Characterizing Contactless Side-channel Eavesdropping on Wireless Chargers

Tao Ni, *Member, IEEE*, Chaoshun Zuo, Jianfeng Li, *Member, IEEE*, Wubing Wang, Weitao Xu, *Senior Member, IEEE*, Xiapu Luo, *Senior Member, IEEE*, Qingchuan Zhao*, *Member, IEEE*

***Abstract*—Today, there are an increasing number of smartphones equipped with wireless charging capabilities that use electromagnetic induction to transfer power from a wireless charger to devices that are being charged. In this paper, we unveil a novel *contactless* and *context-aware* side-channel attack in wireless charging, which harnesses two physical phenomena, *i.e.*, the coil whine and the magnetic field perturbations, emanating from the wireless charging process and further infers user interactions on the charging smartphone. To validate the feasibility of this new side channel, we design and implement a three-stage attack framework, dubbed WISERS+, that first captures the coil whine and the magnetic field perturbation emitted by the wireless charger, then infers *(i)* inter-interface switches (*e.g.*, switching from the home screen to an app interface) and *(ii)* intra-interface activities (*e.g.*, keyboard inputs inside an app) to build *user interaction contexts*, and further reveals sensitive information. We extensively evaluate the effectiveness of our proposed attacks with different commercial-off-the-shelf (COTS) smartphones and wireless chargers. Our evaluation results suggest that WISERS+ can achieve over 90.4% accuracy in inferring sensitive information, such as the unlocking passcode on the screen and the launch of mobile apps. In addition, our study also demonstrates that WISERS+ is resilient to several practical impact factors, and presents its potential to be extended to attack the fast charging mode. Finally, we propose effective countermeasures and mitigate threats from the WISERS+ attack.**

***Index Terms*—Wireless charging, Contactless side channels.**

I. Introduction

Recent years have witnessed significant advancements in wireless charging technology for smartphones. Wireless charging standards, *e.g.*, Qi [1], introduced by the Wireless Power Consortium (*WPC*), have seen widespread adoption, and support for wireless charging has become an almost indispensable feature for newly released smartphones. By the end of 2023, the market saw the release of more than 25 billion smartphones equipped with a wireless charging module [2]

In this paper, we present a novel side channel targeting wireless chargers that can be leveraged to uncover *fine-grained* user interactions with charging smartphones and reveal sensitive information (*e.g.*, screen-unlocking passcode and keyboard input). Specifically, this new side-channel attack utilizes the emitted coil whine and perturbations in the ambient magnetic field when a smartphone is charged wirelessly. Unlike existing side-channel works in *wired* charging [3]–[7] and *wireless* charging [8]–[10] that require physical access to obtain current or voltage traces, this attack can work *contactlessly* and does not require knowledge of the power traces inside the wireless charger. It also makes no assumptions about compromising the victim's smartphones (*e.g.*, installing a malicious app [11]–[14]), and an attacker can launch the attack by placing a measurement device (*e.g.*, a smartphone) in close proximity (*e.g.*, 8in or 20cm) to the victim's smartphone. Additionally, our discovered side channels can be adapted to smartphones with different battery levels and various wireless charging protocols, including the widely-used Qi protocol [1] and newly-introduced fast charging protocols such as AirVOOC [15] and SuperVOOC [16].

Our newly discovered side-channel attack arises from two inevitable physical phenomena: coil whine and magnetic field perturbation, which occur during power transmission between a wireless charger and a smartphone. A user's interaction with the smartphone during wireless charging, such as typing text, can alter the displayed content on the touchscreen. These changes often affect the power supply (the amount of current) in the wireless charger, according to current charging standards (*e.g.*, Qi [1]). Fluctuations in the charger's internal coil current, following Ampere's force law, induce electromagnetic forces that cause slight deformation and vibration of the coil, resulting in coil whine and magnetic field perturbations surrounding the wireless charger, which can be detected by nearby sensing devices.

To validate the feasibility of this novel side-channel attack, we introduce WISERS+, a *WI*reles*S* charg*ER* *S*ensing system that aims to uncover user interactions in a *context-aware* manner based on the collected coil whine and magnetic field perturbations. To this end, we introduce a novel concept of *user interaction context* to comprehensively describe a series of user interactions with the smartphone in two orthogonal aspects: *(i) inter-interface switches* that represent every switch from one interface (*e.g.*, the home screen) to another (*e.g.*, an arbitrary app UI interface); *(ii) intra-interface activities* that represent actions performed within a UI interface (*e.g.*, typing on a soft keyboard). Specifically, WISERS+ comprises three

*The corresponding author.

T. Ni is with the Computer, Electrical and Mathematical Sciences and Engineering (CEMSE) Division, King Abdullah University of Science and Technology (KAUST), Thuwal, Saudi Arabia. E-mail: tao.ni@kaust.edu.sa

C. Zuo is with the Department of Computer Science and Engineering, The Ohio State University, USA. E-mail: zuo.118@osu.edu

J. Li is with the School of Cyber Science and Engineering, Xi'an Jiaotong University, Shaanxi, China. E-mail: jfli.xjtu@gmail.com

X. Luo is with the Department of Computing, The Hong Kong Polytechnic University, Hong Kong SAR. E-mail: csxluo@comp.polyu.edu.hk

W. Wang is with DBAPPSecurity Co., Ltd, Hangzhou, China. E-mail: wubing.wang@dbappsecurity.com.cn

W. Xu, and Q. Zhao are with the Department of Computer Science, City University of Hong Kong, Hong Kong SAR, China. Email: {weitaoxu, qizhao}@cityu.edu.hk

TABLE I: Comparison with related attacks.

Related Work	Attack Surface	Protocol	Non-intrusive
Cour *et al.* [8]	Current traces in power line	Qi	✗
Wu *et al.* [17]	Inductive current traces	Qi	✓
EM-Surfing [9]	Voltage traces in coils	Qi	✗
VoltSchemer [10]	Current traces in coils	Qi	✗
Dai *et al.* [18]	Current traces in coils	Qi	✗
Charger-Surfing [5]	Voltage traces in USB cables	USB 2.0	✗
GhostTalk [7]	Voltage traces in USB cables	USB 2.0/3.0	✗
WISERS+	Coil whine/Magnetic field	Qi/AirVOOC	✓

stages. Initially, it detects a range of features (*e.g.*, smartphone battery level) that influence the measurement of coil whine and magnetic field perturbation. Subsequently, it configures itself accordingly in preparation for an attack. Next, it leverages the coil whine to infer inter-interface switches and utilizes the magnetic field perturbations to uncover intra-interface activities. Based on inferred switches and uncovered activities, WISERS+ builds the *user interaction context* and finally interprets particular user interactions to reveal specific sensitive information (*e.g.*, typing the username and password in a particular app). Table I shows comparisons between WISERS+ and related side-channel attacks, and our attack framework leverages our newly discovered contactless side channels to launch a non-intrusive attack on Qi and AirVOOC fast-charging protocols.

We have developed a prototype of WISERS+ and conducted a comprehensive evaluation to assess its performance at various stages, including individual effectiveness analyzes and end-to-end attack demonstrations. Our prototype utilizes an iPhone to capture coil whine through its microphone and detect magnetic field perturbations via its magnetometer. As a proof-of-concept, this prototype mainly targets three specific intra-interface activities (*i.e.*, app launch, keyboard open, and keystroke) and four types of user interfaces (*i.e.*, off screen, lock screen, home screen, and app interface). These activities and interfaces are instrumental in uncovering sensitive information such as screen-unlocking passcodes, cross-app searching content, and app-specific sensitive user inputs. Accordingly, we prepared eight datasets consisting of data traces collected from the top 15 apps in each of the 24 categories (360 in total) in the Apple Store and Google Play. WISERS+ achieves an accuracy of 92.5% to infer inter-interface switches, 91.8% and 87.9% to recognize an app at launch in the closed-world and open-world setting, respectively, and 99.0% to identify a keyboard open. In respect of uncovering keystrokes ranging from 1 to 15 in length on the screen-unlocking keyboard, the numeric-only keyboard, and the full-size keyboard, WISERS+ also reaches the accuracy of 94.4%, 92.6%, and 90.6%, respectively, within five attempts.

In addition, we conducted 40 end-to-end attack trials to reveal the three types of sensitive information mentioned above from a series of user interactions. Each series starts by unlocking the screen and ends with typing sensitive information in one of the eight popular apps such as WHATSAPP, PAYPAL, and SAFARI. WISERS+ captures each user interaction context and reveals sensitive information with a 100% success rate within five attempts. Furthermore, we also present an extensive analysis of practical impact factors, such as different chargers and smartphones. Our results show that WISERS+ is robust to a variety of impact factors, indicating that WISERS+ can be applied to different wireless chargers, battery levels, users, smartphones, charging protocol, distances, and under different scenarios with interference from background apps, phone calls, and Bluetooth connections. In addition, we propose effective passive and proactive countermeasures to obfuscate signals and prevent privacy leakage.

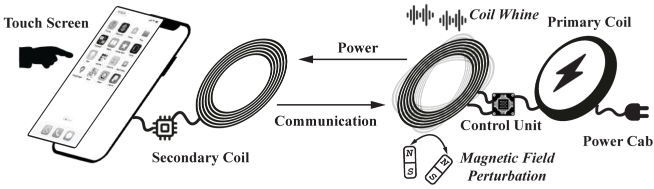


Fig. 1: Wireless charging principle.

Contributions. We make the following contributions:

- **New side-channel attack vectors.** We introduce a new side-channel attack that exploits the emitted coil whine and changes in the ambient magnetic field during the wireless charging process to infer fine-grained and sensitive user interactions on smartphones in a *contactless* manner.
- **A new attack framework.** We propose WISERS+, a three-stage, and context-aware attack framework, and implement a prototype to demonstrate the feasibility of the new side channel. Our prototype introduces a novel concept of user interaction contexts to reveal sensitive information such as screen-unlocking passcode and sensitive user inputs.
- **Extensive evaluation and countermeasures.** WISERS+ is extensively evaluated and the results show that it can effectively construct *user interaction contexts* based on the coil whine and the magnetic field perturbation traces. Furthermore, our study shows that the demonstrated attack is resilient to a list of impact factors, and can be extended to both Qi and fast-charging protocols. In addition, we also propose effective countermeasures to mitigate threats from the uncovered wireless charging side channel.

II. BACKGROUND

A. Wireless Charging on Smartphones

Wireless chargers utilize electromagnetic induction to charge smartphones, adhering to the widely adopted Qi protocol [19] (5–15W) or using advanced fast charging standards (*e.g.*, AirVOOC [15] (50W), SuperVOOC [16] (65W), recently proposed by smartphone manufacturers such as OPPO and Vivo. An illustration of this wireless charging process is presented in Fig. 1. When a wireless charger detects that a smartphone is put on, the charger initiates a series of communications with the smartphone for power transfer configuration, and its control unit converts the DC input to power its coil (primary coil). The primary coil runs an alternating current that incurs alternating voltages in the built-in coil (secondary coil) of the smartphone to achieve charging purposes. In particular, during this power transfer phase, the wireless charging unit in the smartphone continuously talks to the control unit in the wireless charger to change the power supply by adjusting the current running in the primary coil. Changes in power supply are coordinated with the

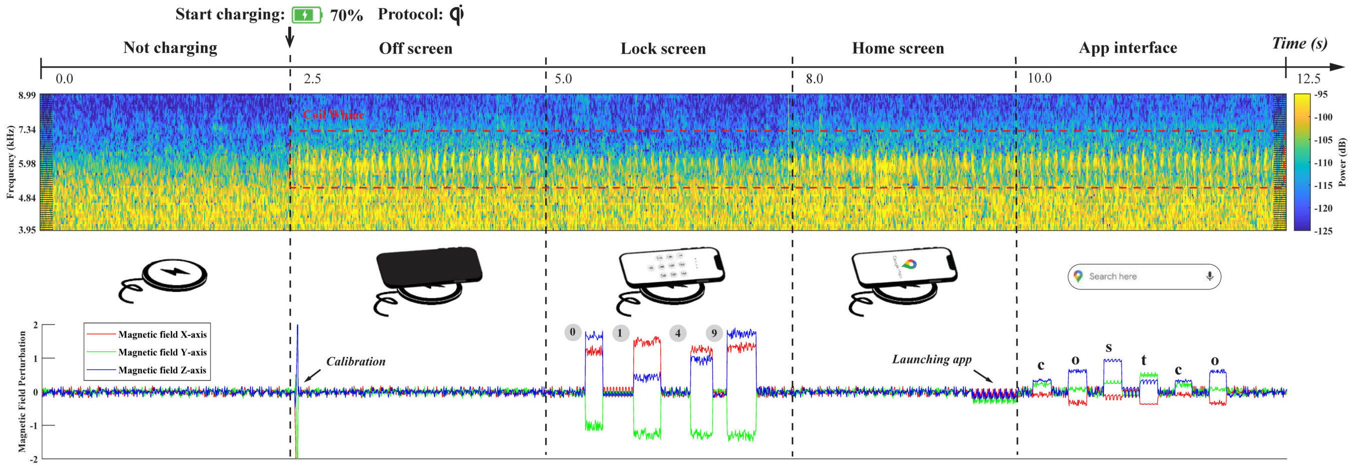


Fig. 2: An real-world attack scenario: a user places a smartphone with 70% battery left on a Qi wireless charger, unlocks the screen with the passcode (*i.e.*, 0149), clicks app icon to open GOOGLE MAP, and types “costco” to search for nearby supermarket locations. Upper Figure: the corresponding power spectrum of the coil whine; Lower Figure: the strength and directions in three dimensions of the magnetic field.

(a) Cafe (b) Airport (c) Hotel (d) Train

Fig. 3: Public wireless charging facility examples.

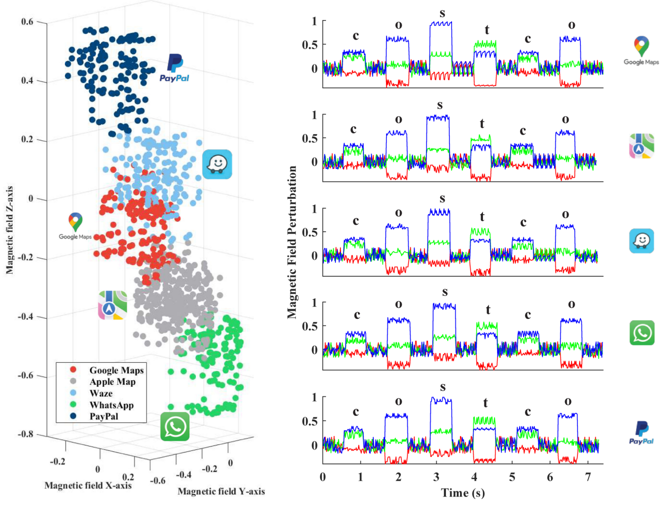
different power requirements of activities performed by the smartphone when charging [1]. Activities that consume more power force the smartphone to request more power from the wireless charger [1], [17]. This charging process terminates if the smartphone is taken away or it sends messages to the charger to stop charging, *e.g.*, the battery is fully charged.

B. Physical Phenomena in Wireless Charging

Given that various charging systems employ electromagnetic induction to facilitate power transfer from the primary coil in the wireless charger to the secondary coil in charging devices, they inherently generate an ambient magnetic field [20]. The fluctuating current levels during this charging process may cause the coils to vibrate, leading to both coil whine and perturbations in the surrounding magnetic field.

Coil whine. Coil whine, *a.k.a.*, electromagnetically induced acoustic noise, is a microphonic phenomenon. As shown in Fig. 1, it is generated by the vibration or deformation of coil materials under the excitation of a series of electromagnetic forces, including the Maxwell stress tensor, magnetostriction, and Lorentz force [21]. The coil whine can be present in different frequency ranges, making it either audible (between 20 Hz and 20 kHz) or inaudible [22] to human ears.

Magnetic field perturbation. The dynamic current changes during the wireless charging process can influence the ambient magnetic field and result in magnetic field perturbations. As such, these perturbations can be quantified by observing changes in the magnetic field over time. At a specific time point, the magnetic field can be represented by a vector comprising coordinates in a 3D Cartesian space.



(a) Launching apps. (b) Typing “costco”.

Fig. 4: Magnetic field perturbation in five different apps.

III. THREAT MODEL AND FUNDAMENTAL PRINCIPLES

A. Threat Model

Attack scenarios. We consider a common scenario that user places a smartphone on a wireless charger in a public space as shown in Fig. 3, then unlocks the screen with the passcode, and clicks the app icon to open GOOGLE MAP to search for wholesale stores by typing “costco” into the search bar. As mentioned in § II-A, these user interactions with the smartphone could impact the current in both the primary coil in the wireless charger and the secondary coil in the smartphone, resulting in coil whine [23] and magnetic field perturbations in ambient environments.

Remarkably, both the coil whine and magnetic field perturbations seem to reflect user interactions accurately. We utilize the microphone and magnetometer of another smartphone to capture these physical phenomena resulting from user interactions with the target smartphone, demonstrating that the recorded data correspond closely to the user interactions depicted in Fig. 2. The middle part of Fig. 2 illustrates the sequence of user interactions, the upper part shows the power spectrum of the coil whine, and the lower part presents the magnetic field perturbations. As can be seen, switches between interfaces (*e.g.*, screen off to lock screen) are more observable in the power spectrum of the coil whine, and finer-grained activities in an interface, such as the app launch and keystrokes, are more noticeable from the ambient magnetic field perturbations. Note that, since it could result in a significant magnetic field perturbation if a smartphone is put on the wireless charger, as shown in Fig. 2, we calibrate the ambient magnetic field to better illustrate the association between the following user interactions and magnetic field perturbations.

The observation that magnetic field perturbations could show finer-grained activities raises two additional questions, *i.e.*, *(i)* whether the launches of different apps result in different magnetic field perturbations and *(ii)* whether the same keyboard input in different apps leads to similar patterns of perturbations. To answer these questions, we further conduct a feasibility study on four other popular iOS apps, including two map apps (*i.e.*, APPLE MAP and WAZE) and two apps that provide totally different services (*i.e.*, one financial app, PAYPAL, and one chat app, WHATSAPP), and present their results in Fig. 4. Specifically, Fig. 4a presents the magnetic field perturbation resulting from the first five seconds after launching different apps, and Fig. 4b shows the perturbation of typing the same word, *i.e.*, "costco", in different apps. Obviously, launching different apps results in different magnetic field perturbations; the same keystroke produces very similar perturbations across different apps. Therefore, coil whine and magnetic field perturbations could potentially construct a new contactless side channel to infer user interactions with the smartphone when it is being charged on a wireless charger.

Attacker's capability. We assume that the adversary can place the attacking device in close proximity (*e.g.*, 8in or 20cm) to the target wireless charger and be aware of the distance and the relative angle between them. The attacking device can record environmental sounds to extract the coil whine and measure the ambient magnetic field, and it is placed before the victim puts the smartphone on the charger. In addition, the attacking device is not required to be professional, but could be a commodity smartphone. Because most smartphones have built-in magnetometers that can measure the ambient magnetic field accurately [24], and their microphones are sensitive enough with a sampling rate of $44.1\,\mathrm{kHz}$ -$48\,\mathrm{kHz}$ [25] to capture most coil whine generated in charging a smartphone with commercial off-the-shelf (*COTS*) wireless chargers (*e.g.*, Apple MagSafe Charger). Furthermore, placing this monitoring device in close proximity [17], [26]–[28] could also be achieved in public facilities (Fig. 3).

In addition, while assuming the adversary can observe the type of target wireless charger and the initial orientations, we also assume that the adversary *cannot* compromise *(i)* the charging station to monitor current traces in the power cable of a wireless charger before the power conversion, *(ii)* the wireless charger to monitor the current traces in the primary coil after the conversion, and *(iii)* the victim smartphone, including modifying hardware or leveraging an installed malicious app or any software vulnerabilities.

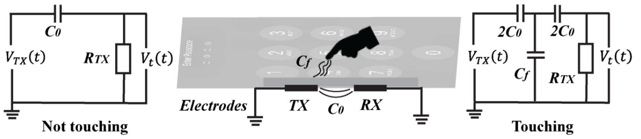


Fig. 5: Illustration of finger-coupling effects in a touching event.

B. The Fundamental Principle

The principle of wireless charging. Typically, wireless chargers leverage electromagnetic induction to transfer power from their primary coil to the secondary coil of the smartphone. First, the primary coil in the charger generates an inductive electromagnetic field, *i.e.*, $\Phi_s(t)$, in the secondary coil based on the Biot-Savart law (Equation 1). The inductive electromagnetic field produces an induced voltage $V_s(t)$ to power the smartphone following Faraday's law (Equation 2).

$$\Phi_s(t) = \eta \Phi_p(t) = \eta \frac{\mu_0 N_p I_p(t)}{2r_p}, \tag{1}$$

$$V_s(t) = N_s \frac{\Delta \Phi_s(t)}{\Delta t} = \eta \frac{N_s}{N_p} \cdot \frac{\mu_0 \Delta I_p(t)}{2r_s \Delta t}, \tag{2}$$

where $\Phi_p(t)$ and $I_p(t)$ are the electromagnetic field and the running current in the primary coil, N_p and r_p are the turns and radius of the primary coil, N_s and r_s are the turns and radius of the secondary coil, η and μ_0 represents the energy transmission ratio and the magnetic constant.

The principle of the associations between user interactions and the coil whine. The running current in the coil generates electromagnetic forces (*e.g.*, Lorentz force [21]) that incur vibration and deformation in the coil, resulting in the coil whine. In particular, a user interaction could result in a change in the current in the primary coil, $\Delta I_p(t)$, which then changes the electromagnetic forces exerted on the coil, $\Delta F_p(t)$, according to the Ampere's force law (Equation 3).

$$\Delta F_p(t) = \Delta I_p(t) L_p \times \Phi_p(t), \tag{3}$$

where L_p is the length of the primary coil. Therefore, $\Delta F_p(t)$ distorts the amplitude A_{cw} and frequency f_{cw} of the coil whine $\Delta \mathcal{S}(A_{cw}, f_{cw})$ emitted from the wireless charging coil.

$$\Delta F_p(t) \Rightarrow \Delta \mathcal{S}(A_{cw}, f_{cw}), \tag{4}$$

where $\Delta \mathcal{S}(A_{cw}, f_{cw})$ reflects the variations of shape, intensity and intervals of the yellow jagged patterns on the spectrogram as shown in the upper part of Fig. 2. In particular, it further demonstrates the correlations of user interactions and the induced coil whine, which makes it a distinctive physical side-channel leakage in the wireless charging process.

The principle of the associations between user interactions and magnetic field perturbations. User interactions with a smartphone continuously and dynamically change the current in the coils for wireless charging, leading to magnetic field perturbations. Specifically, for user interaction, such as pressing a button, both changes in the load of $\Delta R(t)$ on the secondary coil [1] and the finger-coupling effects [28] induce magnetic field perturbations because the capacitance touchscreen consists of a grid of touch sensors (electrodes). As illustrated by the equivalent circuits in Fig. 5, when a

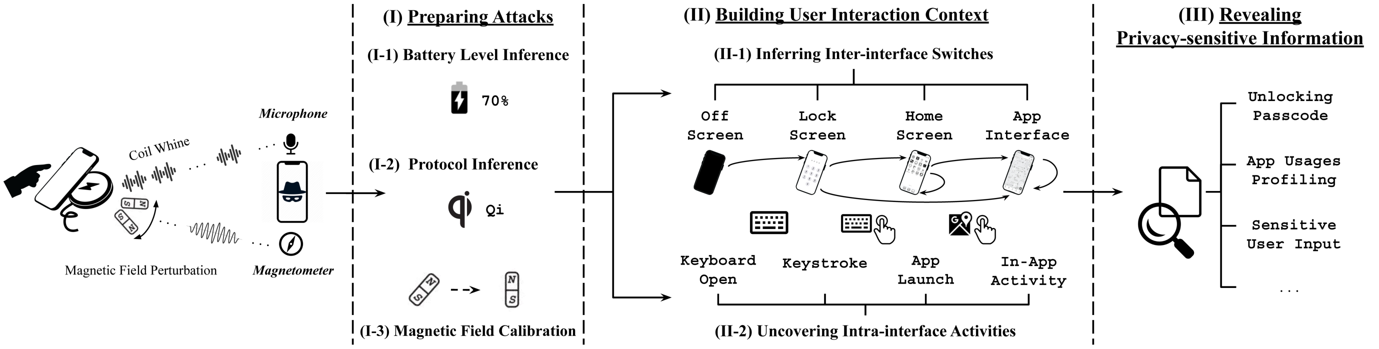

Coil Whine
Microphone
Magnetic Field Perturbation
Magnetometer
(I) Preparing Attacks
(I-1) Battery Level Inference
70%
(I-2) Protocol Inference
Qi
(I-3) Magnetic Field Calibration
N
S
(II) Building User Interaction Context
(II-1) Inferring Inter-interface Switches
Off Screen
Lock Screen
Home Screen
App Interface
Keyboard Open
Keystroke
App Launch
In-App Activity
(II-2) Uncovering Intra-interface Activities
(III) Revealing Privacy-sensitive Information
Unlocking Passcode
App Usages Profiling
Sensitive User Input
...

Fig. 6: Overview of WISERS+.

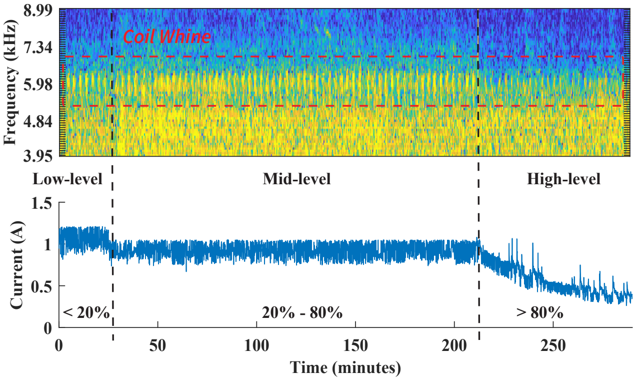


Fig. 7: Power spectrum of coil whine from different battery levels.

finger touches a button, the finger-coupling effect changes the local capacitance of $\Delta C_f(t)$ and results in the changing voltages $V_t(t)$ of this button (Equation 5), which perturbs the corresponding magnetic field. Note that $V_{TX}(t)$ and R_{TX} are the driven voltage and resistor of the electrode grid.

$$\begin{cases} V_t(t) = V_{TX}(t) \cdot \frac{R_{TX}}{R_{TX}+1/(j2\pi f C_0)} \text{ (Not touching)} \\ V_t(t) = V_{TX}(t) \cdot \frac{R_{TX}}{R_{TX}+1/(j4\pi f C_0)+\Delta Z_f(t)} \text{ (Touching)} \\ \Delta Z_f(t) = 1/(\frac{1}{1/(j2\pi f \Delta C_f(t))} + \frac{1}{1/(j4\pi f C_0)}) \text{ (Impedance)} \end{cases} \quad (5)$$

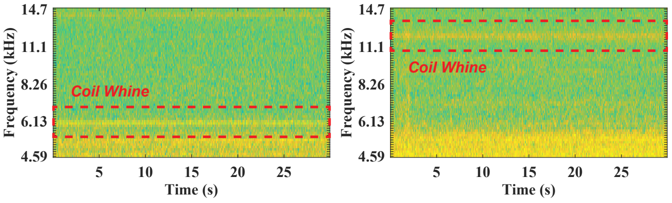
Since the key-pressing animation and finger-coupling effects occur together, the change of current $\Delta I(t)$ and the induced electromagnetic field $\Delta\Phi(t)$ at a certain touching point (Equation 6) finally produce perturbations on the inductive electromagnetic field, $\Phi_s(t)$.

$$\Delta I(t) = \frac{V_s(t) + \Delta V_t(t)}{\Delta R(t)} \Rightarrow \Delta\Phi(t) = \frac{\mu_0 N_s \Delta I(t)}{2r_s}, \quad (6)$$

where $\Delta\Phi(t)$ reflects the perturbations on the ambient magnetic fields. As shown in the curves with triplet colors in the lower part of Fig. 2, these magnetic fluctuations can be captured by magnetometers that are widely integrated in the IMU sensing module of COTS smartphones. In addition, the captured magnetic signals present distinctive patterns when launching different mobile apps and typing keystrokes, which leak fine-grained user privacy from the wireless chargers.

IV. Attack Framework

This section presents the details of our proposed three-stage attack framework, WISERS+. As shown in Fig. 6, *(i)* the first stage is to prepare an attack, which includes inferring the battery level left in the charging smartphone and the charging protocol, as well as calibrating the magnetic field (§ IV-A); *(ii)* the second stage is to build the user interaction context from both the inter-interface switches inferred from the traces of coil whine and the intra-interface activities uncovered from the traces of magnetic field perturbations (§ IV-B); and *(iii)* the last stage is to utilize the established user interaction context to uncover user privacy (§ IV-C). The implementation of the WISERS+ prototype is detailed in § IV-D.



(a) Qi.

(b) AirVOOC fast charging.

Fig. 8: Power spectrum of coil whine at different charging protocols.

A. Preparing Attacks

WISERS+ aims to discern user interactions with a smartphone by analyzing unique patterns in the coil whine and magnetic field perturbations generated during wireless charging. Consequently, its accuracy hinges on the precision of pattern recognition from these traces. Several factors affect pattern recognition, including the *(i)* battery level of the charging smartphone, *(ii)* charging protocol, and *(iii)* relative positions between the wireless charger and the measurement device (*e.g.*, magnetometer in a smartphone). Changes in these factors can produce different patterns for the same user interaction. As such, WISERS+ can *identify the trigger condition* to initiate subsequent attacks (*i.e.*, when wireless charging begins), infer the battery level of the charging smartphone, determine the charging protocol, and calibrate the magnetic field around the wireless charger.

Identifying the trigger condition. The triggering condition for initiating an attack with WISERS+ occurs when a smartphone is placed on a wireless charger. Although this action produces both coil whine and disturbs the magnetic field of the charger, neither signal alone is indicative of the trigger condition due to environmental noise. Specifically, a range of environmental factors may disrupt the magnetic field and/or emit sounds within the frequency range of the coil whine from a wireless charger. For example, the frequency of the coil

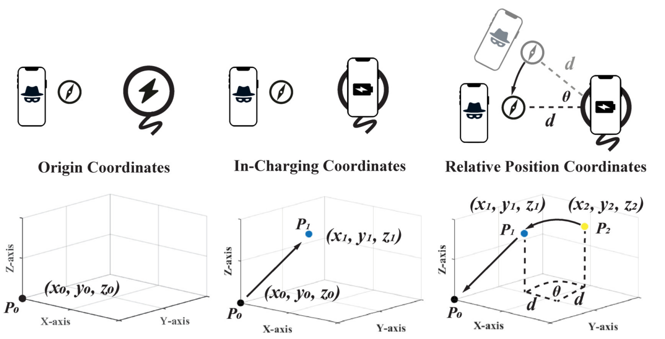


Fig. 9: Magnetic field calibration.

whine in a wireless charger is between 4 kHz and 9 kHz, which is in the same range as the sounds of cutting metal or birds chirping. Therefore, WISERS+ identifies the trigger condition by capturing an abrupt change in coil whine and the magnetic field perturbation simultaneously. Specifically, it first uses the magnetometer to log the direction and strength of the magnetic field in a time series to identify a significant perturbation and applies a high-pass filter to remove environmental noises, such as low-frequency noise resulting from screen touching or pressing, based on the frequency range of the coil whine of a particular wireless charger. Next, it leverages the Short-term Fourier Transform (*STFT*) and a periodic Hann window function to recognize the abrupt change in the filtered power spectrum because STFT combines time-frequency localization and can isolate transient events while balancing spectral leakage and temporal resolution. Unlike the standard Fourier Transform, which lacks time-domain granularity, or wavelet-based methods that prioritize variable resolutions, STFT offers a computationally efficient and interpretable framework ideal for real-time analysis of non-stationary signals, such as sudden shifts in coil whine or intermittent interface interactions.

Inferring the battery level. After identifying the trigger condition, WISERS+ next infers the battery level of the charging smartphone. In charging, the smartphone actively communicates with the wireless charger to adapt the power supply according to the battery level of the smartphone following the wireless charging protocols [1], [15], [16]. Currently, COTS wireless chargers typically divide the charging process into various stages determined by the battery level, each offering distinct power supplies (*i.e.*, varying current amounts) during these stages. However, the specific number of stages can vary among different chargers. For example, as shown in Fig. 7, our 10 W Gikfun charger separates the whole charging process into three stages associated with the battery level, *i.e.*, low-level (below 20%), mid-level (between 20% and 80%), and high-level (more than 80%). WISERS+ infers the battery level by classifying the signal power of the coil whine into different charging stages because different amounts of current generate different patterns of the coil whine. Specifically, after reviewing the acoustic features describing the signal power of a sound, we decide to use all 86 relevant features to model a coil whine trace as a 1×86 vector and adopt the random forest classification algorithm due to its advances in handling high-dimensional feature vectors.

Inferring the charging protocol. Some newly-released smartphones from OPPO, Vivo, and OnePlus support the Qi protocol and the fast charging protocols (*e.g.*, AirVOOC [15], SuperVOOC [16]). For example, when charging a OnePlus 10 Pro with Gikfun wireless chargers, the Qi protocol is used to charge the smartphone (15W, charging current 1–2A). In contrast, if using the AirVOOC 50W wireless charger to charge the smartphone, the fast charging model will be turned on (50W, charging current 3–4A) as supported by smartphone vendors. As different charging currents result in different vibrational strengths in the coils, the coil whine presents different strength and frequency patterns in the Qi charging mode and the AirVOOC fast charging mode, as shown in Fig. 8. Therefore, the coil whine presents the unique feature of being a distinctive feature that reflects different charging currents induced by different charging protocols. Hence, WISERS+ utilizes the coil whine to infer the charging protocol by extracting the acoustic features mentioned and training random forest classifiers.

Calibrating the magnetic field. As outlined in § III-B, user interaction with a smartphone can induce different magnetic field perturbations. However, the pattern of perturbation for a specific user interaction can vary depending on the relative positions of a wireless charger and the magnetic field measuring device. To simplify the task of correlating different patterns of the same interaction across the potentially infinite space of relative positions, WISERS+ calibrates the magnetic field coordinates measured from all possible relative positions between the two devices to the coordinates of a predetermined position.

As shown in Fig. 9, before putting a smartphone on the wireless charger, we first place the magnetic field measuring device at a specific position with an attacking distance d and an initial relative angle θ to the wireless charger as the pre-setting position, and set its measured magnetic field coordinates $P_0 = (x_0, y_0, z_0)$ as the origin of the coordinate. After a smartphone is put on the charger, we use a direction vector $\overrightarrow{P_0P_1}$ to represent the magnetic field drifts, where $P_1 = (x_1, y_1, z_1)$ is the new measured magnetic field coordinates. Next, we use the circular arc interpolation method [29] to calibrate the coordinates in the X-Y plane using Equation 7, where θ is the deviation of position from a random position $P_2 = (x_2, y_2, z_2)$ to P_1, to calibrate the coordinates of a measuring device.

$$\begin{cases} x_1 = x_2 - d(1 - cos\theta) \\ y_1 = y_2 - dsin\theta \\ z_1 = z_2 \end{cases} \tag{7}$$

After calibrating the coordinates to our pre-setting position, we leverage $\overrightarrow{P_0P_1}$ to reset the coordinates to the origin of the coordinate in the pre-setting position by deducting the offsets to accomplish the magnetic field calibration. In addition, user's interactions with the charging smartphone could cause subtle movements or rotations, resulting in the shift of magnetic fields to $P_1^{'} = (x_1^{'}, y_1^{'}, z_1^{'}) = (x_1 + \Delta x_1, y_1 + \Delta y_1, z_1 + \Delta z_1)$, where the calibrated coordinates shifted from $(0, 0, 0)$ to $\overrightarrow{P_0P_1^{'}} - \overrightarrow{P_0P_1}$. To mitigate its impact, WISERS+ continue to monitor the coordinate changes and deduct $\overrightarrow{P_1P_1^{'}}$ from the captured magnetic field to achieve real-time calibration.

B. Building User Interaction Context

WISERS+ constructs a comprehensive user interaction context to discern user interactions, which integrates two orthogonal aspects of user interaction, *i.e.*, inter-interface switches and intra-interface activities. These components are derived from both the coil whine and the magnetic field perturbation.

Inferring inter-interface switches. An inter-interface switch denotes the transition between various interfaces displayed on the screen of a smartphone, such as moving from the home screen to any app interface. WISERS+ exploits coil whine to deduce inter-interface switches because these transitions induce more significant changes in the power spectrum of the coil whine compared to the magnetic field perturbation, as illustrated in the attack scenarios (§ III-A).

Types of inter-interface switches. At a high level, we first systematically categorize smartphone interfaces into four groups: *(i)* off screen interface, *(ii)* lock screen interface, *(iii)* home screen interface, and *(iv)* app interface. Note that the app interface refers to the general interface of any app. According to these categories, while a series of interactions could involve multiple switches of different lengths in practice, these four types of interfaces could systematically compose six atomic and feasible switches: *(i)* off screen to lock screen, *(ii)* lock screen to home screen, *(iii)* lock screen to app interface, *(iv)* home screen to app interface, *(v)* home screen to home screen, and *(vi)* app interface to app interface.

Inferring inter-interface switches. As depicted in Fig. 2, the power spectrum of the coil whine exhibits variations across different interface types. Consequently, akin to the approach employed for inferring the battery level (as detailed in § IV-A), WISERS+ utilizes the distinct patterns present in the power spectrum of the coil whine to discern specific interface types and subsequently infer the corresponding inter-interface switches. Each interface type is represented by an 86-dimensional acoustic feature vector (details in §IV-D), and classification is performed using the random forest algorithm to identify interface types and transitions between them.

Uncovering intra-interface activities. Alongside inferring inter-interface switches, WISERS+ is designed to uncover intra-interface activities. These activities are detailed responses to user interactions within a single interface, encompassing actions such as app launch, soft keyboard activation, and keyboard typing. As mentioned in §III-A, magnetic field perturbations could reflect user interactions in granularity much finer than coil whine. Therefore, this component aims to achieve the objective by monitoring magnetic field perturbations. In addition, since recovering these activities could be formed as a classification problem, we leverage one of the state-of-the-art classification approaches [30] that trains an Attention-Based Bidirectional LSTM (*AttnBiLSTM*) model, turn it into an embedding model by removing the layers after the embedding layer, and uses the embedding model with a Cosine distance to classify magnetic field perturbations into different patterns, each associated with a distinctive intra-interface activity.

Data pre-processing. WISERS+ first utilizes a Savitzky–Golay (*S-G*) filter [31] to remove noise in sequential magnetic field perturbations collected without distorting the signals. Next, considering that each activity may last a different length of time in every attempt (*e.g.*, a single keystroke may take 0.05–0.2 second [32]), it also normalizes each activity attempt into the same length of time via property-preserving up-sampling (*e.g.*, nearest neighbor interpolation [33]) or down-sampling (*e.g.*, decimation factor [34]) algorithms.

Training model. The training of the AttnBiLSTM model requires sequences as input, which requires additional data processing to convert a series of magnetic field perturbations into a sequence that accurately reflects a specific user interaction. Considering the magnetic field at a specific time is usually described in a 3D-Cartesian coordinate system, (x, y, z), and the magnetic field perturbation could be modeled as a sequence of traces of the magnetic field in a time-series; each magnetic field dimension of a magnetic field perturbation sequence contains 1D time-series data points. As such, we adopt an approach similar to the one proposed in [35] by applying a 1D convolutional neural network (*CNN*) to extract features from the time-series data. To this end, an 1D filter is used to capture temporal correlations on each magnetic field dimension, a max-pooling layer is adopted to reduce the dimension by half, and a flatten layer is adopted to reshape the feature map to one-dimensional sequences. These sequences are the required legitimate input to train an AttnBiLSTM model. In the AttnBiLSTM model, the embedding layer takes the input sequences and generates a numerical vector. Next, the bidirectional LSTM layers learn the predictive features from the embedded vectors, and the attention layer captures the dependencies between the features and the output. After that, a full-connected layer produces the classification vectors with the same size as profiled classes. Finally, a soft-max layer generates the probability of each class and outputs the predicted class with the highest probability.

Applying classification. Having obtained the embedding model, WISERS+ will generate the embedding (e_t) of a new sequence of magnetic field perturbation (s_t), and calculate its Cosine distance to each sequence (s_i^j) of a class C_i. s_t will be classified into class C_i if one of the cosine distances between s_t and s_i^j is lower than the threshold.

C. Revealing Sensitive Information

After inferring inter-interface transitions and identifying intra-interface actions, WISERS+ is capable of constructing the user-interaction context to reveal private data. Tailored to uncover specific sensitive information during user interactions, it operates on attack strategies devised by analysts. This flexibility allows WISERS+ to adapt and facilitate various privacy-compromising attacks. The intricacy of user interaction context is crucial for extracting detailed user privacy insights; individual switches or actions alone may not suffice to decipher the nuanced semantics of user engagement with the device. For instance, while inter-interface switches may indicate an app interface switch, identifying the specific app remains elusive. Likewise, determining the meaning behind a 4-digit input discovered during an intra-interface activity is challenging without context. However, contextual analysis, such as tracing the sequence from an off-screen to a lock-screen and then to a home screen or app interface, can clarify that a 4-digit input serves as a screen unlock passcode.

D. Implementation

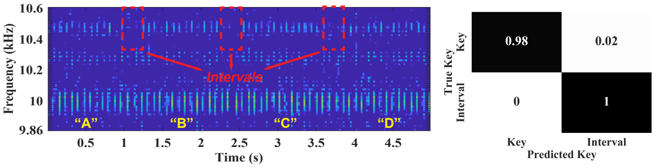
Our prototype of WISERS+ mainly targets three specific intra-interface activities: app launch, keyboard opening, and keystroke, yet it has the capacity to extend to additional activities. The development of this prototype uses a comprehensive toolkit, the details of which are discussed in the following sections.

Processing the coil whine. WISERS+ leverages the coil whine to infer the battery level left in the charging smartphone to prepare attacks and inter-interface switches. To achieve these two objectives, it extracts acoustic features of the coil whine and applies the random forest classification algorithm.

Acoustic feature extraction. As mentioned before, following a recent study [36], we extract a set of 86 acoustic features and coefficients to describe the power spectrum of the coil whine, including 26 Mel-frequency cepstral coefficients (MFCCs) [37] (*i.e.*, 13 standard MFCC features and 13 ΔMFCC dynamic features), 26 Gammatone cepstral coefficients (GTCCs) [38] (*i.e.*, 13 standard GTCC features and 13 ΔGTCC dynamic features), 24 linear prediction cepstrum coefficients (LPCCs) [39] (*i.e.*, 12 standard LPCC features and 12 ΔLPCC dynamic features), 10 spectral power patterns [36], *i.e.*, high power frequency features: *(i)* the number of peaks, *(ii)* relative frequencies corresponding to peaks, and *(iii)* standard deviations of high power frequency locations, *etc.* To extract these features, we exploit on the MATLAB Audio Toolbox (version 3.0), which provides reliable algorithms (*e.g.*, STFT) and effective toolkits (*e.g.*, high-pass and S-G filters).

Random forest classification. WISERS+ uses the random forest classification algorithm to classify different battery levels, detect typing intervals, and types of interfaces because it robustly handles noisy, high-dimensional data, *i.e.*, fluctuating magnetic signals or irregular coil whine patterns while minimizing overfitting. Compared with other lightweight machine learning algorithms (*e.g.*, Decision Tree, SVM, kNN), its inherent feature importance analysis and scalability also streamline prioritization of critical variables and enable efficient real-time performance in dynamic applications [40]. In particular, we set the number of estimators as 100, limit the maximum depth to 32, and adopt a 10-fold cross-validation.

Monitoring magnetic field perturbations. WISERS+ uncovers intra-interface activities through magnetic field perturbations using an AttnBiLSTM classification algorithm. Initially, a 1D CNN algorithm extracts features from each magnetic field perturbation sample containing six sequential magnetic field states. These sequences are then transformed into one-dimensional sequences that are compatible with the input of the AttnBiLSTM algorithm. The CNN algorithm is specifically set up with three input and output channels, utilizes ReLU activation functions, and has a kernel size of three with a stride of one. Regarding the configuration of the AttnBiLSTM classification model, we set its batch size as 128, embedding dimension as six, hidden size as 50, and use the Cross-Entropy Loss and Adam optimizer with an initial learning rate of 0.001 and epoch of 300.



(a) (b)

Fig. 10: Interval detection and keystrokes segmentation. (a) The power spectrum of the coil whines when clicking keys "ABCD" on the charging smartphone. (b) recognition results in interval detection.

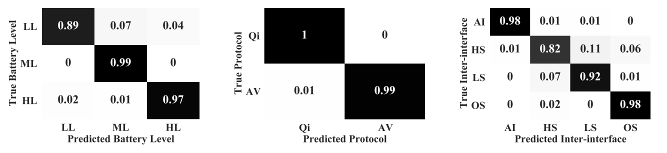
Configurations for specific intra-interface activities. The current prototype focuses mainly on three specific intra-interface activities. Accordingly, our prototype applies a set of configurations particular to each of these activities.

Adaptive threshold in launching app recognition under closed-world and open-world settings. To recognize an app that is being launched, we consider both the closed-world and open-world settings. Since our algorithm requires a threshold on the Cosine distance to classify an app, we propose an adaptive threshold mechanism to ensure its scalability. Following the closed-world and open-world settings proposed in similar works [41]–[43], we let $A_T = \{app_T^i\}_{i=1}^{m_T}$ (resp. $A_I = \{app_I^i\}_{i=1}^{m_I}$) be the set consisting of all apps in the training stage (resp. the identification stage). In particular, A_I is the subset of A_T ($A_I \subseteq A_T$) in the closed-world setting, while it could contain apps that are unmonitored in the training stage ($A_I \nsubseteq A_T$) in the open-world setting. Next, we choose the threshold based on the training set property. Specifically, we first produce the threshold set $T = \{threshold_T^i\}_{i=1}^{m_T}$ for each closed-world app_T^i class in the training stage. Next, we select the maximum threshold value $T_{max} = maximum\{T\}$ as the approximate open-world threshold. Accordingly, if the Cosine distance of a new app exceeds T_{max}, the embedding model will classify it as an unmonitored app; otherwise, it will be classified as a monitored closed-world class app_T^i if their distance is shorter than T_i.

System keyboards opening recognition. In practice, the WISERS+ prototype is designed to work with three default system-level soft keyboards, addressing scenarios where specific input fields require the use of these rather than custom keyboards. These include the keyboard to unlock the screen, a numeric-only keyboard, and the full-size QWERTY keyboard.

Keystroke segmentation via coil whine features. In practice, users often type a single word consisting of a sequence of characters of different lengths. Considering the duration of a typing practice contains both key presses and intervals between two presses, we extract the acoustic features from the coil whine between two key presses, and Fig. 10a shows that the coil whine presents different patterns when typing the touchscreen or not because of the extra induced electromagnetic field resulting from the finger-coupling effect (§III-B). As such, we leverage the extracted 86 acoustic-based features from the coil whine to distinguish the key-pressing intervals, and Fig. 10b shows it achieves 99% accuracy in recognizing the intervals.

Data normalization. As mentioned in §IV-B, a keystroke can last between 0.05 to 0.2 seconds on average [32]. Similarly,



(a) (b) (c)

Fig. 11: Effectiveness evaluation on coil whine based inference. (a) Battery level inference. LL: low-level, ML: mid-level, HL: high-level. (b) Charging protocol inference. Qi: Qi protocol, AV: AirVOOC fast-charging. (c) Inter-interface switches. OS: off screen, LS: lock screen, HS: home screen, AI: app interface.

users may spend different durations on each interface, and the smartphone could launch an app at different speeds. Therefore, we normalize each coil whine and magnetic field perturbation trace as time series with 0.1-second intervals by applying down-sampling and up-sampling, then use these traces of the same length in both training and testing.

V. Evaluation

A. Evaluation Setup

Our evaluation involves two sets of equipment to collect data[1] and process the collected data for training and testing. To collect data, we use an iPhone 11 as the data collector (the attack device) to collect data from an iPhone 13 Pro (the victim device) charging on a 10 W Gikfun wireless charger at a distance of 8 inches (20 cm). Except for the analysis on inferring the battery level, the battery of the victim device is set in the *mid-level (20% to 80%)* in all of our evaluations. Note that, we force close all background third-party apps on the victim device while the remaining system services that provide fundamental functionality. In particular, our iPhone 11 uses two free apps from Apple's App Store to collect data, *i.e.*, Audio Recorder [44] (version 1.8.1) that uses the iPhone's microphone to record the coil whine, and Sensor Logger [45] (version 1.2.5) that utilizes the iPhone's magnetometer to record the magnetic field perturbations. In respect of data processing, we run all experiments on a desktop that runs Windows 10 with 32GB memory on an Intel i7-9700K CPU and an NVIDIA GeForce RTX 2080Ti GPU.

Datasets. We first build a mobile app dataset (D_{app}) by collecting 360 apps from Apple's App Store, which include the top 15 popular free apps in each app category (24 in total) based on statistics provided by *similarweb* [46] as of mid February 2022, since App Store does not provide such statistics. Based on D_{app}, we next build eight datasets to evaluate the effectiveness of WISERS+ in every stage that is elaborated below.

B. Coil-Whine Based Inferences

Inferring battery levels. To evaluate the effectiveness of battery level inference, we build the dataset $D_{battery}$ by collecting coil whine traces from each of the three charging statuses identified in our wireless charger (shown in Fig. 7 in § IV-A). Specifically, we put the iPhone on the wireless charger, turn off its screen, wait until the coil whine becomes stable, and collect one-second data. This procedure is repeated 50 times for each of the three cases (3×50 traces in total). Note that, since they are all stable coil whine data, we further divide them into 1,500 traces, each of which lasts 0.1 seconds, for data normalization (§ IV-D), and split these traces into the training set and the testing set with the ratio of $8 : 2$.

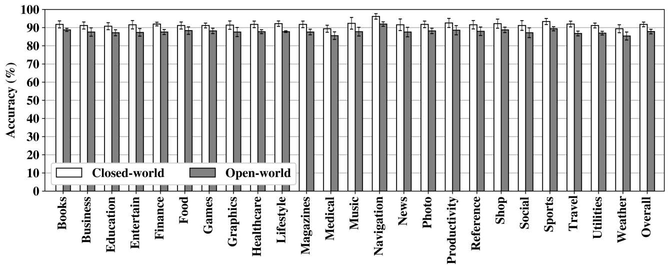
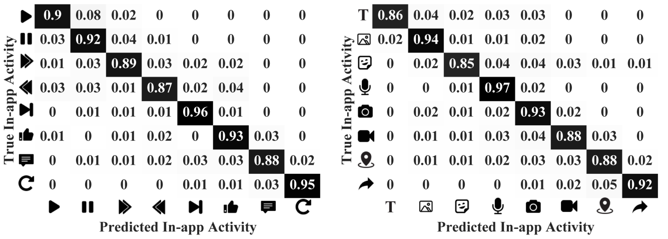


Fig. 12: Effectiveness evaluation on app recognition at launch.



(a) YouTube. (b) WhatsApp.

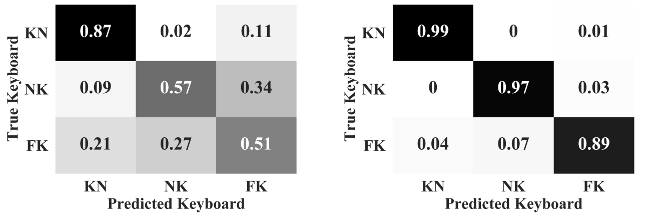
Fig. 13: Effectiveness evaluation of in-app activity recognition. ▶–Play videos, ⏩–Fast forward, ⏪–Backward, ⏸–Pause, ⏭–Switch to next video, 👍–Like, 💬–Comment, ↻–Refresh, T–Send text, 🖼–Send images, 🙂–Send stickers, 🎤–Voice call, 📷–Take photo, 📹–Video call, 📍–Share location, ➦–Share link.

Results. As shown in Fig. 11a, the overall accuracy of battery level inference is 95.0%. Specifically, the low-level, mid-level, and high-level accuracy are 98.7%, 89.3%, and 97.0%, respectively. In particular, since the data traces collected for training and testing are well balanced, the relatively lower accuracy when inferring the battery at the low-level is due to the nature of the wireless charging strategy and protocol. As shown in Fig. 7 in § IV-A, it is less stable at this battery level than that in the other two intervals, which leads to more misclassifications.

Inferring charging protocol. To evaluate the effectiveness of the inference of the charging protocol, we built the dataset $D_{protocol}$ by collecting coil whine traces when charging the OnePlus 10 Pro with the AirVOOC 50W wireless charger, which supports both the Qi protocol and the AirVOOC fast charging protocol for wireless charging. In particular, we follow the same signal preprocessing procedure as illustrated above and split these traces into the training set and the testing set with the ratio of $8 : 2$ to develop the random forest classifier. Note that a wireless charger typically supports only one fast-charging protocol.

Results. As shown in Fig. 11b, the overall accuracy of the charging protocol inference is 99.5%. Specifically, the Qi protocol and the AirVOOC protocol accuracy are 100%

[1] **Ethical Considerations.** We take ethical considerations seriously in this study. Data collection from volunteer participants was conducted with IRB approval. Unlocking passcodes, cross-app searching content, and privacy-sensitive user input were randomly generated for evaluation only, and we only use our own accounts to evaluate keystrokes uncovered inside apps. WISERS+ has never been released to any other parties.The full list of test cases in keystroke inference and more detailed experiment results of § VI (end-to-end attacks) are available at: https://github.com/CityuSeclab/WISERS_SP23



(a) Without keystroke. (b) With one keystroke.

Fig. 14: Effectiveness evaluation of keyboard open detection: KN for keyboard not open, KO for keyboard open, NK for numeric-only keyboard, FK for full-size keyboard.

and 99%, respectively. It shows that coil whine could be a robust indicator for attackers to distinguish the specific charging protocol in the wireless charging process and apply corresponding pre-trained models for uncovering user privacy.

Inferring inter-interface switches. To evaluate the effectiveness of recognizing inter-interface switches, we build the dataset D_{switch}. We also collect the coil whine traces of each type of interface for one second after it is stable for 50 rounds. Specifically, these coil whine traces are collected from (i) one testing case of the off screen and lock screen, respectively, (ii) six testing cases of the home screen, each of which shows a home screen displaying different lines of apps ranging from one to six lines excluding the dock, and (iii) 24 testing cases of app interfaces that are randomly picked from 24 apps in D_{app}, each of which is the most popular app in its category. Similarly, these traces are split into the training and test set with a ratio of $8:2$.

Results. Since the recognition of the inter-interface switch depends on the identification of the type of interfaces(§ IV-B), we evaluate the accuracy of recognizing different types of interfaces. As shown in Fig. 11c, the overall accuracy of interface type recognition is 92.5%. Specifically, the recognition accuracy for the off screen is 98.0%, lock screen is 92.0%, home screen is 82.0%, and app interface is 98.0%. The accuracy of the home screen is lower than that of other types of interfaces. After investigation, the main reason lies in the similarity between the home screen and the lock screen, where we use the same background that consumes the most power, making these two interfaces appear similar in power consumption.

C. Magnetic-Field Based Recognition

Recognizing an app at launch. We build the dataset D_{applch} for this evaluation. Considering that apps on the smartphone are launched one by one, a static image will usually be displayed when an app is being launched, and different apps may vary in launching duration, we choose to collect the magnetometer readings for the first one second after clicking the app icon on the screen to represent the magnetic field perturbations during an app launch. Similar to collecting coil whine traces, each magnetic field perturbation trace also lasts for 0.1 seconds; each app in D_{app} will be repeated 100 times, resulting in 100 traces for each app launch. Since we have 360 apps, D_{applch} consists of 36,000 traces. We also use the $8:2$ split to generate the training and test set.

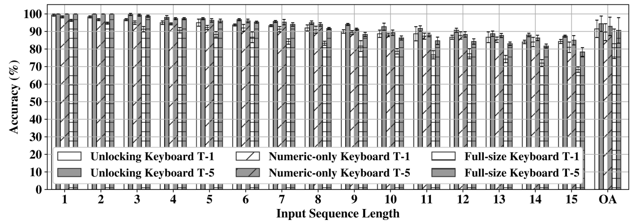


Fig. 15: Effectiveness of uncovering keystrokes on three soft keyboards (UK, NK, FK). Specifically, we evaluate WISERS+ by typing the randomly generated keystrokes with lengths from 1 to 15 (OA: Overall) on the iPhone 13 Pro charged by the Gikfun wireless charger.

Results. As shown in Fig. 12, the effectiveness of recognizing an app at its launch is evaluated in both the closed-world and open-world settings defined. As mentioned in § IV-D, in both settings, we use the same dataset to train the model, and this dataset is built by randomly selecting 80% of traces in 120 apps. Accordingly, to evaluate its effectiveness in the closed-world setting, the test set consists of the rest 20% traces in those 120 apps; and in the open-world setting, the test set includes (i) all traces in the rest 240 apps whose traces have not been used to train the model and (ii) the test set in the closed-world setting. Overall, the recognition accuracy in this closed-world setting is 91.8% with a standard deviation of 1.28%, and WISERS+ achieves an overall 87.9% recognition accuracy with a standard deviation of 1.27% in the open-world setting. The high accuracy and small standard deviation in both the closed-world and the open-world settings indicate the consistency of WISERS+ performance across apps in different categories, and this consistency is also observed in the recognition accuracy shown per category in Fig. 12.

Specifically, among 24 categories, WISERS+ performs the best in recognizing apps in "Navigation" (96.2% with 1.48% SDV (SDV means Standard Deviation)) and worst in "Medical" (89.4% with 1.95% SDV) in the closed-world setting, and best in "Navigation" apps (92.0% with 1.22% SDV) and worst in "weather" (85.4% with 2.19% SDV) apps in the open-world setting. As such, WISERS+ can robustly and consistently recognize apps at app launch in 0.1 seconds in both closed-world and open-world settings.

Recognizing in-app activities. We further explore WISERS+'s capability in uncovering fine-grained in-app activities. Specifically, we build the dataset D_{inapp} by collecting magnetometer data when performing eight in-app activities in YouTube (*e.g.*, play the video, fast forward) and WhatsApp (*e.g.*, send text, video meeting). We follow the same procedure to collect magnetic field perturbation in building D_{applch} and utilize the $8:2$ ratio to obtain the training and test sets. Note that due to the numerous number of in-app activities in a single app, we select the two most popular apps to study in-app activity recognition in WISERS+ as a proof-of-concept work.

Results. Fig. 13a and Fig. 13b show the results in recognizing eight common in-app activities in YouTube and WhatsApp, where we know WISERS+ achieves overall 91.3% and 90.4% accuracy, respectively. In particular, WISERS+ performs better in distinguishing in-app activities such as "Switch to next video" and "Refresh" in streaming apps like YouTube, as well as "Voice call" and "Share link"

TABLE II: End-to-end attack results. BL: battery level, CP: charging protocol OS: off screen, LS : lock screen, HS: home screen, AI: app interface, AR: app recognition, KO: keyboard opening, UK: unlock screen keyboard, NK: numeric-only keyboard, and FK: full-size keyboard, PRE: prediction results, T1: one attempt, T5: five attempts, "●": involved, "○": not involved.

# of Trial				Screen-unlocking Passcode							Cross-app Searching Content											App-specific Sensitive Inputs												
					Inter-interface Switch				PRE (✓/✗)			Inter-interface Switch				Intra-interface Activity				PRE (✓/✗)			Inter-interface Switch				App	Intra-interface Activity					PRE (✓/✗)	
	🔋 %	BL	CP	Input	OS	LS	HS	AI	T1	T5	Input	OS	LS	HS	AI	KO	UK	NK	FK	T1	T5	Input	OS	LS	HS	AI		AR	KO	UK	NK	FK	T1	T5
1	64	M	Qi	0149	●	●	●	○	✓	✓	whats	○	○	●	○	●	○	○	●	✓	✓	hello world	○	○	●	●	WhatsApp	●	●	○	○	●	✓	✓
2	31	M	Qi	0975	●	●	●	○	✓	✓	whatsap	○	○	●	○	●	○	○	●	✓	✓	nice day	○	○	●	●		●	●	○	○	●	✓	✓
3	46	M	Qi	032918	●	●	●	○	✓	✓	what	○	○	●	○	●	○	○	●	✓	✓	never mind	○	○	●	●		●	●	○	○	●	✓	✓
4	58	M	Qi	310867	●	●	●	○	✓	✓	whatsa	○	○	●	○	●	○	○	●	✓	✓	its freezing	○	○	●	●		●	●	○	○	●	✗	✓
5	87	H	Qi	1642185	●	●	●	○	✓	✓	wha	○	○	●	○	●	○	○	●	✓	✓	i really appreciate it	○	○	●	●		●	●	○	○	●	✗	✓
6	54	M	Qi	1896	●	●	●	○	✓	✓	teleg	○	○	●	○	●	○	○	●	✓	✓	hello world	○	○	●	●	Telegram	●	●	○	○	●	✗	✓
7	41	M	Qi	8261	●	●	●	○	✓	✓	telegram	○	○	●	○	●	○	○	●	✓	✓	nice day	○	○	●	●		●	●	○	○	●	✓	✓
8	51	M	Qi	033496	●	●	●	○	✓	✓	tele	○	○	●	○	●	○	○	●	✓	✓	never mind	○	○	●	●		●	●	○	○	●	✓	✓
9	68	M	Qi	3179826	●	●	●	○	✓	✓	tel	○	○	●	○	●	○	○	●	✓	✓	its freezing	○	○	●	●		●	●	○	○	●	✓	✓
10	12	L	Qi	0123456789	●	●	●	○	✗	✓	telegra	○	○	●	○	●	○	○	●	✓	✓	i really appreciate it	○	○	●	●		●	●	○	○	●	✗	✓
11	65	M	Qi	2537	●	●	●	○	✓	✓	payp	○	○	●	○	●	○	○	●	✓	✓	nfawst@gmail.com	○	○	●	●	PayPal	●	●	○	○	●	✓	✓
12	47	M	Qi	129540	●	●	●	○	✓	✓	pay	○	○	●	○	●	○	○	●	✓	✓	jfdrgcd@gmail.com	○	○	●	●		●	●	○	○	●	✓	✓
13	90	H	Qi	482359	●	●	●	○	✓	✓	pal	○	○	●	○	●	○	○	●	✓	✓	sgjczpoe@gmail.com	○	○	●	●		●	●	○	○	●	✗	✓
14	31	M	Qi	4682319	●	●	●	○	✓	✓	paypal	○	○	●	○	●	○	○	●	✓	✓	mcarxbyn@gmail.com	○	○	●	●		●	●	○	○	●	✓	✓
15	85	H	Qi	0022446688	●	●	●	○	✗	✓	pa	○	○	●	○	●	○	○	●	✓	✓	oxmlwuyi@gmail.com	○	○	●	●		●	●	○	○	●	✗	✓
16	14	L	Qi	3671	●	●	●	○	✓	✓	venmo	○	○	●	○	●	○	○	●	✓	✓	nfawst@gmail.com	○	○	●	●	Venmo	●	●	○	○	●	✓	✓
17	82	H	Qi	9430	●	●	●	○	✓	✓	ven	○	○	●	○	●	○	○	●	✓	✓	jfdrgcd@gmail.com	○	○	●	●		●	●	○	○	●	✓	✓
18	46	M	Qi	185437	●	●	●	○	✓	✓	venm	○	○	●	○	●	○	○	●	✓	✓	mcarxbyn@gmail.com	○	○	●	●		●	●	○	○	●	✓	✓
19	32	M	Qi	7342	●	●	●	○	✓	✓	v	○	○	●	○	●	○	○	●	✓	✓	sgjczpoe@gmail.com	○	○	●	●		●	●	○	○	●	✗	✓
20	71	M	Qi	8413620	●	●	●	○	✓	✓	ve	○	○	●	○	●	○	○	●	✓	✓	oxmlwuyi@gmail.com	○	○	●	●		●	●	○	○	●	✗	✓
21	73	M	Qi	4869	●	●	●	○	✓	✓	chrom	○	○	●	○	●	○	○	●	✓	✓	www.google.com	○	○	●	●	Chrome	●	●	○	○	●	✓	✓
22	99	H	Qi	159628	●	●	●	○	✓	✓	chro	○	○	●	○	●	○	○	●	✓	✓	www.yahoo.com	○	○	●	●		●	●	○	○	●	✗	✓
23	44	M	Qi	694330	●	●	●	○	✓	✓	chr	○	○	●	○	●	○	○	●	✓	✓	www.youtube.com	○	○	●	●		●	●	○	○	●	✗	✓
24	45	M	Qi	47526401	●	●	●	○	✗	✓	chrome	○	○	●	○	●	○	○	●	✓	✓	www.amazon.com	○	○	●	●		●	●	○	○	●	✓	✓
25	55	M	Qi	976013672	●	●	●	○	✗	✓	ch	○	○	●	○	●	○	○	●	✓	✓	www.walmart.com	○	○	●	●		●	●	○	○	●	✓	✓
26	68	M	Qi	5198	●	●	●	○	✓	✓	safa	○	○	●	○	●	○	○	●	✓	✓	www.google.com	○	○	●	●	Safari	●	●	○	○	●	✗	✓
27	72	M	Qi	257813	●	●	●	○	✓	✓	safari	○	○	●	○	●	○	○	●	✓	✓	www.yahoo.com	○	○	●	●		●	●	○	○	●	✓	✓
28	88	H	Qi	751943	●	●	●	○	✓	✓	safar	○	○	●	○	●	○	○	●	✓	✓	www.youtube.com	○	○	●	●		●	●	○	○	●	✓	✓
29	17	L	Qi	78787878	●	●	●	○	✗	✓	saf	○	○	●	○	●	○	○	●	✓	✓	www.amazon.com	○	○	●	●		●	●	○	○	●	✓	✓
30	33	M	Qi	643185310	●	●	●	○	✗	✓	sa	○	○	●	○	●	○	○	●	✓	✓	www.walmart.com	○	○	●	●		●	●	○	○	●	✗	✓
31	13	L	Qi	6263	●	●	●	○	✓	✓	swiss	○	○	●	○	●	○	○	●	✓	✓	013468764189	○	○	●	●	SwissCovid	●	●	○	●	○	✓	✓
32	36	M	Qi	330522	●	●	●	○	✓	✓	swi	○	○	●	○	●	○	○	●	✓	✓	167983578654	○	○	●	●		●	●	○	●	○	✓	✓
33	87	H	Qi	462183	●	●	●	○	✓	✓	swis	○	○	●	○	●	○	○	●	✓	✓	296794641236	○	○	●	●		●	●	○	●	○	✗	✓
34	41	M	Qi	843250	●	●	●	○	✓	✓	swissc	○	○	●	○	●	○	○	●	✓	✓	358784645231	○	○	●	●		●	●	○	●	○	✓	✓
35	17	L	Qi	987474501	●	●	●	○	✗	✓	sw	○	○	●	○	●	○	○	●	✓	✓	431654651568	○	○	●	●		●	●	○	●	○	✗	✓
36	24	M	Qi	2360	●	●	●	○	✓	✓	lh	○	○	●	○	●	○	○	●	✓	✓	84532761	○	○	●	●	LHSafe	●	●	○	●	○	✓	✓
37	10	L	Qi	950718	●	●	●	○	✓	✓	lhsa	○	○	●	○	●	○	○	●	✓	✓	76831025	○	○	●	●		●	●	○	●	○	✓	✓
38	35	M	Qi	825134	●	●	●	○	✓	✓	lhs	○	○	●	○	●	○	○	●	✓	✓	68543102	○	○	●	●		●	●	○	●	○	✓	✓
39	60	M	Qi	5253	●	●	●	○	✓	✓	lhsaf	○	○	●	○	●	○	○	●	✓	✓	53681279	○	○	●	●		●	●	○	●	○	✓	✓
40	89	H	Qi	47654432	●	●	●	○	✗	✓	lhsafe	○	○	●	○	●	○	○	●	✓	✓	46531640	○	○	●	●		●	●	○	●	○	✗	✓

in chat apps like WHATSAPP, as these activities cost more power consumption and induce distinctive patterns in captured magnetic field perturbations during wireless charging.

Identifying keyboard open. This experiment involves two datasets (*i.e.*, D_{skopen} and D_{sktype}) to evaluate the effectiveness of identifying whether a keyboard is open and recognizing the type of keyboard. In particular, these two datasets consist of data traces collected when opening the numeric-only keyboard and full-size keyboard in the same 24 apps that are used to build D_{switch}. Note that the screen-unlocking keyboard can only be opened on the lock screen. Specifically, to build D_{skopen}, we collect one-second data traces in a static interface and one-second data traces after the soft keyboard is open and becomes stable in the same interface, and the collection for each app is repeated 100 times. With respect to build D_{sktype}, we also collect one-second data traces in a static interface and one-second stable data traces after the soft keyboard is opened with one keystroke, and this process is repeated 100 times for each app. The collected traces in both datasets will be normalized as 0.1 seconds trace (§ IV-D). Therefore, in total, each D_{skopen} and D_{sktype} has 2,400 traces, which are split into a training and a testing set with the ratio of 8 : 2.

Results. As shown in Fig. 14a, while WISERS+ achieves a precision of 87.0% if there is no keyboard open, it cannot effectively distinguish between the numeric-only keyboard and full-size keyboard with less than 60% accuracy. However, if it involves a keystroke, as shown in Fig. 14b, it can successfully recognize whether a keyboard is open with 99.0% precision, the numeric-only keyboard with 97.0% accuracy, and the full-size keyboard with 89.0% correctness. Specifically, the main reason why its performance in recognizing the type of keyboard with and without a single keystroke varies is that these two keyboards almost occupy the same area and consume similar amounts of power. Due to the different size of each key, one keystroke could result in an energy consumption burst that could be significant enough to separate these two keyboards.

Inferring keystrokes. This evaluation consists of three datasets for different keyboards: *(i)* screen-unlocking keyboard (D_{kbds}), *(ii)* system numeric-only keyboard (D_{kbdn}), and *(iii)* system full-size keyboard (D_{kbdf}). To build the training set, each key including the static key used to separate keystrokes is clicked 100 times, and each time forms a magnetic field perturbation trace which is normalized to 0.1 as described in § IV-B. To create the test set, we randomly generate a sequence of characters[1] for each keyboard ranging from one character to 15 characters in length, and each sequence generates three testing cases, each of which is repeated 100 times. For example, the three testing cases with one character of the full-size keyboard are "u", "a", and "n". Note that test cases of a keyboard include all its individual keys.

Results. Fig. 15 shows the evaluation results on three different soft keyboards where the length of keystrokes ranges from one to 15; the overall accuracy for them are 91.5%, 89.6%, and 83.0%, respectively, with only one guess attempt. The accuracy can increase to 94.4%, 92.9%, and 90.6%, respectively, within five attempts. At a high level, within five attempts, the uncovering success rate of all three keyboards reaches the highest when there is only one character and decreases with length grows. In particular, within five attempts, WISERS+ can 100% correctly recover one-character keystroke in all three keyboards, while it achieves a precision of 87.3%, 85.0%, and 78.3% to uncover 15-character keystroke sequence from the screen unlocking keyboard, numeric-only keyboard, and full-size keyboard, respectively. Note that this accuracy is comparable to other works [6], which shows the ability of WISERS+ to accurately uncover keystrokes. Moreover, five attempts can significantly increase the accuracy in keystroke inference than the one-time attempt, especially in recovering 15-character keystroke sequence on the full-size keyboard with 10% increase from 68.3% to 78.3%.

VI. End-to-End Attacks

End-to-end attack scenarios. The end-to-end success rate is crucial for assessing an attack framework's efficacy. In WISERS+, due to the interdependence of components, the overall success rate is not simply a product of the accuracies of the individual components. Therefore, to accurately gauge the success rate, we have carried out end-to-end attack experiments. Each experiment aims to accurately identify all user interactions in a sequence, starting with screen unlocking using a passcode, progressing to cross-app searches in the home screen, and culminating in app launch and input of sensitive information. The end-to-end success rate can be calculated as *success rate = (number of success trials) / (number of all trials)*. In particular, we have prepared 40 screen unlocking passcodes that are randomly generated where there are 13 four-digit, 15 six-digit, and 12 custom length passcodes. With respect to cross-app search keywords, we provide 40 different keywords particular to eight popular apps (five keywords of different lengths for each app): two chat apps (WhatsApp and Telegram), two financial apps (Paypal and Venmo), two browsers (Chrome and Safari), and two Covid-19 apps (SwissCovid and LHSafe). Specifically, we prepared five arbitrary sentences to type in these two chatting apps, five randomly generated gmail addresses as user accounts to use in two financial apps, five popular website URLs to visit in two browsers, five randomly generated 12-digit Covid case serial numbers for SwissCovid, and five 8-digit mobile numbers for LHSafe. Note that these two Covid-19 apps pop up the numeric-only keyboard when asking for sensitive unique information. In total, there are 40 attack trials, and each trial involves a screen unlocking passcode, one cross-app search keyword, one app, and one app-specific user input.

End-to-end attack results. Table II presents the detailed results of our end-to-end attack. In this attack, WISERS+ achieves a 100% overall success rate in the 40 end-to-end attacks in at most five attempts to recover user input without a single wrong character. In addition, even under the strictest standard where only one attempt is allowed to recover user input, more than half of all trials (*i.e.*, 22 out of 40) can still succeed without a single mispredicted character. Each failed case in the remaining trials is in the length of 14 on average, and each only contains *one* mispredicted character. The detailed analysis of each stage is elaborated in the following.

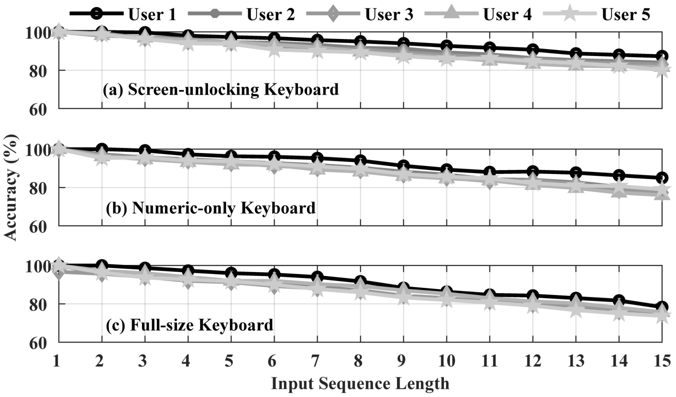


Fig. 16: Impact factor of users on uncovering keyboard inputs.

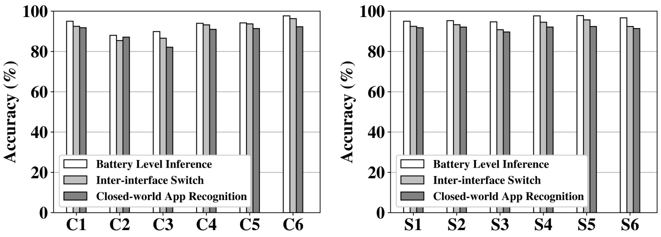
Revealing screen-unlocking passcode. WISERS+ determines the screen-unlocking passcode by analyzing the unlocking sequence, which comprises a keystroke and two series of interface transitions. The first transition sequence progresses from the off-screen to the lock screen and concludes at the home screen, while the second follows a similar path but ends at an app interface. The transition to the after-lock screen is crucial, as it confirms successful unlocking and the accuracy of the entered passcode. As shown in Table II, WISERS+ has successfully inferred all inter-interfaces switches and recovered all passcodes within at most five attempts. In particular, for failed ones if using only one attempt, there are four eight-digit, two nine-digit, and two ten-digit passcodes, and all eight failed attempts mispredict only one digit[1].

Revealing cross-app searching content. To capture the essence of cross-app search activities, the context is characterized by a series of inter-interface transitions across home screens, combined with intra-interface actions like keyboard activation and typing. Table II show the summary of the attack results[1]. In particular, all inter-interface switches, keyboard opening, and search content have been successfully recognized and recovered without a single failure, achieving a 100% success rate.

Revealing app-specific sensitive inputs. The user interaction context encompasses navigating from the home screen to an app's interface, which may include transitions between various sections within the app. For precise identification of actions, three intra-interface activities are essential: launching the app, opening the keyboard, and typing keystrokes. As shown in Table II, all inter-interface switches and keyboard openings in these attempts are accurately recognized and identified[1]. In addition, WISERS+ needs more attempts to successfully recover a typing word, especially when there is a character in such a word appearing consecutively, such as "ee" in the word "freezing", or the corresponding key of a character has a relatively smaller space than the other keys, such as "." in the middle of "gmail.com". Similarly, within at most five attempts, all user chat content is recovered.

VII. Impact Factors

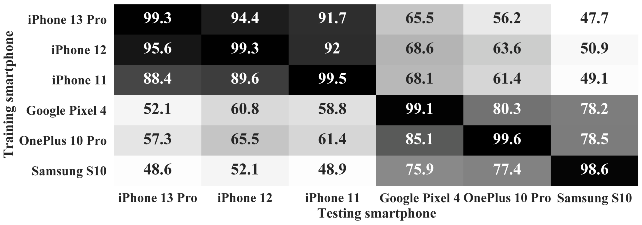
Impacts from different users. Given prior research indicating variability in user keystroke durations, WISERS+ adjusts each keystroke trace to mitigate these discrepancies in



(a) Different wireless chargers.

(b) Different smartphones.

Fig. 17: Impact factor analysis of wireless chargers and smartphones.



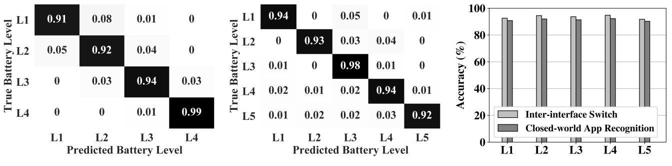
Training smartphone	iPhone 13 Pro	iPhone 12	iPhone 11	Google Pixel 4	OnePlus 10 Pro	Samsung S10
iPhone 13 Pro	99.3	94.4	91.7	65.5	56.2	47.7
iPhone 12	95.6	99.3	92	68.6	63.6	50.9
iPhone 11	88.4	89.6	99.5	68.1	61.4	49.1
Google Pixel 4	52.1	60.8	58.8	99.1	80.3	78.2
OnePlus 10 Pro	57.3	65.5	61.4	85.1	99.6	78.5
Samsung S10	48.6	52.1	48.9	75.9	77.4	98.6

Fig. 18: Transferability of different smartphones.

keystroke recovery analysis. Furthermore, we conducted a user study approved by the IRB to assess its efficacy. In particular, we have recruited four more volunteers (two males and two females) to join this study in the analysis of keystroke recovery, and these volunteers were asked to conduct the same experiments following the same procedures when we build our keystroke evaluation dataset (*i.e.*, D_{kbds}, D_{kbdn}, and D_{kbdf}) in § V-A. By using the same classification model trained on these three datasets, as shown in Fig. 16, the accuracy rates for different lengths of sequential keystrokes show a similar trend that decreases slightly as the length grows, and the accuracy difference is within 8% at most between those of these volunteers and us. *Therefore, the empirical results indicate* WISERS+ *is practical for cross-user attacks.*

Impacts from different wireless chargers. Commercially available wireless chargers exhibit varying levels of noise suppression, particularly coil whine, and magnetic field shielding. These variations can significantly influence the effectiveness of analyses based on coil whine and magnetic field perturbations. To evaluate such impacts, we evaluate WISERS+ on another five popular wireless chargers, *i.e.*, C_1: Apple MagSafe Charger (A2140), C_2: Samsung Wireless Charger Stand (EP-N5200TBEGGB), C_3: Mophie Snap+ 15W (SNP-WRLS-CHGR), C_4: Belkin BOOSTUP 7.5W (F7U054), and C_5: Baseus Mini Magnetic 15W (BS-W522), and compare the results with those obtained from 10 W Gikfun charger used in our previous effectiveness evaluation. Specifically, WISERS+ was improved by training new models with data from five commercially available products. We adhered to the same data collection and evaluation methodology to evaluate the system's capability to determine battery levels, detect interface switches, and identify applications upon their initiation, as detailed in § V.

As shown in Fig. 17a, WISERS+ can achieve high precision in all six evaluations across different COTS wireless chargers. In particular, the accuracy of these chargers in battery level inference ranges from 89.9% to 98.0% (2.96% SDV (SDV means Standard Deviation)); in inter-interface switch recognition ranges from 85.4% to 96.3% (4.31% SDV); and in the app recognition at launching with a closed-world setting ranges from 82.1% to 92.3% (3.98% SDV), respectively. The findings indicate that Apple's MagSafe outperforms in mitigating coil whine noise, while the Samsung Wireless Charger Stand is more effective in minimizing magnetic field perturbations. *As such,* WISERS+ *can be applied to various COTS wireless chargers and achieve similar performance.*



	L1	L2	L3	L4
L1	0.91	0.08	0.01	0
L2	0.05	0.92	0.04	0
L3	0	0.03	0.94	0.03
L4	0	0	0.01	0.99

	L1	L2	L3	L4	L5
L1	0.94	0	0.05	0	0.01
L2	0	0.93	0.03	0.04	0
L3	0.01	0	0.98	0.01	0
L4	0.02	0.01	0.02	0.94	0.01
L5	0.01	0.02	0.02	0.03	0.92

(a) Mophie Snap+.

(b) Belkin BOOSTUP.

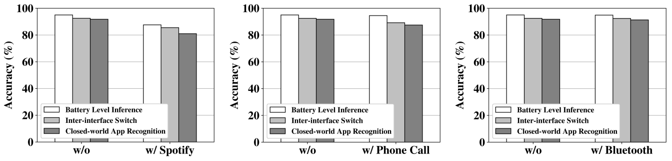
(c) Evaluation results.

Fig. 19: (a) and (b): Battery level inference of wireless chargers with more than three levels. (c) Effectiveness of inter-interface switch and closed app recognition when we evaluate WISERS+ on the Belkin BOOSTUP wireless charger at different battery levels (L_1–L_5) during the process of charging the iPhone 13 Pro wirelessly.

Impacts from different smartphone models. Smartphones, even from the same manufacturer, can exhibit varying energy consumption and management due to differing hardware and software strategies between models (*e.g.*, different iPhones). To investigate its impact, we follow the same procedures as before to evaluate six different iOS and Android smartphones, *i.e.*, S_1: iPhone 13 Pro, S_2: iPhone 12, S_3: iPhone 11, S_4: Google Pixel 4, S_5: OnePlus 10 Pro, and S_6: Samsung S10. The results in Fig. 17b show that these six classification models of different smartphones achieve accuracy from 94.7% to 97.8% (1.38% SDV) in battery level inference, 90.8% to 95.7% (1.72% SDV) in inter-interface switch, and 89.7% to 92.4% (0.98% SDV) in closed-world app recognition.

In addition, we also evaluate the model transferability by training a classification model for each smartphone and applying each trained model to all six smartphones. As shown in Fig. 18, if applying the classification model to the smartphone this model is trained from, these six models all achieve similar high accuracy ranging from 98.6% to 99.6%; however, if applying a model for other smartphones, the accuracy will decrease to different degrees. Within smartphones running the same OS, the accuracy of the classification model trained for iPhone 13 Pro and iPhone 12 slightly decreases by around 5%, while the model of iPhone 11 decreases roughly 10%. The differences may result from the varying screen techniques used in them, where iPhone 11 uses LCD while the other two use OLED. On the other hand, the accuracy of the model trained from the three Android smartphones decreases by around 14–23% due to their different keyboard layouts. Furthermore, the accuracy decreases with 30% as we transfer the model trained from an iOS smartphone to an Android smartphone because their screen techniques and UI layouts are extremely different. In short, *though performance might decrease,* WISERS+ *can also work for cross-device attacks.*

Impacts from different battery levels. Based on the wireless charging protocol, the same user activity could induce different



(a) Background app. (b) Phone call. (c) Bluetooth.

Fig. 20: Effectiveness of WISERS+ under impact factors such as running background apps (*e.g.*, music apps like SPOTIFY), interacting during a phone call, and Bluetooth connections.

patterns of coil whine and magnetic field perturbations if the smartphone is at different battery levels [8]. We comprehensively investigate 26 commodity wireless chargers with their battery levels in the charging process [47], and there are 17 wireless chargers that have three levels or less (*e.g.*, Anker 10W charger), eight have four levels (*e.g.*, Mophie Snap+ 15W), and only Belkin 7.5W wireless charger has five levels. Therefore, similar to the experiment in battery level inference, we also evaluated WISERS+ in wireless chargers with more than three charging battery levels. As shown in Fig. 19a and Fig. 19b, WISERS+ achieves 94.0% and 94.2% accuracy in recognizing battery levels from Mophie Snap+ 15W (four levels) and Belkin 7.5W BOOSTUP (five levels), respectively. We also collected data at different battery levels from Belkin 7.5W BOOSTUP to train five models. As shown in Fig. 19c, models trained from level one ($L1$) to level five ($L5$), achieve an accuracy ranging from 91.8% to 94.8% (1.27% SDV) in recognizing inter-interface switches, and an accuracy ranging from 90.3% to 92.3% (0.91% SDV) in app launch, respectively. *Therefore,* WISERS+ *could be used to launch attacks on different battery levels of a charger regardless of the number of levels a charger has.*

Impacts from background apps, phone calls and Bluetooth connections. To further demonstrate WISERS+'s robustness, we then evaluated its performance on the iPhone 13 Pro charged by the Gikfun wireless charger under three common scenarios. First, we conducted experiments when running another music app, SPOTIFY, in the background, and Fig. 20a shows that the performance of WISERS+ decreases by approximately 8.4%, revealing that background apps may have impacts on WISERS+ because they bring extra energy consumption and affect the charging process. Second, Fig. 20b shows the performance when we make a phone call to the charging smartphone, where WISERS+ only decreases 2.7% on average and a phone call presents a subtle influence as it has no impact on the electromagnetic field. Third, Fig. 20c indicates that Bluetooth connections have almost no impact because BLE modules consume extremely low power that cannot affect the charging current. *Hence,* WISERS+ *is resilient to these impact factors in real-world scenarios.*

Analysis of other impact factors. Apart from analyzing the above impact factors, we also analyze several other impact factors, such as the app interfaces when typing on a keyboard, the distance and orientation angles between the wireless charger and the attacking smartphone, and the number of acoustic features. Our evaluation shows that WISERS+ is also practically resilient to these factors. In particular, inferring keystrokes is resilient to the impacts of different app interfaces, and there is only an average 5% decrease in accuracy. Furthermore, if enlarging the attack distance from 8in (20cm) to 12in (30cm) and 16in (40cm), WISERS+ can still achieve the accuracy of 86.4% and 80.8% in using coil whine to infer inter-interface switches and 90.2% and 77.3% in using magnetic field perturbations to infer keystrokes, respectively. Next, WISERS+ achieves an accuracy of 85.9% on average in recognizing inter-interface switches and 91.4% in recovering keystrokes when the relative orientation angle between the charging devices and the attacking smartphone is changed to 30°, 60°, and 90°. In addition, we evaluated inter-interface switches with a different number of acoustic features, and the results show that it achieves the accuracy of 83.9%, 91.4%, and 92.5% if applying 39, 78, and 86 features, respectively.

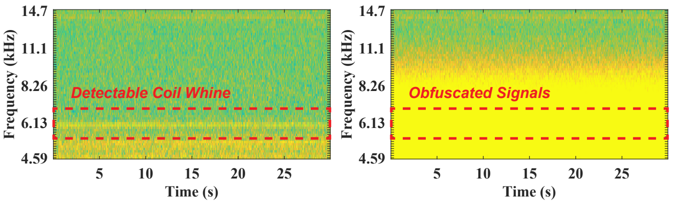
TABLE III: Empirical results of extended attacks of WISERS+ on fast charging AirVOOC 50W wireless charger. BL: battery level inference, IIS: inter-interface switch, AF: app fingerprinting.

Protocol	Smartphone	Coil Whine Freq.	BL	IIS	AF
Qi	OnePlus 10 Pro	5.98–6.16 kHz	97.8%	95.7%	92.4%
	OPPO Find X3 Pro		95.6%	94.0%	90.3%
AirVOOC	OnePlus 10 Pro	12.20–12.45 kHz	88.3%	87.5%	80.1%
	OPPO Find X3 Pro		85.5%	86.4%	78.7%

VIII. DISCUSSION

A. Extend Attacks on Fast Charging

To demonstrate that WISERS+ can be extended to launch attacks on the fast charging mode of wireless chargers, we use the AirVOOC 50W wireless charger to charge a OnePlus 10 Pro and an OPPO Find X3 Pro, and then collect data samples for evaluation. In practice, we detect the expected leakages of coil whine and magnetic field perturbations by an iPhone 11's microphone with a sampling frequency 44.1kHz. Then, we apply a moving-variance sliding window to divide the recorded signals and obtain the frequency range of the coil whines in the AirVOOC fast charging mode. Table III shows the results that include the frequency of coil whine detected in the AirVOOC mode and recognition accuracy in battery level inference (85.5%–88.3%), inter-interface switches (86.4%–87.5%), and intra-interface activities such as app fingerprinting (78.7%–80.1%), which demonstrated that WISERS+ is still effective with the detected coil whine in the AirVOOC protocol. Compared with attack results in § V, WISERS+'s performance degrades about 10% when switching to the fast-charging mode. That is because the AirVOOC fast charging requires a transmitting power in a large current in the primary coil, which induces a strong electromagnetic field with a lot of noise. For example, the frequency range of the coil whine increases from 5.98–6.16 kHz to 12.20–12.45 kHz when switching the charging protocol from Qi to AirVOOC fast charging mode. Hence, the distinctive patterns inside the coil whine and magnetic field perturbations generated by users' interactions could be overwhelming. However, WISERS+ still shows the feasibility of extending attacks on wireless chargers with fast charging protocols.



(a) Original coil whine.

(b) After obfuscation.

Fig. 21: Power spectrograms of coil whine before/after obfuscation.

B. Countermeasures

To defend against this new side-channel attack, our proposed countermeasure is to protect the coil whine and magnetic field perturbations from being eavesdropped and exploited. Below, we have designed and implemented several countermeasures from the perspective of passive and proactive.

Passive countermeasures. To mitigate the risks of eavesdropping, utilizing advanced materials to build coils to minimize noise, such as coil whine, and enhance magnetic field shielding can effectively obscure signals, making them challenging to intercept and analyze. However, implementing this strategy in existing products presents challenges, and its effectiveness and cost-effectiveness require further scrutiny [48].

Proactive countermeasures. To enhance wireless chargers' security, we propose the integration of proactive protection methods. This necessity stems from user interactions leaving unique signatures in coil whine and magnetic field disturbances. By intentionally injecting random noise, such as Gaussian white noise [17], into the coil whine, we can effectively disrupt potential malicious analyses. Illustrative evidence from power spectrograms, like those depicted in Fig. 21, demonstrates that these measures significantly obscure the characteristic patterns of coil whine when we apply a 6 kHz Gaussian white noise, makes distinct traits less discernible. Furthermore, varying the amplitude and frequency of the primary coil's voltage can similarly obscure the magnetic field signals. However, it is important to note that these alterations may compromise the efficiency of wireless charging.

In addition, to protect users from WISERS+ in uncovering sensitive keystrokes, it has been shown that shuffling soft keyboards [5], [28], [49] could be an effective approach, as the attacker cannot know the randomized layout of the keyboard and cannot further infer sensitive keystrokes. Hence, WISERS+ cannot effectively recover the specific keystrokes from the input of a shuffled keyboard. However, deploying shuffling virtual keyboards for each interface inside mobile devices such as smartphones could increase the time spent typing the keyboard, further lower functionality and usability for smartphone users, and consume more battery power to run the shuffling algorithm and generate redundant keyboards.

C. Limitations and Future Works

We have demonstrated that WISERS+ is resilient to various impact factors (§ VII) Nevertheless, the performance could be degraded when the attack distance is increased due to the signal attenuation of both coil whine and magnetic field perturbations. However, it is worth noting that close proximity attack scenarios are common in daily lives where adversaries can exploit this newly-discovered side channel to launch attacks. Additionally, WISERS+'s performance will decrease if directly transferring pre-trained models from one mobile OS to another (*i.e.*, iOS to Android), and we will tackle this problem in future endeavors. Although WISERS+ does not take into account the magnetic interference of neighborhood devices, recent studies have demonstrated that its effect only appears at a very close distance (*e.g.*, less than 1cm [50]). Similarly, the background apps might mislead WISERS+ in a few specific circumstances (*e.g.*, high run-sleep ratio [51]). We will evaluate and address them in future work. Moreover, the current prototype may not uncover sensitive user inputs if a system-level auto-filling mechanism exists. For example, the system may autofill the user input "ve" to "venmo". Using commodity large language models (LLMs), *e.g.*, ChatGPT, could generate several word candidates from inferred incomplete keys to address this challenge [52]. The current prototype also assumes an awareness of the distance and relative angle between the attacking device and the target charger. This limitation might be alleviated by tweaking existing solutions proposed to detect distance and angle by using magnetometers [53] and additional sensors [54], such as the accelerometer and gyroscope, which have already been integrated into most commodity smartphones, which are also listed as another future work.

IX. Related Work

Wireless charging side-channel attacks. Recently, side-channel attacks on wireless chargers have been widely studied. However, existing works require either compromise the wireless charger's power line to acquire current traces [8] or inductive voltages [9], or placing a hidden coil in a close distance of 3.2cm to monitor the inductive currents [17], which are impractical in real-life scenarios. Furthermore, recent studies have shown the feasibility of injecting inaudible audio commands into voice assistants through a maliciously designed wireless charger [10], [18]. Unlike these works, we propose the first *contactless and context-aware* framework, dubbed WISERS+, that leverages the emitted coil whine and magnetic field perturbations appearing in the wireless charging process to uncover fine-grained user-smartphone interactions in more practical scenarios with much looser assumptions. In addition, WISERS+ presents the ability to attack chargers with not only the Qi protocol but also other fast charging protocols (*e.g.*, AirVOOC [15], SuperVOOC [16]).

Power-based side-channel attacks. Numerous research endeavors focus on investigating side channels from the power traces of smartphones. These types of attacks are launched under the assumption that adversaries can compromise the victim's smartphones (*e.g.*, by installing malware [4], [11], [12], [14]) or gain access to power cables (*e.g.*, USB cables) or power stations to monitor current/voltage traces and perform activities such as app fingerprinting [55], [56], location tracking [12], and extracting privacy-sensitive information [5], [57]–[59]. Recently, AppListener [40] has taken advantage of the available Wi-Fi power to infer app activities. In contrast to these existing approaches, WISERS+ takes a different approach by eliminating the need for power profiles and making no assumptions about the victim's device.

Electromagnetic (EM) side-channel attacks. Many research studies on electromagnetic (EM) side channels require the use of specialized EM probes to capture EM signals for purposes such as reverse engineering neural networks [60], monitoring program execution on microcontrollers [61], [62], detecting microphone status [63], extracting secret keys [64]–[67], recognizing security codes from touchscreens [50], extracting credit card tokens [68], inferring running apps keystrokes from PCs [69] and smartphones [28], [49], injecting/eavesdropping audios [70], [71], and recovering 2D fingerprints [72] from in-display fingerprint sensors or images from embedded cameras [73]. These EM-based attacks are typically conducted in close proximity, ranging from less than 1cm [50], [60]–[62], [64]–[67], [72]–[74] to 2.5–5cm [17], [68], [69], [75] or even 20–90cm [28]. Unlike these existing approaches, WISERS+ introduces a novel side channel and uses commodity smartphones to analyze user interactions on another victim's smartphone that is being wirelessly charged.

X. Conclusion

In this paper, we present a novel wireless charging side channel that utilizes in a *contactless* and *context-aware* manner. Using coil whine and magnetic field perturbations generated during the wireless charging process, our approach enables the inference of sensitive user interactions on the charging smartphone. To demonstrate the feasibility of this side channel, we have developed WISERS+, a three-stage attack framework. To our knowledge, WISERS+ becomes the *first* attack that exploits contactless physical signals emitted from wireless charging to extract sensitive information from smartphones. Based on comprehensive evaluations, we demonstrate the effectiveness of WISERS+ in inferring user interactions while remaining resilient to various practical impact factors and presenting the ability to attack fast charging systems.

Acknowledgment

This paper was supported by the Research Grants Council (RGC) of Hong Kong under Grants ECS (CityU 21201420), CRF (8730087), GRF (CityU 11201422), GRF (CityU 11202124), and CityU SRG-Fd funding (CityU 11205323 and 7005984), and by the Hong Kong ITF Project (PRP/037/23FX, MHP/072/23, GHP/052/19SZ), the Research and Development Program of Shenzhen under Grants SGDX20190918101201696, NSFC (62202405), and the Post-Doctoral Science Foundation (2023M732791). Any opinions, findings, and conclusions in this paper are those of the authors and are not necessarily of supported organizations.

References

[1] D. Van Wageningen and T. Staring, "The Qi wireless power standard," in *Proceedings of 14th International Power Electronics and Motion Control Conference (EPE-PEMC)*. IEEE, 2010, pp. S15–25.

[2] Fortune Business Insight, "Wireless charging market size," https://www.fortunebusinessinsights.com/wireless-charging-market-105183.

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

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

[5] P. Cronin, X. Gao, C. Yang, and H. Wang, "Charger-surfing: Exploiting a power line side-channel for smartphone information leakage," in *Proceedings of the 30th USENIX Security Symposium*, 2021.

[6] P. Cronin, X. Gao, H. Wang, and C. Cotton, "An exploration of ARM system-level cache and GPU side channels," in *Proceedings of the Annual Computer Security Applications Conference (ACSAC)*, 2021.

[7] Y. Wang, H. Guo, and Q. Yan, "Ghosttalk: Interactive attack on smartphone voice system through power line," in *Proceedings of the Network and Distributed Systems Security (NDSS) Symposium*, 2022.

[8] A. S. La Cour, K. K. Afridi, and G. E. Suh, "Wireless charging power side-channel attacks," in *Proceedings of the ACM SIGSAC Conference on Computer and Communications Security (CCS)*, 2021, pp. 651–665.

[9] J. Liu, X. Zou, L. Zhao, Y. Tao, S. Hu, J. Han, and K. Ren, "Privacy leakage in wireless charging," *IEEE Transactions on Dependable and Secure Computing*, 2022.

[10] Z. Zhan, Y. Yang, H. Shan, H. Wang, Y. Jin, and S. Wang, "Voltschemer: Use voltage noise to manipulate your wireless charger," *arXiv preprint arXiv:2402.11423*, 2024.

[11] Y. Chen, X. Jin, J. Sun, R. Zhang, and Y. Zhang, "Powerful: Mobile app fingerprinting via power analysis," in *Proceedings of the International Conference on Computer Communications (INFOCOM)*, 2017.

[12] Y. Michalevsky, A. Schulman, G. A. Veerapandian, D. Boneh, and G. Nakibly, "Powerspy: Location tracking using mobile device power analysis," in *Proceedings of the USENIX Security Symposium*, 2015.

[13] N. Matyunin, Y. Wang, T. Arul, K. Kullmann, J. Szefer, and S. Katzenbeisser, "Magneticspy: Exploiting magnetometer in mobile devices for website and application fingerprinting," in *Proceedings of the 18th ACM Workshop on Privacy in the Electronic Society*, 2019, pp. 135–149.

[14] X. Zhou, S. Demetriou, D. He, M. Naveed, X. Pan, X. Wang, C. A. Gunter, and K. Nahrstedt, "Identity, location, disease and more: Inferring your secrets from Android public resources," in *Proceedings of the ACM SIGSAC conference on Computer and Communications Security (CCS)*, 2013, pp. 1017–1028.

[15] OPPO, "Airvooc – world leading wireless charging," https://www.oppo.com/en/newsroom/stories/airvooc-world-leading-wireless-charging/.

[16] Calvin Wankehede, "Supervooc fast charging technology: Everything you need to know," https://www.androidauthority.com/supervooc-fast-charging-686000/.

[17] 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 *Proceedings of the Annual Computer Security Applications Conference (ACSAC)*, 2021, pp. 916–929.

[18] D. Dai, Z. An, and L. Yang, "Inducing wireless chargers to voice out for inaudible command attacks," in *Proceedings of the IEEE Symposium on Security and Privacy (SP)*, 2023, pp. 503–520.

[19] Wireless Power Consortium, "Download the Qi specifications," https://www.wirelesspowerconsortium.com/knowledge-base/specifications/download-the-qi-specifications.html.

[20] Wikipedia, "Inductive charging," https://en.wikipedia.org/wiki/Inductive_charging.

[21] A. Belahcen *et al.*, *Magnetoelasticity, magnetic forces and magnetostriction in electrical machines*. Helsinki University of Technology, 2004.

[22] Wikipedia, "Electromagnetically induced acoustic noise," https://en.wikipedia.org/wiki/Electromagnetically_induced_acoustic_noise.

[23] Claire, "Should you be concerned if your wireless charger makes an unusual noise," https://global.ipitaka.com/blogs/news/should-you-be-concerned-if-your-wireless-charger-makes-an-unusual-noise?, 2020.

[24] Y. Cai, Y. Zhao, X. Ding, and J. Fennelly, "Magnetometer basics for mobile phone applications," *Electron. Prod. (Garden City, New York)*, vol. 54, no. 2, 2012.

[25] G. Zhang, C. Yan, X. Ji, T. Zhang, T. Zhang, and W. Xu, "Dolphinattack: Inaudible voice commands," in *Proceedings of the ACM SIGSAC Conference on Computer and Communications Security (CCS)*, 2017.

[26] S. Maruyama, S. Wakabayashi, and T. Mori, "Tap'n ghost: A compilation of novel attack techniques against smartphone touchscreens," in *Proceedings of the IEEE Symposium on Security and Privacy (S&P)*. IEEE, 2019, pp. 620–637.

[27] M. Li, Y. Meng, J. Liu, H. Zhu, X. Liang, Y. Liu, and N. Ruan, "When CSI meets public WiFi: inferring your mobile phone password via WiFi signals," in *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security (CCS)*, 2016.

[28] W. Jin, S. Murali, H. Zhu, and M. Li, "Periscope: A keystroke inference attack using human coupled electromagnetic emanations," in *Proceedings of the ACM SIGSAC Conference on Computer and Communications Security (CCS)*, 2021, pp. 700–714.

[29] X. Yang, "Efficient circular arc interpolation based on active tolerance control," *Computer-Aided Design*, vol. 34, no. 13, pp. 1037–1046, 2002.

[30] M. Tan, J. Wan, Z. Zhou, and Z. Li, "Invisible probe: Timing attacks with PCIe congestion side-channel," in *Proceedings of the IEEE Symposium on Security and Privacy (S&P)*. IEEE, 2021, pp. 322–338.

[31] 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 sensing of Environment*, vol. 91, no. 3-4, pp. 332–344, 2004.

[32] B. Yang, R. Chen, K. Huang, J. Yang, and W. Gao, "Eavesdropping user credentials via GPU side channels on smartphones," in *Proceedings of the ACM International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS)*, 2022.

[33] O. Rukundo and H. Cao, "Nearest neighbor value interpolation," *arXiv preprint arXiv:1211.1768*, 2012.

[34] D. M. Kreindler and C. J. Lumsden, "The effects of the irregular sample and missing data in time series analysis," in *Nonlinear Dynamical Systems Analysis for the Behavioral Sciences Using Real Data*, 2016.

[35] R. Ning, C. Wang, C. Xin, J. Li, and H. Wu, "Deepmag+: Sniffing mobile apps in magnetic field through deep learning," *Pervasive and Mobile Computing*, vol. 61, p. 101106, 2020.

[36] 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 *Proceedings of the 29th USENIX Security Symposium*, 2020, pp. 2685–2702.

[37] M. Sahidullah and G. Saha, "Design, analysis and experimental evaluation of block based transformation in MFCC computation for speaker recognition," *Speech communication*, vol. 54, no. 4, pp. 543–565, 2012.

[38] A. Adiga, M. Magimai, and C. S. Seelamantula, "Gammatone wavelet cepstral coefficients for robust speech recognition," in *Proceedings of the IEEE TENCON*, 2013.

[39] S. McCandless, "An algorithm for automatic formant extraction using linear prediction spectra," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 22, no. 2, pp. 135–141, 1974.

[40] T. Ni, G. Lan, J. Wang, Q. Zhao, and W. Xu, "Eavesdropping mobile app activity via {Radio-Frequency} energy harvesting," in *32nd USENIX Security Symposium (USENIX Security 23)*, 2023, pp. 3511–3528.

[41] J. Li, H. Zhou, S. Wu, X. Luo, T. Wang, X. Zhan, and X. Ma, "FOAP: Fine-grained open-world android app fingerprinting," in *Proceedings of the 31st USENIX Security Symposium*, 2022.

[42] P. Sirinam, N. Mathews, M. S. Rahman, and M. Wright, "Triplet fingerprinting: More practical and portable website fingerprinting with n-shot learning," in *Proceedings of the ACM SIGSAC Conference on Computer and Communications Security (CCS)*, 2019, pp. 1131–1148.

[43] T. Wang, "High precision open-world website fingerprinting," in *Proceedings of the IEEE Symposium on Security and Privacy (S&P)*, 2020.

[44] App Store, "Audio Recorder," https://apps.apple.com/us/app/audio-recorder-wav-m4a/id14544888.

[45] ——, "Sensor Logger," https://apps.apple.com/us/app/sensorlogger-csv-export/id15052035.

[46] Similarweb, "Top Apps Ranking," https://www.similarweb.com/apps/top/apple/store-rank/us/all/top-free/iphone/.

[47] ChargingLab, https://www.chargerlab.com/.

[48] T. Gluck, R. Puzis, Y. Oren, and A. Shabtai, "The curious case of the curious case: Detecting touchscreen events using a smartphone protective case," in *Proceedings of the IEEE European Symposium on Security and Privacy Workshops (EuroS&PW)*, 2017.

[49] T. Ni, X. Zhang, C. Zuo, J. Li, Z. Yan, W. Wang, W. Xu, X. Luo, and Q. Zhao, "Uncovering user interactions on smartphones via contactless wireless charging side channels," in *2023 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2023, pp. 3399–3415.

[50] Z. Liu, N. Samwel, L. Weissbart, Z. Zhao, D. Lauret, L. Batina, and M. Larson, "Screen gleaning: A screen reading TEMPEST attack on mobile devices exploiting an electromagnetic side channel," in *Proceedings of the Network and Distributed Systems Security (NDSS) Symposium*, 2020.

[51] H. Pan, L. Yang, H. Li, C.-W. You, X. Ji, Y.-C. Chen, Z. Hu, and G. Xue, "Magthief: Stealing private app usage data on mobile devices via built-in magnetometer," in *Proceedings of the International Conference on Sensing, Communication, and Networking (SECON)*, 2021.

[52] J.-B. Mouret, "Large language models help computer programs to evolve," 2024.

[53] M. Wang, Q. Luo, Y. Iravantchi, X. Chen, A. Sample, K. G. Shin, X. Tian, X. Wang, and D. Chen, "Automatic calibration of magnetic tracking," in *Proceedings of the 28th Annual International Conference on Mobile Computing And Networking (MobiCom)*, 2022, pp. 391–404.

[54] H. Huang, H. Chen, and S. Lin, "Magtrack: Enabling safe driving monitoring with wearable magnetics," in *Proceedings of the International Conference on Mobile Systems, Applications, and Services*, 2019.

[55] 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 *Proceedings of the 29th Annual International Conference on Mobile Computing and Networking*, 2023, pp. 1–15.

[56] T. Ni, J. Li, X. Zhang, C. Zuo, W. Wang, W. Xu, X. Luo, and Q. Zhao, "Exploiting contactless side channels in wireless charging power banks for user privacy inference via few-shot learning," in *Proceedings of the 29th Annual International Conference on Mobile Computing and Networking*, 2023, pp. 1–15.

[57] T. Ni, Y. Chen, K. Song, and W. Xu, "A simple and fast human activity recognition system using radio frequency energy harvesting," in *Adjunct Proceedings of the 2021 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2021 ACM International Symposium on Wearable Computers*, 2021.

[58] Z. Sun, T. Ni, H. Yang, K. Liu, Y. Zhang, T. Gu, and W. Xu, "Flora+: Energy-efficient, reliable, beamforming-assisted, and secure over-the-air firmware update in lora networks," *ACM Transactions on Sensor Networks*, 2024.

[59] R. Spolaor, H. Liu, F. Turrin, M. Conti, and X. Cheng, "Plug and power: Fingerprinting usb powered peripherals via power side-channel," in *Proceedings of the IEEE Conference on Computer Communications (INFOCOM)*, 2023.

[60] L. Batina, S. Bhasin, D. Jap, and S. Picek, "CSI NN: Reverse engineering of neural network architectures through electromagnetic side channel," in *Proceedings of the 28th USENIX Security Symposium*, 2019.

[61] A. Nazari, N. Sehatbakhsh, M. Alam, A. Zajic, and M. Prvulovic, "Eddie: Em-based detection of deviations in program execution," in *Proceedings of the Annual International Symposium on Computer Architecture (ISCA)*, 2017, pp. 333–346.

[62] Y. Han, S. Etigowni, H. Liu, S. Zonouz, and A. Petropulu, "Watch me, but don't touch me! contactless control flow monitoring via electromagnetic emanations," in *Proceedings of the ACM SIGSAC Conference on Computer and Communications Security (CCS)*, 2017.

[63] S. Ramesh, G. S. Hadi, S. Yang, M. C. Chan, and J. Han, "Ticktock: Detecting microphone status in laptops leveraging electromagnetic leakage of clock signals," in *Proceedings of the ACM SIGSAC Conference on Computer and Communications Security (CCS)*, 2022.

[64] D. Genkin, L. Pachmanov, I. Pipman, E. Tromer, and Y. Yarom, "ECDSA key extraction from mobile devices via nonintrusive physical side channels," in *Proceedings of the ACM SIGSAC Conference on Computer and Communications Security (CCS)*, 2016, pp. 1626–1638.

[65] M. Alam, B. B. Yilmaz, F. Werner, N. Samwel, A. G. Zajic, D. Genkin, Y. Yarom, and M. Prvulovic, "Nonce@ Once: A single-trace EM side channel attack on several constant-time elliptic curve implementations in mobile platforms." in *Proceedings of the IEEE European Symposium on Security and Privacy (EuroS&P)*, 2021, pp. 507–522.

[66] P. Belgarric, P.-A. Fouque, G. Macario-Rat, and M. Tibouchi, "Side-channel analysis of weierstrass and koblitz curve ecdsa on android smartphones," in *Cryptographers' Track at the RSA Conference*. Springer, 2016, pp. 236–252.

[67] M. Alam, H. A. Khan, M. Dey, N. Sinha, R. Callan, A. Zajic, and M. Prvulovic, "One&Done: A Single-Decryption EM-Based attack on OpenSSL'sConstant-Time blinded RSA," in *Proceedings of the USENIX Security Symposium*, 2018, pp. 585–602.

[68] M. Choi, S. Oh, I. Kim, and H. Kim, "Magsnoop: listening to sounds induced by magnetic field fluctuations to infer mobile payment tokens," in *Proceedings of the 20th Annual International Conference on Mobile Systems, Applications and Services*, 2022, pp. 409–421.

[69] Y. Cheng, X. Ji, W. Xu, H. Pan, Z. Zhu, C.-W. You, Y.-C. Chen, and L. Qiu, "Magattack: Guessing application launching and operation via smartphone," in *Proceedings of the ACM Asia Conference on Computer and Communications Security (AsiaCCS)*, 2019, pp. 283–294.

[70] T. Liu, F. Lin, Z. Wang, C. Wang, Z. Ba, L. Lu, W. Xu, and K. Ren, "Magbackdoor: Beware of your loudspeaker as a backdoor for magnetic injection attacks," in *2023 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2023, pp. 3416–3431.

[71] Q. Liao, Y. Huang, Y. Huang, Y. Zhong, H. Jin, and K. Wu, "Magear: eavesdropping via audio recovery using magnetic side channel." in *Proceedings of the International Conference on Mobile Systems, Applications, and Services (MobiSys)*, 2022, pp. 371–383.

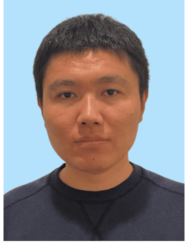
[72] T. Ni, X. Zhang, and Q. Zhao, "Recovering fingerprints from in-display fingerprint sensors via electromagnetic side channel," in *Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security*, 2023, pp. 253–267.

[73] Y. Long, Q. Jiang, C. Yan, T. Alam, X. Ji, W. Xu, and K. Fu, "EM Eye: Characterizing electromagnetic side-channel eavesdropping on embedded cameras," in *Proceedings of the Network and Distributed Systems Security (NDSS) Symposium*, 2024.

[74] J. Li, Y. Meng, L. Zhang, G. Chen, Y. Tian, H. Zhu, and X. S. Shen, "Magfingerprint: A magnetic based device fingerprinting in wireless charging," in *Proceedings of the IEEE Conference on Computer Communications (INFOCOM)*, 2023.

[75] R. Xiao, T. Li, S. Ramesh, J. Han, and J. Han, "Magtracer: Detecting gpu cryptojacking attacks via magnetic leakage signals," in *Proceedings of the 29th Annual International Conference on Mobile Computing and Networking (MobiCom)*, 2023, pp. 1–15.

Tao Ni is an Assistant Professor at the Computer, Electrical and Mathematical Sciences and Engineering (CEMSE) Division, King Abdullah University of Science and Technology (KAUST). He was a Postdoctoral Research Fellow at the Department of Computer Science, City University of Hong Kong, where he obtained his PhD. Before that, he received his Master's degree from the Australian National University in 2020 and his Bachelor's degree from Shanghai Jiao Tong University in 2018. His research interests include cyber-physical systems (CPS) security, AI security, and mobile computing. His work received the 2024 Cybersecurity Best Practical Paper Award and was also named an ACM MobiSys Rising Star (2024) and an AIoTSys Rising Star (2025).



Chaoshun Zuo is a postdoctoral scholar at the Department of Computer Science and Engineering at the Ohio State University. His research focuses on mobile and IoT security with an emphasis on applying both static and dynamic program analysis on mobile applications to systematically identify various security and privacy vulnerabilities in the mobile app-centric ecosystem, including vulnerable access controls that result in massive data leakages in the cloud and weak input validations that lead to memory corruption in IoT devices. Many of his works have been published in flagship conferences, such as CCS, NDSS, Usenix Security, IEEE S&P, and WWW. He received his Ph.D. in 2020 from the Ohio State University, advised by Professor Zhiqiang Lin, Master's degree advised by Professor Shanqing Guo, and Bachelor's degree in Computer Science from Shandong University in 2016 and 2013, respectively.

Jianfeng Li is currently an Assistant Professor with the MOE Key Laboratory for Intelligent Networks and Network Security, Faculty of Electronic and Information Engineering, Xi'an Jiaotong University. He received his Ph.D. degree in control science and engineering from Xi'an Jiaotong University, China, in March 2018. He was a Post-Doctoral Fellow at The Hong Kong Polytechnic University from September 2019 to June 2022. He has published a number of research papers in top conferences and journals, such as S&P, CCS, USENIX Security, NDSS, INFOCOM, IEEE/ACM Transactions on Networking, and IEEE Transactions on Information Forensics and Security. His research interests include traffic analysis, the privacy of mobile platforms, network monitoring, AI security, and large-scale cyber security.

Wubing Wang is the technical director of privacy-preserving computing at DBAppSecurity Co., Ltd. As a core member, he participated in the design and development of privacy-preserving computing products, the "Digital Certificate of Data Compliance Circulation" in "China Digital Valley". His research interests include data security, data element market, system security, trusted execution environment, side-channel attack, binary analysis, mobile security, etc.

Weitao Xu is a tenured Associate Professor at the Department of Computer Science at City University of Hong Kong. Before that, he was a Postdoctoral Research Associate at the School of Computer Science and Engineering (CSE) at UNSW from June 2017 to August 2019. He obtained his PhD degree from the University of Queensland in 2017 (advised by Prof. Neil Bergmann and Prof. Wen Hu). His research areas include mobile computing, sensor networks, and IoT security. His research in these areas has been recognized with a Best Paper Runner-up Award (IPSN'16), a Best Demo Runner-up Award (IoTDI'18), and the Mark Weiser Best Paper Award (PerCom'23). He is also the winner of the 2016 Google PhD Fellowship Award (1/52 all over the world) and the 2021 ACM SIGBED China Rising Star Award. He is a senior member of IEEE.

Xiapu Luo is a Professor at the Department of Computing and the director of the Research Centre for Blockchain Technology of the Hong Kong Polytechnic University. His research focuses on Blockchain and Smart Contracts Security, Mobile and IoT Security, Network Security and Privacy, and Software Engineering with papers published in top-tier security, software engineering, and networking conferences and journals. His research led to ten best/distinguished paper awards, including ACM SIGSOFT Distinguished Paper Awards in ISSTA'22, ICSE'21, and ICSE'24, Best Paper Award in INFOCOM'18, Best Research Paper Award in ISSRE'16, etc., and several awards from the industry. His research uncovered many severe vulnerabilities in critical infrastructures and applications, such as blockchain systems and smart contracts, mobile platforms and apps, IoT devices, and vehicles. He regularly serves in the program committees of top security and software engineering conferences and received the Top Reviewer Award from CCS'22 and the Distinguished TPC Member Award from INFOCOM'23. He also served as the program committee/general chair of several international conferences, including RAID, SECURECOM, ICICS, etc. He is an associate editor for IEEE/ACM Transactions on Networking (ToN) and IEEE Transactions on Dependable and Secure Computing (TDSC). He is a senior member of IEEE.

Qingchuan Zhao is an Assistant Professor in the Department of Computer Science at the City University of Hong Kong. Prior to joining the department in 2021, he completed his Ph.D. at the Ohio State University in the same year, following his M.S. degree from the University of Florida in 2015, and a B.E. degree from South China University of Technology in 2009. His research focuses on the security and privacy practices in the Android appified ecosystem. He employs both static and dynamic data flow analysis on mobile apps and delves into hardware side channels to uncover a variety of vulnerabilities, including privacy leakage, privilege escalation, and vulnerable access controls. 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.