

When Good Becomes Evil: Exploring Crosstalk Attack Surfaces on Multi-Port USB Chargers

Tao Ni^{ID}, Member, IEEE, Zehua Sun^{ID}, Student Member, IEEE, Yongliang Chen^{ID}, Member, IEEE, Yihe Zhou^{ID}, Student Member, IEEE, Jiayimei Wang^{ID}, Student Member, IEEE, Weitao Xu^{ID}, Senior Member, IEEE, Qingchuan Zhao^{ID}, Member, IEEE, and Cong Wang^{ID}, Fellow, IEEE

Abstract—Multi-port chargers, designed to simultaneously charge multiple mobile devices such as smartphones, have gained significant popularity, with millions of units sold in recent years. However, this multi-device charging feature introduces security and privacy risks. If not properly designed and implemented, these chargers can enable communication between connected devices because they are inherently interconnected, which leads to crosstalk voltage leakages. Despite their widespread use, these risks have not been thoroughly investigated. We have identified novel attack surfaces in the circuit design of multi-port chargers that allow an adversary who shares the multi-port charger with the target victim in close proximity to exploit one port to (i) recognize fine-grained user activities of other devices being charged, (ii) eavesdrop on secret audio transmission from USB-C audio pins, and (iii) inject malicious audio commands into built-in voice assistants of charging devices (e.g., Siri, Google Assistant). In this paper, we design and implement XPORTHEFT, a novel system to analyze and demonstrate the uncovered security and privacy threats in multi-port chargers. Specifically, it leverages changes in voltage signals in one neighbor port to monitor voltage changes in the charging port induced by user activities in various user interfaces, such as recognizing running apps and detecting keystrokes. Moreover, XPORTHEFT can also achieve audio transmission eavesdropping and launch inaudible audio injection attacks from the neighbor port to the charging mobile device via the USB-C interface. We extensively evaluate the effectiveness of XPORTHEFT using five commercial multi-port chargers and five mobile devices. The evaluation results show its high effectiveness in recognizing the launch of 20 mobile apps (88.7%) and revealing unlocking passcodes (98.8%), as well as eavesdropping on the audios of numeric digits (97.1%) and alphabetic characters (98.0%). Furthermore, XPORTHEFT achieves

100% success rates in inaudible audio injection attacks on three commercial voice assistants. In addition, our study also shows that XPORTHEFT is resilient to various impact factors and presents the potential to attack multiple victims.

Index Terms—Crosstalk voltage leakage, multi-port chargers, eavesdropping attacks, inaudible audio injection attacks.

I. INTRODUCTION

THE rapid proliferation of mobile devices, such as smartphones and tablets, has driven the development of various battery charging accessories, with the market for these products projected to reach approximately 3.09 billion USD by the end of 2033 [1]. Among these accessories, multi-port chargers have become a prominent choice, offering multiple ports (e.g., two or more USB-C/USB-A ports) to support charging several mobile devices simultaneously. Over the past five years, multi-port chargers have gained significant popularity, fueled by the growing demand for charging multiple devices with varying charging specifications. For instance, Fig. 1 illustrates four typical real-world scenarios showcasing the use of commercial off-the-shelf (COTS) multi-port chargers.

However, the multi-device charging feature of these well-designed multi-port chargers introduces a significant attack surface, enabling one device to perform malicious actions on other devices charging simultaneously. This vulnerability arises from the fundamental design of multi-port chargers, where all charging ports are connected in parallel and share the same voltage. As a result, any voltage fluctuation in one port can propagate to other parallel-connected ports, creating opportunities for attacks that can eavesdrop or inject voice commands into other connected devices. Previous studies have shown that voltage changes in a charging mobile device can reveal sensitive information, such as pressing buttons, unlocking screen keystrokes, and running smartphone applications [2], [3], [4], [5]. In addition, these voltage fluctuations (*a.k.a.*, crosstalk voltage leakage) can be exploited to manipulate the voice assistant of a charging device, enabling the injection of malicious voice commands, which can result in misinterpretation of information or unintended actions [6].

Unfortunately, these severe security and privacy risks associated with multi-port chargers have largely been overlooked. One possible reason for this neglect is the perception that multi-port chargers are immune to such threats since they are not primarily designed for data transfer—a critical

Received 19 December 2024; revised 9 June 2025; accepted 3 July 2025. Date of publication 8 July 2025; date of current version 5 November 2025. This work was supported in part by the Research Grants Council (RGC) of Hong Kong SAR under Grant CityU 11202124, Grant 21219223, Grant 11218521, Grant 11218322, Grant R6021-20F, Grant R1012-21, Grant RFS2122-1S04, Grant C2004-21G, Grant C1029-22G, Grant C6015-23G, and Grant N_CityU139/21, in part by the Innovation and Technology Commission of Hong Kong (ITC) under Mainland-Hong Kong Joint Funding Scheme (MHKJFS) under Grant MHP/135/23, in part by the InnoHK initiative, The Government of the HK-SAR, and in part by the Laboratory for AI-Powered Financial Technologies (AIFT). Recommended for acceptance by A. A. Nayak. (*Corresponding authors:* Qingchuan Zhao; Cong Wang.)

This work involved human subjects or animals in its research. Approval of all ethical and experimental procedures and protocols was granted by the Human Subjects Ethics Sub-Committee of City University of Hong Kong under Application No. HU-STA-00000169, and performed in line with the IEEE Code of Ethics.

The authors are with the Department of Computer Science, City University of Hong Kong, Hong Kong 123456 Hong Kong (e-mail: taoni2@cityu.edu.hk; zehuasun2-c@my.cityu.edu.hk; yonglchen5@cityu.edu.hk; yihezhou2@cityu.edu.hk; jwang2664@cityu.edu.hk; weitaoxu@cityu.edu.hk; cs.qczhao@cityu.edu.hk; congwang@cityu.edu.hk).

Digital Object Identifier 10.1109/TMC.2025.3587292

1536-1233 © 2025 IEEE. All rights reserved, including rights for text and data mining, and training of artificial intelligence and similar technologies. Personal use is permitted, but republication/redistribution requires IEEE permission. See <https://www.ieee.org/publications/rights/index.html> for more information.



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

attack surface commonly exploited for eavesdropping and voice command injection attacks on other devices (e.g., USB hubs [7], [8], [9]). To fill in this gap, we aim to investigate three typical side-channel attacks when the adversary shares the same multi-port charger with the target victim in close proximity: (i) eavesdropping on user activities at the User-Interface (UI) level, (ii) stealing sensitive audio information, and (iii) launching inaudible audio injection attacks. This analysis serves as a crucial first step in uncovering the previously overlooked threats posed by multi-port chargers and contributes to the development of improved security measures for these devices.

We design and implement XPORTHEFT, a novel attack system to investigate eavesdropping and audio injection attacks arising from communication across charging ports of a multi-port USB charger. For eavesdropping attacks, XPORTHEFT detects the leakage of voltage signals from neighboring ports, processes the signals to extract informative voltage clips, and trains models to recognize user activities, ultimately inferring sensitive information from other connected devices. For audio eavesdropping and inaudible audio injection attacks, XPORTHEFT exploits the crosstalk voltage leakage of the audio pins on the USB-C charging interface to reconstruct audio transmissions, activate the voice assistant on the target device while bypassing its speech verification system, then injects malicious voice commands via a compromised multi-port charger.

We have implemented XPORTHEFT with a custom-built attacking device to demonstrate the feasibility of the three attacks mentioned above. As a proof of concept, first, XPORTHEFT aims to eavesdrop on three particular types of UI-level 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 signals of crosstalk voltage leakage from 20 popular mobile apps and two soft keyboards (i.e., unlocking numeric keyboard and full-size QWERTY keyboard) running on five mobile devices that are charging with five commodity multi-port chargers from different vendors. Our evaluation results for eavesdropping attacks show high effectiveness of XPORTHEFT where it achieves 98.8% in recognizing the unlocking passcode, 88.7% in fingerprinting the 20 mobile apps, and 83.0% in revealing the alphabetic keystrokes of a QWERTY keyboard. Furthermore, XPORTHEFT presents effectiveness in eavesdropping audio from USB-C audio pins that could contain sensitive information, such as medical conditions, where it achieves 97.1% and 98.0% in eavesdropping audio of numeric digits and alphabetic characters, respectively. The empirical results also demonstrate that XPORTHEFT is resilient to various practical impact factors, including different multi-port

chargers, mobile devices, and battery levels of charging devices. In addition, we show the potential for launching attacks on multiple victims' devices and provide efficient software- and hardware-based countermeasures to smooth out voltage leakages to defend against XPORTHEFT.

In respect of validating the inaudible audio injection attack, we evaluate it over three commercial voice assistants integrated into popular smartphones, including Apple Siri, Google Assistant, and OnePlus Breeno. Specifically, the attacking device can receive voice commands remotely from the attacker through wireless communications (e.g., Wi-Fi or 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 on commodity mobile devices. Finally, 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 voice-controllable IoT devices (e.g., Apple HomeKit). In particular, extensive evaluation shows that XPORTHEFT achieves 100% attacking success rates in activating the three voice assistants, injecting different voice commands, and 12 trials of end-to-end audio injection attacks.

Key Advancements in XPORTHEFT : As an extension work of the MobiCom'23 paper [10], XPORTHEFT presents the following key differences: (i) Introducing a new exploration of potential audio eavesdropping attack through the crosstalk leakage across USB-C ports, (ii) Proposing a new end-to-end pipeline to systematically illustrate the attack surface, (iii) Providing software-defined countermeasures and a comprehensive user study of USB charging security, and (iv) Extensively reconstructing and expanding based on the latest insights and research progress in relevant works.

Contributions: We summarize the contributions as follows:

- *Novel Attack Surfaces:* We comprehensively explore crosstalk attack surfaces that can be exploited to attack mobile devices charged by a commercial multi-port charger. It uses changes in voltage leakage between neighboring USB charging ports to reveal sensitive information and characteristics of the USB-C interface to eavesdrop on sensitive audio conversations and inject malicious voice commands into other charging devices across ports.
- *End-to-end Attack Exploration:* We propose and implement an end-to-end attack system, XPORTHEFT, to demonstrate the feasibility of a collection of the proposed attacks. Specifically, it exploits the crosstalk voltage leakage to recognize UI-level user privacy. In addition, it exploits the audio pins of the USB-C interface to reconstruct sensitive audio transmission, inaudibly activate the voice assistant, and inject modulated malicious voice commands from the neighbor USB-C port to other charging devices.
- *Comprehensive Evaluation:* We comprehensively evaluate the effectiveness of XPORTHEFT with five commodity multi-port chargers and five mobile devices. The results indicate that it effectively performs eavesdropping on various user activities and audio information. Moreover, XPORTHEFT achieves a 100% success rate in activating different voice assistants and inaudibly injecting different

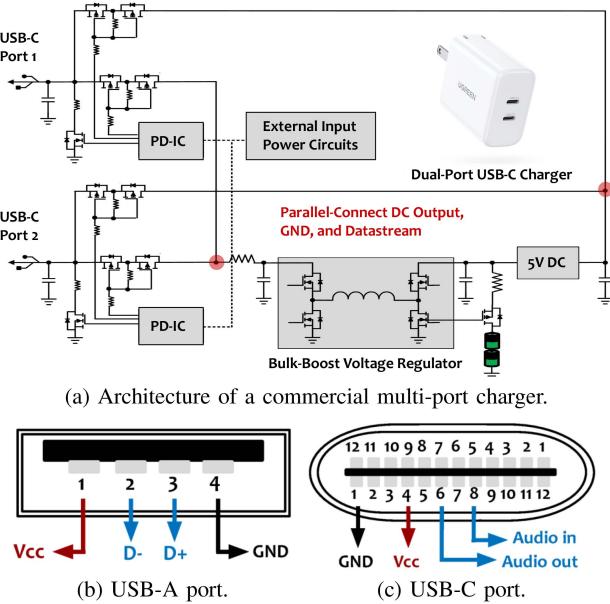


Fig. 2. Architecture of a multi-port charger and USB ports: (a) Circuit of a typical multi-port charger, (b) USB-A port (4 pins), and (c) USB-C port (24 pins on two sides).

voice commands. Next, we also show the potential to attack multiple charging devices and provide effective countermeasures to defend against it. In addition, we provide a user study to show the social influence of USB security on public awareness.

II. BACKGROUND

A. Primer on Multi-Port Chargers

Commercial multi-port chargers enable users to charge multiple mobile devices (e.g., smartphones, tablets) simultaneously. For example, Fig. 2(a) depicts a commercial multi-port charger, the UGREEN 40 W dual USB-C charger, that are integrated with two USB-C ports. Typically, the hardware schematic architecture of a multi-port charger consists of an AC voltage step-down transformer, a rectification circuit, a filtration circuit, a voltage regulation module, and multiple output USB 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. To support simultaneous charging, the output ports are connected in parallel, ensuring that each port provides the same voltage (e.g., 5 V). Consequently, voltage fluctuations on one port can affect the voltage levels of neighboring ports during the charging process.

B. USB Type-A and USB Type-C Ports

The USB-A port is commonly used in various mobile device accessories (e.g., chargers, USB hubs), as shown in Fig. 2(b).

In this structure, two pins (pins 1 and 4) are designated for battery charging, while two other pins handle data transfer. On the other hand, USB-C ports have been widely adopted in most Android smartphones and will be mandatory for all smartphones, including iPhones, sold within the European Union by the end of 2024 due to recently passed legislation [11]. The structure of a USB-C port, shown in Fig. 2(c), includes 24 pins distributed on two sides. Its rotational symmetry ensures identical pin functions on both sides, eliminating the need to align the connector in a specific orientation for plugging. Additionally, USB-C ports support not only battery charging and data transmission, but also audio input (pin 8) and audio output (pin 6). Because USB-A and USB-C ports support battery charging, power traces can be analyzed to infer user activities on the connected smartphone. Furthermore, the advanced capabilities of USB-C introduce potential security risks, such as the possibility of audio eavesdropping and injecting audible voice commands, as we demonstrate in this work.

C. Fundamental Principles of Potential Attack Surfaces

Below, we illustrate the fundamental principles of potential attack surfaces resulting in crosstalk user privacy leakage and inaudible audio injection between two neighbor USB ports of a multi-port charger from the aspect of physics.

Crosstalk Voltage Leakage: Similar to the crosstalk leakage identified in multi-port USB hubs [7], [12], the pins on USB charging ports inevitably share the same voltage input, and such a fundamental hardware imperfection (i.e., parallel-connected architecture) leads to crosstalk 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 (1) as follows:

$$V_x(t) \propto C \cdot V_c(t), \quad (1)$$

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 [7], [13]. 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 $I_c(t)$ to charge the smartphone in (2):

$$I_c(t) = V_c(t)/R_s(t) \propto 1/R_s(t), \quad (2)$$

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 [2], which require extra energy consumption of the battery that results in load changes $\Delta R_s(t)$ in the battery of the charging smartphone [5], [14]. As such, these load changes induce the changes of voltage $\Delta V_c(t)$ on the charging port, as well as voltage changes $\Delta V_x(t)$ on the neighbor port because of the leakage across

ports, which is shown in (3):

$$\Delta V_x(t) \propto C \cdot \Delta V_c(t) \propto C \cdot I_c(t) \cdot \Delta R_s(t). \quad (3)$$

Therefore, it is feasible to exploit the crosstalk voltage leakage from a neighboring port to monitor voltage fluctuations on the charging smartphone, which allows inference of fine-grained user activity at the UI level. In particular, the output voltage contain ripples generated by the AC-DC conversion circuitry and other signal conditioning components embedded in the charger shared by the multiple USB ports. Nevertheless, the strength of the ripple is much weaker than $\Delta V_x(t)$, which only causes subtle fluctuations on $V_x(t)$ and $V_c(t)$, and the induced noise can be mitigated through signal filters (Section IV-B).

In addition, when the victim's smartphone is charged through a USB-C port, audio is routed through a virtual headphone output rather than the loudspeaker. This enables attackers to capture audio signals by measuring crosstalk voltage leakage on the audio pins, bypassing the need for direct access to audible sound. Notably, the USB-C interface includes separate pins for the left and right audio channels, although the attacker only needs to measure the voltage of one of the audio pins.

Inaudible Audio Injection: As USB-C ports support audio transmission (Section II-B), they reveal the possibility of an inaudible audio injection attack on a smartphone's voice assistant. Typically, activating the voice assistant requires a verified owner's voice to pass a speech recognition check. However, the USB-C interface provides an alternative activation method [6]. Many smartphones permit the inline control button of the earphone to trigger the voice assistant when held for about 1 to 2 seconds, which is a function embedded within the USB-C capabilities. 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:

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

where $x(t)$ is the original audio clip that contains 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 the audio pin of the USB-C interface can recognize.

III. MOTIVATION AND THREAT MODEL

A. Motivating Examples

In this section, we present three motivating examples of launching UI-level eavesdropping, audio eavesdropping, and inaudible audio injection attacks through a commercial multi-port charger. That is, the user connects the smartphone to one port of the charger to charge the battery, 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 changes the energy consumption of the smartphone battery and further changes the power line's running current and the charger's output voltage. As mentioned in Section II-A, the voltage changes in one port can induce crosstalk voltage changes in other neighboring ports, and these changes present detectable patterns and predictable features that can be exploited to infer corresponding user activities at the UI level. In addition, the attacker can exploit the integrated audio pin in the USB-C interface to eavesdrop on audio in a private phone call, as well as activate the voice assistant (e.g., Apple Siri) and then inject malicious audio commands (e.g., "Where is my home?").

In Fig. 3, we present the voltage changes 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. Furthermore, we demonstrate that crosstalk voltage leakage on the audio pins of the USB-C interface can be exploited by attackers to eavesdrop on audio in conversations that contain sensitive information (e.g., medical health information). In addition, we also show the patterns on 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 "Where is my home?" into it. As such, Siri will respond to the requirement to send the victim's home address to the malicious attacker.

B. Threat Model

We consider a common scenario where multi-port chargers are used to charge mobile devices (e.g., smartphones and tablets), with target victims connecting their devices and performing various activities (e.g., unlocking the phone, running apps). In this setup, an attacker can share the multi-port charger with the victims in close proximity and execute three types of attacks: a UI-level eavesdropping attack, an audio eavesdropping attack, and an inaudible audio injection attack. Such scenarios are prevalent in public facilities and shared spaces, such as offices and airports, and it is practical for attackers to connect the hacking device to access crosstalk leakage or deploy a modified multi-port charger in advance.

I) UI-level Eavesdropping Attack: When launching the UI-level eavesdropping attack, the attacker monitors the crosstalk voltage changes of a neighbor port and leverages these voltage traces to infer sensitive information, including (i) the digits of the smartphone's unlock password, (ii) the victim's app usage and related activities, and (iii) sensitive keystrokes on a QWERTY keyboard. We assume that the attacker has access to a neighboring port on a shared multi-port charger with the victim, but *cannot* compromise (i) the multi-port charger itself by installing malicious firmware, (ii) the victim's USB power line to monitor the charging current traces, or (iii) the victim's smartphone by installing malicious apps.

II) Audio Eavesdropping Attack: When launching the audio eavesdropping attack, the attacker can exploit the crosstalk

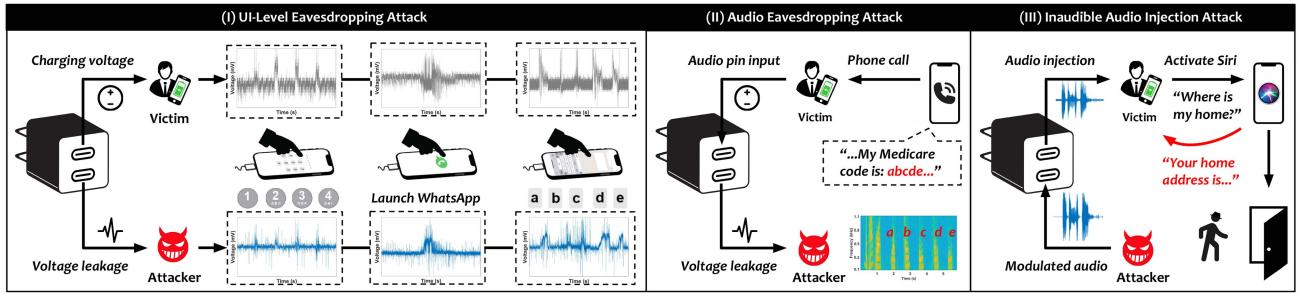


Fig. 3. Motivating examples of three attacks via XPORTHEFT. **(I) UI-level Eavesdropping Attack:** When a multi-port charger is charging the smartphone, the victim performs various UI-level activities (e.g., unlocking the smartphone, opening an app, and typing keystrokes), which induces voltage changes (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. **(II) Audio Eavesdropping Attack:** the attack leverages the crosstalk voltage leakage on the audio pin of the USB-C port to recognize sensitive information (e.g., medical health information) when the victim is making a phone call. **(III) Inaudible Audio Injection Attack:** The attacker can compromise the multi-port charger to achieve audio injection by activating the voice assistant of the victim’s smartphone and then injecting malicious voice commands through the audio pin of USB-C to obtain user privacy (e.g., “Where’s my home?”) and conduct malicious intrusions based on the response from the VCS.

voltage leakage of audio pins on the USB-C interface to recognize private information (e.g., bank account, medical condition) in secret conversations, when the victim is making a phone call on the charging smartphone. Considering some manufacturers may not connect all pins in parallel in the multi-port charger, the unconnected audio pins present no crosstalk leakages. In this scenario, following the assumptions in previous relevant studies [6], [7], we assume that the attacker can first compromise a multi-port charger by connecting the audio output pins of the two neighboring USB-C ports. Nevertheless, we also assume that the attacker *cannot* compromise the victim’s smartphone to acquire the audio by malware or place small microphones in close proximity to record audio directly, which could raise suspicions from the victim.

III) Inaudible Audio Injection Attack: When launching the inaudible voice injection attack, the attacker can use the USB-C interface to bypass speech verification and activate the voice assistant (e.g., Apple Siri, Google Assistant) of the victims’ smartphones to inject modulated audio commands through the audio signal pin of the neighbor USB-C port. Similar to the previous audio eavesdropping attack, we also assume that the attacker can first compromise a multi-port charger with USB-C ports by connecting the audio input pins together. Then, the attacker shares the multi-port charger with the victims in a shared place and leverages a customized attacking device to achieve the inaudible audio injections. In addition, the attacker can utilize speech synthesis [15] tools (e.g., Google WaveNet [16]) to generate modulated audio commands.

IV. ATTACK DESIGN

A. Overview of XPORTHEFT

Fig. 4 presents the overview of XPORTHEFT in launching the UI-level eavesdropping attack, audio eavesdropping attack, and inaudible audio injection attack. Specifically, an attacker first shares the multi-port charger with the victim and obtains the crosstalk voltage leakage from a neighbor USB port. Then, the recorded voltage traces will be processed and normalized for UI-level user activity recognition to eavesdrop on privacy information, i.e., unlocking passcode, running app activities, and in-app keystrokes. Moreover, the attack leverages the leakage on

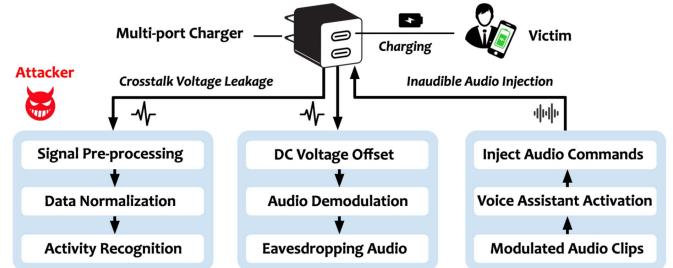


Fig. 4. Overview of XPORTHEFT.

the USB-C audio pins to eavesdrop on secret audio, i.e., bank account and medical conditions. In addition, the attacker can also exploit the integrated audio pins in the USB-C interface 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).

B. UI-Level Eavesdropping Attack

We first present the design and implementation of the UI-level eavesdropping attacks in XPORTHEFT, which consists of three components as follows: *(i)* signal pre-processing, *(ii)* data normalization, and *(iii)* activity recognition.

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, XPORTHEFT first exploits a Savitzky-Golay (S-G) filter to remove high-frequency electromagnetic noise induced by charging current’s variations, AC-DC conversions, and other components in the collected voltage signals (lines 2-6) without distorting the signal shapes [17]. 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 determined by empirical results) to find the start and end indices of the activity patterns and then segment the signal with privacy information of specific user-smartphone interactions (lines 8-18). Note that it is feasible

Algorithm 1: Crosstalk Leakage Processing.

Input: $\mathcal{V} = [v_{c_1}(t_1), v_{c_2}(t_2), \dots, v_{c_m}(t_m)]$: obtained signals from the voltage leakage. o, f : order and frequency of the S-G filter. τ : threshold of the variance.

Output: $\mathcal{S} = [S_1, S_2, \dots, S_n]$: filtered voltage signal clips containing specific smartphone activities.

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

to adjust the threshold between 0.05–0.10 to achieve adaptive signal segments in different attack scenarios.

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 [18] to reshape these voltage signals to fixed length vectors (e.g., 128×1), and then leverage the dynamic time warping (DTW) algorithm [19] 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 (5):

$$DTW_q(S_i, S'_i) = \min_{\pi \in \mathcal{P}(S_i, S'_i)} \left(\sum_{(i,j) \in \pi} d(S_i, S'_j)^q \right)^{\left(\frac{1}{q}\right)}, \quad (5)$$

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

To resolve this optimization problem, we need to obtain the quantity $R_{i,j}$ [20] between two timestamps i and j as:

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

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

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

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 UI-level user activity recognition.

③ **Activity Recognition:** As the processed voltage signals are time series, XPORTHEFT adopts a one-dimensional convolutional neural network (CNN) cascaded with a Long Short-Term Memory (LSTM) [21] layer to build a classifier for various activity recognition (e.g., app launching, individual key-pressing inference). Specifically, CNN-based neural networks are utilized in various side-channel attacks [2], [4] 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 [22]. 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. In practice, we implement the first two components, signal pre-processing, and data normalization, by leveraging the MATLAB R2022a Signal Processing Toolbox (version 3.0) that supports reliable toolkits. Then, we implement 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

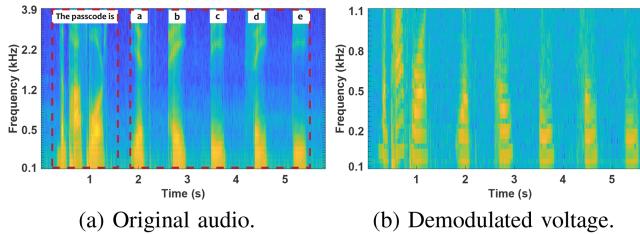


Fig. 5. Spectrograms of the original audio and the demodulated voltage signal of eavesdropping voice mail “The passcode is abcde” through the USB-C interface.

mobile apps (20 classes), and the alphabetic keys on the full-size QWERTY keyboard (26 classes).

C. Audio Eavesdropping Attack

As discussed in Section II-C, the audio pins of a USB-C port also support audio output, allowing the acquisition of audio data through the analysis of charging power patterns. Therefore, XPORTHEFT can be extended to obtain crosstalk 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 from a compromised multi-port charger, following three steps: (i) DC voltage offset, (ii) audio demodulation, and (iii) eavesdropping on audio content.

① DC Voltage Offset: To address the background noise in the crosstalk voltage leakage from the USB-C audio pin, we first apply a high-pass filter to remove the DC voltage offset (e.g., ~ 1.5 V) and some low-frequency acoustic noise induced by the ambient environment and human speech from the collected voltage signals, $V_x(t)$, to recover the raw audio, $A_n(t)$. Following the method proposed in [23], we then use spectral subtraction to enhance the speech audio signal in $A_n(t)$. First, Specifically, we perform a Fast Fourier Transformation (FFT) to obtain the frequency-domain spectrum, $\mathcal{A}_n(\omega)$. In practice, we set the default frame length in FFT as 1000 and the MCU sampling rate as 10 kHz. By monitoring the idle smartphone charging current, we estimate the strength of the noise signal, $\mathcal{N}(\omega)$ and subtract it from $\mathcal{A}_n(\omega)$ to obtain the filtered spectrum, $\mathcal{A}_c(\omega)$, and transform the denoised frequency spectra $\mathcal{A}_c(\omega)$ back to the time-domain audio signals $A_c(t)$.

② Audio Demodulation: After removing the low-frequency noise from the voltage signals of audio pins on the USB-C interface, we use an amplifier to add an initial voltage offset ΔV_{offset} (e.g., ~ 1.5 V) to obtain an absolutely positive voltage $\Delta V_o(t)$ for the audio input, since the attacker’s ADC can only process the signals with positive voltage and then demodulate the audio signals $A_e(t)$ as:

$$A_e(t) = \frac{V_o(t) - \Delta V_{offset}}{k}, \quad (8)$$

where k represents the maximum difference value of $|V_{offset} - \Delta V_{offset}|$, and we set $k = 0.1$ to balance the audio quality and SNR value in launching real-world attacks. Fig. 5(a) and (b) individually present the spectrograms of the original audio conversations and the obtained voltage output after applying the demodulation methods through XPORTHEFT of the voice

mail “The passcode is abcde”, where we can also find similar patterns that contain sensitive information that are presented in the crosstalk voltage signals. Hence, the attacker can also exploit voltage leakage, as shown in XPORTHEFT, to uncover the conversation content in voice email and phone calls in a more stealthy way.

③ Eavesdropping Audio: Although the audio signals reconstructed from crosstalk voltage leakage retain essential information, parts of the recovered audio remain unrecognizable due to signal loss. To address this, deep learning models such as CNNs are applied to identify complex patterns in voice signals, enabling speech recognition even with low-frequency audio. As a proof of concept, we focus on recognizing digits and alphabet characters to demonstrate the potential to extract sensitive information, such as bank account numbers, passwords, and verification codes. Specifically, the CNN input is a 100×100 spectrogram matrix of denoised audio signals, generated by the Short-Time Fourier Transform (STFT). The CNN model includes two convolutional layers with ReLU activation, along with two 2×2 max-pooling layers. To improve classification performance and prevent overfitting, two dense layers with a dropout rate of 0.5 are used. Finally, a softmax layer provides the probability distribution across ten digits. With this trained model, an attacker could potentially infer spoken digits and alphabet characters from the audio clips reconstructed by crosstalk voltage leakages.

D. Inaudible Audio Injection Attack

Apart from the UI-level and audio eavesdropping attacks, we also present the design and implementation of XPORTHEFT in launching an inaudible audio injection attack in this subsection. As mentioned in Section III-B, 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. The attacker then connects the attacking device (details in Section IV-E) to the neighbor USB-C port and conducts three steps to achieve malicious audio injection: (i) audio modulation, (ii) voice assistant activation, and (iii) audio commands injection.

① Audio Modulation: As discussed in Section II-B, 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 (4) in Section II-C 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.

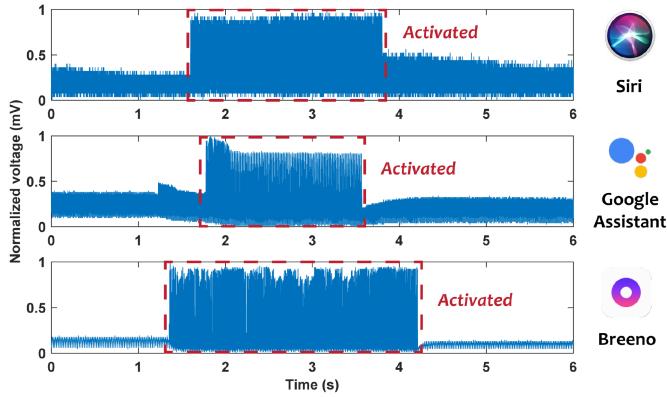


Fig. 6. Audio pin voltage signals when activating three commodity voice assistants (Apple Siri, Google Assistant, and Breeno) through the USB-C interface. The red boxes present the voltage changes when the voice assistants are activated.

② Voice Assistant Activation: Previously, inaudible audio injection attacks [24], [25] 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 [26], [27], [28]. Therefore, in Section II-C, 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 the voice assistants of smartphones and present the results in Fig. 6. In practice, we tested it on three commodity voice assistants (Apple Siri, Google Assistant, and OnePlus Breeno), and we 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 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.

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”. Fig. 7(a) and (b) individually present the spectrograms of the modulated audio clip and the injected signal received by

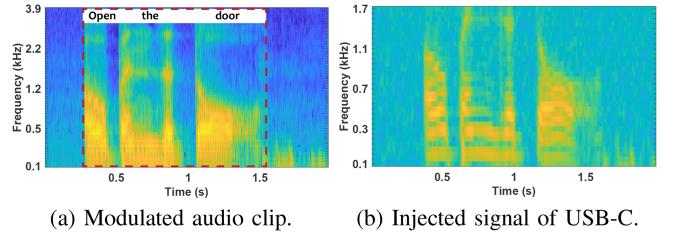
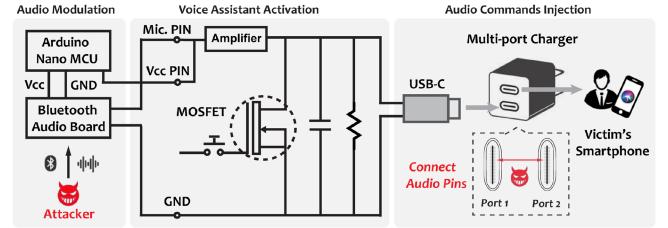
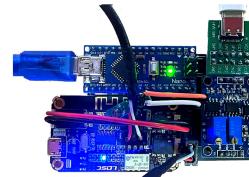


Fig. 7. Spectrograms of the modulated audio clip and the voltage signal of the USB-C audio pin when injecting the voice command “Open the door” to Siri through XPORTHEFT.



(a) Circuit design of the attacking device.



(b) Prototype.



(c) Attack scenarios.

Fig. 8. Attacking device in launching eavesdropping and audio injection attacks through a multi-port charger.

the USB-C audio pin when sensing the voice command “Open the door” to Siri through XPORTHEFT. In particular, despite the modulated audio being distorted in voice command injection, Siri can recognize the command and conduct corresponding responses because the patterns containing the most important information are maintained as the two spectrograms present (yellow part).

E. Custom-Built Attacking Device

We design and implement a portable attacking device to achieve eavesdropping and inaudible audio injection attacks in XPORTHEFT, where Fig. 8(a), (b), and (c) show the circuit design, prototype outlook, and attack scenarios, respectively. First, the device allows an attacker to record voltage leakages from an adjacent USB port, facilitating UI-level and audio eavesdropping attacks. Second, under the assumption that the attacker can compromise a multi-port charger by connecting the audio pins of neighboring USB-C ports in parallel, they can connect this device to one of the USB-C ports. This setup enables the attacker to remotely activate the voice assistant on the victim’s mobile device and send inaudible audio clips that contain malicious voice commands, potentially exposing sensitive information.

In the prototype, we utilize an Arduino Nano microcontroller to record voltage leakages and control the MOSFET transistor from a CX-729 wire control board [29] for voice assistant activation, a Bluetooth audio board [30] for receiving voice

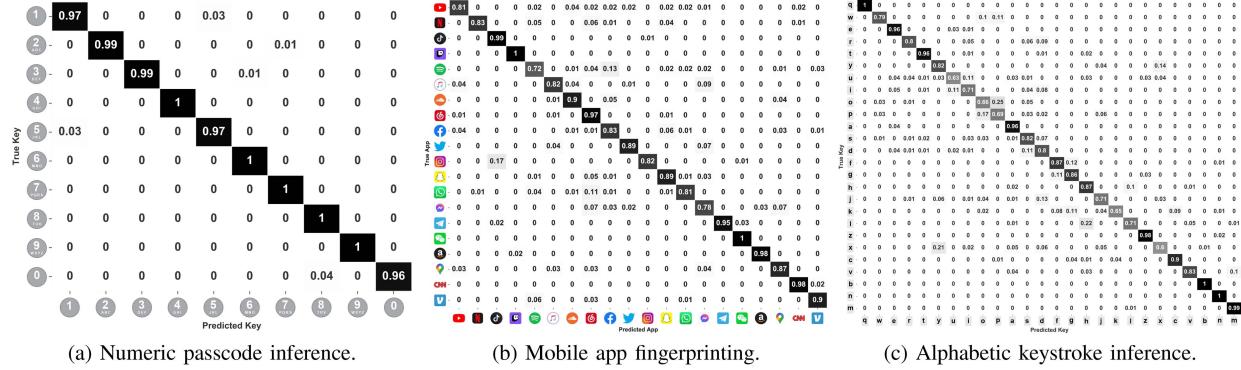


Fig. 9. Effectiveness of UI-level eavesdropping attack. (a) Recognizing 10 different numeric passcode pins (from 0 to 9, 98.8% accuracy) on the unlocking screen. (b) Fingerprinting 20 mobile apps (88.7% accuracy). (c) Uncovering 26 different alphabetic keys on a QWERTY keyboard (from “a” to “z”, 83.0% accuracy). 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.

commands, an AD620 amplifier module [31] to adjust the amplitude of the recorded voltage signals or modulated audio signals. Since the audio transmission of the USB-C interface requires no speech verification, the configurations on the audio modulation and encoding have no impact on the effectiveness of the actual inaudible audio injection attacks through XPORTTHEFT. As a proof-of-concept, we integrate these components in a custom 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.

V. EVALUATION

A. Effectiveness of Eavesdropping Attack

Experimental Setup: In the primary setting to evaluate the effectiveness of eavesdropping attacks, we use the UGREEN 40 W USB-C port charger¹, 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 that 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. All participants were informed that only the crosstalk voltage leakage from the multi-port charger would be recorded to infer their user activities like unlocking passcodes, running apps, and keystrokes. Furthermore, we also leverage audio samples from open-sourced datasets [32], [33] to conduct experiments in the audio eavesdropping attacks. We follow the

¹Note that dual-port chargers are also marketed as multi-port chargers. We adopt it to verify the feasibility of XPORTTHEFT, and also to show the potential of attacking multiple charging devices in Section VI-A. This work takes ethical considerations seriously and has been approved by the IRB of our institution, and we only use our own multi-port chargers and smartphones to collect data samples of crosstalk voltage leakage and launch inaudible audio injection attacks. In particular, the confidential datasets and our customized attacking device have never been released to any other parties.

same procedure and separately conduct experiments on four other commercial multi-port chargers from different vendors, four other mobile devices, and four other battery levels of the charging device. Moreover, all data processing and model training processes are performed on a desktop with 32 GB memory and an Intel i7-9700 K CPU, and an NVIDIA GeForce RTX 2080Ti GPU.

Effectiveness of Unlocking Password Inference: To evaluate the effectiveness of XPORTTHEFT in inferring the 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 0.5 s. Then we use 80% data samples to train the proposed CNN-LSTM classifier to determine each input key of the unlocking password and the remaining 20% data samples to evaluate the performance of the trained model with a 10-fold cross-validation. Fig. 9(a) shows the confusion matrix of the evaluation results, where XPORTTHEFT achieves 98.8% accuracy in recognizing the ten passcode pins (from 0 to 9) on the unlock 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.

Effectiveness of App Fingerprinting: To evaluate the effectiveness of XPORTTHEFT in recognizing mobile app activities, we follow the same data collection procedure and record traces when the charged smartphone launches different mobile apps. Specifically, we select 20 of the most popular mobile apps and launch each of them 50 times and obtain the first one-second voltages as data samples for app fingerprinting. Similarly, we also utilize 80% data samples to train the classifier and the rest of 20% data to evaluate model performance. Fig. 9(b) shows the confusion matrix of the evaluation results, where XPORTTHEFT presents an overall 88.7% accuracy in fingerprinting 20 popular mobile apps.

Moreover, we find XPORTTHEFT 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 voltage signals. On the contrary, XPORTTHEFT performs the worst in recognizing apps such as Spotify (72%)

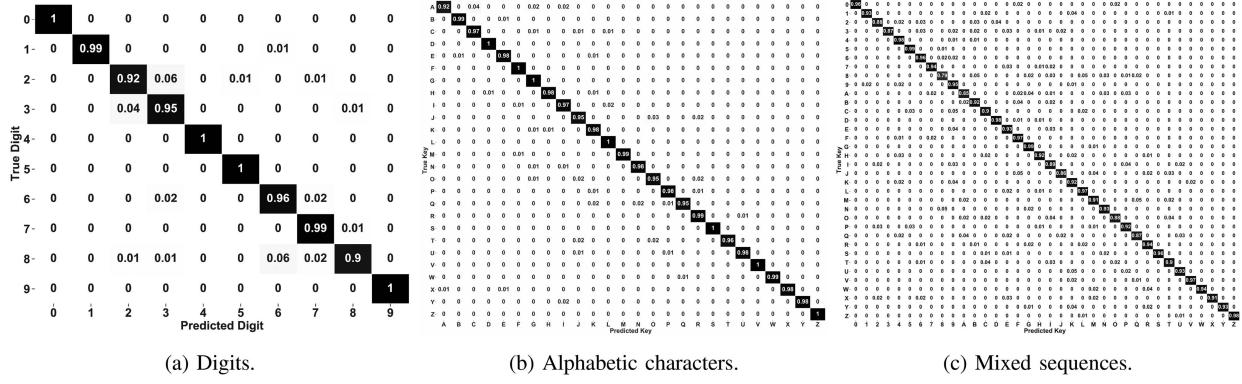


Fig. 10. Effectiveness of audio eavesdropping attack. (a) Audios with only numeric digits (97.1% accuracy), (b) Audios with only alphabetic characters (98.0% accuracy), (c) Audios with both mixed sequences (92.6% accuracy).

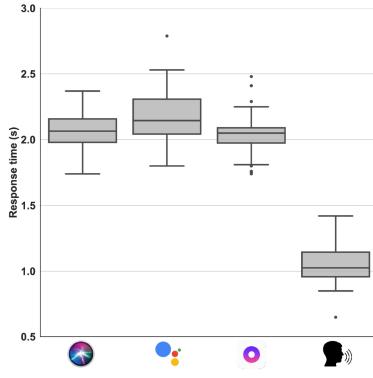


Fig. 11. Response time of three assistants and human speaking.

and Messenger (78%) because they adopt the default app launch setup (i.e., white background with a static icon) and consume the lowest energy consumption as they have fewer network requirements and animations on the screen. Therefore, the changes in the voltage incurred by app launching are milder than other apps, which further impacts the performance of XPORTHEFT in recognizing these apps. Nevertheless, XPORTHEFT still shows high accuracy in detecting the victim's app usage information stealthily during the charging process, especially apps containing sensitive information, e.g., Facebook, and WhatsApp contain the contact and address information of users.

Effectiveness of Keystroke Recovery: We also evaluate XPORTHEFT in recovering input keystrokes to achieve finer-grained eavesdropping attacks. Specifically, we collect data samples by typing the keys on the QWERTY full-size keyboard and repeating each key 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. Fig. 9(c) shows the confusion matrix of the evaluation results, where XPORTHEFT achieves overall 83.0% accuracy in recognizing 26 alphabetic keys (from "a" to "z") on a full-size QWERTY keyboard. In particular, we find that most misclassification always occurs 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, XPORTHEFT can detect keys on edge with high accuracy rates, such as "q" (100%), "a" (96%), and "z" (98%), which present distinctive patterns because they have fewer neighbor keys. In short, XPORTHEFT 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 chat apps such as WhatsApp and the password for payment in financial apps such as PayPal.

B. Effectiveness of Audio Eavesdropping Attack

Experiment Setup: To evaluate the effectiveness of the audio eavesdropping attack, we collect 30000 audio samples from the Audio MNIST dataset [33] that contain information about "0–9" digits and 10000 audio samples from CSLU: Alphadigit dataset [32] that contain alphabetic characters of "A–Z". Specifically, we play these speech samples with the maximum volume setting of the victim's smartphone, and the attacking device collects crosstalk voltage leakage from the audio pins of the neighbor USB-C port. Then, we extract the informative parts from the audio samples and convert them to spectral images for digit and alphabetic recognition. Finally, we used 80% data samples to train CNN classification models and then used the rest of 20% data for testing.

Effectiveness of Digits Eavesdropping: Fig. 10(a) shows the confusion matrix in the evaluation of XPORTHEFT in eavesdropping audio with "0–9" digits, where the proposed attack achieves 97.1% accuracy. The matrix reveals some misclassifications, particularly among the digits "two", "three", and "eight", which are frequently mistaken for "six" or "seven" due to similar low-frequency patterns. The classification accuracy is affected by the low signal-to-noise ratio (SNR) of the leaked audio signal and the limited frequency band, which reduces the distinguishability of the extracted digit utterances compared to the original ones. These empirical results also align with the discovery of previous eavesdropping attacks from compromised USB cables (e.g., [2], [6], [34]).

Effectiveness of Alphabetic Characters Eavesdropping: Fig. 10(b) depicts the evaluation results of the eavesdropping

on audios that contain “A–Z” alphabetic characters. In general, XPORTHEFT achieves an overall 98.0% accuracy when distinguishing the different pronounced alphabetic characters. In particular, XPORTHEFT can recognize characters like “D”, “G”, and “Z” with an accuracy of 100%, and there are also misclassified cases in the results due to the interference and low SNRs of the leaked audio signals in the crosstalk voltage leakage. Nevertheless, XPORTHEFT still performs well in eavesdropping on these privacy-related audios from the USB-C interface, further validating its potential threats of privacy leakage in daily mobile charging scenarios.

Effectiveness of Mixed Sequence Eavesdropping: Fig. 10(c) shows the evaluation results of the eavesdropping on audios that contain both “A–Z” alphabetic characters and “0–9” digits by playing these letters in mixed mode. Specifically, XPORTHEFT achieves an overall 92.6% accuracy when recognizing different pronounced sequences mixed with alphabetic characters and numeric digits. Compared with the empirical results above, the performance of audio eavesdropping in mixed mode decreases by approximately 4.5%–5.4%, while XPORTHEFT still shows acceptable performance, further demonstrating the threats of uncovered vulnerabilities in the USB-C interface.

C. Effectiveness of Audio Injection Attack

Experiment Setup: To evaluate the effectiveness of the audio injection attack, we compromised the UGREEN 40 W 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. Note that we only ask participants to speak the given voice commands for evaluation, which do not involve private information about themselves, and the collected confidential data samples are not released to any third parties.

Effectiveness of Voice Assistant Activation: We conduct experiments on smartphones with different voice assistants to demonstrate XPORTHEFT’s ability to activate the smartphone’s voice assistant while bypassing the speech verification system. Specifically, we utilize three smartphones (i.e., iPhone 13 Pro, Google Pixel 4, and OnePlus 10 Pro) with different commodity voice assistant systems (i.e., 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 for activating these voice assistants by speaking hotwords such as “Hey Siri”, “Hello Google”, and “Hey Breeno”. Fig. 11 shows the box plot of the response time of the three voice assistants and human speaking, and we know that it takes an average of 2.07, 2.18, and 2.05 seconds to activate Siri, Google Assistant, and Breeno through XPORTHEFT, respectively. On the other hand, it only takes approximately 1.04 seconds to activate voice assistants by human speaking. Although more time is required to activate the voice assistant, XPORTHEFT can bypass the speech

verification mechanisms that have been widely implemented in commercial mobile devices, which makes XPORTHEFT more practical in a real-world scenario. Moreover, since the injected audio commands are voltage signals, XPORTHEFT cannot be detected and countered by existing defense approaches [26], [27], [28] that are proposed to defend against inaudible audio injections through acoustic signals.

Effectiveness of Audio Commands Injection: To evaluate the effectiveness of inaudible voice commands injection attacks through XPORTHEFT, we exploit the Google WaveNet API [16] to generate 20 voice commands that have been widely used with high frequency in a quiet environment ($\text{SNR} \leq 25 \text{ dB}$), 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 a successful attack trial. Table I 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, XPORTHEFT achieves 100% success rate in activating the three voice assistants, and 100% success rate in injecting different voice commands to compromise user privacy. Therefore, XPORTHEFT shows competitive performance compared to other state-of-the-art inaudible voice injection attacks [6], [24] and fills the gap between eavesdropping and injection attacks through a multi-port charger.

D. Impact of Practical Factors

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 XPORTHEFT can be launched to other multi-port chargers, we conduct further experiments by separately collecting data and training models from four other different commercial multi-port chargers: Apple 35 W USB-C compact charger (A2579) [35], Anker 65 W smart charger (A2668) [36], Belkin 65 W USB-C charger (WCH013) [37], and ROMOSS 2.1 A USB-A charger [38]. Fig. 12(a) shows the evaluation results of launching eavesdropping attacks on the five multi-port chargers, where we find that XPORTHEFT achieves high eavesdropping accuracy across different multi-port chargers in inferring unlocking passcode (93.9% accuracy), recognizing app launch (86.6% accuracy), and uncovering keystrokes (82.2% accuracy). In particular, the results show that XPORTHEFT shows a relatively lower eavesdropping accuracy when applied to the Apple 35 W USB-C charger, as it presents a relatively high voltage ripple [39] in the charging process so that the voltage changes induced by user activities are overwhelmed. However, the results demonstrate that voltage leakage is a fundamental design flaw existing in different multi-port chargers, and XPORTHEFT presents a promising performance in inferring fine-grained user privacy across different commercial multi-port chargers. In addition, recent studies [40], [41] show that the manufacturing process leads to subtle differences in hardware circuits such as USB ports (e.g., power and audio modules). However, based on the empirical results, XPORTHEFT

TABLE I
EFFECTIVENESS OF LAUNCHING INAUDIBLE AUDIO INJECTION ATTACKS VIA XPORTHEFT

#	Voice Command	SNR (dB)	Act.	Inj.	Act.	Inj.	Act.	Inj.	#	Voice Command	SNR (dB)	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	✓	✓	✓	✓

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).

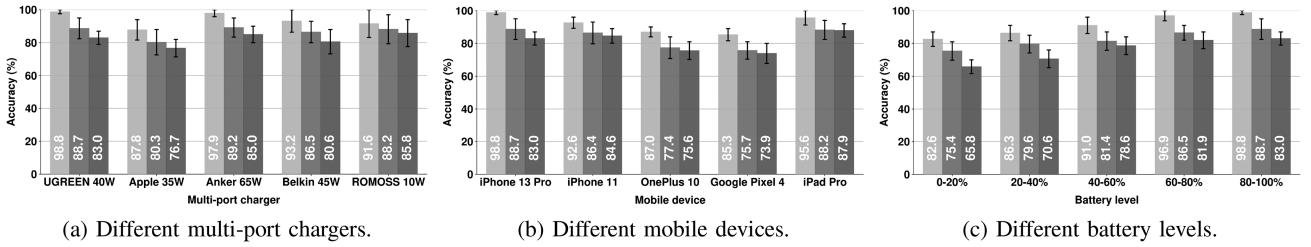


Fig. 12. Evaluation results of three practical impact factors on UI-level eavesdropping attacks of XPORTHEFT: (a) Impact of different commodity multi-port chargers, (b) Impact of different mobile devices, (c) Impact of different battery levels of the in-charging device. — Unlocking passcode inference, — App recognition, — Keystroke recovery.

achieves promising performance across multi-port chargers with different brands. As long as the multi-port chargers adopt the parallel-connecting USB structures, XPORTHEFT presents an adaptive generalization ability to launch these attacks on a broad range of USB charging accessories.

Impact of Different Mobile Devices: We use five commodity devices, including four smartphones, i.e., iPhone 13 Pro (battery volume: 3095 mA h), iPhone 11 (battery volume: 3046 mA h), OnePlus 10 Pro (battery volume: 5000 mA h), Google Pixel 4 (battery volume: 2800 mA h), and one tablet iPad Pro 2019 (battery volume: 9720 mA h), to assess the impact of different mobile devices. Fig. 12(b) shows the results of launching eavesdropping attacks on different charging devices, where we find that XPORTHEFT 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, XPORTHEFT can be scaled to different mobile devices with high recognition accuracy rates to recognize the unlocking passcode (91.9% accuracy), the running app (83.3% accuracy), and the keystrokes (81.0% accuracy), respectively. Note that we select both Android and iOS smartphones as a proof-of-concept to demonstrate the generalization of the proposed attacks.

Impact of Different Battery Levels: In practice, the mobile device may have different battery levels when connected to the port to charge the battery. To evaluate the impact of different battery levels on the performance of XPORTHEFT, we follow the

same procedure and perform experiments when the iPhone 13 Pro is at five different battery levels: 0–20%, 20–40%, 40–60%, 60–80%, and 80–100%, and Fig. 12(c) shows the experimental results of inferring the three user activities. Specifically, we know when the charging mobile device is at a high battery level (e.g., $\geq 60\%$), XPORTHEFT’s performance of the eavesdropping attack is approximately 15% higher than the lower battery levels (e.g., $\leq 40\%$). This is because most of the output voltage of the plugged USB port is used to charge 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. In 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 [2], [4], [14]. Despite the impact caused by the different battery levels of the charging device, XPORTHEFT still achieves promising accuracy rates in inferring the unlocking passcodes (91.1% accuracy), the running app (82.3% accuracy), and the keystrokes (76.0% accuracy) at the five battery levels.

Impact Factors on Audio Eavesdropping and Injection Attacks: Table II shows the evaluation results of audio eavesdropping attacks on digits and alphabetic characters and end-to-end inaudible audio injection attacks with 16 combinations of different impact factors. Specifically, XPORTHEFT achieves promising performance in eavesdropping audio from crosstalk voltage leakage of audio pins on USB-C ports. Furthermore, the results indicate that XPORTHEFT achieves a 100% success rate in activating voice assistants and a 100% success rate in injecting various voice commands into different multi-port chargers

Algorithm 2: Signal Separation Algorithm.

Input: N : Number of desired components (victims).
 $Y \in \mathbb{R}^{N \times L}$: Observed L -length crosstalk voltage signals from N charging devices.

Output: $A^{-1} \in \mathbb{R}^{N \times N}$: Inverse mixing matrix. $X \in \mathbb{R}^{N \times L}$: Independent crosstalk voltage signals.

```

1 Initialize an empty array  $A^{-1}$ 
2 for  $i \leftarrow 1$  to  $N$  do
3   Initialize a random  $N$ -length vector  $a_i$ 
4   while  $a_i$  is not converged do
5      $a_i^* = \frac{1}{L} Y g(a_i^T Y)^T - \frac{1}{L} g'(a_i^T Y) 1_L a_i$  //  $1_L$  is
       a  $L$ -dimension column vector of 1's
6      $a_i^* = a_i - \sum_{j=1}^{i-1} (a_i^T a_j) a_j$ 
7      $a_i = \frac{a_i^*}{\|a_i^*\|}$ 
8    $A^{-1} = [a_1, a_2, \dots, a_N]$ , if converged, add to  $A^{-1}$ 
9  $A^{-1} = [a_1, a_2, \dots, a_N]$ , obtain inverse mixing matrix.
10  $X = A^{-1} Y$ , calculate independent voltage signals.

```

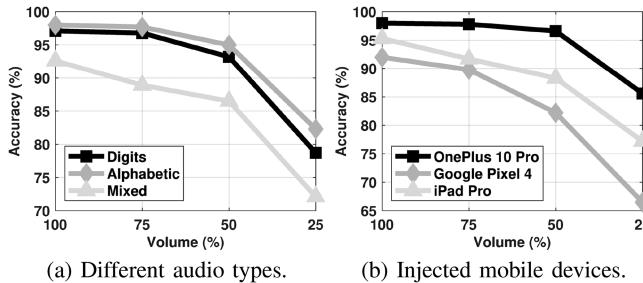


Fig. 13. Effectiveness of XPORTHEFT’s audio eavesdropping attack at different audio volumes.

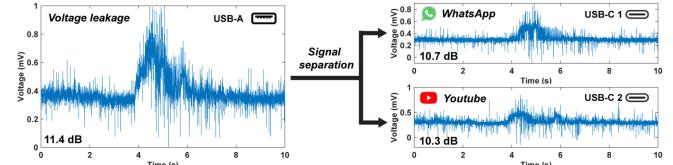
and mobile devices with different battery levels. Therefore, XPORTHEFT is resilient to the three practical factors in launching the injection attacks and achieves a high success rate of the attack. Next, as shown in Table II, we evaluated the effectiveness of XPORTHEFT on two additional multi-port chargers, i.e., Belkin 200W [42] (four USB-C ports) and Xpower 240 W chargers [43] (six USB-C ports), in the launch of audio eavesdropping and inaudible audio injection attacks, where XPORTHEFT still shows high performance.

In addition, we also conducted experiments to evaluate the effectiveness of the audio eavesdropping attack at different volumes, i.e., using UGREEN 40 W charger and the three mobile devices with USB-C charging ports. Fig. 13(a) and (b) show the evaluation results when eavesdropping on different types of audio and different mobile devices at the volume of 100%, 75%, 50%, and 25%, respectively. Specifically, we find that XPORTHEFT maintains a high volume performance over 50% and decreases by about 12.4%–25.5% when switching to the low volume mode at 25%.

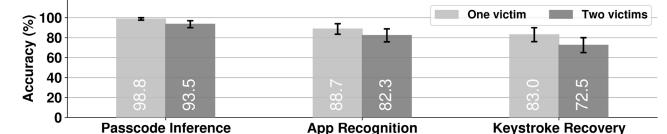
VI. DISCUSSION

A. Extending Attacks

Attacks on Multiple Victims: To explore the feasibility of attacking multiple victims, we use the Anker 65 W smart charger ($2 \times$ USB-C, $1 \times$ USB-A) to charge two iPhone 13 Pro, and play the two charging smartphones simultaneously (e.g., launching two different apps) while recording voltage leakages from the



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



(b) Effectiveness of eavesdropping two victims.

Fig. 14. Evaluation of attacking multiple victims.

neighbor USB-A port. Then, since the voltage leakage is a one-dimensional signal, we design and implement a source separation method (Algorithm 2) based on FastICA [44] to separate the mixed voltage signal into individual signals to determine the activities of each victim. Fig. 14(a) 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 conducted extensive experiments to evaluate the effectiveness of eavesdropping on two victims, and Fig. 14(b) shows the results. The accuracy decreases by approximately 5.3%–10.5% due to the increase in noise in the individual signals after the source separation, but XPORTHEFT still achieves acceptable accuracy in uncovering different user activities.

Meanwhile, the empirical results also demonstrate that the interference or noise from other charging mobile devices could impact the performance of XPORTHEFT. In addition, multi-victim audio injection attacks can be orchestrated by interconnecting the audio signal pins (e.g., audio input and output) of USB-C ports within a compromised multi-port USB charger. Using the analog audio channel capability, an attacker could remotely trigger voice assistant activation and inject covert voice commands into multiple connected devices simultaneously, bypassing user interaction.

Other Types of Attacks on Multi-port USB Chargers: In this paper, we comprehensively explored the most common eavesdropping and injection attacks on multi-port chargers induced by the crosstalk USB architectures. In fact, there are many other attack vectors existing in the multi-port chargers, such as potential crosstalk USB deanonymization [41], [45], off-path injections into USB communication [12], and eavesdropping attacks on other emerging cyber-physical systems (e.g., VR headsets [34]). In our future work, we plan to explore these attack vectors in specific real-world scenarios and propose effective countermeasures.

B. Countermeasures

Software-based Countermeasures: To mitigate audio eavesdropping injection attacks such as those of XPORTHEFT, a software-based solution is to disable audio transmission via the system-level API [46], preventing the voice control system from

TABLE II
EVALUATION OF AUDIO EAVESDROPPING AND INAUDIBLE AUDIO INJECTION ATTACKS WITH DIFFERENT IMPACT FACTORS' COMBINATIONS

Multi-port Charger	# of Ports	Type of Ports	Mobile Device	Voice Assistant	Battery Level	Eav. SR. (D)	Eav. SR. (A)	Act. SR.	Inj. SR.
UGREEN 40 W	2	2× USB-C	iPhone 13 Pro	●	80%–100%	97.1%	98.0%	100%	100%
Anker 65W	3	1× USB-A 2× USB-C	iPhone 13 Pro	●	40%–60%	92.3%	90.5%	100%	100%
Belkin 65W	2	2× USB-C	iPhone 13 Pro	●	60%–80%	95.5%	94.8%	100%	100%
UGREEN 40 W	2	2× USB-C	Google Pixel 4	●	20%–40%	96.2%	95.3%	100%	100%
Anker 65W	3	1× USB-A 2× USB-C	Google Pixel 4	●	60%–80%	91.6%	89.3%	100%	100%
Belkin 65W	2	2× USB-C	Google Pixel 4	●	0%–20%	94.5%	95.0%	100%	100%
UGREEN 40 W	2	2× USB-C	OnePlus 10 Pro	●	80%–100%	95.6%	94.7%	100%	100%
Anker 65W	3	1× USB-A 2× USB-C	OnePlus 10 Pro	●	60%–80%	91.9%	90.8%	100%	100%
Belkin 65W	2	2× USB-C	OnePlus 10 Pro	●	0%–20%	95.6%	95.0%	100%	100%
UGREEN 40 W	2	2× USB-C	iPad Pro	●	60%–80%	97.5%	97.9%	100%	100%
Anker 65W	3	1× USB-A 2× USB-C	iPad Pro	●	80%–100%	92.0%	91.6%	100%	100%
Belkin 65W	2	2× USB-C	iPad Pro	●	20%–40%	94.0%	93.3%	100%	100%
Belkin Pro 200W	4	4× USB-C	iPhone 13 Pro	●	60%–80%	96.0%	92.6%	100%	100%
Belkin Pro 200W	4	4× USB-C	Google Pixel 4	●	80%–100%	93.2%	88.9%	100%	100%
Xpower 240W	8	2× USB-A 6× USB-C	iPhone 13 Pro	●	20%–40%	95.0%	90.7%	100%	100%
Xpower 240W	8	2× USB-A 6× USB-C	OnePlus 10 Pro	●	40%–60%	96.2%	93.5%	100%	100%

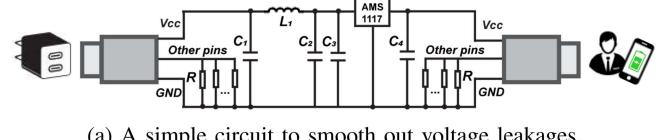
Act. SR.: Activation success rate, Inj. SR.: Injection success rate, Eav. SR: Eavesdropping success rate.



Fig. 15. Software-based defense in three smartphone systems via pop-up notification for USB access restriction.

detecting unauthorized voice commands. Following reports of USB charging vulnerabilities, several smartphone manufacturers have implemented such restrictions in their operating systems. For instance, as shown in Fig. 15(a), when connected to untrusted USB charging peripherals, smartphones like the OnePlus 10 Pro display a prompt asking users to authorize charging and data transmission, thereby helping to prevent attacks during the charging process. In contrast, iPhones and Samsung phones only provide a notification regarding the reliability of the charging device (Fig. 15(b) and (c)), without further pin-level restrictions. Hence, users could select the “Charing Only” or “Don’t Trust” option to restrict or deny access to the audio pins on the USB-C interface when charging with a multi-port charger to mitigate threats from XPORTHEFT. In addition, since eavesdropping attacks are based on captured crosstalk voltage leakages, random noise (e.g., dummy traffic packets [44], [47]) could be introduced to obscure voltage traces without affecting the user experience. However, such methods inevitably increase energy consumption and may affect charging efficiency, which has been illustrated in previous studies [48].

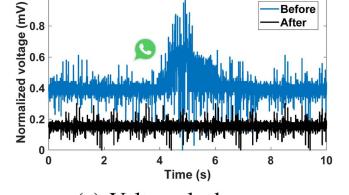
Hardware-based Countermeasures: One of the straightforward ways to mitigate inferences and injections from XPORTHEFT is to eliminate voltage leakages in multi-port chargers. In practice, we can connect a physical peripheral between the multi-port charger and the charging devices to smooth out the voltage leakages. For instance, we implement a simple circuit



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



(b) Prototype.



(c) Voltage leakages.

Fig. 16. Hardware-based defense against XPORTHEFT via a simple circuit to smooth out voltage leakages.

prototype as shown in Fig. 16(a) and (b) with resistors $R = 10\text{ k}\Omega$, capacitors $C_1 = 10\text{ }\mu\text{F}$, $C_2 = 1\text{ }\mu\text{F}$, $C_3 = 100\text{ }\mu\text{F}$, $C_4 = 22\text{ }\mu\text{F}$, and inductor $L_1 = 0.1\text{ H}$, and an AMS1117 low-dropout regulator [49]. Fig. 16(c) shows it can smooth the voltage patterns induced by smartphone activities so that the attacker cannot exploit voltage leakages to infer user privacy through XPORTHEFT. Thus, the manufacturer of commercial multi-port chargers could minimize the size of the prototype (e.g., using SMD/SMT resistors, capacitors, and inductors) and integrate it into their products to smooth out the potential crosstalk leakage to mitigate the eavesdropping attacks from XPORTHEFT. 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 in 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.

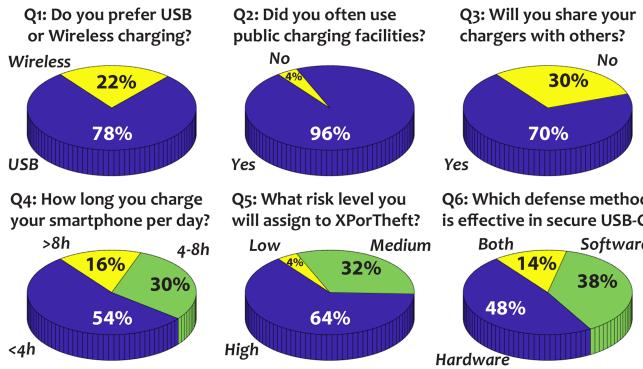


Fig. 17. Results of user study questionnaire.

VII. USER STUDY IN USB CHARGING SECURITY

To comprehensively investigate the security issues of USB interfaces and their social influence, we recruit 50 participants ranging from 18 to 55 years of age to answer a questionnaire based on their daily experience charging the batteries of their smartphones. We first showed the participants of the XPORTTHEFT attacks and other reports about USB threats and asked them about their personal experience using these charging devices, and then calculated the statistics of their answers. Fig. 17 depicts the results of the user study, where we provide six common questions in the questionnaire. We found that 78% of the participants prefer to use USB chargers to charge their smartphones while other 22% prefer to use wireless chargers. Surprisingly, 96% of the participants often use public charging facilities at airports or shared spaces to charge their smartphones, and nearly 70% of the participants would probably share their charging devices with others, making XPORTTHEFT a more influential attack in real-world scenarios. Furthermore, 30% of the participants charge their smartphones in 4–8 hours per day, and 16% of them even charge over 8 hours, which demonstrates the necessity of building secure and reliable charging accessories. Regarding attacks and defenses we have illustrated in XPORTTHEFT, 64% of the participants define the three proposed attacks as high-risk attacks, and 36% of them think the potential threats are medium or low. In addition, all participants think that our proposed defense methods are effective, where 48% of them prefer hardware-based countermeasures and 38% believe software-based solutions could be more convenient and general. As a proof-of-concept study on the security of multi-port chargers, the defense methods may not be immediately effective in all USB chargers but provide the directions for securing the USB charging. We plan to address this issue by collaborating with industrial smartphone and charger manufacturers in our future works. Note that before our survey, only 12% of the participants had realized the threats from USB charging ports and hubs, and only 3% of them had encountered or suspected any USB-related attacks. After our investigation, all of them notice the severe user privacy from these crosstalk attacks existing in multi-port chargers.

VIII. LIMITATIONS AND FUTURE WORKS

Although we demonstrate the above vulnerabilities in multi-port chargers, there still exist several limitations in the current

work. First, to validate the feasibility of the inaudible audio injection attack, we follow the research line of previous studies [6], [7] and assume that the attacker has to physically access the target multi-port charger and connect the audio pins of USB-C ports together. That is because not all manufacturers implement the connection of audio pins in their products and we adopt this assumption as a proof-of-concept study. Second, we evaluated audio injections on smartphones with speech verification mechanisms in VCS activation and injected modulated voice samples from ourselves, whereas XPORTTHEFT may require audio samples from the victim through social media or public speech to achieve the speech synthesis via Google WaveNet API [16]. Last, the current prototype requires the attacker to train multiple inference models to launch eavesdropping attacks on different chargers and smartphones with different VCS. Theoretically, XPORTTHEFT is feasible to determine the types of charging devices through the data transmission (e.g., charging protocol information) in other pins and then select the corresponding models for uncovering privacy.

IX. RELATED WORKS

A. Attacks via Charging Facilities

In Table III, we summarize the quantified comparisons between XPORTTHEFT and other state-of-the-art attacks via peripheral charging devices, i.e., USB cables [2], [6], [34], wireless chargers [4], [5], [48], [51], [52], [53], and USB hubs [7], [12]. For instance, VoltSchemer [52] demonstrates that voltage noise in chargers could be exploited to induce a hazardous charging process that impairs the battery life of the mobile device. In particular, XPORTTHEFT is the first work to explore the essential design drawbacks of a popular charging interface, the multi-port chargers, and investigate their vulnerabilities in eavesdropping and voice injection. XPORTTHEFT outperforms these works in three aspects: (i) Unlike prior works that require compromise cables [2], [6] or chargers [4], [5] to launch attacks, XPORTTHEFT has no need to compromise victim devices to achieve fine-grained eavesdropping attacks that loosen the assumptions of attackers' ability in [2], [4], [5]. It also reduces the attack efforts to inject malicious voice commands than the prior work [6] 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) XPORTTHEFT is an orthogonal attack system that can launch both eavesdropping attacks and inaudible voice injections through novel attack surfaces of the emerging charging platform and (iii) XPORTTHEFT presents the potential of attacking multiple charging devices simultaneously as we have demonstrated in Section VI-A.

B. Attacks via Other Power Traces

Recent research efforts have shown that the power consumption of a device's battery can also be used to infer user privacy [48], [53], [54], [55], [56], [57], [58], [59]. 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 [60] exploits the smartphone's battery

TABLE III
QUANTIFIED COMPARISON WITH RELATED WORKS VIA CHARGING DEVICES

Related Works	Target Device	No Need to Compromise		Eavesdropping Attacks (Acc.)			Audio Injection Potential of Attacking attacks (SR.)	
		Eavesdropping	Injection	App/Web	Keystroke (UK/FK)	Speech	Multiple Victims	
Charger-Surfing [2]	USB cables	○	○	○	● (98.7%/NA)	○	○	○
GhostTalk [6]	USB cables	○	○	○	○	● (93.3%)	● (100%)	○
Su <i>et al.</i> [7]	USB hubs	●	○	○	● (97.0%)	○	○	○
Dumitru <i>et al.</i> [12]	USB hubs	●	○	○	○	○	○	●
EM-Surfing [5]	Power line of wireless chargers	○	○	● (95.0%)	● (98.3%/96.4%)	● (81.0%)	○	○
Cour <i>et al.</i> [4]	Power line of wireless chargers	○	○	● (91.5%)	○	○	○	○
Dai <i>et al.</i> [51]	Wireless chargers	○	○	○	○	○	● (100%)	●
VoltSchemer [52]	Power adapter of wireless chargers	○	○	○	○	○	● (97.2%)	○
Wu <i>et al.</i> [14]	Wireless chargers	●	○	● (85.8%)	○	○	○	○
WISERS [53]	Wireless chargers	●	○	● (91.8%)	● (94.4%/90.6%)	○	○	○
BankSnoop [49]	Wireless charging power banks	●	○	● (93.1%)	● (94.9%/86.9%)	○	○	○
XPORTHEFT (Our method)	Multi-port USB chargers	●	○	● (88.7%)	● (98.8%/83.0%)	● (98.0%)	● (100%)	●

“●”: Yes, “○”: No, Acc.: Classification accuracy, SR.: Injection success rate, UK: Unlocking keyboard, FK: Full-size QWERTY keyboard, NA: Not available or evaluated.

consumption data to recognize the usage and activities of mobile apps. PowerSpy [61] uses two battery profiles in Android smartphones (*voltage_now* and *current_now*) to determine the motion of the smartphone to track the user’s location. DeepTheft [62] leverages the CPU energy consumption profile to reconstruct the architecture of on-device deep neural networks. Furthermore, AppListener [44] and REHSense [63] use RF energy harvesting to capture RF energy emitted by a Wi-Fi router to recognize multi-level mobile app activities of multiple connected mobile devices and fine-grained human activities within diffraction zones, respectively. In addition, recent studies [56], [64], [65] have demonstrated potential power side-channel attacks in CPUs, GPUs, and SoCs, leading to severe user privacy leakage. In comparison, XPORTHEFT presents crosstalk voltage leakages across USB ports on commercial multi-port chargers, which could be maliciously exploited to launch effective and practical eavesdropping and injection attacks in various real-world scenarios.

C. Attacks via USB Interfaces

There are many recent studies that have demonstrated different attack surfaces through USB interfaces in various embedded systems and charging facilities [8], [66], [67], [68], [69], [70]. For instance, previous works [2], [10], [45] reveal that the charging current of USB ports could be maliciously exploited to eavesdrop on fine-grained user activities such as device types, unlocking passcodes, running apps, and keystrokes. Furthermore, Su *et al.* [7] presents the crosstalk data leakage across USB ports that could leak user privacy, such as keystrokes on a mechanical keyboard on desktop computers. Dumitru *et al.* [12] presents off-path USB injection attacks that achieve injecting adversarial keystrokes or replacing file content through the USB interfaces. SpyUSB [71] leverages backscatter communication to create a covert wireless channel to steal data from the host computer. In addition, research has also shown that USB charging cables can be modified to inject intentional electromagnetic interference (IEMI) to produce “Ghost Touch” on the capacitive touchscreen to manipulate the user’s charging smartphones [72], [73]. Our work, XPORTHEFT, comprehensively explores crosstalk attack surfaces on a new multi-port charging platform and validates its feasibility and potential threats.

X. CONCLUSION

In this paper, we comprehensively explore crosstalk attack surfaces 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 XPORTHEFT, an attack system that leverages the crosstalk voltage leakage of the neighbor ports to infer sensitive UI-level user activities and secret audios, 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 architecture. Our extensive evaluation demonstrates that XPORTHEFT is effective in inferring fine-grained UI-level user privacy, and also achieves promising performance in launching inaudible audio eavesdropping and injection attacks on commercial multi-port chargers across various impact factors.

ACKNOWLEDGMENT

Any opinions, findings, and conclusions in this paper are those of the authors and are not necessarily of the supported organizations.

REFERENCES

- [1] FactMR, “USB wall charger market,” 2022. [Online]. Available: <https://www.factmr.com/report/2471/usb-wall-charger-market>
- [2] P. Cronin, X. Gao, C. Yang, and H. Wang, “Charger-Surfing: Exploiting a power line side-channel for smartphone information leakage,” in *Proc. 30th USENIX Secur. Symp.*, 2021, pp. 681–698.
- [3] Q. Yang, P. Gasti, K. Balagani, Y. Li, and G. Zhou, “USB side-channel attack on Tor,” *Comput. Netw.*, vol. 141, pp. 57–66, 2018.
- [4] A. S. La Cour, K. K. Afridi, and G. E. Suh, “Wireless charging power side-channel attacks,” in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, 2021, pp. 651–665.
- [5] J. Liu *et al.*, “Privacy leakage in wireless charging,” *IEEE Trans. Dependable Secure Comput.*, vol. 21, no. 2, pp. 501–514, Mar./Apr. 2024.
- [6] Y. Wang, H. Guo, and Q. Yan, “GhostTalk: Interactive attack on smartphone voice system through power line,” in *Proc. Netw. Distrib. Syst. Secur. Symp.*, 2022, pp. 1–15.
- [7] Y. Su, D. Genkin, D. Ranasinghe, and Y. Yarom, “USB snooping made easy: Crosstalk leakage attacks on USB hubs,” in *Proc. 26th USENIX Secur. Symp.*, 2017, pp. 1145–1161.
- [8] J. Tian, N. Scaife, D. Kumar, M. Bailey, A. Bates, and K. Butler, “SoK: “plug & pray” today—understanding USB insecurity in versions 1 through C,” in *Proc. IEEE Symp. Secur. Privacy*, 2018, pp. 1032–1047.
- [9] F. Grisioli, M. Pizzonia, and M. Sacchetti, “USBCheckIn: Preventing BadUSB attacks by forcing human-device interaction,” in *Proc. 14th Annu. Conf. Privacy Secur. Trust*, 2016, pp. 493–496.

- [10] T. Ni, Y. Chen, W. Xu, L. Xue, and Q. Zhao, "XPorter: A study of the multi-port charger security on privacy leakage and voice injection," in *Proc. 29th Annu. Int. Conf. Mobile Comput. Netw.*, 2023, Art. no. 78.
- [11] B. Ashworth, "A new EU law would force iPhones to adopt USB-C charging," 2022. [Online]. Available: <https://www.wired.com/story/eu-law-usb-c-iphones-lightning/>
- [12] R. Dumitru, D. Genkin, A. Wabnitz, and Y. Yarom, "The impostor among US(B): Off-path injection attacks on USB communications," in *Proc. 32nd USENIX Secur. Symp.*, 2023, pp. 5863–5880.
- [13] Y. Sun and J. Fidler, "Design method for impedance matching networks," *IEE Proc.-Circuits Devices Syst.*, vol. 143, pp. 186–194, 1996.
- [14] Y. Wu, Z. Li, N. Van Nostrand, and J. Liu, "Time to rethink the design of Qi Standard? Security and privacy vulnerability analysis of Qi wireless charging," in *Proc. Annu. Comput. Secur. Appl. Conf.*, 2021, pp. 916–929.
- [15] TensorSpeech, "Real-time state-of-the-art speech synthesis for TensorFlow 2," 2021. [Online]. Available: <https://github.com/TensorSpeech/TensorflowTTS>
- [16] A. V. D. Oord et al., "WaveNet: A generative model for raw audio," 2016, *arXiv:1609.03499*.
- [17] J. Chen, P. Jönsson, M. Tamura, Z. Gu, B. Matsushita, and L. Eklundh, "A simple method for reconstructing a high-quality NDVI time-series data set based on the Savitzky–Golay filter," *Remote Sens. Environ.*, vol. 91, pp. 332–344, 2004.
- [18] D. M. Kreindler and C. J. Lumsden, "The effects of the irregular sample and missing data in time series analysis," *Nonlinear Dyn. Psychol. Life Sci.*, vol. 10, pp. 187–214, 2006.
- [19] P. Senin, "Dynamic time warping algorithm review," Information and Computer Science Department, University of Hawaii at Manoa, Honolulu, USA, vol. 855, no. 1–23, 2008, Art. no. 40.
- [20] C. Tralie and E. Dempsey, "Exact, parallelizable dynamic time warping alignment with linear memory," 2020, *arXiv: 2008.02734*.
- [21] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, pp. 1735–1780, 1997.
- [22] H. Ismail Fawaz, G. Forestier, J. Weber, L. Idoumghar, and P.-A. Muller, "Deep learning for time series classification: A review," *Data Mining Knowl. Discov.*, vol. 33, pp. 917–963, 2019.
- [23] S. Sami, Y. Dai, S. R. X. Tan, N. Roy, and J. Han, "Spying with your robot vacuum cleaner: Eavesdropping via LiDAR sensors," in *Proc. 18th Conf. Embedded Netw. Sensor Syst.*, 2020, pp. 354–367.
- [24] G. Zhang, C. Yan, X. Ji, T. Zhang, T. Zhang, and W. Xu, "DolphinAttack: Inaudible voice commands," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, 2017, pp. 103–117.
- [25] T. Liu et al., "MagBackdoor: Beware of your loudspeaker as a backdoor for magnetic injection attacks," in *Proc. 44th IEEE Symp. Secur. Privacy*, 2023, pp. 3416–3431.
- [26] 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 *Proc. 29th USENIX Secur. Symp.*, 2020, pp. 2685–2702.
- [27] Z. Li et al., "Robust detection of machine-induced audio attacks in intelligent audio systems with microphone array," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, 2021, pp. 1884–1899.
- [28] G. Zhang, X. Ji, X. Li, G. Qu, and W. Xu, "EarArray: Defending against DolphinAttack via acoustic attenuation," in *Proc. Netw. Distrib. Syst. Secur. Symp.*, 2021, pp. 1–14.
- [29] AliExpress, "New original replacement wire control board volume button PCB for PB3 powerbeat earphone," 2022. [Online]. Available: <https://www.aliexpress.com/item/1005003525462944.html>
- [30] DROK, "Bluetooth board, DROK 12V audio receiver bluetooth," 2022. [Online]. Available: <https://www.amazon.com/Bluetooth-DROK-Receiver-Electronics-Headphone/dp/B07P94Z9XR>
- [31] ProtoSupplies, "AD620 instrumentation amplifier module," 2022. [Online]. Available: <https://protosupplies.com/product/ad620-instrumentation-amplifier-module/>
- [32] R. Cole, M. Noel, T. Lander, and T. Durham, "CSLU: Alphadigit version 1.3," 2008. [Online]. Available: <https://catalog.ldc.upenn.edu/LDC2008S06>
- [33] S. Srinivasan, "Audio MNIST: Audio samples of spoken digits (0-9) of 60 different speakers," 2020. [Online]. Available: <https://www.kaggle.com/datasets/sripaadsrinivasan/audio-mnist>
- [34] J. Li, Y. Meng, Y. Zhan, L. Zhang, and H. Zhu, "Dangers behind charging VR devices: Hidden side channel attacks via charging cables," *IEEE Trans. Inf. Forensics Secur.*, vol. 19, pp. 8892–8907, 2024.
- [35] Apple, "35 W dual USB-C port compact power adapter," 2025. [Online]. Available: <https://www.apple.com/shop/product/MW2H3AM/A/35w-dual-usb-c-port-compact-power-adapter>
- [36] Amazon, "Belkin BoostCharge pro 4-Port USB-C GaN wall charger, 200 W," 2025. [Online]. Available: https://www.ebay.com/itm/186863178008?chn=ps&mkevt=1&mkcid=28&google_free_listing_action=view_item
- [37] Belkin, "BoostCharge pro dual USB-C GaN wall charger with PPS 65W," 2025. [Online]. Available: <https://www.belkin.com/hk/en/p/dual-usb-c-gan-wall-charger-with-pps-65w/WCH013myWH.html>
- [38] ROMOSS, "ROMOSS TK12S 10.5 W 2.1 A double USB port fast charging wall charger," 2025. [Online]. Available: <https://www.sunsky-online.com/p/TBD0603942401/-ROMOSS-TK12S-10.5W-2.1A-Double-USB-Port-Fast-Charging-Wall-Charger-CN-Plug.htm>
- [39] ChargerLAB, "Review of Apple 35 W dual USB-C compact power adapter," 2022. [Online]. Available: <https://www.youtube.com/watch?v=aHdZu-m9y64>
- [40] W. E. Cobb, E. D. Laspe, R. O. Baldwin, M. A. Temple, and Y. C. Kim, "Intrinsic physical-layer authentication of integrated circuits," *IEEE Trans. Inf. Forensics Secur.*, vol. 7, no. 1, pp. 14–24, Feb. 2012.
- [41] S. Liao, H. Chen, and Z. Yang, "SecurityHub: Electromagnetic fingerprinting USB peripherals using backscatter-assisted commodity hardware," in *Proc. Annu. Comput. Secur. Appl. Conf.*, 2024, pp. 425–438.
- [42] eBay, "Anker PowerPort III 65 W 3 port wall charger black," 2025. [Online]. Available: <https://www.amazon.com/dp/B0D4GGF75W>
- [43] Centralfield, "Xpower XP-PC240GaN 6xType-C 2xUSB-A USB charger station," 2025. [Online]. Available: https://www.centralfield.com/product/POWXPO_XP-PC240GaN-BK/
- [44] T. Ni, G. Lan, J. Wang, Q. Zhao, and W. Xu, "Eavesdropping mobile app activity via radio-frequency energy harvesting," in *Proc. 32nd USENIX Secur. Symp.*, 2023, pp. 3511–3528.
- [45] R. Spolaor, H. Liu, F. Turrin, M. Conti, and X. Cheng, "Plug and power: Fingerprinting USB powered peripherals via power side-channel," in *Proc. IEEE Conf. Comput. Commun.*, 2023, pp. 1–10.
- [46] A. Developer, "Documentation of manifest permission," 2022. [Online]. Available: https://developer.android.com/reference/android/Manifest.permission#MODIFY_AUDIO_SETTINGS
- [47] J. Li et al., "FOAP: Fine-grained open-world android app fingerprinting," in *Proc. 31st USENIX Secur. Symp.*, 2022, pp. 1579–1596.
- [48] T. Ni et al., "Exploiting contactless side channels in wireless charging power banks for user privacy inference via few-shot learning," in *Proc. 29th Annu. Int. Conf. Mobile Comput. Netw.*, 2023, Art. no. 73.
- [49] SHIKUES, "Ams1117 1A bipolar linear regulator," 2022. [Online]. Available: https://datasheet.lcsc.com/szlcsc/2001081204_Shikues-AMS1117-1-2_C475600.pdf
- [50] J. Li et al., "MagFingerprint: A magnetic based device fingerprinting in wireless charging," in *Proc. IEEE Conf. Comput. Commun.*, 2023, pp. 1–10.
- [51] D. Dai, Z. An, and L. Yang, "Inducing wireless chargers to voice out for inaudible command attacks," in *Proc. IEEE Symp. Secur. Privacy*, 2023, pp. 1789–1806.
- [52] Z. Zhan, Y. Yang, H. Shan, H. Wang, Y. Jin, and S. Wang, "VoltSchemer: Use voltage noise to manipulate your wireless charger," in *Proc. 33rd USENIX Secur. Symp.*, 2024, pp. 3979–3995.
- [53] T. Ni et al., "Uncovering user interactions on smartphones via contactless wireless charging side channels," in *Proc. IEEE Symp. Secur. Privacy*, 2023, pp. 3399–3415.
- [54] T. Ni, Y. Chen, K. Song, and W. Xu, "A simple and fast human activity recognition system using radio frequency energy harvesting," in *Proc. 2021 ACM Int. Joint Conf. Pervasive Ubiquitous Comput. Proc. 2021 ACM Int. Symp. Wearable Comput.*, 2021, pp. 666–671.
- [55] T. Ni, X. Zhang, and Q. Zhao, "Recovering fingerprints from in-display fingerprint sensors via electromagnetic side channel," in *Proc. ACM SIGSAC Conf. Comput. Commun.*, 2023, pp. 253–267.
- [56] Y. Wang, R. Paccagnella, E. T. He, H. Shacham, C. W. Fletcher, and D. Kohlbrenner, "Hertzbleed: Turning power side-channel attacks into remote timing attacks on x86," in *Proc. 31st USENIX Secur. Symp.*, 2022, pp. 679–697.
- [57] T. Ni, "Sensor security in virtual reality: Exploration and mitigation," in *Proc. 22nd Annu. Int. Conf. Mobile Syst. Appl. Serv.*, 2024, pp. 758–759.
- [58] A. Brighente, M. Conti, D. Donadel, and F. Turrin, "EVScout2.0: Electric vehicle profiling through charging profile," *ACM Trans. Cyber-Phys. Syst.*, vol. 8, 2024, Art. no. 11.
- [59] W. Wang, M. Li, Y. Zhang, and Z. Lin, "PwrLeak: Exploiting power reporting interface for side-channel attacks on AMD SEV," in *Proc. Int. Conf. Detection Intrusions Malware Vulnerability Assessment*, 2023, pp. 46–66.

- [60] Y. Chen, X. Jin, J. Sun, R. Zhang, and Y. Zhang, "POWERFUL: Mobile app fingerprinting via power analysis," in *Proc. Int. Conf. Comput. Commun.*, 2017, pp. 1–9.
- [61] Y. Michalevsky, A. Schulman, G. A. Veerapandian, D. Boneh, and G. Nakkibly, "PowerSpy: Location tracking using mobile device power analysis," in *Proc. 24th USENIX Secur. Symp.*, 2015, pp. 785–800.
- [62] Y. Gao et al., "DeepTheft: Stealing DNN model architectures through power side channel," in *Proc. IEEE Symp. Secur. Privacy*, 2024, pp. 3311–3326.
- [63] T. Ni et al., "REHSense: Towards battery-free wireless sensing via radio frequency energy harvesting," in *Proc. Int. Symp. Theory Algorithmic Found. Protocol Des. Mobile Netw. Mobile Comput.*, 2024, pp. 211–220.
- [64] M. Lipp et al., "PLATYPUS: Software-based power side-channel attacks on x86," in *Proc. IEEE Symp. Secur. Privacy*, 2021, pp. 355–371.
- [65] H. Taneja, J. Kim, J. J. Xu, S. Van Schaik, D. Genkin, and Y. Yarom, "Hot pixels: Frequency, power, and temperature attacks on GPUs and arm SoCs," in *Proc. 32nd USENIX Secur. Symp.*, 2023, pp. 6275–6292.
- [66] Ü. Meteriz-Yıldırın, N. F. Yıldırın, A. Awad, and D. Mohaisen, "A keylogging inference attack on air-tapping keyboards in virtual environments," in *Proc. IEEE Conf. Virtual Reality 3D User Interfaces*, 2022, pp. 765–774.
- [67] K. Sridhar, S. Prasad, L. Punitha, and S. Karunakaran, "EMI issues of universal serial bus and solutions," in *Proc. Int. Conf. Electromagn. Interference Compat.*, 2003, pp. 97–100.
- [68] D. Oswald, B. Richter, and C. Paar, "Side-channel attacks on the Yubikey 2 one-time password generator," in *Proc. 16th Int. Symp. Res. Attacks Intrusions Defenses*, 2013, pp. 204–222.
- [69] M. Neugschwandtner, A. Beitler, and A. Kurmus, "A transparent defense against USB eavesdropping attacks," in *Proc. 9th Eur. Workshop Syst. Secur.*, 2016, Art. no. 6.
- [70] M. Guri, M. Monitz, and Y. Elovici, "USBee: Air-gap covert-channel via electromagnetic emission from USB," in *Proc. Annu. Conf. Privacy Secur. Trust*, 2016, pp. 264–268.
- [71] S. Li, S. Li, Q. Liu, Y. Song, C. Zhang, and L. Lu, "Watch out your thumb drive: Covert data theft from portable data storage via backscatter," *IEEE Trans. Dependable Secure Comput.*, vol. 21, no. 4, pp. 2434–2447, Jul./Aug. 2024.
- [72] Y. Jiang et al., "WIGHT: Wired ghost touch attack on capacitive touchscreens," in *Proc. IEEE Symp. Secur. Privacy*, 2022, pp. 984–1001.
- [73] H. Zhu, Z. Yu, W. Cao, N. Zhang, and X. Zhang, "PowerTouch: A security objective-guided automation framework for generating wired ghost touch attacks on touchscreens," in *Proc. 41st IEEE/ACM Int. Conf. Comput.-Aided Des.*, 2022, pp. 1–9.



Tao Ni (Member, IEEE) received the bachelor's degree from Shanghai Jiao Tong University, in 2018, the master's degree from Australian National University, in 2020, and the PhD degree from the City University of Hong Kong. He is a postdoctoral researcher with the Department of Computer Science, City University of Hong Kong. His research interests include cyber-physical system security, contactless side channels, and low-power wireless sensing. His works received the Cybersecurity Best Practical Paper Award 2024, and he was also named an ACM MobiSys Rising Star.



Zehua Sun (Student Member, IEEE) received the bachelor's degree from the School of Computer Science, Wuhan University, in 2021. He is currently working toward the final-year PhD degree (advised by Prof. Weitao Xu) with the Department of Computer Science, City University of Hong Kong. He was also a research intern (advised by Prof. Jun Liu) with the Singapore University of Technology and Design (SUTD) from March 2020 to January 2021. His research interests include satellite networking, mobile computing, and machine learning.



Yongliang Chen (Member, IEEE) received the PhD degree from the Department of Computer Science, City University of Hong Kong, in 2024. He is currently a postdoctoral researcher with the Department of Computer Science, City University of Hong Kong. His primary research interests are on the Internet of Things (IoT) and mobile security, especially in automatic vulnerability discovery and program analysis. His works received the ACM SIGSOFT Distinguished Paper Award of ICSE'24.



Yihe Zhou (Student Member, IEEE) received the bachelor's degree from Hong Kong Metropolitan University, in 2024. She is a research assistant with the Department of Computer Science, City University of Hong Kong. Her research interests include computer security, AI safety, and privacy risks in machine learning.



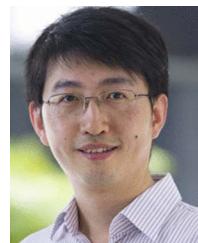
Jiayimei Wang (Student Member, IEEE) received the BE degree from the School of Cyber Science and Engineering (SCSE), Wuhan University. She is currently working toward the MPhil degree with the Department of Computer Science, City University of Hong Kong. She participated in the SOC Summer Workshop 2023 organized by the School of Computing, National University of Singapore, and won the first prize in the topic of visual computing. Her research interests include public key cryptography and software security.



Weitao Xu (Senior Member, IEEE) received the PhD degree from the University of Queensland, in 2017 (advised by Prof. Neil Bergmann and Prof. Wen Hu). He is an assistant professor with the Department of Computer Science, City University of Hong Kong. Before that, he was a postdoctoral research associate with the School of Computer Science and Engineering (CSE), UNSW from 2017 to 2019. His research areas include mobile computing, sensor networks, and IoT security.



Qingchuan Zhao (Member, IEEE) received the BE degree from the South China University of Technology, in 2009, the MS degree from the University of Florida, in 2015, and the PhD degree from Ohio State University, in 2021. He is an assistant professor with the Department of Computer Science, City University of Hong Kong. His research focuses on the security and privacy practices in the Android appified ecosystem. His works received the ACM SIGSOFT Distinguished Paper Award of ICSE'24, and have been granted bug bounties from industry-leading companies and have garnered significant media attention.



Cong Wang (Fellow, IEEE) is a professor with the Department of Computer Science, City University of Hong Kong. His research interests include data and network security, blockchain and decentralized applications, and privacy-enhancing technologies. He was a co-recipient of the IEEE INFOCOM Test of Time Paper Award 2020, the Best Paper Award of IEEE ICDCS 2020, ICPADS 2018, and MSN 2015, and the Best Student Paper Award of IEEE ICDCS 2017. At CityU, he received the Outstanding Researcher Award (2019), the Outstanding Supervisor Award (2017), and the President's Awards (2016 and 2019). He is a founding member of the Young Academy of Sciences of Hong Kong and a research fellow of the Hong Kong Research Grants Council.