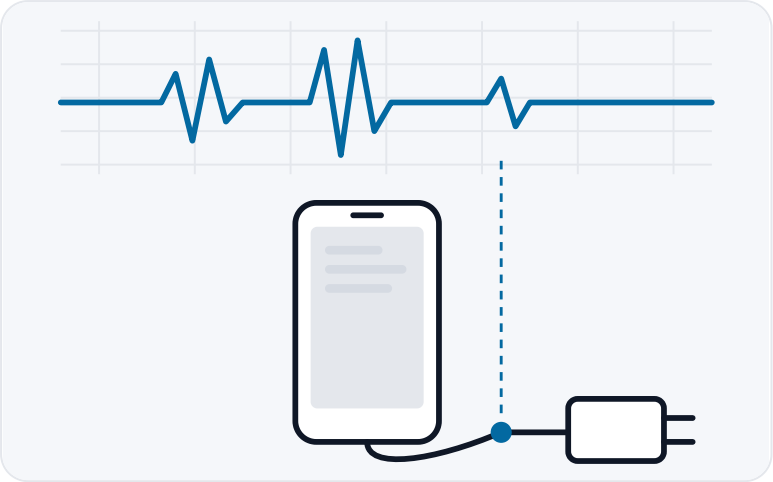
A-SIT Technology

E-Government Innovation Center

2 November 2026 Analysis · Mobile Security · 22 min read

Power Side Channels on Mobile Devices

Florian Draschbacher



Authors	Florian Draschbacher
Version	2026-10-05
Source	https://technology2.a-sit.at/en/publications/power-side-channels-on-mobile-devices/
Note	Machine translation.

Key findings

- A tampered charger can infer activity from a smartphone's charging current, without any app and without user interaction.
- The board developed (shunt resistor, amplifier, RP2040) records the charging current and at the same time simulates a smartphone charger.
- On the Xiaomi 12, the start and end of a call are visible in the charging current through characteristic consumption peaks and a raised mean.
- A CNN detected call start and call end with an accuracy of just under 93 %; a few other activities were wrongly detected as call end.
- Pass-through charging is likely to reduce signal noise; the report expects power line side channels to become more attractive to attackers.

Side channels have increasingly become a focus of research in recent years. These are attacks that extract information from a computer system by observing (mostly physical) effects that arise indirectly from the processing of that information. Such side channels also exist on mobile devices and can be used to extract various kinds of information.

This project first provides an overview of the existing scientific literature on side channels on mobile devices. The focus is on physical side channels and, within that area, particularly on side channels based on fluctuations in power consumption (power line side channels). A dedicated evaluation platform was then developed, and a side channel based on charging current consumption was implemented for a specific mobile device.

1. Introduction

Within research on IT security problems, side channels are a relatively young field. It had been known for decades that electronic devices can leak information through unintended emanations. However, side channels only became a focus of academic cryptography in the mid-1990s. The starting point was work showing that the execution time of cryptographic operations allows conclusions about secret keys. This was soon followed by the finding that a device's power consumption during a computation also correlates with the data being processed. The basic principle is always the same: an attacker does not attack the algorithm itself, which may be mathematically secure. Instead, the attacker observes physical or system-level side effects of its concrete implementation. In the following years, the field expanded considerably. In addition to execution time and power consumption, electromagnetic emanation, acoustic emissions and finally the behaviour of processor caches were identified as sources of information. At the latest with the publication of Spectre and Meltdown in 2018, the wider public also became aware that side channels are not a purely theoretical problem. They can undermine fundamental security assumptions of modern computer systems.

In this context, mobile devices are a particularly interesting target. Smartphones accompany their users almost constantly and process a wide range of sensitive information, from private communication and location data to credentials for banking and government services. At the same time, they have numerous sensors and interfaces that produce side effects potentially usable by attackers. Accordingly, a large part of previous research has focused on software-based side channels. For example, it was shown that apps without special permissions can reconstruct keyboard input via motion sensors, or infer the activities of other apps from publicly accessible system statistics. Operating system vendors have reacted to such findings and have gradually restricted access to many of these information sources in recent years.

Hardware side channels that require no software at all on the target device have so far received less attention. Classic attacks of this kind, such as power analysis, usually require physical access to the device and elaborate measurement setups. Smartphones, however, have a special characteristic: because of their limited battery capacity, they must be charged regularly, and increasingly on third-party infrastructure, such as public USB charging stations in airports, trains or cafés. The charging cable thus creates a direct electrical connection between the device and a potentially untrusted counterpart. The charging current drawn depends not only on the battery's state of charge but also on the momentary power demand of the

hardware. Device activities are therefore reflected in this current. Such a power line side channel requires the attacker neither to install an app nor to interact with the user. It only requires a manipulated charging station, and it is practically undetectable for the user.

This project therefore designed and implemented an evaluation platform that can precisely record the charging current of mobile devices and systematically examine it for information leaks. Using this platform, a side channel was identified that makes it possible to detect when a user is making a phone call, based on the charging current alone. The aim of this report is to describe in detail the design of the platform, the measurements performed and the side channel found, and to assess its practical relevance. Finally, possible countermeasures are discussed, both on the part of hardware and operating system vendors and in the form of recommendations for end users.

2. Background

This section explains the technical foundations needed to understand the later sections.

2.1. Side Channels

A side channel is an unintended flow of information. It does not arise from the actual function of a system, but from observable side effects of its execution. An attacker does not read protected data directly. Instead, the attacker observes physical or logical quantities related to that data and draws conclusions from them.

Side channels also exist in everyday life. A burglar does not need to look into a house to know whether the occupants are away: an overflowing letterbox or shutters that have been closed for days give it away. Another well-known example is the safecracker who listens to the clicking of the combination lock instead of knowing the combination. In both cases, the actual secret is well protected. However, it is revealed through a channel the defender did not consider.

In computers, such channels include execution time, the use of shared resources such as caches, electromagnetic emanation, sound and power consumption.

2.2. Causes of Side Channels

Many side channels are a direct consequence of optimisations. A system that always performs every task in the same time and with the same energy reveals little about the data it processes. This is not efficient, however. Modern software and hardware therefore save time and energy wherever possible. As a by-product, their behaviour becomes dependent on inputs and state.

At the software level, for example, a naive password comparison stops as soon as the first wrong character is found. The execution time thus reveals how many characters were correct. Caches, particularly at the operating system level, keep frequently used data available, so an access is faster or slower depending on prior history. Power-saving mechanisms are particularly relevant for mobile devices: processors dynamically adapt clock frequency and voltage to the load (Dynamic Voltage and Frequency Scaling, DVFS), unused circuit parts are switched off, and radio modules switch between sleep and transmit states. Such mechanisms are necessary for a smartphone to last a full day on one battery charge. At the same time, they cause power consumption to closely follow the device's actual activity, and this is exactly what makes it so informative as a side channel.

2.3. Physical Foundations of Power Side Channels

Today, digital circuits consist almost exclusively of CMOS transistors. A CMOS gate draws only little current in its idle state, but each state change requires a somewhat higher current. The more computation is required, the more gates switch, and the higher the power consumption. This explains how the computations executed on a processor leave measurable traces in power consumption.

A smartphone has further large consumers whose power draw depends directly on usage behaviour. With OLED displays, consumption depends on the brightness and colour of each individual pixel. The cellular modem needs considerably more energy when transmitting than when idle. Speaker amplifiers and the vibration motor draw current in proportion to their output. A phone call, for example, puts the modem, the audio path and the processor into a characteristic operating state at the same time, which can differ clearly from other types of use. Discussions of these side effects usually refer to energy consumption as the measured quantity that reveals user activity. In practice, however, for simplicity, it is usually the magnitude of the current flowing that is measured, on the assumption that the voltage remains relatively constant.

3. Side Channels on Mobile Devices

Side-channel attacks do not exploit the actual function of a system, but unintended side effects of its execution. These include timing behaviour, the use of shared resources, electromagnetic emanation and power consumption. Originally, such attacks were mainly directed at cryptographic implementations on smartcards [1]. Today, smartphones are among the most intensively studied targets: they process a wide range of sensitive data, are equipped with numerous sensors and are regularly connected to third-party infrastructure such as public charging stations. Spreitzer et al. give a systematic overview of side channels on mobile devices. Among other things, they distinguish between attacks that require physical access to or proximity to the device, and attacks that are possible purely from software [2]. Below, we first give a brief overview of software side channels. We then discuss hardware side channels in more detail, in particular power line side channels.

3.1. Software Side Channels

Software side channels are exploited by a malicious app that is already installed on the device. Often without special permissions, it evaluates publicly accessible information from the operating system. Early work showed that the memory statistics of other processes visible in the procfs file system allow conclusions about visited websites [3]. Zhou et al. derived users' identity, location and even illnesses from public resources such as data usage statistics [4]. Chen et al. used shared memory counters to detect the current UI state of other apps and were thus able to carry out targeted phishing attacks [5]. Finally, ProcHarvester automated the search for such information leaks [6].

A second large group consists of motion sensors, which apps were for a long time able to read without permission and at a high sampling rate. Gyroscope and accelerometer can be used, for example, to reconstruct speech signals [7]. AccelEve even achieved this for the output of the device's own speaker [8]. In addition, microarchitectural attacks such as cache timing attacks, first known from desktop and server systems, were transferred to ARM-based smartphones [9]. Notably, power consumption can also be observed from software. PowerSpy used the battery data provided by Android to determine a user's location [10]. Oberhuber et al. recently showed that the current and power readings accessible via the Android Sensor Framework enable power side-channel attacks without any physical access [11].

In response to this work, platform operators have gradually restricted access to many of the information sources mentioned. Examples are the restricted access to procfs entries of other processes and the limit of 200 Hz on the sampling rate of motion sensors from Android 12 onwards. In addition, software side channels always require the attacker to be able to execute code on the target device. Hardware side channels avoid this requirement because they measure physical quantities outside the control of the operating system.

3.2. Hardware Side Channels and Power Line Side Channels

In classic power analysis, a device's power consumption is measured in order to infer the data being processed. Genkin et al. showed that this also works for complex systems such as smartphones. They extracted ECDSA keys from iOS and Android devices, both with an electromagnetic probe near the device and via a measurement on the USB charging cable [12]. Most recent work, however, targets user activities rather than cryptographic keys. The threat model is usually a manipulated charger or charging station that covertly records the power consumption of the connected smartphone. This power consumption depends largely on components such as the display, processor, radio modules and speaker. Their activity, in turn, is directly determined by how the device is used.

In 2017, Yang et al. were among the first to show that a USB charging station can detect from power consumption which website is being opened on a charging smartphone. In many configurations, they achieved an accuracy of over 90 % [13]. Matovu et al. demonstrated further attacks on user activities using current meters integrated into charging stations. At the same time, they proposed countermeasures that randomly add noise to power consumption during charging [14]. Cronin et al. went a step further with Charger-Surfing: the visual feedback when a screen key is tapped measurably changes the display's power consumption. Using this, they were able to reconstruct entered unlock PINs key by key with a malicious charger [15]. Lifshits et al. examined a related scenario in which a manipulated battery, rather than the charger, records power consumption. From this, they were able to derive, among other things, keyboard input, visited websites and incoming calls [16]. Power line side channels also reveal acoustic signals. With GhostTalk, Wang et al. showed that a modified charging cable can be used to eavesdrop on a smartphone's speaker output while simultaneously injecting inaudible voice commands [17].

Further work extends the threat model beyond the directly connected charger. With XPorter, Ni et al. showed that in multi-port chargers, an attacker device on the adjacent port can observe a smartphone's activity through voltage fluctuations. This makes it possible to detect running apps and keyboard input [18]. As Oberhuber et al. have shown, many of these side channels can also be exploited indirectly from software [11]. Wireless charging does not protect either: La Cour et al. demonstrated website fingerprinting via the power consumption of a wireless charger [19]. With BankSnoop, Ni et al. were even able to determine, without contact, the running app and the keyboard input of a smartphone charging on a wireless power bank. To do so, they analysed coil whine and disturbances of the magnetic field [20].

4. Implementation and Evaluation

Building on the existing state of the art, this project developed a dedicated evaluation platform that can be used to analyse power line side-channel attacks on smartphones in detail.

4.1. Attack Scenario

For our evaluation, we assumed a malicious charger as the attack scenario, as already known from earlier attacks (such as JuiceJacking). Here, an attacker manipulates, for example, a charger in a public space (airports, railway stations, museums, cafés, … are conceivable) by replacing its electronics. The manipulation cannot be detected from the outside. If users connect their device to the manipulated charger, the device is still supplied with power. In addition, however, the charger secretly measures the device's power consumption and can thus draw conclusions about the actions the user performs on the device.

4.2. Implementation

To study power line side channels in the scenario described above, researchers usually use a laboratory setup based on an oscilloscope. It measures power consumption from the voltage drop across a low-resistance resistor. An oscilloscope allows very high measurement resolutions and frequencies. However, experiments carried out this way inevitably differ from real attack environments. Firstly, oscilloscopes are expensive measuring instruments and therefore unsuitable for an attack that achieves real impact mainly by scaling to a large number of manipulated chargers. Secondly, because of their physical size, oscilloscopes are difficult to integrate into a manipulated charger. Apart from that, using an oscilloscope is impractical even for research into power line side channels, simply because of its purchase cost and physical dimensions.

For our research, we therefore designed and built our own evaluation platform, which simulates a malicious charger in the form of a compact circuit board. A shunt resistor sits between the smartphone as the load and the ground line (Ground / GND). The voltage drop across the resistor is captured by the analogue-to-digital converter (ADC) of an RP2040 microcontroller and, if required, passed on to a computer via USB. The computer can thus perform elaborate real-time analyses of the measured values in order to identify possible power line side channels. To execute attacks that have already been captured, the board can also be used on its own, with only the microcontroller as the processing unit.

The shunt resistor must have a very low resistance (specifically 15 milliohms) so as not to significantly affect the charging voltage for the smartphone. However, this also results in only a very small voltage drop across the resistor, which lies below the measurement range of the ADC of typical microcontrollers. Our evaluation platform therefore contains an INA301A2 amplifier, designed for exactly such use cases, which amplifies the measured voltage drop by a factor of 50. The corresponding excerpt from the circuit diagram is shown in Figure 1.

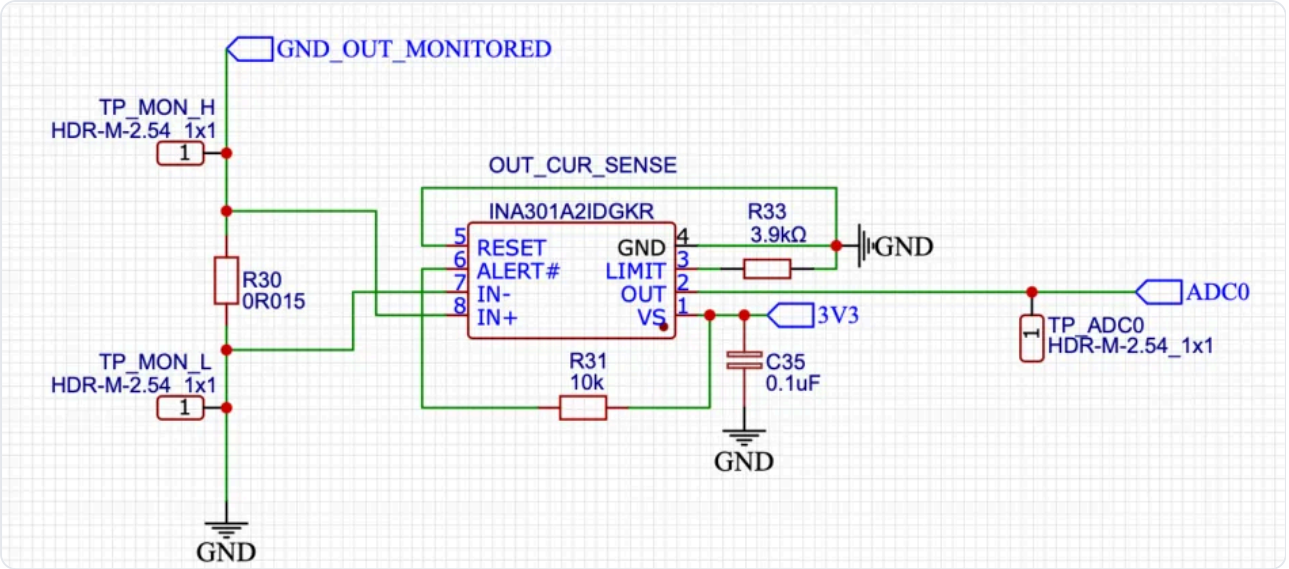


Figure 1: Excerpt of the circuit diagram with shunt resistor and amplifier

The RP2040 was chosen as the microcontroller for several reasons. Firstly, it has an ADC with a high sampling rate (up to 1.2 million samples per second) and relatively high resolution (12 bit). Secondly, the microcontroller has programmable peripherals (PIO) that can be used to implement various additional interfaces. This allowed us to equip the board with 2 working USB interfaces, which enable simultaneous communication with both a computer and a smartphone. In addition, this Raspberry Pi microcontroller has been widely used in recent years, so a large number of libraries and tooling are available.

In addition to measuring current, our board also behaves like a modern smartphone charger. It uses USB Power Delivery controllers to communicate with the smartphone over the Control Channel (CC) in order to agree on a profile of charging current and charging voltage. For this purpose, an FUSB302 controller was fitted, which is controlled by the RP2040 microcontroller via I2C. The complete board is shown in Figure 2.

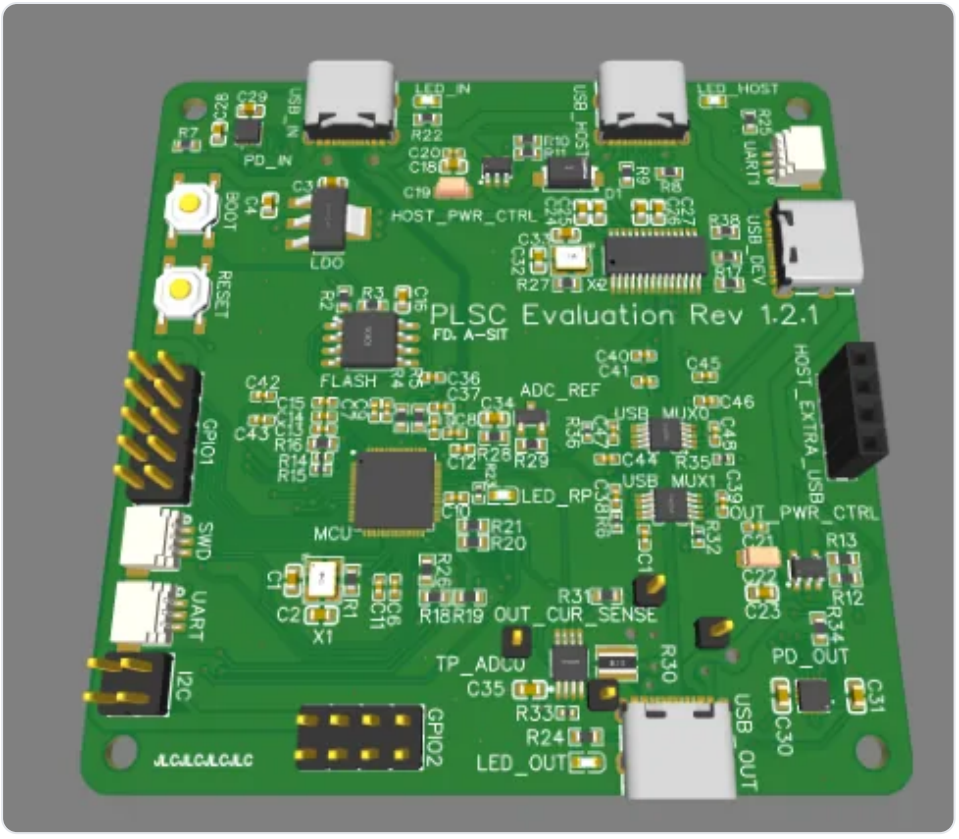


Figure 2: Rendering of the circuit board of our evaluation platform

As soon as the board is powered, the firmware on the microcontroller configures the USB PD controller and starts the USB interface. If the board is connected to a computer via USB, the computer can, for example, set some settings via a virtual serial interface. When the controller reports that a smartphone has been connected, the microcontroller starts reading the ADC at an interval specified by the computer and passes the measured values on to the computer.

For further analysis, we implemented a Python framework on the computer that abstracts the communication with the board. The ADC values read can also be stored here and loaded later for analysis without the board connected. They can also be output as a real-time plot to identify first visual starting points for power line side channels.

4.3. Evaluation

The aim of evaluating our platform was to identify a side channel that allows conclusions about the activity profile of the connected smartphone. As an illustrative example, we used the detection of moments in which the user is using the smartphone to make a phone call. Since particularly power-hungry cellular communication is used here, we expected especially good chances of a distinct power line side channel. A Xiaomi 12 served as the test device. We chose the low sampling rate of 1 kHz, as the amount of data then remains manageable and coarse patterns are also easy to recognise visually.

In this setup, we connected the test device to our evaluation platform and observed the power consumption in a real-time plot while starting a call on the device. As hoped, a pattern emerged. As can be seen in Figure 3, the start and end of the call are marked by characteristic consumption peaks and a slightly increased mean.

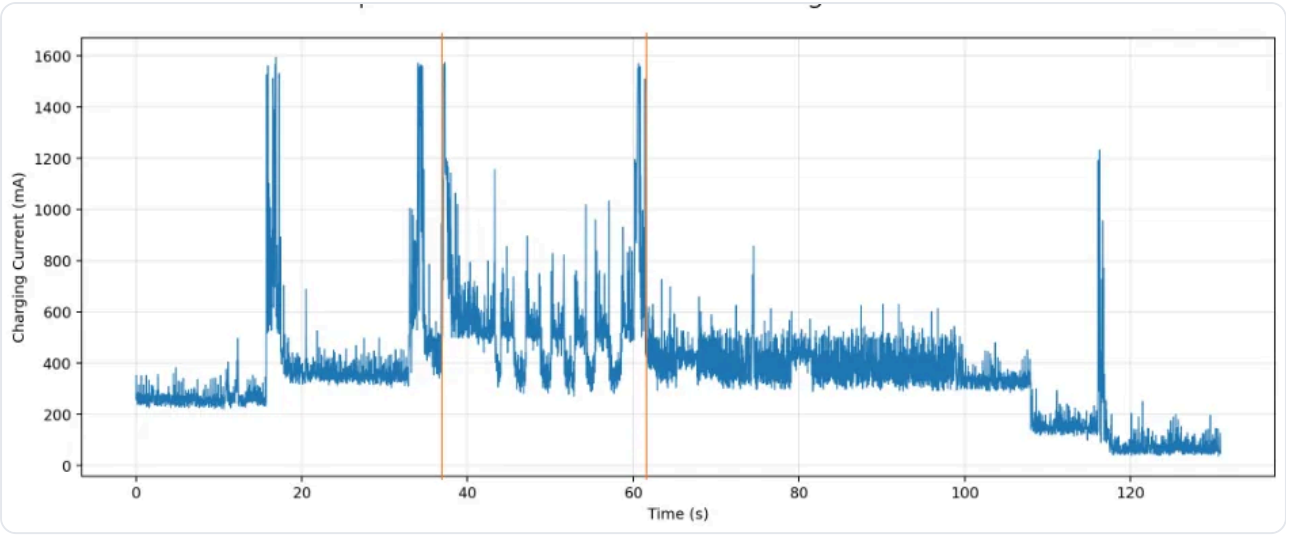


Figure 3: Power consumption of a Xiaomi 12. The red lines mark the start and end of the call.

In the next step, we trained a machine learning model (specifically a convolutional neural network; CNN) to automatically detect from power consumption when the smartphone is in a call. To do this, we first had to collect additional recordings of alternative activities on the smartphone, to make sure that no similar patterns can be recognised in power consumption. Specifically, we automatically collected power consumption recordings of 16 typical user activities, including:

- Web browsing (using Google Chrome)
- Online shopping (on Amazon)
- Listening to music (Spotify)
- Video streaming (Youtube)
- Social media (Facebook, Snapchat, TikTok)
- Calculator
- File management (via OneDrive)

For each of the activities performed, we collected at least 15 recordings. Together with the same number of recordings of call start and call end, we used them to train the CNN. We then evaluated the classifier using 10 new recordings for each of the activities listed above. This resulted in an accuracy of just under 93% for the model. The start and end of calls were always detected correctly. However, a few other activities were in individual cases also wrongly detected as call end. We expect that the accuracy can be increased further through more complex models and more extensive training. Nevertheless, the experiment already showed that the power consumption of a charging smartphone can be used to distinguish between different typical activities.

5. Outlook

In this project, we were able to show that power line side channels, as a concrete example of a hardware-based side channel, affect current Android devices. Indeed, current devices are not only affected; a look at current developments also shows that such side channels are becoming increasingly interesting for attackers:

Broad awareness of JuiceJacking. In the recent past, the mass media have repeatedly reported on the danger posed by public charging stations. Not least because of the novel ChoiceJacking attacks we found last year, public awareness of these attacks has been renewed and deepened. As a result, at least particularly vulnerable users rely on additional protective measures such as USB data blockers. These block the USB data lines, so that at the hardware level the USB connection can only be used for charging.

More and more technical measures against classic USB data attacks. Not least because of our research on ChoiceJacking, all major smartphone manufacturers have now also integrated further software-based protective measures intended to prevent malicious chargers from gaining access to user data via USB. Both iOS and Android now offer optional lockdown modes for particularly vulnerable users, which disable USB communication at a low software level.

Ubiquitous availability of charging infrastructure. At the same time, smartphone charging options in public spaces continue to increase strongly. Even buses or museums, for example, now offer charging sockets or rentable power banks. It can be assumed that, owing to this strong exposure, the use of public charging infrastructure is increasing overall. Not least, the USB data blockers and lockdown modes mentioned above lead even particularly vulnerable users, believing in the effectiveness of these protective measures, to make use of the charging offers.

Pass-through charging. More and more smartphone manufacturers are integrating pass-through charging into their devices. Put simply, when pass-through charging is enabled, the smartphone is powered directly from the charging current. Until now, roughly speaking, the charging current was usually used to charge the smartphone's battery, which in turn supplied the other components in the smartphone with power. The consumers in the smartphone slow down charging here, because they continuously draw current from the battery at the same time. Firstly, this has negative consequences for the battery's lifetime. Secondly, this inefficient approach generates heat, which can lead to throttling of the CPU speed, particularly while running computationally intensive applications. Pass-through charging has consequences for power side channels as we have shown them in this project: earlier studies and our own experiments have already shown that such attacks work regardless of the exact state of charge, even on devices without pass-through charging. We did not investigate why exactly activity-induced fluctuations in consumption can be measured in the charging current across the battery, because this would have required analysing the complex charging electronics in the device. In any case, it seems likely that considerably less signal noise can be expected with pass-through charging. In this case, the measured power consumption is attributable exclusively to the components directly involved in the user activity (in particular the CPU, display and communication modules).

6. Summary

This project examined side channels, and in particular power side channels, on current smartphones. After an overview of existing work in the areas of software and hardware side channels, a dedicated evaluation platform was developed. On this basis, an illustrative side channel was demonstrated that makes it possible to detect from the power consumption during charging when the smartphone is being used for a phone call. This side channel could not only be captured visually, but also automated and evaluated with the help of a convolutional neural network. Finally, a short outlook highlighted the future relevance of power side channels. Owing to technical developments and ever better measures against classic USB-based attacks, power line side channels are becoming increasingly attractive to attackers. Not least for this reason, research must also increasingly address these attacks and possible countermeasures.

References

[1] P. C. Kocher, J. Jaffe and B. Jun, "Differential Power Analysis," in CRYPTO, 1999.

[2] R. Spreitzer, V. Moonsamy, T. Korak and S. Mangard, "Systematic Classification of Side-Channel Attacks: A Case Study for Mobile Devices," IEEE Communications Surveys & Tutorials, vol. 20, no. 1, 2017.

[3] S. Jana and V. Shmatikov, "Memento: Learning Secrets from Process Footprints," in IEEE S&P, 2012.

[4] 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 ACM CCS, 2013.

[5] Q. A. Chen, Z. Qian and Z. M. Mao, "Peeking into Your App without Actually Seeing It: UI State Inference and Novel Android Attacks," in USENIX Security, 2014.

[6] R. Spreitzer, F. Kirchengast, D. Gruss and S. Mangard, "ProcHarvester: Fully Automated Analysis of Procfs Side-Channel Leaks on Android," in AsiaCCS, 2018.

[7] Y. Michalevsky, D. Boneh and G. Nakibly, "Gyrophone: Recognizing Speech from Gyroscope Signals," in USENIX Security, 2014.

[8] Z. Ba, T. Zheng, X. Zhang, Z. Quin, B. Li, X. Liu and K. Ren, "Learning-based Practical Smartphone Eavesdropping with Built-in Accelerometer," in NDSS, 2020.

[9] M. Lipp, D. Gruss, R. Spreitzer, C. Maurice and S. Mangard, "ARMageddon: Cache Attacks on Mobile Devices," in USENIX Security, 2016.

[10] Y. Michalevsky, A. Schulman, G. A. Veerapandian and D. Boneh, "PowerSpy: Location Tracking Using Mobile Device Power Analysis," in USENIX Security, 2015.

[11] M. Oberhuber, M. Unterguggenberger, L. Maar, A. Kogler and S. Mangard, "Power-Related Side-Channel Attacks using the Android Sensor Framework," in NDSS, 2025.

[12] D. Genkin, L. Pachmanov, I. Pipman, E. Tromer and Y. Yarom, "ECDSA Key Extraction from Mobile Devices via Nonintrusive Physical Side Channels," in ACM CCS, 2016.

[13] Q. Yang, P. Gasti, G. Zhou, Farajidavar, Aydin 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. 1, 2017.

[14] R. Matovu, A. Serwadda and A. V. G.-S. I. Bilbao, “Defensive Charging: Mitigating Power Side-Channel Attacks on Charging Smartphones,” in CODASPY, 2020.

[15] P. Cronin, X. Gao, C. Yang and H. Wang, “Charger-Surfing: Exploiting a Power Line Side-Channel for Smartphone Information Leakage,” in USENIX Security, 2021.

[16] P. Lifshits, R. Forte, Y. Hoshen, M. Halpern, M. Philipose, M. Tiwari and M. Silberstein, “Power to peep-all: Inference Attacks by Malicious Batteries on Mobile Devices,” in PoPETS, 2018.

[17] Y. Wang and H. Y. Q. Guo, “GhostTalk: Interactive Attack on Smartphone Voice System Through Power Line,” in NDSS, 2022.

[18] 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 ACM MobiCom, 2023.

[19] A. S. La Cour, K. K. Afridi and G. E. Suh, “Wireless Charging Power Side-Channel Attacks,” in ACM CCS, 2021.

[20] 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 ACM MobiCom, 2023.