International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 08 Issue: 07 | July - 2024 SJIF Rating: 8.448 ISSN: 2582-3930

Threats due To Side Channel Attacks (SCA) in Smartphones

Anushka Naik[1]

[1]Department of Information Technology Goa College of Engineering,Goa, India

---

Abstract - Today's mobile devices contain densely packaged system-on-chips (SoCs) with multi-core, high frequency CPUs and complex pipelines. In parallel, sophisticated SoC-assisted security mechanisms have become commonplace for protecting device data, such as trusted execution environments, full disk and file-based encryption. Both advancements have dramatically complicated the use of conventional physical attacks, requiring the development of specialised attacks Our proposed classification system allows to analyze side-channel attacks systematically, and facilitates the development of novel countermeasures. Besides this new categorization system, the extensive survey of existing attacks and attack strategies provides valuable insights into the evolving field of side-channel attacks, especially when focusing on mobile devices..

***Key Words*:** Side-channel Attack, active, passive

1.INTRODUCTION

Twenty years later, mobile devices are a necessary component of every person's everyday existence. In one way or another, these devices make life easier every day. Examples of these jobs include figuring out the quickest route to get there, making quick payments without using plastic cards, managing our time well, sending information instantly, taking pictures and videos, and recording sounds. Additionally, they make it possible to process a lot of work procedures, which compels the user to carry them around at all times. These "electronic companions" consequently turn into repositories of private information, guardians of state and business secrets, and observers of private discussions. Thus, the issue of having access to all information resources arises.

Modern smart devices come equipped with a multitude of sensors, which opens up a wide range of possibilities.Many sensors are integrated into modern smart devices, opening up a world of possibilities in the areas of productivity, mobile health, context awareness, and activity identification. Regrettably, these sensors can also serve as entry points for accidental disclosure of user activity data.[1] Unauthorized access to sensor data containing user preferences, habits, behaviors, and personal information has long been a serious security and privacy risk.

Users must give their express consent for mobile operating systems like Android and iOS to allow an application access to sensitive data (such as GPS positions, microphone recordings, and camera images). On the other hand, information gathered from more general sensors—like magnetometers, accelerometers, and gyroscopes—is less controlled and frequently accessible by programs without the consent of users. Recent times have seen the emergence of numerous side-channel attacks due to unmonitored access to the data from these sensors. Previous attempts have shown security and privacy issues arising from programs using the data from motion sensors.

These publications suggest attacks aimed at deducing sensitive user privacy information, including fingerprinting of websites and applications, touch actions and keystrokes, passwords, and PIN codes. Unfortunately, because software is updated frequently and new devices are released into the market, side-channel assaults remain a key source of concern for developers and users.

Especially the following key enablers enable more devastating attacks on mobile devices.

1. Always-on and portable: To begin with, mobile devices are always carried about because of their portability and are always turned on. They are therefore deeply ingrained in our daily existence.

2. Bring your own device (BYOD): Employees utilize personal devices to process company data and access corporate infrastructure in an effort to reduce the number of devices they carry about, which emphasizes the significance of secure mobile devices

3. Software installation ease: Because of the appification [15] of mobile devices—that is, the fact that there is an app for nearly anything—new software may be loaded with ease through reputable app marketplaces. As a result, rogue apps can also proliferate quickly..

4. 4. Linux kernel-based OS: The Linux kernel serves as the foundation for many contemporary mobile operating systems (OS), including Android. However, the Linux kernel was first created for desktop computers, and while features or information deemed innocuous on these platforms may not be so on mobile devices, they may pose a serious risk to privacy and/or security ([16]). Attack surface for cellphones, cloud computing systems, and smart cards.

5. Last but not least, these devices have a plethora of functions and sensors that are absent from conventional platforms. Because mobile devices are inherently always-on and portable, connected, inherent input methods, etc.), these characteristics often enable damaging side-channel attacks. These sensors have also been used to take down 3D printers [21], [22], infer content from TV shows [20], and breach computer hard drives and keyboards [17]–[19]. All of these examples clearly demonstrate the immense capability of mobile devices.

A greater awareness of side-channel attacks has emerged as a result of the previously listed key enablers. Most modern side-channel assaults are primarily non-invasive and depend on the execution of malicious software within the targeted environment. In light of these advancements, we note that the classification approach designed to examine side channel assaults on smart cards is no longer suitable for these modern attack scenarios and tactics. For this reason, side-channel assaults on mobile devices as well as other contemporary side-

channel attacks cannot be systematically categorized using the current classification method.

We bridge this gap in this work by developing a new classification scheme for contemporary side-channel assaults against mobile devices. As a result, we examine current side-channel attacks and find similarities among them. The newfound understanding enables researchers to decide on future lines of inquiry and to counteract these attacks more broadly.

2. Background

In this section, we introduce the basics of mobile security, define the general notion of side-channel attacks, and we estab lish the boundaries between side-channel attacks and other attacks on mobile devices. We stress that side-channel attacks do not exploit specific software vulnerabilities of the OS or any specific library, but instead exploit available information that either leaks unintentionally or that is (in some cases) pub lished for benign reasons in order to infer sensitive information indirectly. Finally, we also discuss related work.

A. A Primer on Smartphone Security

Mobile devices, such as tablet computers and smartphones, are powerful multi-purpose computing platforms that enable many different application scenarios. Third-party applications can be easily installed in order to extend the basic function ality of these devices. Examples include gaming applications that make use of the many different sensors, office applica tions, banking applications, and many more. These examples clearly demonstrate that mobile devices are already tightly integrated into our everyday lives, which leads to sensitive data and information being stored and processed on these devices. In order to protect this information properly, modern mobile operating systems rely on two fundamental security concepts, i.e., the concept of application sandboxing and the concept of permission systems. For instance, on Android the underlying Linux kernel ensures the concept of sandboxed applications. Each application is assigned a user ID (UID), which allowsthe kernel to prevent applications from accessing resources of other applications. The permission system on the other hand allows applications to request access to specific resources out side of its sandbox, which typically includes resources that are considered as being sensitive or privacy relevant. Android also categorizes permissions depending on so-called protection lev els. The two important categories of Android permissions are normal permissions and dangerous permissions, respectively. While normal permissions are granted automatically during the installation procedure, dangerous permissions must be explicitly granted by the user. Other mobile operating systems such as Apple's iOS rely on similar protection mechanisms. Besides these basic security concepts on the OS level, applications themselves rely on cryptographic primitives, cryp tographic protocols, and dedicated security mechanisms to protect sensitive resources. For instance, applications rely on encryption primitives to protect sensitive information being stored on the device or when transmitting data over the Internet. Another example of a dedicated security mechanism is a personal identification number (PIN) required to access a specific service such as a banking application.

B. Side-Channel Attacks

Although the above mentioned concepts are secure (or are typically considered as being secure) in theory, a specific implementation of such a mechanism is not necessarily secure in practice. Since side-channel attacks have been extensively used to attack cryptographic implementations, let us consider the following illustrative example. In an ideal world, an imple mentation of a cryptographic algorithm takes a specific input and produces a specific (intended) output. For example, an encryption algorithm takes the plaintext as well as crypto graphic key material to produce the ciphertext. However, in practice, an implementation of an encryption algorithm usu ally also "outputs" unintended information as a byproduct of the actual computations. Such unintended information leak age might be a different power consumption or a different execution time due to instructions being conditionally exe cuted depending on the processed data. Attacks exploiting such unintended information leaks are denoted as side-channel attacks and have been impressively used to bypass or break protection mechanisms such as encryption algorithms. Subsequently, we discuss the general notion of side-channel attacks. We distinguish between passive side-channel attacks, as in the example above, and active side-channel attacks.

Passive Side-Channel Attacks: The general notion of a pas sive side-channel attack can be described by means of three main components, i.e., target, side channel, and attacker. A target represents anything of interest to possible attack ers. During the computation or operation of the target, it influences a side channel (physical or logical properties) and thereby emits potential sensitive information. An attacker who is able to observe these side channels potentially learns useful information related to the actual computations or operations performed by the target. Therefore, an attacker models possi ble effects of specific causes. Later on, careful investigations of observed effects can then be used to learn information about possible causes.

Active Side-Channel Attacks: An active attacker tampers with the device or modifies/influences the targeted device via a side channel, e.g., via an external interface or envi ronmental conditions. Thereby, the attacker influences the computation/operation performed by the device in a way that allows to bypass specific security mechanisms directly or that leads to malfunctioning, which in turn enables possible attacks, e.g., indirectly via the leaking side-channel information or directly via the (erroneous) output of the targeted device.

A target emits sensitive information as it influences specific side channels. For example, physically operating a smartphone via the touchscreen, i.e., the touchscreen input represents the target, causes the smartphone to undergo specific movements and accelerations in all three dimensions. In this case, one possible side channel is the acceleration of the device (a physical property), which can be observed via the embedded accelerometer sensor and accessed by an app via the official Sensor API.

Differentiation From Other Attacks: Irrespective of whether an attacker is passive or active, we only consider side-channel attacks. Side-channel attacks do not exploit software bugs or anomalies within the OS or apps that, for example, allow to access the main communication channel directly. For example,

buffer overflow attacks allow to access the main communica tion channel directly (i.e., the main memory) and, thus, do not represent side-channel attacks.

Similarly, we also do not consider other attacks that learn information that is available from the main channel. For exam ple, Luzio et al. [28] exploited Wi-Fi probe-requests, which contain the service set identifier (SSID) of preferred Wi-Fi hotspots in clear. These probe-requests allow mobile devices to determine nearby Wi-Fi hotspots in order to preferably connect to already known hotspots. These attacks do not represent side channel attacks as the learned information is directly available from the main channel.

Furthermore, we also do not survey covert channels where two entities (e.g., processes) communicate over a channel that is not explicitly provided by the platform or the operating system. Although identified side channels can in general also be used as a covert channel, i.e., as a means to stealthily communicate between two processes whereby one process influences the side channel and the other one observes it, we do not explicitly survey covert channels such as [29] in this paper. Nevertheless, our newly introduced classification system can also be used to classify covert channels.

C. Related Surveys

In this section, we discuss surveys on mobile security, as well as side-channel attacks on smart cards, PCs, cloud infrastructures, and smartphones.

Surveys on Mobile Security: Most surveys on mobile secu rity primarily focused on malware in general, and many of these surveys only mention side-channel attacks as a side node. Enck [30] surveyed possible protection mechanisms beyond the standard protection mechanisms provided by Android. These include tools that analyze permissions and action strings (within the Android Manifest) to assess the risk of Android apps, policy-based approaches that allow a more fine-grained protection of Android apps, as well as static and dynamic code analysis tools to perform application analysis, which in turn allows to detect malware.

Polla et al. [31] surveyed threats and vulnerabilities (i.e., botnets, Trojans, viruses, and worms) with a focus on work published from 2004 until 2011. Suarez-Tangil et al. [32] and Faruki et al. [33] continued this line of research for the period from 2010 until 2013, and from 2010 until 2014, respectively.

Rashidi and Fung [34] surveyed techniques (e.g., based on static and dynamic code analysis) to cope with mal ware on mobile devices and Sadeghi et al. [35] surveyed tools and analysis techniques to identify malware. In addition, Sadeghi et al. provided a “survey of surveys” discussing surveys and their main contributions in more detail. We refer to their work for a more detailed investigation of mal ware analysis techniques and further literature on this topic. Tam et al. [36] surveyed mobile malware analysis techniques (static, dynamic, hybrid) as well as malware tactics to hinder analysis (obfuscation).

Surveys on Side-Channel Attacks: The survey of Tunstall [37] focused on smart card security, in particular side-channel attacks against cryptographic algorithms.

Zander et al. [38] surveyed covert channels via computer network protocols, and Biswas et al. [39] conducted an in depth study on network timing channels (remote timing side channels) as well as in-system timing channels (focusing on hardware-based timing channels such as cache attacks) on commodity PCs. They surveyed timing channels according to their suitability for covert channels, timing side channels, and network flow watermarking (e.g., to de-anonymize Tor).

Regarding cloud computing platforms, Ge et al. [7] and Szefer [40] surveyed microarchitectural attacks with a focus on cache attacks. Ullrich et al. [41] focused on network based covert channels and network-based side channels in cloud settings. Betz et al. [42] focused on covert chan nels and mentioned a few side-channel attacks in the cloud setting.

The focus of our paper is on side-channel attacks against mobile devices. Surveys about this topic are quite scarce and consider specific types of side-channel attacks only. Xu et al. [43] surveyed attacks and defenses on Android at a broader scale and thereby provide a comprehensive overview of the research landscape. They considered system privilege escalation, issues in the permission model, side channels and covert channels (a high-level overview of exploits consid ering the accelerometer, the CPU cache, and the procfs), feature abuses, malware detection, and app repackaging. Hussain et al. [44] and Nahapetian [45] surveyed sensor-based keylogging attacks. However, a systematic survey and classi f ication of all existing categories of side-channel attacks on mobile devices does not exist so far.

3. Local side-channel attacks

In this section, we survey side-channel attacks that require a local adversary. Some of these attacks will show that the transition between local attacks and vicinity attacks is seamless as the distance between the victim (device) and the attacker can be increased, especially in case of some passive attacks.

A. Passive Attacks

We start with traditional side-channel attacks that aim to break insecure cryptographic implementations (of mathemati cally secure primitives). Besides, we discuss attacks that target the user’s interaction with the device as well as the user’s input on the touchscreen, i.e., attacks that result from the inherent nature of mobile devices.

Power Analysis Attacks: The actual power consumption of a computing device or implementation depends on the processed data and the executed instructions. Power analysis attacks exploit this information leak to infer sensitive information. As the name suggests, the power consumption, typically measured as the voltage drop across a resistor inserted in the supply line, serves as the side channel. State-of-the-art printed circuit board designs (PCB-designs), including multi-layer routing as well as surface mounted devices (SMD), and packaging techniques (e.g., ball-grid array) make it hard to access the appropriate power supply lines in modern smartphones without permanent modifications. Therefore, in contrast to smart cards, measuring the power consumption became less relevant for side-channel attacks targeting smartphones. Depending on whether a single measurement trace or multiple traces are required, we

distinguish between simple power analysis (SPA) attacks and differential power analysis (DPA) attacks, as defined by Kocher et al. [2]. SPA attacks rely on the interpretation of power traces in order to reveal, for example, the sequence of executed instructions, which allows to break implementations where the executed instruc tions depend on secret data. However, the power consumption also depends on the processed data, although the variations are smaller. Therefore, DPA attacks rely on statistical investi gations of multiple traces in order to infer information about the processed data.

Attacks: Messerges et al. [48] exploited the power con sumption of a smart card to attack the Data Encryption Standard (DES) algorithm. Hardly any side-channel attacks using a similar setup for measuring the power consumption targeting smartphones are published. Nevertheless, a coarse grained power-consumption monitoring of smartphones allows to identify running apps, as demonstrated by Yan et al. [49].

Electromagnetic Analysis Attacks: Exploiting electromagnetic emanations is another method of attacking the power leakage of computing systems. These emanations are typically easier to obtain because the power line is generally not accessible directly. These attacks are typically referred to as differential power analysis attacks, regardless of whether the power trace is received directly from the power line or through electromagnetic emanations. In this regard, it's also important to note that aiming the measurements at a particular spot above the chip may enhance the signal-to-noise ratio, depending on the equipment being used (EM probes for catching the electromagnetic radiation). Using spatial information allows for a reduction in the number of measures needed for an attack to be successful.

Attacks: On mobile devices, conventional side-channel attacks that take advantage of smart cards' electromagnetic emissions have also been used. Elliptic Curve Cryptography (ECC) and the Advanced Encryption Standard (AES) were both shown to be vulnerable to attacks by Gebotys et al. [50] on Java-based PDA software implementations. Subsequently, the Elliptic Curve Digital Signature Algorithm (ECDSA) implementation of Android's Bouncy Castle was attacked by Belgarric et al. [53], Goller and Sigl [52], and Nakano et al. [51] against the RSA and ECC implementations of the default crypto provider (JCE) on Android smartphones. Genkin et al. [54] similarly targeted the CommonCrypto implementation of ECDSA on iOS and the OpenSSL implementation of ECDSA on Android.

Differential Computation Analysis: Embedding the secret key inside the software implementation in a way that makes it impossible for an adversary to extract it—even if they have access to the source code—is the fundamental concept behind white box crypto systems. As a result, the algorithm and the key are combined in such a way that the key is concealed inside the code and difficult to find. According to the white-box assault concept, the adversary controls the device and the execution environment in its entirety.

Attacks: Binary instrumentation can be used to monitor and manage the intermediate state of white-box crypto systems, as demonstrated by Bos et al. [55]. As a result, the apparatus enables accurate program execution monitoring and allows for the profiling of program activity through the observation of, among other things, the intermediate state and read/write memory accesses. Bos et al. classified these attacks as differential computation analysis (DCA) attacks due to their resemblance to DPA attacks. However, DCA assaults do not have to contend with measurement noise like DPA attacks do. Even while attacks against white-box crypto implementations haven't been used on mobile devices yet, these devices can also be targets of this type of attack..

Smudge Attacks: The touchscreen on mobile devices is the most widely used input mechanism; users swipe and tap the screen with their fingers. Users always leave traces of their fingerprints and smudges on touch screens because of this intrinsic feature.

Attacks: According to Aviv et al. [56], some interactions with the smartphone or touchscreen-based devices in general can result in side-channel attacks. More precisely, unlock patterns can be deduced by forensic examination of smudges—oily fingerprint residues—on the touchscreen. Even when the phone has been cleaned and the When placing the phone in the pocket, smudges seem to stay most of the time. Smudges are therefore quite persistent, which raises the possibility of smudge attacks. Zhang et al. [57] have provided follow-up work that takes into account an attacker using fingerprint powder to infer keypad inputs. Additionally, an examination using thermal cameras has been conducted into the heat traces left on the screen as a result of finger touches [58].

Shoulder Surfing and Reflections: Touchscreens of mobile devices optically/visually emanate the displayed content. Often these visual emanations are reflected by objects in the envi ronment, such as sunglasses and tea pots [59], [60].

Attacks: Maggi et al. [61] observed that touchscreen input can be recovered by monitoring the visual feedback (pop-up characters) on soft keyboards during the user input. Therefore, they rely on cameras that are pointed directly on the targeted screen. Raguram et al. [62], [63] observed that reflections, e.g., on the user's sunglasses, can also be used to recover input typed on touchscreens. However, the attacker needs to point the camera, used to capture the reflections, directly on the targeted user. Subsequently, they rely on computer vision techniques and machine learning techniques to infer the user input from the captured video stream. Xu et al. [64] extended the range of reflection-based attacks by considering reflections of reflections. Although, they do not rely on the visual feed back of the soft keyboard but instead track the user's fingers on the smartphone while interacting with the device. By increasing the distance between the attacker and the victim, e.g., by relying on more expensive and sophisticated cameras, some of these attacks might as well be considered as vicinity attacks.

Hand/gadget Movements: A lot of input techniques on different devices rely on the user using her hands and fingers to control the gadget. For example, while using their fingers to operate the device, users typically hold it in their hands.

Attacks: Shukla et al. [65] suggested using hand and finger gestures as well as reflections to infer inputted PIN inputs without actually aiming the camera at the intended screen. While Yue et al. [67] presented an attack where the input on touch-enabled devices may be predicted from a video of a

victim tapping on a touch screen, Sun et al. [66] monitored the backside of tablets during user input and observed small motions that can be utilized to infer keystrokes.

Once more, these attacks might also be classified as vicinity attacks due to the increased distance between the attacker and the victim, indicating the smooth transition between local and vicinity attacks for these kinds of attacks.

B. Active Attacks

Additionally, an active attacker modifies the target, its input, or its surroundings in order to directly circumvent security measures or to watch information leakage through the target's aberrant behavior. Active assaults always presume that the attacker is in possession of the device (at least temporarily), whereas passive attacks seamlessly switch between local and nearby attackers.

The origins of active attacks against cryptographic implementations can be traced back to the work of Boneh et al. [68] (also known as the Bellcore attack), who exploited random hardware flaws to attack RSA crypto systems, particularly those based on the Chinese Remainder Theorem (CRT . Subsequently, Biham and Shamir [69] introduced the term differential fault analysis (DFA) attacks and showed that the secret key of symmetric primitives can be recovered by introducing faults and noticing variations in the output ciphertext. These assaults' fundamental goal is to solve algebraic equations using both valid and erroneous outputs.

Clock/Power Glitching: In the past, it has been demonstrated that clock signal variations, such as overclocking, are a useful technique for fault injection on embedded systems. An external source of clock is one of the requirements for this attack. Because most microcontrollers used in smartphones have an internal clock generator, clock tampering is challenging. In addition to tampering with the clock, deliberate changes to the power supply provide another way to introduce faults. Most microcontroller platforms allow for power-supply manipulation with only small hardware modifications.

Attacks: Skorobogatov and Anderson [75] published the first optical fault-injection attacks in 2002, aimed at an 8-bit microcontroller. Many optical fault-injection attacks, primarily aimed at smart cards or low-resource embedded devices, have been reported in the years that followed, largely as a result of their work (e.g., [76] and [77]). The application of optical fault injection on contemporary microprocessors seen in smartphones is challenging due to the growing number of metal layers on top of the silicon, the shrinking feature size (small process technology), and the high decapsulation effort.

Temperature Variation: When a device is operated outside of its designated temperature range, malfunctioning behavior may result. When a device is heated above the maximum recommended temperature, memory cell defects may occur. The pace at which RAM disappears once a device is powered off can be affected by cooling it down (RAM remanence effect).

Attacks: An AVR microprocessor was the victim of thermal fault attacks described by Hutter and Schmidt [78]. Their ability to successfully attack an RSA implementation on a designated microcontroller demonstrates the practicability of this approach. In contrast, FROST [79] is a program that uses cold-boot techniques to get disc encryption keys from RAM on Android devices. Here, the writers profit from the longer period of validity that data in RAM has following a power outage because of low temperature.

Differential Computation Analysis: The white-box approach, as previously stated, is predicated on the attacker having complete control over the execution environment. This also implies that by changing intermediate values throughout the process, the attacker may generate incorrect or flawed outputs.

Attacks: It has been shown by Sanfelix et al. [74] that fault injection assaults are also a viable tactic for attackers in the white-box model. The attacker can alter data while the program is running or the execution's control flow since she has complete control over the execution environment and the binary that is being executed. As with other fault attacks, the goal is to break the cryptographic implementations by looking for variations between the binary's normal and erroneous outputs.

NAND Mirroring: Data mirroring is the process of replicating data storage across several sites. These methods enable the restoration of a prior system state in addition to being utilized for the recovery of crucial data following disasters.

Attacks: By encrypting data, the Apple iPhone safeguards users' privacy. Therefore, several keys that can be used to safeguard the data on the device are derived using a hardware-based key and a passcode. Brute-force attempts must be made on the attacked device because these keys are derived using a specialized hardware-based key. Additionally, by progressively lengthening the interval between incorrectly entered passcodes until the phone is erased, brute-force efforts are discouraged. Regarding the FBI v. Apple case, Skorobogatov [80] shown that the passcode may be brute-forced by using NAND mirroring to reset the phone's state. Given that the attacker actively modifies (resets) the device's state, it is obvious that this strategy also constitutes an active attack.

4. CONCLUSIONS

Considering the immense threat arising from side-channel attacks on mobile devices, a thorough understanding of infor mation leaks and possible exploitation techniques is necessary. Based on this open issue, we surveyed existing side-channel attacks and identified commonalities between these attacks in order to systematically categorize all existing attacks. With the presented classification system we aim to provide a thor ough understanding of information leaks and hope to spur further research in the context of side-channel attacks as well as countermeasures and, thereby, to pave the way for secure computing platforms.

REFERENCES

1. P. C. Kocher, "Timing attacks on implementations of Diffie–Hellman, RSA, DSS, and other systems," in Advances in Cryptology— CRYPTO 1996 (LNCS 1109). Heidelberg, Germany: Springer, 1996, pp. 104–113.

2. P. C. Kocher, J. Jaffe, and B. Jun, "Differential power analysis," in Advances in Cryptology—CRYPTO 1999 (LNCS 1666). Heidelberg, Germany: Springer, 1999, pp. 388–397.

3. J.-J. Quisquater and D. Samyde, "Electromagnetic analysis (EMA): Measures and counter-measures for smart cards," in Smart Card Programming and Security—E-Smart 2001 (LNCS 2140). Heidelberg, Germany: Springer, 2001, pp. 200–210.

4. S. Mangard, E. Oswald, and T. Popp, Power Analysis Attacks—Revealing the Secrets of Smart Cards. New York, NY, USA: Springer, 2007.

5. E. Tromer, D. A. Osvik, and A. Shamir, "Efficient cache attacks on AES, and countermeasures," J. Cryptol., vol. 23, no. 1, pp. 37–71, 2010.

6. Y. Yarom and K. Falkner, "FLUSH+RELOAD: A high resolution, low noise, L3 cache side-channel attack," in Proc. USENIX Security Symp., San Diego, CA, USA, 2014, pp. 719–732.

7. Q. Ge, Y. Yarom, D. Cock, and G. Heiser, "A survey of microarchitec tural timing attacks and countermeasures on contemporary hardware," J. Cryptograph. Eng., pp. 1–27, 2016.

8. P. Pessl, D. Gruss, C. Maurice, M. Schwarz, and S. Mangard, "DRAMA: Exploiting DRAM addressing for cross-CPU attacks," in Proc. USENIX Security Symp., 2016, pp. 565–581.

9. L. Cai and H. Chen, "TouchLogger: Inferring keystrokes on touch screen from smartphone motion," in Proc. USENIX Workshop Hot Topics Security (HotSec), San Francisco, CA, USA, 2011. [Online]. Available: https://www.usenix.org/conference/hotsec11/touchlogger inferring-keystrokes-touch-screen-smartphone-motion

10. A. J. Aviv, B. Sapp, M. Blaze, and J. M. Smith, "Practicality of accelerometer side channels on smartphones," in Proc. Annu. Comput. Security Appl. Conf. (ACSAC), Orlando, FL, USA, 2012, pp. 41–50.

11. L. Simon, W. Xu, and R. Anderson, "Don't interrupt me while I type: Inferring text entered through gesture typing on Android keyboards," Proc. Privacy Enhancing Technol., vol. 2016, no. 3, pp. 136–154, 2016.

12. M. Mehrnezhad, E. Toreini, S. F. Shahandashti, and F. Hao, "TouchSignatures: Identification of user touch actions and PINs based on mobile sensor data via JavaScript," J. Inf. Security Appl., vol. 26, pp. 23–38, Feb. 2016.

13. Y. Michalevsky, A. Schulman, G. A. Veerapandian, D. Boneh, and G. Nakibly, "PowerSpy: Location tracking using mobile device power analysis," in Proc. USENIX Security Symp., 2015, pp. 785–800.

14. X. Zhou et al., "Identity, location, disease and more: Inferring your secrets from Android public resources," in Proc. Conf. Comput. Commun. Security (CCS), Berlin, Germany, 2013, pp. 1017–1028.

15. Y. Acar et al., "SoK: Lessons learned from Android security research for appified software platforms," in Proc. IEEE Symp. Security Privacy (S P), San Jose, CA, USA, 2016, pp. 433–451.

16. N. Zhang, K. Yuan, M. Naveed, X. Zhou, and X. Wang, "Leave me alone: App-level protection against runtime information gathering on Android," in Proc. IEEE Symp. Security Privacy (S P), San Jose, CA, USA, 2015, pp. 915–930.

17. P. Marquardt, A. Verma, H. Carter, and P. Traynor, "(sp)iPhone: Decoding vibrations from nearby keyboards using mobile phone accelerometers," in Proc. Conf. Comput. Commun. Security (CCS), Chicago, IL, USA, 2011, pp. 551–562.

18. T. Zhu, Q. Ma, S. Zhang, and Y. Liu, "Context-free attacks using key board acoustic emanations," in Proc. Conf. Comput. Commun. Security (CCS), Scottsdale, AZ, USA, 2014, pp. 453–464.

19. S. Biedermann, S. Katzenbeisser, and J. Szefer, "Hard drive side-channel attacks using smartphone magnetic field sensors," in Financial Cryptography—FC 2015 (LNCS 8975). Heidelberg, Germany: Springer, 2015, pp. 489–496.

20. L. Schwittmann, V. Matkovic, M. Wander, and T. Weis, "Video recogni tion using ambient light sensors," in Proc. Pervasive Comput. Commun. Workshops (PerCom), Sydney, NSW, Australia, 2016, pp. 1–9.

21. C. Song et al., "My smartphone knows what you print: Exploring smartphone-based side-channel attacks against 3D printers," in Proc. Conf. Comput. Commun. Security (CCS), Vienna, Austria, 2016, pp. 895–907.

22. A. Hojjati et al., "Leave your phone at the door: Side channels that reveal factory floor secrets," in Proc. Conf. Comput. Commun. Security (CCS), Vienna, Austria, 2016, pp. 883–894.

23. D. Gruss, R. Spreitzer, and S. Mangard, "Cache template attacks: Automating attacks on inclusive last-level caches," in Proc. USENIX Security Symp., Washington, DC, USA, 2015, pp. 897–912.

24. D. Gruss, D. Bidner, and S. Mangard, "Practical memory deduplication attacks in sandboxed JavaScript," in Proc. Eur. Symp. Res. Comput. Security (ESORICS), vol. 9326. Vienna, Austria, 2015, pp. 108–122.

25. S. Jana and V. Shmatikov, "Memento: Learning secrets from process footprints," in Proc. IEEE Symp. Security Privacy (S P), San Francisco, CA, USA, 2012, pp. 143–157.

26. R. Spreitzer, S. Griesmayr, T. Korak, and S. Mangard, "Exploiting data usage statistics for website fingerprinting attacks on Android," in Proc. Security Privacy Wireless Mobile Netw. (WISEC), Darmstadt, Germany, 2016, pp. 49–60.

27. Global Market Share Held by the Leading Smartphone Operating Systems in Sales to End Users From 1st Quarter 2009 to 1st Quarter 2017, Gartner, Stamford, CT, USA, accessed: Jun. 13, 2017.

28. A. D. Luzio, A. Mei, and J. Stefa, "Mind your probes: De-anonymization of large crowds through smartphone WiFi probe requests," in Proc. IEEE INFOCOM, San Francisco, CA, USA, 2016, pp. 1–9.

29. 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 Applied Cryptography and Network Security— ACNS 2017. Cham, Switzerland: Springer, 2017.

30. W. Enck, "Defending users against smartphone apps: Techniques and future directions," in Information Systems

Security—ICISS (LNCS 7093). Heidelberg, Germany: Springer, 2011, pp. 49–70.

31. M. L. Polla, F. Martinelli, and D. Sgandurra, "A survey on security for mobile devices," IEEE Commun. Surveys Tuts., vol. 15, no. 1, pp. 446–471, 1st Quart., 2013.

32. G. Suarez-Tangil, J. E. Tapiador, P. Peris-Lopez, and A. Ribagorda, "Evolution, detection and analysis of malware for smart devices," IEEE Commun. Surveys Tuts., vol. 16, no. 2, pp. 961–987, 2nd Quart., 2014.

33. P. Faruki et al., "Android security: A survey of issues, malware pen etration, and defenses," IEEE Commun. Surveys Tuts., vol. 17, no. 2, pp. 998–1022, 2nd Quart., 2015.

34. B. Rashidi and C. J. Fung, "A survey of Android security threats and defenses," J. Wireless Mobile Netw. Ubiquitous Comput. Depend. Appl., vol. 6, no. 3, pp. 3–35, 2015.

35. A. Sadeghi, H. Bagheri, J. Garcia, and S. Malek, "A taxonomy and qualitative comparison of program analysis techniques for security assessment of Android software," IEEE Trans. Softw. Eng., vol. 43, no. 6, pp. 492–530, Jun. 2017.

36. K. Tam, A. Feizollah, N. B. Anuar, R. Salleh, and L. Cavallaro, "The evolution of Android malware and Android analysis techniques," ACM Comput. Surveys, vol. 49, no. 4, pp. 1–41, 2017.

37. M. Tunstall, Smart Card Security. Cham, Switzerland: Springer Int., 2017, pp. 217–251.

38. S. Zander, G. J. Armitage, and P. Branch, "A survey of covert channels and countermeasures in computer network protocols," IEEE Commun. Surveys Tuts., vol. 9, no. 3, pp. 44–57, 3rd Quart., 2007.

39. A. K. Biswas, D. Ghosal, and S. Nagaraja, "A survey of timing chan nels and countermeasures," ACM Comput. Surveys, vol. 50, no. 1, pp. 1–39, 2017.

40. J. Szefer, "Survey of microarchitectural side and covert channels, attacks, and defenses," IACR Cryptology ePrint Archive, Report 2016/479, 2016.

41. J. Ullrich, T. Zseby, J. Fabini, and E. R. Weippl, "Network-based secret communication in clouds: A survey," IEEE Commun. Surveys Tuts., vol. 19, no. 2, pp. 1112–1144, 2nd Quart., 2017.

42. J. Betz, D. Westhoff, and G. Müller, "Survey on covert channels in virtual machines and cloud computing," Trans. Emerg. Telecommun. Technol., vol. 28, no. 6, 2017, Art. no. e3134.

43. M. Xu et al., "Toward engineering a secure Android ecosystem: A survey of existing techniques," ACM Comput. Surveys, vol. 49, no. 2, pp. 1–47, 2016.

44. M. Hussain et al., "The rise of keyloggers on smartphones: A survey and insight into motion-based tap inference attacks," Pervasive Mobile Comput., vol. 25, pp. 1–25, Jan. 2016.

45. A. Nahapetian, "Side-channel attacks on mobile and wearable systems," in Proc. Consum. Commun. Netw. Conf. (CCNC), Las Vegas, NV, USA, 2016, pp. 243–247.

46. Q. Xiao, M. K. Reiter, and Y. Zhang, "Mitigating storage side channels using statistical privacy mechanisms," in Proc. Conf. Comput. Commun. Security (CCS), Denver, CO, USA, 2015, pp. 1582–1594.

47. C. Lin, H. Li, X. Zhou, and X. Wang, "Screenmilker: How to milk your Android screen for secrets," in Proc. Netw. Distrib. Syst. Security Symp. (NDSS), 2014, doi: 10.14722/ndss.2014.23049.

48. T. S. Messerges, E. A. Dabbish, and R. H. Sloan, "Investigations of power analysis attacks on smartcards," in Proc. Workshop Smartcard Technol. (Smartcard), 1999, p. 17.

49. L. Yan, Y. Guo, X. Chen, and H. Mei, "A study on power side chan nels on mobile devices," in Proc. Symp. Internetware (Internetware), Wuhan, China, 2015, pp. 30–38.

50. C. H. Gebotys, S. Ho, and C. C. Tiu, "EM analysis of Rijndael and ECC on a wireless Java-based PDA," in Cryptographic Hardware and Embedded Systems—CHES (LNCS 3659). Heidelberg, Germany: Springer, 2005, pp. 250–264.

51. Y. Nakano et al., "A pre-processing composition for secret key recov ery on Android smartphone," in Information Security Theory and Practice—WISTP 2014 (LNCS 8501). Heidelberg, Germany: Springer, 2014, pp. 76–91.

52. G. Goller and G. Sigl, "Side channel attacks on smartphones and embedded devices using standard radio equipment," in Constructive Side-Channel Analysis and Secure Design—COSADE 2015 (LNCS 9064). Cham, Switzerland: Springer, 2015, pp. 255–270.

53. P. Belgarric, P. Fouque, G. Macario-Rat, and M. Tibouchi, "Side channel analysis of weierstrass and Koblitz curve ECDSA on Android smartphones," in Topics in Cryptology—CT-RSA 2016 (LNCS 9610). Cham, Switzerland: Springer, 2016, pp. 236–252.

54. D. Genkin, L. Pachmanov, I. Pipman, E. Tromer, and Y. Yarom, "ECDSA key extraction from mobile devices via nonintrusive phys ical side channels," in Proc. Conf. Comput. Commun. Security (CCS), Vienna, Austria, 2016, pp. 1626–1638.

55. J. W. Bos, C. Hubain, W. Michiels, and P. Teuwen, "Differential com putation analysis: Hiding your white-box designs is not enough," in Cryptographic Hardware and Embedded Systems—CHES 2016 (LNCS 9813). Heidelberg, Germany: Springer, 2016, pp. 215–236.

56. A. J. Aviv, K. L. Gibson, E. Mossop, M. Blaze, and J. M. Smith, "Smudge attacks on smartphone touch screens," in Proc. Workshop Offensive Technol. (WOOT), Washington, DC, USA, 2010.

57. Y. Zhang et al., "Fingerprint attack against touch-enabled devices," in Proc. Security Privacy Smartphones Mobile Devices SPSM@CCS, Raleigh, NC, USA, 2012, pp. 57–68.

58. P. Andriotis, T. Tryfonas, G. C. Oikonomou, and C. Yildiz, "A pilot study on the security of pattern screen-lock methods and soft side chan nel attacks," in Proc. Security Privacy Wireless Mobile Netw. (WISEC), Budapest, Hungary, 2013, pp. 1–6.

59. M. Backes, M. Dürmuth, and D. Unruh, "Compromising reflections or-how to read LCD monitors around the corner," in Proc. IEEE Symp. Security Privacy (S P), Oakland, CA, USA, 2008, pp. 158–169.

60. M. Backes, T. Chen, M. Dürmuth, H. P. A. Lensch, and M. Welk, "Tempest in a teapot: Compromising reflections revisited," in Proc. IEEE Symp. Security Privacy (S P), Berkeley, CA, USA, 2009, pp. 315–327.

61. F. Maggi, S. Gasparini, and G. Boracchi, "A fast eavesdropping attack against touchscreens," in Proc. Inf. Assurance Security (IAS), 2011, pp. 320–325.

62. R. Raguram, A. M. White, D. Goswami, F. Monrose, and J.-M. Frahm, "iSpy: Automatic reconstruction of typed input from compromising reflections," in Proc. Conf. Comput. Commun. Security (CCS), Chicago, IL, USA, 2011, pp. 527–536.

63. R. Raguram et al., “On the privacy risks of virtual keyboards: Automatic reconstruction of typed input from compromising reflec tions,” IEEE Trans. Depend. Secure Comput., vol. 10, no. 3, pp. 154–167, May/Jun. 2013.

64. Y. Xu, J. Heinly, A. M. White, F. Monrose, and J.-M. Frahm, “Seeing double: Reconstructing obscured typed input from repeated compro mising reflections,” in Proc. Conf. Comput. Commun. Security (CCS), Berlin, Germany, 2013, pp. 1063–1074.

65. D. Shukla, R. Kumar, A. Serwadda, and V. V. Phoha, “Beware, your hands reveal your secrets!” in Proc. Conf. Comput. Commun. Security (CCS), Scottsdale, AZ, USA, 2014, pp. 904–917.
66. J. Sun et al., “VISIBLE: Video-assisted keystroke inference from tablet backside motion,” in Proc. Netw. Distrib. Syst. Security Symp. (NDSS), San Diego, CA, USA, 2016, doi: 10.14722/ndss.2016.23.

67. Q. Yue et al., “Blind recognition of touched keys on mobile devices,” in Proc. Conf. Comput. Commun. Security (CCS), Scottsdale, AZ, USA, 2014, pp. 1403–1414.

68. D. Boneh, R. A. DeMillo, and R. J. Lipton, “On the importance of checking cryptographic protocols for faults (extended abstract),” in Advances in Cryptology—EUROCRYPT 1997 (LNCS 1233). Heidelberg, Germany: Springer, 1997, pp. 37–51.

69. E. Biham and A. Shamir, “Differential fault analysis of secret key cryptosystems,” in Advances in Cryptology—CRYPTO 1997 (LNCS 1294). Heidelberg, Germany: Springer, 1997, pp. 513–525.

70. Fault Injection Raspberry PI, NewAE Technol. Inc., Halifax, NS, Canada, accessed: Aug. 3, 2016. [Online]. Available: https://wiki.newae.com

71. C. O’Flynn, “Fault injection using crowbars on embedded systems,” IACR Cryptology ePrint Archive, Report 2016/810, 2016. [Online]. Available: https://eprint.iacr.org/2016/810

72. S. Ordas, L. Guillaume-Sage, and P. Maurine, “Electromagnetic fault injection: The curse of flip-flops,” J. Cryptograph. Eng., vol. 7, no. 3, pp. 183–197, 2017.

73. L. Rivière et al., “High precision fault injections on the instruction cache of ARMv7-M architectures,” in Proc. Hardw. Orient. Security Trust (HOST), Washington, DC, USA, 2015, pp. 62–67.

74. E. Sanfelix, C. Mune, and J. de Haas, “Unboxing the white-box: Practical attacks against obfuscated ciphers,” Blackhat, 2015.

75. S. P. Skorobogatov and R. J. Anderson, “Optical fault induction attacks,” in Cryptographic Hardware and Embedded Systems—CHES 2002 (LNCS 2523). Heidelberg, Germany: Springer, 2002, pp. 2–12.

76. J. G. J. van Woudenberg, M. F. Witteman, and F. Menarini, “Practical optical fault injection on secure microcontrollers,” in Proc. Fault Diagnosis Tolerance Cryptograp. (FDTC), Nara, Japan, 2011, pp. 91–99.

77. C. Roscian, A. Sarafianos, J. Dutertre, and A. Tria, “Fault model anal ysis of laser-induced faults in SRAM memory cells,” in Proc. Fault Diagnosis Tolerance Cryptograph. (FDTC), Santa Barbara, CA, USA, 2013, pp. 89–98.

78. M. Hutter and J.-M. Schmidt, “The temperature side channel and heating fault attacks,” in Smart Card Research and Advanced Applications—CARDIS 2013 (LNCS 8419). Cham, Switzerland: Springer, 2013, pp. 219–235.

79. T. Müller and M. Spreitzenbarth, “FROST—Forensic recovery of scrambled telephones,” in Applied Cryptography and Network Security—ACNS 2013 (LNCS 7954). Berlin, Germany: Springer, 2013, pp. 373–388.

80. S. Skorobogatov, “The bumpy road towards iPhone 5c NAND mir roring,” arXiv ePrint Archive, Report 1609.04327, 2016.

81. Randolph, M.; Diehl, W. Power Side-Channel Attack Analysis: A Review of 20 Years of Study for the Layman. Cryptography 2020, 4, 15. https://doi.org/10.3390/cryptography4020015