attack vector that can be exploited to attack mobile devices charged by a commodity multi-port charger. It leverages the changes of the voltage leakage between the neighbor USB charging ports to reveal sensitive information and the characteristics of the USB-C interface to inject malicious voice commands to charging devices across ports.
- **A new framework.** We propose and implement a new attack framework, XPorter, to demonstrate the feasibility of the proposed attacks. Specifically, it exploits the leakage of the voltage signal to recognize the unlocking passcode, running apps, and sensitive keystrokes. In addition, it exploits the audio pins of the USB-C interface to inaudibly activate the voice assistant and inject malicious voice commands from the neighbor USB-C port to other charging devices.
- **Comprehensive evaluation.** We comprehensively evaluate the effectiveness of XPorter with five commodity multi-port chargers and five mobile devices. The results indicate that it performs effectively in eavesdropping on various user activities. Moreover, XPorter achieves a 100% success rate in activating different voice assistants and inaudibly injecting different voice commands. In addition, we also show the potential of attacking multiple victims and further provide effective countermeasures.

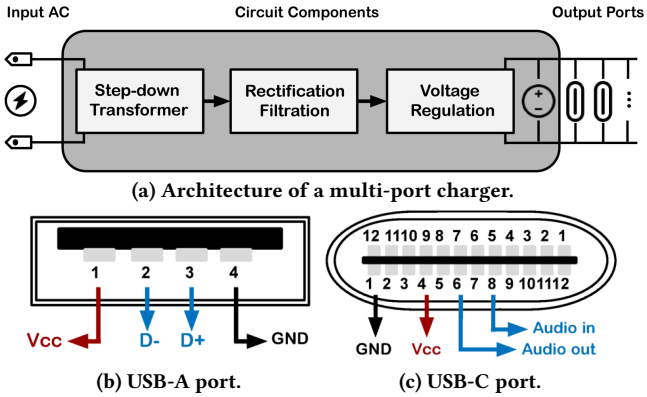
2 BACKGROUND

2.1 Multi-port Charger

Nowadays, the multi-port charger allows users to charge multiple mobile devices (*e.g.*, smartphones, tablets) at the same time. Figure 2a shows the basic architecture of a typical multi-port charger, which includes an AC voltage step-down transformer, a rectification circuit, a filtration circuit, a voltage regulation module, and multiple outputs charging ports. First, the step-down transformer converts the high input AC voltage (*e.g.*, 110 V AC) to low AC voltage (*e.g.*, 9 V AC). Then, the rectification circuit removes the negative part of the downgraded AC voltage to produce a partial DC with oscillations, and a filtration circuit suppresses such oscillations to generate a proper DC voltage. Finally, a voltage regulation module eliminates other noise and outputs the DC voltage (*e.g.*, 5 V DC) to the charging ports for powering multiple mobile devices. In particular, as the multi-port charger needs to power two or more devices simultaneously, the output charging ports are parallel connected together so that each of them obtains the same voltage (*e.g.*, 5 V). That is, in a charging process, the voltage changes on one port can induce voltage changes in its neighbor ports.

2.2 USB Type-A and USB Type-C Ports

Most commodity multi-port chargers adopt two types of Universal Serial Bus (USB) standards: USB Type-A (*a.k.a.*, USB-A) and USB Type-C (*a.k.a.*, USB-C). The USB-A port is commonly used in different mobile device accessories (*e.g.*, charger, USB-hub), and Figure 2b shows its structure, where two pins (pin 1 and pin 4) are used for charging the battery, and two pins support data transfer. Moreover, the USB-C port has been widely deployed in most Android smartphones and will be mandatorily applied to all smartphones (including iPhone) sold in the European Union by the end of 2024 based on a newly passed legislation [4]. Figure 2c shows the structure of a USB-C port that consists of 24 pins on two sides, and pins on both sides have the same functions because of its rotationally symmetrical structure. Therefore, users have no need to find the correct side to plug into the USB-C port. Furthermore, the USB-C port supports not only battery charging and data transmission but also audio input (pin 8) and audio output (pin 6). As both ports support battery charging, the power traces can be used for inferring user activities on the charging smartphone. In addition, due to the integrated features of USB-C ports, the smartphone is also threatened by potentially injecting inaudible voice commands, as we will demonstrate in this work.



(a) Architecture of a multi-port charger.

(b) USB-A port. (c) USB-C port.

Figure 2: Architecture of a multi-port charger and USB ports: (a) Circuit of a typical multi-port charger, (b) USB-A port (4 pins), and (c) USB-C port (24 pins on two sides).

2.3 Fundamental Principles

Below, we illustrate the fundamental principles of voltage leakage and audio injection between two neighbor USB ports of a multi-port charger from the aspect of physics.

Voltage leakage. Due to the parallel-connected architecture of USB ports in multi-port chargers, which allows them to simplify the circuit design by sharing a common output DC voltage, a fundamental design flaw arises, resulting in voltage leakage across neighboring USB ports. In a common battery charging scenario, we denote the output voltage of the charging port as $V_c(t)$ and the voltage of another neighbor port as $V_x(t)$. As these two ports are parallel connected, their relations are shown in Equation 1 as follows:

$$V_x(t) \propto C \cdot V_c(t), \tag{1}$$

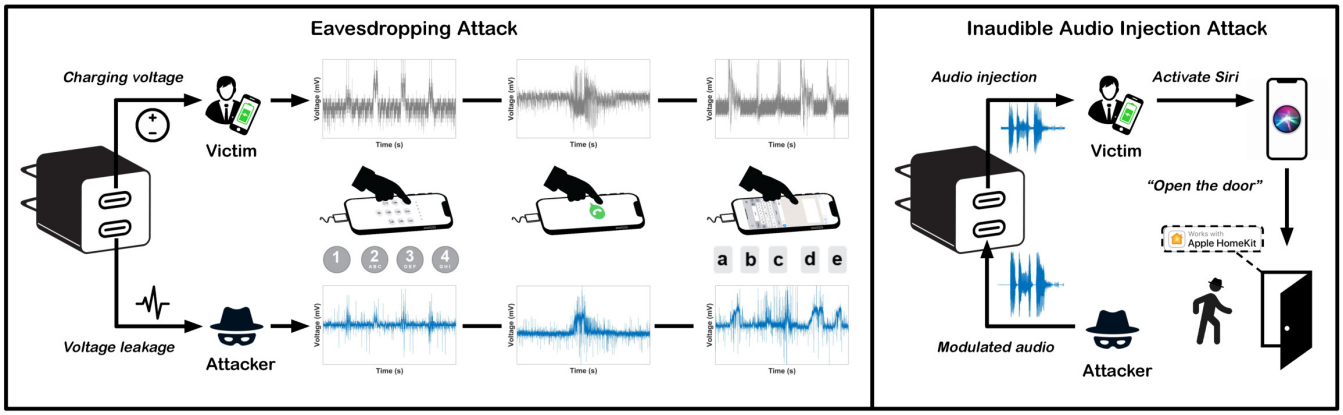


Figure 3: Motivating example scenario of two attacks via XPorter. *(i)* Eavesdropping attack (left-hand): When the smartphone is being charged by a multi-port (2-port) charger, the victim performs various activities (*e.g.*, unlocking the smartphone, opening an app, and typing keystrokes), which induces voltage changes (grey color) on the charging port as well as the neighbor port. Meanwhile, the attacker acquires the voltage leakage (blue color) and utilizes it to uncover private information (*e.g.*, passcode, app usage, and sensitive keystroke input). *(ii)* Inaudible audio injection attack (right-hand): Based on a compromised multi-port charger, the attacker can achieve audio injection by activating the voice assistant of the victim's smartphone through the audio pin of the USB-C interface and then injecting malicious voice commands to obtain further user privacy (*e.g.*, "Where's my home?") or control the voice-controlled IoT devices (*e.g.*, "Open the door").

where C is a mapping factor that reflects the $V_x(t)$ changes with the $V_c(t)$ based on the circuit design between the neighbor USB ports. Note that the magnitude and shape of $V_x(t)$ and $V_c(t)$ may be different, but the mapping factor C only depends on the design of the hardware circuit [32, 33]. That is, for a specific multi-port charger, C is a constant factor between the two neighbor USB ports.

We assume the load of the smartphone is $R_s(t)$ when being charged by a multi-port charger through a USB powerline. Based on Ohm's law, we can present the running current $I_c(t)$ for charging the smartphone in Equation 2:

$$I_c(t) = V_c(t)/R_s(t) \propto 1/R_s(t), \tag{2}$$

When the user performs different smartphone activities (*e.g.*, running apps, pressing buttons on keyboards), these activities induce different displays of lighter/darker pixels on an OLED touchscreen that consume different amounts of power [9], resulting in load changes $\Delta R(t)$ on the battery of the charging smartphone [20, 38]. As such, these load changes induce the changes of voltage $\Delta V_c(t)$ on the charging port, as well as voltage changes $\Delta V_x(t)$ the neighbor port because of the leakage across ports, which is shown in Equation 3:

$$\Delta V_x(t) \propto C \cdot \Delta V_c(t) \propto C \cdot I_c(t) \cdot \Delta R(t). \tag{3}$$

Therefore, it is feasible to exploit the voltage leakage of a neighbor USB port to monitor the voltage changes of the charging smartphone and further infer user privacy.

Inaudible audio injection. Since USB-C ports support audio transmission as discussed in §2.2, it is feasible to achieve an inaudible audio injection attack towards the voice assistant of the victim's smartphone. The first step is to activate the voice assistant, but most commodity smartphones have a speech verification mechanism that can deny the activation request of a non-owner activating command. Fortunately, the USB-C interface provides a solution to activate the voice assistant [37]. That is, commodity smartphones allow the wire control board of the earphone to activate voice assistant by pressing the button for nearly 1 to 2 seconds, which is also integrated into the functions of a USB-C port. Therefore, the attacker can manipulate the audio pin's voltage changes to simulate a button-pressing event to inaudibly activate the voice assistant of the victim's smartphone while bypassing the speech recognition system.

After activating the voice assistant through the above method, one can inject a modulated audio signal that contains malicious voice commands to the victim's smartphone across the neighbor USB-C ports of a commodity multi-port charger. Specifically, the modulated audio signal $A(t)$ for injecting voice commands can be denoted as follows:

$$A(t) = \alpha \cdot x(t) + V_{offset}, \tag{4}$$

where $x(t)$ is the original audio clip containing the voice command, α is a factor to adjust the amplitude, and V_{offset} is an extra DC offset to compensate for the initial voltage of the port. Then an analog-to-digital converter (ADC) will take the modulated signal and convert it to a digital signal that can be recognized by the audio pin of the USB-C interface.

3 MOTIVATION AND THREAT MODEL

3.1 A Motivating Example

We present a motivating example of launching eavesdropping and audio injection attacks through a commodity multi-port charger in this subsection. That is, the user connects the smartphone to one port of the charger for battery charging,

unlocks the smartphone with a password (*e.g.*, "1234"), and then launches the app WhatsApp to send a message to others (*e.g.*, "abcde"). This series of activities change the energy consumption of the smartphone battery and further changes the running current in the power line as well as the output voltage provided by the charger. As mentioned in §2.1, the voltage changes in one port can induce voltage changes of other neighbor ports, and these changes present detectable and predictable features that can be exploited for inferring corresponding user activities. In addition, the attacker can exploit the integrated audio pin in the USB-C interface to activate the voice assistant (*e.g.*, Apple Siri) and then inject malicious audio commands (*e.g.*, "open the door").

In Figure 3, we present the changes of voltages in both the user's charging port and a neighbor port when the user performs different activities. Specifically, we show the voltage changes of unlocking password input, app launching, and QWERTY keystrokes. As can be seen, both the voltages of the charging port (grey curve) and the neighbor port (blue curve) present distinctive and synchronized changes when pressing a button to unlock the smartphone, launch apps, or enter keystrokes. In addition, we also show the patterns at the audio input pin of the user's charging port and the neighbor port when we activate the voice assistant Siri and inject the malicious voice command "open the door" into it. As such, Siri will then follow the voice command to open the door of a smart home that is equipped with Apple HomeKit [3].

3.2 Threat Model

We consider a common scenario of using multi-port chargers to charge mobile devices (*e.g.*, smartphones) where victims connect their devices to the ports and perform different activities (*e.g.*, unlocking the smartphone, running apps). An attacker can share the multi-port charger with the victims and launch two types of attacks: *eavesdropping attack* and *inaudible audio injection attack*. Such a scenario is ubiquitous in public facilities and shared space, *e.g.*, offices and airports.

Eavesdropping. When launching an *eavesdropping attack*, the attacker monitors the voltage changes of a neighbor port and exploits the voltage traces to infer privacy-sensitive information, *i.e.*, *(i)* digits of the smartphone's unlocking password, *(i)* the victims' app usage and corresponding activities, and *(iii)* sensitive keystrokes of QWERTY keyboard. We assume the attacker can share the neighbor port of a multi-port charger with the victims, but *cannot* compromise *(i)* the commodity multi-port charger to install malicious firmware, *(ii)* the victims' USB power line to monitor current traces, and *(iii)* the victims' smartphone including malware installation.

Inaudible audio injection. When launching an *inaudible voice injection attack*, the attacker can use the USB-C interface to bypass the speech verification and activate the voice assistant (*e.g.*, Apple Siri, Google Assitant) of the victims' smartphones to inject modulated audio commands through the audio signal pin of the neighbor USB-C port. As such, we assume that the attacker can first compromise a multi-port charger that has USB-C ports by connecting the audio pins of the two neighbor USB-C ports together. Then, the attacker shares the multi-port charger with the victims in a shared place and then leverages a customized attacking device to achieve the inaudible audio injections. In addition, the attacker can utilize speech synthesis [34] tools (*e.g.*, Google WaveNet [27]) to generate modulated audio commands.

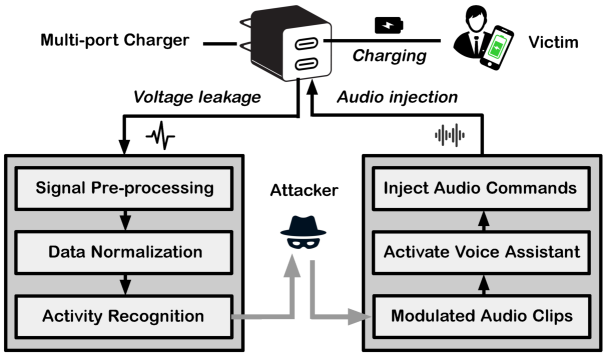


Figure 4: Overview of XPorter.

4 ATTACK DESIGN

4.1 Overview of XPorter

Figure 4 presents the overview of XPorter in launching the eavesdropping attack and audio injection attack. Specifically, an attacker first shares the multi-port charger with the victim and obtains the voltage leakage from a neighbor port. Then, the recorded voltage traces will be processed and normalized for user activity recognition to eavesdrop on privacy information, *i.e.*, unlocking passcode, running app activities, and in-app keystrokes. Moreover, the attacker can exploit the integrated audio pins in USB-C ports to inaudibly activate and inject modulated audio commands into the victim's charging smartphone to maliciously access the voice assistant systems (*e.g.*, Apple Siri, Google Assistant).

4.2 Eavesdropping Attack

Below, we present the design and implementation of XPorter in launching an eavesdropping attack, which consists of three components as follows: *(i)* signal pre-processing, *(ii)* data normalization, and *(iii)* activity recognition.

4.2.1 Signal Pre-processing. After obtaining the raw voltage signals, we design a signal processing algorithm to handle the acquired voltage leakage as shown in Algorithm 1. Specifically, XPorter first exploits a Savitzky-Golay (S-G) filter to remove high-frequency noise in the collected time-series signals (line 2-6) without distorting the signal shapes [7]. We

then use the average values of the first one-second data as the DC offset and deduce this offset value in the following signals (line 7). Since the captured voltage signals contain both non-activity and activity-induced voltage changes, we apply a moving-variance window with a given threshold τ (*e.g.*, 0.05) to find the start and end indices of the activity patterns and then segment the signal with privacy information of specific user-smartphone interactions (line 8-18).

4.2.2 Data Normalization. To eliminate the impact of the varied output voltages when charging different mobile devices, we apply methods of data normalization on the segmented voltage signals. Specifically, we normalize the amplitude of the processed voltage signals to the range from 0 to 1 and utilize the decimation factor down-sampling algorithm [16] to reshape these voltage signals to fixed length vectors (*e.g.*, 128×1), and then leverage the dynamic time warping (DTW) algorithm [30] to generate the vectors that maintain the informative patterns for training deep learning models that can recognize fine-grained user activities. Specifically, the DTW algorithm maps output voltage signal $\mathcal{S}$ to the down-sampled $\mathcal{S}^{'}$ by optimizing all admissible paths from S_i to $S_i^{'}$ as shown in Equation 5:

$$DTW_q(S_i, S_i^{'}) = \min_{\pi \in \mathcal{P}(S_i, S_i^{'})} \Big(\sum_{(i,j)\in\pi} d(S_i, S_j^{'})^q \Big)^{(\frac{1}{q})}, \tag{5}$$

where π is the alignment path of a sequence of K-length index pairs $((i_0, j_0), (i_1, j_1), \ldots, (i_{K-1}, j_{K-1}))$, $\mathcal{P}(S_i, S_i^{'})$ is the set containing all admissible paths, $d(S_i, S_i^{'})$ is the Euclidean distance between $\mathcal{S}_i$ and $\mathcal{S}_j^{'}$, and q is the power constant.

To resolve this optimization problem, we need to obtain the quantity $R_{i,j}$ [36] between two timestamps i and j as:

$$R_{i,j} = DTW_q(S_{\rightarrow i}, S^{'}_{\rightarrow j})^q, \tag{6}$$

where $S_{\rightarrow i}$ means the time-series voltages obtained up to timestamp i, and we can further obtain $R_{i,j}$ as Equation 5:

$$\begin{aligned} R_{i,j} &= \min_{\pi \in \mathcal{P}(S_{\rightarrow i}, S^{'}_{\rightarrow j})} \sum_{(k,l)\in\pi} d(S_k, S_l^{'})^q \\ &\overset{*}{=} d(S_i, S_j^{'})^q + \min_{\pi \in \mathcal{P}(S_{\rightarrow i}, S^{'}_{\rightarrow j})} \sum_{(k,l)\in\pi[:-1]} d(S_k, S_l^{'})^q \\ &\overset{**}{=} d(S_i, S_j^{'})^q + \min(R_{i-1,j}, R_{i,j-1}, R_{i-1,j-1}), \end{aligned} \tag{7}$$

where $*$ denotes the constraints on all admissible paths π, and we set the target length K as 128 and calculate the each $R_{n-1,m-1}$ to retrieve the corresponding $DTW_q(S_i, S_i^{'})$. After the data normalization process, we then collect the normalized data vectors as the input to train a deep learning classifier for fine-grained user activity recognition.

Algorithm 1: Signal processing of eavesdropping attack

Input: $\mathcal{V} = [v_{c_1}(t_1), v_{c_2}(t_2), \ldots, v_{c_m}(t_m)]$: obtained signals from the voltage leakage. o, f: order and frequency of the S-G filter. τ: threshold of the variance.
Output: $\mathcal{S} = [S_1, S_2, \ldots, S_n]$: filtered voltage signal clips containing specific smartphone activities.

1 $\widetilde{\mathcal{V}} \leftarrow [], \mathcal{S} \leftarrow []$ ▷ initialize the empty array to record filtered signals and segmented voltage signal clips.
2 $filter \leftarrow sgolayfilt(o, f)$ ▷ initialize an S-G filter with the given order o and the frequency f.
3 **for** *each signal* $v_{c_i}(t_i) \in \mathcal{V}$ **do**
4 $\quad \widetilde{v}_{c_i}(t_i) \leftarrow filter(v_{c_i}(t_i))$
5 $\quad \widetilde{\mathcal{V}} \leftarrow [\widetilde{v}_{c_1}(t_1), \widetilde{v}_{c_2}(t_2), \ldots, \widetilde{v}_{c_i}(t_i)]$
6 $\widetilde{\mathcal{V}} \leftarrow [\widetilde{v}_{c_1}(t_1), \widetilde{v}_{c_2}(t_2), \ldots, \widetilde{v}_{c_m}(t_m)]$ ▷ the filtered signals.
7 $\widetilde{\mathcal{V}} \leftarrow \widetilde{\mathcal{V}} - average([\widetilde{v}_{c_1}(t_1), \cdots, \widetilde{v}_{c_f}(t_f)])$ ▷ deduct offset.
8 $window \leftarrow movvar(\tau, f/10)$ ▷ initialize an moving-variance window with the given threshold τ and size of $f/10$.
9 **for** *each filtered signal* $\widetilde{v}_{c_i}(t_i) \in \widetilde{\mathcal{V}}$ **do**
10 $\quad \mathcal{R}_{c_i}(t_i) \leftarrow window(\widetilde{v}_{c_i}(t_i))$ ▷ obtain the time-variance signal from the moving-variance window.
11 $\quad$ **for** *each* $r_i \in \mathcal{R}_{c_i}(t_i)$ **do**
12 $\quad\quad$ **if** $\forall r_j \in [r_i, r_{i+f/10}], r_j < r_{j+1}$ *and* $r_j > \tau$ **then**
13 $\quad\quad\quad k_{start} \leftarrow r_i$ ▷ obtain *start* index of the activity.
14 $\quad\quad$ **else if** $\forall r_j \in [r_i, r_{i+f/10}], r_j > r_{j+1}$ *and* $r_j > \tau$ **then**
15 $\quad\quad\quad k_{end} \leftarrow r_{i+f/10}$ ▷ obtain *end* index.
16 $\quad S_i \leftarrow [\widetilde{v}_{c_i}(k_{start}), \widetilde{v}_{c_i}(k_{start})]$ ▷ voltage signal clip that contains the specific activity.
17 $\quad \mathcal{S} \leftarrow [S_1, S_2, \ldots, S_i]$
18 $\mathcal{S} = [S_1, S_2, \ldots, S_n]$
19 Output voltage signal clips $\mathcal{S}$ that contain user activities.

4.2.3 Activity Recognition. As the processed voltage signals are time series, XPorter adopts a one-dimensional convolutional neural network (CNN) cascaded with a Long Short-term Memory (LSTM) [14] layer to build a classifier for various activity recognition (*e.g.*, app launching, single key-pressing inference). Specifically, CNN-based neural networks are utilized in various side-channel attacks [9, 17] using one-dimensional time-series signals because the convolutional layers can capture both temporal and spatial features from time-series signals and achieve a promising classification accuracy [15]. Furthermore, as the CNN extracts multiple features from the voltage signal, we use an LSTM layer to learn the order dependence and identify these features.

The topology of our CNN-LSTM model consists of three convolutional layers followed by an LSTM layer, a fully-connected layer, and a softmax layer with a single output for each instance (*e.g.*, key, app). For the three convolutional layers, we use the ReLU as the activation function and add

a max-pooling layer to reduce the dimension by half. Then, a flatten layer converts the extracted feature maps to one-dimensional vectors as the valid input for the LSTM layer. After the LSTM layer, a dropout layer with 50% dropout rate is added to regularize the network and prevent overfitting. Finally, the fully-connected layer and the softmax layer output the predicted class with the highest probability.

4.2.4 Implementation Details. In practice, we implement the first two components, signal pre-processing (§4.2.1) and data normalization (§4.2.2) by leveraging MATLAB R2022a Signal Processing Toolbox (version 3.0) that supports reliable toolkits. Then, we implement the CNN-LSTM neural networks for activity recognition in Keras 2.3 on the Tensorflow 2.0 framework. In the training stage, we set the batch size as 32 and use the cross-entropy loss and Adam optimizer with an initial learning of 0.01 and epoch of 100. In particular, the output shape depends on the corresponding task (*e.g.*, the number of apps and the number of keys on a keyboard). Specifically, we study 10 numeric buttons on the unlocking keypad of the touchscreen (10 classes), 20 different mobile apps (20 classes), and the alphabetic keys on the full-size QWERTY keyboard (26 classes).

4.3 Inaudible Audio Injection Attack

Apart from the eavesdropping attack, we also present the design and implementation of XPorter in launching an inaudible audio injection attack in this subsection. As mentioned in §3.2, the attacker can simply compromise the multi-port charger by connecting the audio pins of the output ports together without modifying the packaging, which results in less suspicion for the victim. Then the attacker connects the attacking device (details in §4.4) to the neighbor USB-C port and conducts three steps to achieve malicious audio injection: *(i)* audio modulation (§4.3.1), *(ii)* voice assistant activation (§4.3.2), and *(iii)* audio commands injection (§4.3.3).

4.3.1 Audio Modulation. As discussed in §2.2, the audio signals obtained by the USB-C port are represented by the changing current and voltage of the audio pin. As such, the attacker should first convert the audio clips that contain malicious voice commands to modulated voltage signals and then inject these modulated voltage signals into the victim's smartphone. Specifically, we can exploit Equation 4 in §2.3 to implement the audio modulation from the audio clips to the recognizable voice commands. To achieve automatic audio modulation, we use an audio board with a Bluetooth module to receive the malicious voice command from the attacker remotely and modulate it to a voltage signal that can be received by the audio pins of the USB-C port and recognized by the voice assistant of the victim's mobile device. Moreover, we also apply a differential amplifier module to adjust the amplitude of the modulated audio signal to obtain the best configurations for the injection attacks.

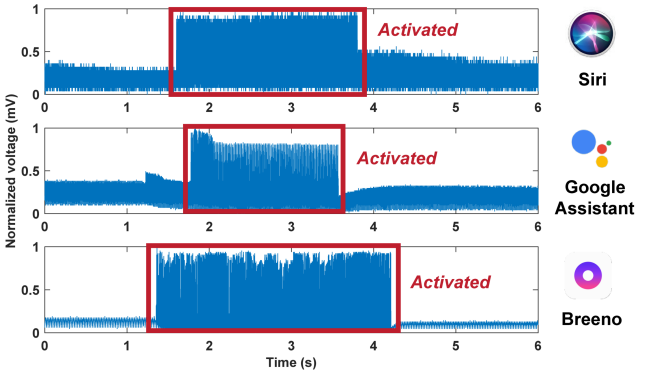
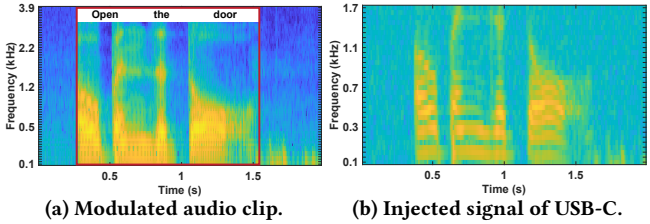


Figure 5: Audio pin voltage signals when activating three commodity voice assistants (Apple Siri, Google Assistant, and Breeno) through the USB-C interface. The red boxes present the voltage changes when the voice assistants are activated.

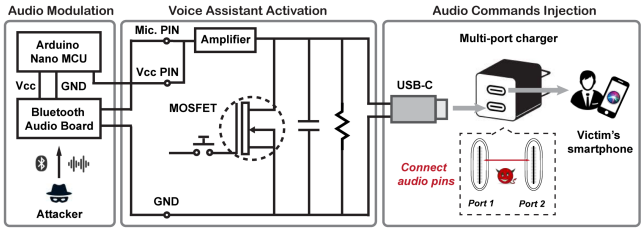


(a) Modulated audio clip. **(b) Injected signal of USB-C.**

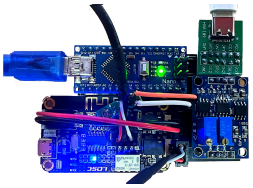
Figure 6: Spectrograms of the modulated audio clip and the voltage signal of the USB-C audio pin when injecting the voice command "Open the door" to Siri through XPorter.

4.3.2 Voice Assistant Activation. Previously, inaudible audio injection attacks [21, 41] on smartphones' voice control systems require voice samples from authorized users to generate hotword commands (*e.g.*, "Hey Siri" or "Hello Google") through virtual microphones and speakers to activate the voice assistants. However, these replaying methods can easily be detected and prevented by state-of-the-art verification approaches [1, 19, 40]. Therefore, in §2.3, we introduced the headphone button-pressing event that can activate the voice assistant while bypassing the speaker verification system. To verify its practicality, we record the voltage signals of the USB-C audio pin when activating smartphone voice assistants and present the results in Figure 5. In practice, we tested it on three commodity voice assistants (Apple Siri, Google Assistant, and OnePlus Breeno), and we can know that the voltage of the audio pin will boost to a high stage when the voice assistant is activated. In particular, we find that different voice assistants require different patterns of input voltage changes on the USB-C audio pin to activate themselves, *e.g.*, different lasting times and amplitudes.

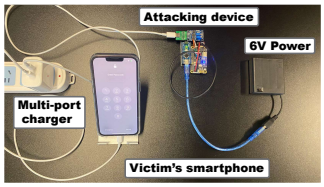
To activate the voice assistant through the introduced method and achieve a more generalized audio injection attack, we use a wire control board that contains a MOSFET transistor to manipulate the voltage received by the audio



(a) Circuit design of the attacking device.



(b) Prototype.



(c) Attack scenarios.

Figure 7: Attacking device in launching eavesdropping and audio injection attacks through a multi-port charger.

pin of a USB-C port. Specifically, the MOSFET transistor is used to control the current flow between the audio pin and the ground to simulate a fake button-pressing event that produces the same pattern of the input voltage for activating the voice assistants of the charging mobile devices.

4.3.3 Audio Commands Injection. After obtaining the modulated audio signals and activating the voice assistant, the attacker can inject malicious audio commands through the compromised multi-port charger to acquire user privacy and perform further attacks. For instance, the attacker can send voice commands like "What's my name?" to obtain the victim's private information, make a ghost phone call by injecting "Call my wife", and hack the smart home equipped with a voice control system (*e.g.*, Apple HomeKit) by sending malicious voice commands like "Open the door". Figure 6a and Figure 6b individually present the spectrograms of the modulated audio clip and the injected signal received by the USB-C audio pin when sensing the voice command "Open the door" to Siri through XPorter. In particular, we find that despite the modulated audio being distorted in voice command injection, Siri can recognize the command and conduct corresponding responses because the patterns that contain the most important information are maintained as the two spectrograms present (yellow part).

4.4 Custom-built Attacking Device

We design and implement a portable attacking device to achieve eavesdropping and inaudible audio injection attacks in XPorter, and Figure 7a–Figure 7c show the circuit design, prototype outlook, and attack scenarios, respectively. First, the attacker can record the voltage leakages from the neighbor USB port to launch various eavesdropping attacks. Second, based on the assumption that the attacker can compromise the multi-port charger by parallel connecting the audio pins of the neighbor USB-C ports, an attacker can connect this attacking device to one of the USB-C ports and then remotely activate the voice assistant of the victim's mobile device and send audio clips that contain malicious voice commands to uncover sensitive information further.

In the prototype, we utilize an Arduino Nano microcontroller to record the voltage leakages and control the MOSFET transistor from a CX-729 wire control board [2] for voice assistant activation, a Bluetooth audio board [11] for receiving voice commands, an AD620 amplifier module [29] for adjusting the amplitude of the recorded voltage signals or modulated audio signals. As a proof-of-concept, we integrate these components in a customized extension PCB board powered by an external battery pack to eavesdrop on user activities as well as inaudibly inject malicious voice commands into the victim's smartphone through the USB-C interface. Note that it is possible to draw power from the charger to support the attacking device by redesigning the prototype, which can also be implemented smaller and stuffed into the compromised charger to launch attacks directly.

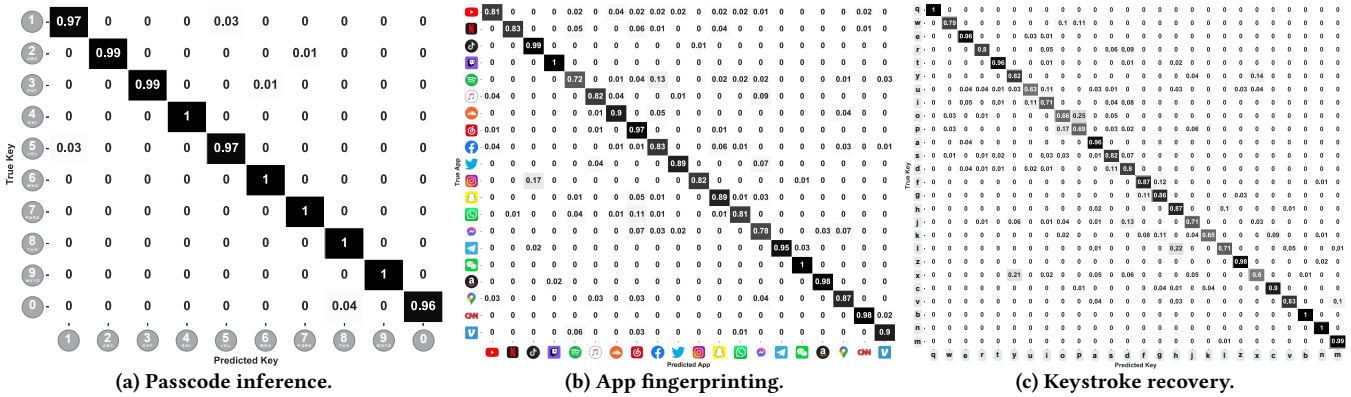
5 EVALUATION

5.1 Effectiveness of Eavesdropping Attack

5.1.1 Experimental Setup. In the primary setting for evaluating the effectiveness of the eavesdropping attacks, we use the UGREEN 40W USB-C port charger[1], which has two USB-C ports for battery charging. Specifically, we first use one port to charge an iPhone 13 Pro as the victim's smartphone and then use the custom-built attacking device to record the voltages of another port when recruiting five participants (three males, two females) to collect data samples perform three common activities: *(i)* entering the password to unlock the smartphone, *(ii)* launching different mobile apps, and *(iii)* typing words in chat apps such as WhatsApp. We follow the same procedure and separately conduct experiments on four other commodity multi-port chargers from different vendors (§5.3.1), four other mobile devices (§5.3.2), and four other battery levels of the charging device (§5.3.3). Moreover, all data processing and model training processes are conducted on a desktop with 32 GB memory and an Intel i7-9700K CPU, and an NVIDIA GeForce RTX 2080Ti GPU.

5.1.2 Effectiveness of Unlocking Password Inference. To evaluate the effectiveness of XPorter in inferring unlocking password, we collect voltage signals and obtain the processed data samples from the neighbor output USB-C port while pressing each button (*i.e.*, from 0 to 9) on the unlocking numeric keyboard for 100 times with a time interval of

[1]Note that dual-port chargers are also marketed as multi-port chargers. We adopt it to verify the feasibility of XPorter, and also show the potential of attacking multiple charging devices in §6.1. This work takes ethical considerations seriously and has been approved by the IRB of our institution.



(a) Passcode inference. (b) App fingerprinting. (c) Keystroke recovery.

Figure 8: Effectiveness evaluation of eavesdropping attack. (a): Confusion matrix of recognizing 10 different passcode pins (from 0 to 9) on the unlocking screen. (b): Confusion matrix of fingerprinting 20 mobile apps. (c): Confusion matrix of uncovering 26 different keys on a full-size QWERTY keyboard (from "a" to "z"). Evaluated app list: –YouTube, –Netflix, –TikTok, –Twitch, –Spotify, –Apple Music, –SoundCloud, –Netease Cloud Music, –Facebook, –Twitter, –Instagram, –Snapchat, –WhatsApp, –Messenger, –Telegram, –WeChat, –Amazon, –Google Map, –CNN News, –Venmo.

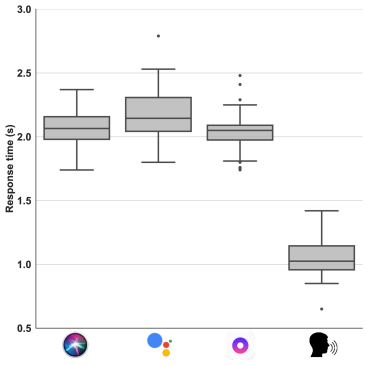


Figure 9: Response time of assistants and human speaking.

#	Voice Command	SNR (dB)						#	Voice Command	SNR (dB)					
			Act.	Inj.	Act.	Inj.	Act.	Inj.			Act.	Inj.	Act.	Inj.	Act.	Inj.
1	Call mom.	20.7	✓	✓	✓	✓	✓	✓	11	Where is my home?	19.0	✓	✓	✓	✓	✓	✓
2	Call my wife.	21.2	✓	✓	✓	✓	✓	✓	12	What's my ETA?	20.7	✓	✓	✓	✓	✓	✓
3	Call Bob.	20.3	✓	✓	✓	✓	✓	✓	13	Open the garage door.	21.5	✓	✓	✓	✓	✓	✓
4	Open Gmail.	19.8	✓	✓	✓	✓	✓	✓	14	Turn on the lights.	19.8	✓	✓	✓	✓	✓	✓
5	Open WhatsApp.	20.3	✓	✓	✓	✓	✓	✓	15	Turn off all alarms.	20.7	✓	✓	✓	✓	✓	✓
6	Open Paypal.	22.3	✓	✓	✓	✓	✓	✓	16	Send a message to...	19.2	✓	✓	✓	✓	✓	✓
7	Check my voicemail.	19.8	✓	✓	✓	✓	✓	✓	17	Send a reply email to...	18.8	✓	✓	✓	✓	✓	✓
8	Check my emails.	20.7	✓	✓	✓	✓	✓	✓	18	Tell Bob where I am.	20.3	✓	✓	✓	✓	✓	✓
9	Check my wallet.	18.5	✓	✓	✓	✓	✓	✓	19	Did I lock the front door?	21.3	✓	✓	✓	✓	✓	✓
10	What's my name?	21.2	✓	✓	✓	✓	✓	✓	20	What's my next schedule?	19.5	✓	✓	✓	✓	✓	✓

Table 1: Effectiveness of launching inaudible audio injection attacks via XPorter. We test voice commands with different SNR values and conduct 20 trials of end-to-end attacks, including the activation (Act.) and injection (Inj.). (✓/✗: success/ fail).

0.5s. Then, we use 80% data samples to train the proposed CNN-LSTM classifier for determining each input key of the unlocking password and the remaining 20% data samples to evaluate the performance of the trained model with 10-fold cross-validation. Figure 8a shows the confusion matrix of the evaluation results, where XPorter achieves 98.8% accuracy in recognizing the ten passcode pins (from 0 to 9) on the unlocking screen. As such, the attacker can precisely detect the victim's unlocking password and then unlock the victim's smartphone to steal more user privacy when the victim's smartphone is left by charging.

5.1.3 Effectiveness of App Fingerprinting. To evaluate the effectiveness of XPorter in recognizing mobile app activities, we follow the same data collection procedure and record traces when the in-charging smartphone launches different mobile apps. Specifically, we select 20 most popular mobile apps and launch each of them for 50 times and obtain the first one-second voltages as the data samples for app fingerprinting. Similarly, we also utilize 80% data samples to train the classifier and the rest of 20% data for evaluating the model performance. Figure 8b shows the confusion matrix of the evaluation results, where XPorter presents an overall accuracy of 88.7% in fingerprinting 20 popular mobile apps.

Moreover, we find XPorter performs the best in recognizing apps such as Twitch and WeChat that have distinguishable voltage patterns due to their customized launching animations that result in more energy consumption, which induces distinctive patterns of the voltage signals. On the contrary, XPorter performs the worst in recognizing apps like Spotify (72%) and Messenger (78%) because they adopt the default app launching setup (*i.e.*, white background with a static icon) and consume the lowest energy consumption as they have fewer network requirements and screen animations. Therefore, the changes in the voltage incurred by app launching are milder than other apps, which further impacts the performance of XPorter in recognizing these apps. Nevertheless, XPorter still demonstrates high accuracy in detecting the app usage information of the victim during the charging process stealthily, especially apps containing sensitive information, e.g., Facebook and WhatsApp contain the contact and address information of the users.

5.1.4 Effectiveness of Keystroke Recovery. To achieve more fine-grained eavesdropping attacks, we also evaluate XPorter in recovering input keystrokes. Specifically, we collect data samples by typing the keys on the QWERTY full-size keyboard and repeating each key for 100 times, including 26 alphabetic keys from "a" to "z". Likewise, 80% of the collected data samples are used to build the CNN-LSTM classifier for recognizing keys, and 20% data samples are used to evaluate the model's effectiveness. Figure 8c shows the confusion matrix of the evaluation results, where XPorter achieves overall 83.0% accuracy in recognizing 26 alphabetic keys (from "a" to "z") on a full-size QWERTY keyboard. In particular, we find most misclassification always happens in two neighbor alphabetic keys, *e.g.*, nearly 11% testing samples are misclassified in recognizing keys "u" (63%) and "i" (71%) as the voltage patterns incurred by these key-pressing events are close. On the other hand, XPorter can detect keys on edge with high accuracy rates, such as "q" (100%), "a" (96%), and "z" (98%) that present distinctive patterns because they have fewer neighbor keys. In short, XPorter has demonstrated the ability to infer the victim's keystrokes through the voltage leakage in the charging process, which may contain fine-grained user privacy such as the conversation in chatting apps like WhatsApp, the password for payment in financial apps like PayPal.

5.2 Effectiveness of Audio Injection Attack

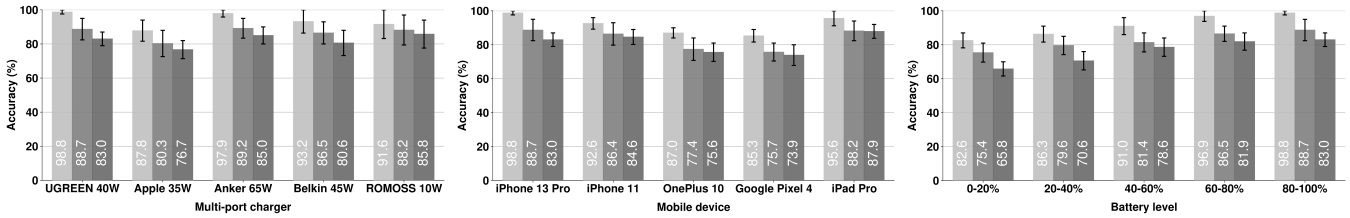
5.2.1 Experiment Setup. To evaluate the effectiveness of the audio injection attack, we compromised the UGREEN 40W USB-C port charger by connecting the audio pins of its two USB-C ports together. We also use one port to charge an iPhone 13 Pro correspondingly as the victim's smartphone and then plug the attacking device into another port. Since the attacking device integrates a Bluetooth module for communication, we conduct the evaluation process by controlling the attacking device to activate the voice assistant and inject different modulated audio commands at a non-line-of-sight (NLOS) distance of 5 m.

5.2.2 Effectiveness of Voice Assistant Activation. We conduct experiments on smartphones with different voice assistants to demonstrate the XPorter's ability to activate the smartphone's voice assistant while bypassing the speech verification system. Specifically, we utilize three smartphones (iPhone 13 Pro, Google Pixel 4, and OnePlus 10 Pro) with different commodity voice assistant systems (Siri, Google Assistant, and Breeno) by plugging them into the compromised charger and then activating each voice assistant for 50 times. Meanwhile, we record the response time of each trial as well as 50 trials of the response time of activating these voice assistants by speaking hotwords such as "Hey Siri", "Hello Google", and "Hey Breeno". Figure 9 shows the box plot of the response time of the three voice assistants and the human speaking, and we know it takes an average of 2.07, 2.18, and 2.05 seconds to activate Siri, Google Assistant, and Breeno through XPorter, respectively. On the other hand, it only needs approximately 1.04 seconds to activate voice assistants by human speaking. Even though more time is required to activate the voice assistant, XPorter can bypass the speech verification mechanisms that have been widely deployed in commodity mobile devices, which makes XPorter more practical in a real-world scenario. Moreover, as the injected audio commands are voltage signals, XPorter cannot be detected and countered by existing defense approaches [1, 19, 40] that are proposed to defend against inaudible audio injections through acoustic signals.

5.2.3 Effectiveness of Audio Commands Injection. To evaluate the effectiveness of inaudible voice commands injection attacks through XPorter, we exploit the Google WaveNet API [27] to generate 20 voice commands that have been widely used with high frequency in a quiet environment (SNR $\leq$25 dB), and each of those voice commands is a sentence that contains 2–10 words. Then, we activate the aforementioned three voice assistants (Siri, Google Assistant, and Breeno) using the proposed method and then inject each voice command into them. Once a voice assistant receives the voice commands and provides corresponding feedback, we consider it as one successful attack trial. Table 1 shows the detailed results of the 20 trials of end-to-end inaudible audio injection attacks on the three voice assistants. In all end-to-end attack trials, XPorter achieves 100% success rate in activating the three voice assistants, and 100% success rate in injecting different voice commands to compromise user privacy. Therefore, XPorter shows competitive performance compared to other state-of-the-art inaudible voice injection attacks [37, 41] and fills the gap between the eavesdropping and the injection attacks via a multi-port charger.

5.3 Impact of Practical Factors

5.3.1 Impact of different multi-port chargers. Due to the variety of different multi-port chargers' circuits, the induced voltage leakage presents different patterns. Thus, to evaluate whether XPorter can be launched to other multi-port chargers, we conduct further experiments by separately collecting data and training models from four other different commodity multi-port chargers: Apple 35W USB-C compact charger (A2579), Anker 65W smart charger (A2668), Belkin 65W USB-C charger (WCH013), and ROMOSS 2.1A USB-A charger. Figure 10a shows the evaluation results of launching eavesdropping attacks on the five multi-port chargers, where we find XPorter achieves high eavesdropping accuracy across different multi-port chargers, *e.g.*, 93.9% in inferring unlocking passcode, 86.6% in recognizing app launching, and 82.2% uncovering keystrokes. In particular, the results show that



(a) Different multi-port chargers. (b) Different mobile devices. (c) Different battery levels.

Figure 10: Evaluation results of three practical impact factors on the eavesdropping attacks of XPorter: (a) Impact of different commodity multi-port chargers, (b) Impact of different mobile devices, (c) Impact of different battery levels of the in-charging device. – Unlocking passcode inference, – App recognition, – Keystroke recovery.

XPorter shows a relatively lower eavesdropping accuracy when applying on the Apple 35W USB-C charger since it presents a relatively high voltage ripple [6] in the charging process so that the voltage changes induced by user activities are overwhelmed. However, the results demonstrate that the voltage leakage is a fundamental design flaw existing in different multi-port chargers, and XPorter presents a promising performance in inferring fine-grained user privacy across different commodity multi-port chargers.

5.3.2 Impact of different mobile devices. We use five commodity devices, including four smartphones (iPhone 13 Pro, iPhone 11, OnePlus 10 Pro, and Google Pixel 4) and one tablet (iPad Pro 2019), to evaluate the impact of different mobile devices. Figure 10b shows the results of launching the eavesdropping attacks on different in-charging devices, where we find XPorter achieves the highest accuracy in inferring privacy from the iPhone 13 Pro and the iPad Pro but the lowest accuracy in smartphones like the OnePlus 10 Pro and the Google Pixel 4. Because user interactions (*e.g.*, launching apps or pressing keys) with an iPad Pro require more energy consumption due to the large touchscreen and UI components, which induces stronger voltage changes in the charger and voltage leakage. Nevertheless, XPorter can be scaled to different mobile devices with average accuracy rates of 91.9%, 83.3%, and 81.0% to recognize the unlocking passcode, the running app, and the keystrokes, respectively.

5.3.3 Impact of different battery levels. In practice, the mobile device may have different battery levels when being plugged into the port for charging the battery. To evaluate the impact of different battery levels on the performance of XPorter, we follow the same procedure and conduct experiments when the iPhone 13 Pro is at five different battery levels: 0–20%, 20–40%, 40–60%, 60–80%, and 80–100%, and Figure 10c shows the experimental results of inferring the three user activities. Specifically, we know when the in-charging mobile device is at a high battery level (*e.g.*, $\geq$ 60%), XPorter's performance of the eavesdropping attack is approximately 15% higher than the lower battery levels (*e.g.*, $\leq$ 40%). Because most of the output voltage of the plugged USB port is used for charging the battery when the device is at a low battery percentage. As such, when the battery is at a low level, the voltage changes induced by user activities could be overwhelmed by the intensive charging voltage. By contrast, when the battery reaches a high level, the charging process slows down, and the charging voltage is constant so that the voltage changes incurred by various user activities would present more distinctive patterns [9, 17, 38]. Despite the impact caused by different battery levels of the charging device, XPorter still achieves an overall accuracy of 91.1%, 82.3%, and 76.0% in inferring the unlocking passcode, the running app, and the keystrokes at the five battery levels.

Table 2: Evaluation of inaudible audio injection attacks with different impact factors' combinations. Act. SR.: activation success rate. Inj. SR.: injection success rate.

Multi-port Charger	# of Ports	Type of Ports	Mobile Device	Voice Assistant	Battery Level	Act. SR.	Inj. SR.
UGREEN 40 W	2	2× USB-C	iPhone 13 Pro		80-100%	100%	100%
Anker 65W	3	1× USB-A 2× USB-C	iPhone 13 Pro		40-60%	100%	100%
Belkin 65W	2	2× USB-C	iPhone 13 Pro		60-80%	100%	100%
UGREEN 40 W	2	2× USB-C	Google Pixel 4		20-40%	100%	100%
Anker 65W	3	1× USB-A 2× USB-C	Google Pixel 4		60-80%	100%	100%
Belkin 65W	2	2× USB-C	Google Pixel 4		0-20%	100%	100%
UGREEN 40 W	2	2× USB-C	OnePlus 10 Pro		80-100%	100%	100%
Anker 65W	3	1× USB-A 2× USB-C	OnePlus 10 Pro		60-80%	100%	100%
Belkin 65W	2	2× USB-C	OnePlus 10 Pro		0-20%	100%	100%
UGREEN 40 W	2	2× USB-C	iPad Pro		60-80%	100%	100%
Anker 65W	3	1× USB-A 2× USB-C	iPad Pro		80-100%	100%	100%
Belkin 65W	2	2× USB-C	iPad Pro		20-40%	100%	100%

5.3.4 Impact factors on audio injection attacks. Table 2 is the evaluation results of 12 end-to-end inaudible audio injection attacks with combinations of different impact factors. The results indicate that XPorter achieves 100% success rate in activating voice assistants and 100% success rate in injecting various voice commands across different multi-port chargers and mobile devices with different battery levels. Therefore, XPorter is resilient to the three practical factors in launching the injection attacks and realizes a high success rate.

6 DISCUSSION

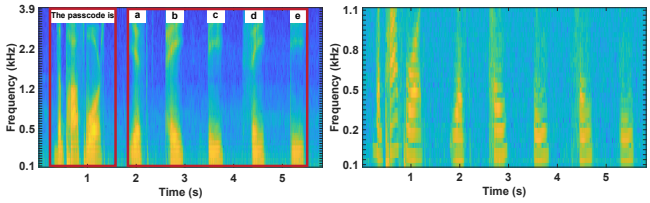
6.1 Extending Attacks

Eavesdropping audio through the voltage leakage. The audio pins of a USB-C port also support audio output, allowing for the acquisition of audio data through the analysis of charging power patterns [37]. Therefore, XPorter can be extended to obtain the voltage leakage from the audio output pins of the USB-C port so that the attacker can further spy on private information such as sensitive conversations in a phone call and secret messages in voice mails. Figure 11a and Figure 11b individually present the spectrograms of the original audio conversations and the obtained voltage output after applying the demodulation methods through XPorter of the voice mail "The passcode is abcde", where we can also find similar patterns that contain sensitive information are presented in the voltage signals. Hence, the attacker can also exploit the voltage leakage as shown in XPorter to uncover the conversation content in a more stealthy way.

Attacks on multiple victims. To explore the feasibility of attacking multiple victims, we leverage the Anker 65W smart charger (2×USB-C, 1×USB-A) to charge two iPhone 13 Pro, and play the two charging smartphones simultaneously (*e.g.*, launching two different apps) while recording the voltage leakages from the neighbor USB-A port. Then, since the voltage leakage is a one-dimensional signal, we apply the blind source separation method (*e.g.*, FastICA [25]) to separate the mixed voltage signal into individual signals to determine the activities of each victim. Figure 12a shows the process of separating the mixed voltage leakage (SNR=11.4 dB) to individual voltage signals when launching WhatsApp (SNR=10.7 dB) and YouTube (SNR=10.3 dB) on the two charging smartphones, respectively. We then conduct extensive experiments to evaluate the effectiveness of eavesdropping on two victims, and Figure 12b shows the results. The accuracy decreases by approximately 5.3–10.5% due to the increase of noise in the individual signals after the source separation, but XPorter still achieves acceptable accuracy in uncovering different user activities. In addition, it is also feasible to launch multi-victim audio injection attacks by connecting all the audio pins of USB-C ports together in the compromised multi-port charger. In this case, the attacker can activate voice assistants and inject malicious voice commands into multiple charging devices simultaneously.

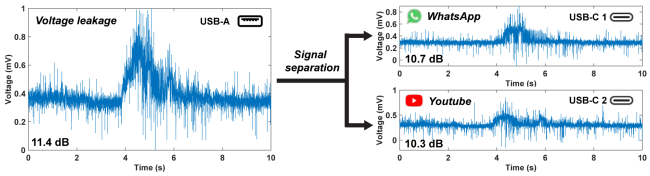
6.2 Countermeasures

Software-based countermeasures. To prevent the audio injection attacks from XPorter, one software-based solution is to disable the audio transmission function through the system-level API [10] so that the voice control system cannot detect the voice commands. In addition, since the eavesdropping attacks depend on the captured voltage signals, we can

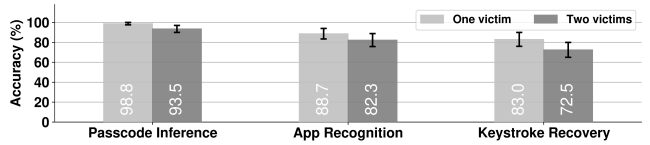


(a) Original audio. (b) Demodulated voltage.

Figure 11: Spectrograms of the original audio and the demodulated voltage signal of eavesdropping voice mail "The passcode is abcde" through the USB-C interface.



(a) Signal separation in the two-victim scenario.



(b) Effectiveness of eavesdropping two victims.

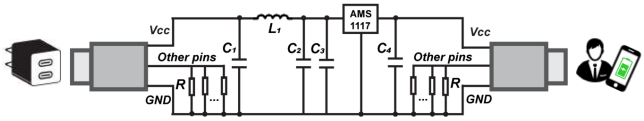
Figure 12: Evaluation of attacking multiple victims.

add random noise (*e.g.*, dummy traffic packets [18, 25]) in the services to introduce extra power consumption to obfuscate the voltage traces without influencing the user experience. However, these methods inevitably bring extra energy consumption and may impact the charging efficiency.

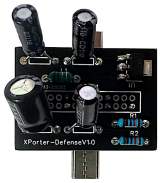
Hardware-based countermeasures. The straightforward way to mitigate the inferences and injections from XPorter is to eliminate the voltage leakages in multi-port chargers. Hence, we can connect a physical peripheral between the multi-port charger and the charging devices to smooth out the voltage leakages. For instance, we implement a simple circuit prototype as shown in Figure 13a and Figure 13b with resistors R =10 kΩ, capacitors C_1 =10 μF, C_2 =1 μF, C_3 =100 μF, C_4 =22 μF, and inductor L_1 =0.1 H, and an AMS1117 low-dropout regulator [31]. Figure 13c shows it can smooth the voltage patterns induced by smartphone activities so that the attacker cannot exploit the voltage leakages to infer user privacy through XPorter. Another method is that the manufacturer could redesign the hardware by modifying the parallel connection mechanism so that the voltage change of one port cannot induce changes on other neighbor ports. Nevertheless, redesigning the hardware circuits can be a costly endeavor and is not feasible for sold multi-port chargers. Even if hardware modifications are made, it is still ambiguous how users would ascertain whether or not a multi-port charger could be trusted. Therefore, raising public awareness and educating users about the threat of untrusted multi-port chargers is a more effective and economical solution to prevent attacks.

Table 3: Quantified comparison with related works via charging devices. "●": yes, "○": no, Acc.: classification accuracy, SR.: injection success rate, UK: unlocking keyboard, FK: full-size QWERTY keyboard, NA: not available or evaluated.

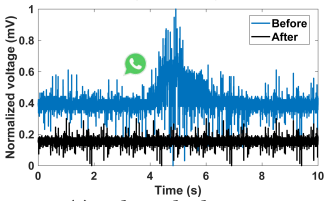
Related works	Target device	No need to compromise		Eavesdropping attacks (Acc.)			Audio injection attacks (SR.)	Potential of attacking multiple victims
		Eavesdropping	Injection	App/Web	Keystroke (UK/FK)	Speech		
Charger-Surfing [9]	USB cable	○	○	○	● (98.7%/NA)	○	○	○
GhostTalk [37]	USB cable	○	○	○	○	● (93.3%)	● (100%)	○
EM-Surfing [20]	Power line of a wireless charger	○	○	● (95.0%)	● (98.3%/96.4%)	● (81.0%)	○	○
Cour *et al.* [17]	Power line of a wireless charger	○	○	● (91.5%)	○	○	○	○
XPorter (Our method)	Multi-port charger	●	○	● (88.7%)	● (98.8%/83.0%)	● (NA)	● (100%)	●



(a) A simple circuit to smooth out voltage leakages.



(b) Prototype.



(c) Voltage leakages.

Figure 13: Defend against XPorter via a simple circuit to smooth out voltage leakages.

7 RELATED WORKS

Attacks via charging devices. In Table 3, we summarize the quantified comparisons between XPorter and other state-of-the-art attacks via peripheral charging devices, *i.e.*, USB cables [9, 37], wireless chargers [17, 20]. In particular, XPorter can launch eavesdropping attacks without compromising devices, but it tampers chargers for audio injection. It is the first work to explore the essential design drawback of a popular charging interface, the multi-port chargers, to investigate their eavesdropping and voice injection vulnerabilities. In particular, XPorter outperforms these works in three-folds: *(i)* Unlike prior works that need to compromise USB cables [9, 37] or chargers [17, 20] to launch attacks, XPorter has no need to compromise victim devices to achieve fine-grained eavesdropping attacks that loosen the assumptions of attackers' ability in [9, 17, 20]. It also reduces the attack efforts to inject malicious voice commands than the prior work [37] because it needs no extra hardware component to be hidden in victims' devices or special USB cable as we have integrated all modules in the custom-built attacking device; *(ii)* XPorter is an orthogonal attack framework that can launch both eavesdropping attacks and inaudible voice injections through a single attack surface of the new charging platform; and *(iii)* XPorter presents the potential of attacking multiple charging devices simultaneously as we have demonstrated in §6.1.

Attacks via other power traces. The power consumption of a smartphone's battery can also be used to infer user privacy [5, 22, 24, 26, 28]. That is, an attacker can use pre-installed malware to obtain the battery profile of the victim's smartphone and further uncover user privacy. For instance, POWERFUL [8] exploits the smartphone's battery consumption data to recognize mobile app usage and activities. PowerSpy [23] uses two battery profiles in Android smartphones (*voltage_now* and *current_now*) to determine the motion of the smartphone for tracking the user's location. Furthermore, AppListener [25] leverages RF energy harvesting to capture the emitted RF energy of a Wi-Fi router to recognize fine-grained mobile app activities of a connected smartphone.

8 CONCLUSION

In this paper, we present a new attack vector for eavesdropping on user privacy and inaudibly injecting voice commands through a commodity multi-port charger. To validate its feasibility and practicality, we design and implement XPorter, an attack framework that leverages the voltage leakage of the neighbor ports to infer sensitive information and exploits the USB-C interface to activate voice assistant and inject modulated voice commands into the victim's charging smartphone across the multi-port interface. Our extensive evaluation demonstrates that XPorter is effective in inferring fine-grained user privacy and also achieves 100% success rate in launching inaudible audio injection attacks across various impact factors such as different multi-port chargers and mobile devices. We hope our finds can raise public awareness of the vulnerability of multi-port chargers and spur research on detecting forthcoming attacks and new defense methods.

ACKNOWLEDGMENT

We sincerely thank our shepherd and all anonymous reviewers for their constructive feedback. This work was supported by CityU APRC grant 9610563, the Research Grants Council of Hong Kong (CityU 21219223, C1029-22G, CityU 21201420, CityU 11201422), CCF-NSFOCUS Kunpeng Fund, NSFC Young Scientists Fund (No. 62002306), and NSFC (No. 62101471), NSF of Shandong province (No. ZR2021LZH010), and Shenzhen Science and Technology Funding Fundamental Research Program (2021Szvup126). Any opinions, findings, and conclusions in this paper are those of the authors and do not necessarily of supported organizations.

REFERENCES

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

[2] AliExpress. 2022. New Original Replacement Wire control board volume button Pcb for Pb3 Powerbeat Earphone. (2022). https://www.aliexpress.com/item/1005003525462944.html.

[3] Apple. 2022. HomeKit Accessories. (2022). https://support.apple.com/en-us/HT208939.

[4] BOONE ASHWORTH. 2022. A New EU Law Would Force iPhones to Adopt USB-C Charging. (2022). https://www.wired.com/story/eu-law-usb-c-iphones-lightning/.

[5] Niels Brouwers, Marco Zuniga, and Koen Langendoen. 2014. Neat: A novel energy analysis toolkit for free-roaming smartphones. In *Proceedings of the 12th ACM Conference on Embedded Network Sensor Systems (SenSys)*. 16–30.

[6] ChargerLAB. 2022. Review of Apple 35W Dual USB-C Compact Power Adapter. (2022). https://www.youtube.com/watch?v=aHdZu-m9y64.

[7] Jin Chen, Per Jönsson, Masayuki Tamura, Zhihui Gu, Bunkei Matsushita, and Lars Eklundh. 2004. A simple method for reconstructing a high-quality NDVI time-series data set based on the Savitzky–Golay filter. *Remote sensing of Environment* 91, 3-4 (2004), 332–344.

[8] Yimin Chen, Xiaocong Jin, Jingchao Sun, Rui Zhang, and Yanchao Zhang. 2017. POWERFUL: Mobile app fingerprinting via power analysis. In *Proceedings of the International Conference on Computer Communications (INFOCOM)*. IEEE, 1–9.

[9] Patrick Cronin, Xing Gao, Chengmo Yang, and Haining Wang. 2021. Charger-Surfing: Exploiting a Power Line Side-Channel for Smartphone Information Leakage. In *Proceedings of the 30th USENIX Security Symposium*. 681–698.

[10] Android Developer. 2022. Documentation of Manifest Permission. (2022). https://developer.android.com/reference/android/Manifest.permission#MODIFY_AUDIO_SETTINGS.

[11] DROK. 2022. BlueTooth Board, DROK 12V Audio Receiver BlueTooth. (2022). https://www.amazon.com/Bluetooth-DROK-Receiver-Electronics-Headphone/dp/B07P94Z9XR.

[12] FactMR. 2022. USB Wall Charger Market. (2022). https://www.factmr.com/report/2471/usb-wall-charger-market.

[13] Federico Griscioli, Maurizio Pizzonia, and Marco Sacchetti. 2016. USBCheckIn: Preventing BadUSB attacks by forcing human-device interaction. In *Proceedings of the 14th Annual Conference on Privacy, Security and Trust (PST)*. IEEE, 493–496.

[14] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation* 9, 8 (1997), 1735–1780.

[15] Hassan Ismail Fawaz, Germain Forestier, Jonathan Weber, Lhassane Idoumghar, and Pierre-Alain Muller. 2019. Deep learning for time series classification: a review. *Data mining and knowledge discovery* 33, 4 (2019), 917–963.

[16] David M Kreindler and Charles J Lumsden. 2006. The Effects of the Irregular Sample and Missing Data in Time Series Analysis. *Nonlinear dynamics, psychology, and life sciences* (2006).

[17] Alexander S La Cour, Khurram K Afridi, and G Edward Suh. 2021. Wireless charging power side-channel attacks. In *Proceedings of the ACM SIGSAC Conference on Computer and Communications Security (CCS)*. 651–665.

[18] Jianfeng Li, Hao Zhou, Shuohan Wu, Xiapu Luo, Ting Wang, Xian Zhan, and Xiaobo Ma. 2022. FOAP: Fine-Grained Open-World Android App Fingerprinting. In *Proceedings of the 31st USENIX Security Symposium*.

[19] Zhuohang Li, Cong Shi, Tianfang Zhang, Yi Xie, Jian Liu, Bo Yuan, and Yingying Chen. 2021. Robust detection of machine-induced audio attacks in intelligent audio systems with microphone array. In *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security (CCS)*. 1884–1899.

[20] Jianwei Liu, Xiang Zou, Leqi Zhao, Yusheng Tao, Sideng Hu, Jinsong Han, and Kui Ren. 2022. Privacy Leakage in Wireless Charging. *IEEE Transactions on Dependable and Secure Computing* (2022).

[21] Tiantian Liu, Feng Lin, Zhangsen Wang, Chao Wang, Zhongjie Ba, Li Lu, Wenyao Xu, and Kui Ren. 2023. MagBackdoor: Beware of Your Loudspeaker as a Backdoor for Magnetic Injection Attacks. In *Proceedings of the 44th IEEE Symposium on Security and Privacy (SP)*. IEEE Computer Society, 3416–3431.

[22] Xiao Ma, Peng Huang, Xinxin Jin, Pei Wang, Soyeon Park, Dongcai Shen, Yuanyuan Zhou, Lawrence K Saul, and Geoffrey M Voelker. 2013. eDoctor: Automatically Diagnosing Abnormal Battery Drain Issues on Smartphones. In *Proceedings of the 10th USENIX Symposium on Networked Systems Design and Implementation (NSDI)*. 57–70.

[23] Yan Michalevsky, Aaron Schulman, Gunaa Arumugam Veerapandian, Dan Boneh, and Gabi Nakibly. 2015. PowerSpy: Location Tracking Using Mobile Device Power Analysis. In *Proceedings of the 24th USENIX Security Symposium*. 785–800.

[24] Tao Ni, Yongliang Chen, Keqi Song, and Weitao Xu. 2021. A Simple and Fast Human Activity Recognition System Using Radio Frequency Energy Harvesting. In *Adjunct Proceedings of the 2021 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2021 ACM International Symposium on Wearable Computers*. 666–671.

[25] Tao Ni, Guohao Lan, Jia Wang, Qingchuan Zhao, and Weitao Xu. 2023. Eavesdropping Mobile App Activity via Radio-Frequency Energy Harvesting. In *Proceedings of the 32nd USENIX Security Symposium*.

[26] Tao Ni, Xiaokuan Zhang, Chaoshun Zuo, Jianfeng Li, Zhenyu Yan, Wubing Wang, Weitao Xu, Xiapu Luo, and Qingchuan Zhao. 2022. Uncovering User Interactions on Smartphones via Contactless Wireless Charging Side Channels. In *Proceedings of the 44th IEEE Symposium on Security and Privacy (SP)*. IEEE Computer Society, 1828–1844.

[27] Aaron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew Senior, and Koray Kavukcuoglu. 2016. WaveNet: A generative model for raw audio. *arXiv preprint arXiv:1609.03499* (2016).

[28] Abhinav Pathak, Y Charlie Hu, and Ming Zhang. 2012. Where is the energy spent inside my app? Fine Grained Energy Accounting on Smartphones with Eprof. In *Proceedings of the 7th ACM European Conference on Computer Systems*. 29–42.

[29] ProtoSupplies. 2022. AD620 Instrumentation Amplifier Module. (2022). https://protosupplies.com/product/ad620-instrumentation-amplifier-module/.

[30] Pavel Senin. 2008. Dynamic time warping algorithm review. *Information and Computer Science Department University of Hawaii at Manoa Honolulu, USA* 855, 1-23 (2008), 40.

[31] SHIKUES. 2022. AMS1117 1A Bipolar Linear Regulator. (2022). https://datasheet.lcsc.com/szlcsc/2001081204_Shikues-AMS1117-1-2_C475600.pdf.

[32] Yang Su, Daniel Genkin, Damith Ranasinghe, and Yuval Yarom. 2017. USB snooping made easy: crosstalk leakage attacks on USB hubs. In *Proceedings of the 26th USENIX Security Symposium*. 1145–1161.

[33] Yichuang Sun and JK Fidler. 1996. Design method for impedance matching networks. *IEE Proceedings-Circuits, Devices and Systems* 143, 4 (1996), 186–194.

[34] TensorSpeech. 2021. Real-Time State-of-the-art Speech Synthesis for Tensorflow 2. (2021). https://github.com/TensorSpeech/TensorflowTTS.

[35] Jing Tian, Nolen Scaife, Deepak Kumar, Michael Bailey, Adam Bates, and Kevin Butler. 2018. SoK:" Plug & Pray" Today–Understanding USB Insecurity in Versions 1 Through C. In *Proceedings of the IEEE Symposium on Security and Privacy (SP)*. IEEE, 1032–1047.

[36] Christopher Tralie and Elizabeth Dempsey. 2020. Exact, parallelizable dynamic time warping alignment with linear memory. *arXiv preprint arXiv:2008.02734* (2020).

[37] Yuanda Wang, Hanqing Guo, and Qiben Yan. 2022. GhostTalk: Interactive Attack on Smartphone Voice System Through Power Line. In *Proceedings of the Network and Distributed System Security Symposium (NDSS)*.

[38] Yi Wu, Zhuohang Li, Nicholas Van Nostrand, and Jian Liu. 2021. Time to Rethink the Design of Qi Standard? Security and Privacy Vulnerability Analysis of Qi Wireless Charging. In *Proceedings of the Annual Computer Security Applications Conference (ACSAC)*. 916–929.

[39] Qing Yang, Paolo Gasti, Kiran Balagani, Yantao Li, and Gang Zhou. 2018. USB side-channel attack on Tor. *Computer Networks* (2018).

[40] Guoming Zhang, Xiaoyu Ji, Xinfeng Li, Gang Qu, and Wenyuan Xu. 2021. EarArray: Defending against DolphinAttack via Acoustic Attenuation.. In *Proceedings of the Network and Distributed System Security Symposium (NDSS)*.

[41] Guoming Zhang, Chen Yan, Xiaoyu Ji, Tianchen Zhang, Taimin Zhang, and Wenyuan Xu. 2017. Dolphinattack: Inaudible voice commands. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security (CCS)*. 103–117.