Poster — Magneto: Leveraging Magnetic Field Changes for Inferring Smartphone App Usage

Meenu Rani Dey*
IIIT Bhubaneshwar, India
a117004@iiit-bh.ac.in

Satadal Sengupta
IIT Kharagpur, India
satadal.sengupta@iitkgp.ac.in

Bhabendu Kr. Mohanta
IIIT Bhubaneshwar, India
c116004@iiit-bh.ac.in

Debasish Jena
IIIT Bhubaneshwar, India
debasish@iiit-bh.ac.in

Sandip Chakraborty
IIT Kharagpur, India
sandipc@cse.iitkgp.ernet.in

ABSTRACT

Side-channel attacks, which exploit deficiencies in the implementations of theoretically secure systems, have been known to take a variety of forms on the mobile platforms. In this work, we present *Magneto*, a magnetic field based app classification mechanism. Magneto captures the Hall effect due to energy consumption by different components in a smartphone, and fingerprints apps based on data captured with a Hall sensor, and a phone magnetometer. We demonstrate that our mechanism can identify magnetic field changes due to varying levels of energy consumption. We further show that Magento can not only classify between apps in the same scenario, but also can tell apart scenarios when the phone is being charged (with an AC adapter, wireless charger, or powerbank) or not. We perform validation experiments with 5 different apps, and achieve $\sim$ 85% accuracy with 3 Android apps, subject to 3 different charging scenarios.

CCS CONCEPTS

• **Security and privacy → Systems security**; Mobile and wireless security;

KEYWORDS

magnetic field changes; power consumption; mobile apps

*The work was done during her internship in IIT Kharagpur.

MobiCom '18, October 29-November 2, 2018, New Delhi, India

ACM ISBN 978-1-4503-5903-0/18/10.
https://doi.org/10.1145/3241539.3267772

1 INTRODUCTION

The proliferation of smartphones in day-to-day usage has engendered fresh challenges in the domains of user security and privacy. Extensive use of mobile devices has led to the addition of a newer dimension to the already challenging area concerning *side-channel attacks*. Side-channel attacks are characerized with gleaning out sensitive data via indirect means, i.e., by exploiting signatures that manifest from implementation specifics of an otherwise secure system. Recent studies have shown that a myriad of passive yet fine-grained side-channel attacks are possible on mobile platforms, due to both the ubiquity of devices, as well as the variety of sources through which the devices emit data [1]. For example, analysis of power consumption data has been shown to reveal clues regarding the application usage behaviour of smartphone users [4].

Non-intrusive fingerprinting of mobile application usage throws open opportunities for a variety of use-cases, from the perspectives of both security and monitoring. While knowledge of app usage can aid and abet privacy breaches and associated security attacks (e.g., if an attacker knows which app is running, he can launch a targetted attack), it can also enable policy enforcers, such as enterprise administrators, to monitor the productivity of employees (measuring time spent on productive apps like email, vs. entertainment apps).

In this abstract, we propose *Magneto*[1], a magnetic field (MF) based mechanism for app usage fingerprinting. Magneto works by identifying the changes in MF in the vicinity of a smartphone, caused due to power consumption during the execution of an app. We capture the MF strength around a device using 2 types of sensors, i.e., (1) a *Hall* sensor, and (2) the magnetometer sensor present in smartphones. We hypothesize and validate (using 5 Android apps from 5 different categories) that the power consumption due to different components of a smartphone (e.g., background CPU, foreground

[1]Magneto is an American comic-book character, who possesses the ability to leverage magnetic fields around himself to his advantage.

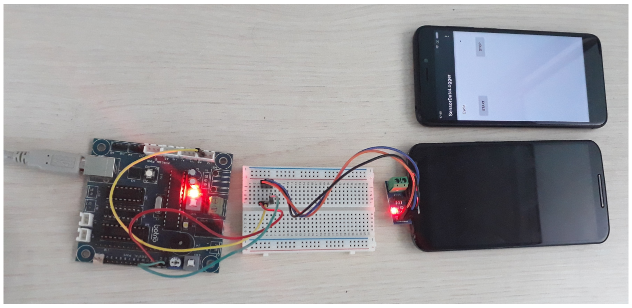

Figure 1: Experimental in-laboratory setup: The Hall sensor connected to an Arduino board collects Hall effect values, while a Moto X2 phone collects magnetometer values from the target smartphone.

CPU, screen, etc.) manifest in the overall energy consumption of an app, which in turn leaves footprints on the MF around the device. Furthermore, we derive features from the collected data, and perform classification using a machine learning based model. We show that not only are we able to differentiate among appss based on their energy footprints, we are also able to segregate scenarios based on whether the phone is being charged (with AC adapter, powerbank, or wireless charger) or not. Our mechanism is able to achieve $\sim 85\%$ classification accuracy overall, while using 3 representative smartphone apps, and 3 phone charging scenarios.

Having demonstrated the efficacy of Magneto in identifying smartphone apps from MF data, we identify 2 immediate directions along which the mechanism can be improved: (1) Identifying apps when multiple apps are executed together, and (2) Handling effects of external factors (i.e., temperature) on the behaviour of sensors. We discuss proposed resolutions as part of future work.

2 EXPERIMENTAL SETUP

We describe the experimental setup in this section.

Hall effect: The *Hall effect*, due to Edwin Hall in 1879, is a phenomenon by which a potential difference is produced across an electrical conductor, when a MF is applied perpendicular to the flow of current inside it [3]. We observed the creation of a Hall voltage in the vicinity of a smartphone, when an app is executed on it.

Sensors: We measured the Hall effect voltage using a *Hall effect sensor* (ACS712).We also used the magnetometer sensor available in commodity smartphones, which measures direction and strength of MF at a particular location.

Microcontroller: We use an *Arduino* board for recording data from the Hall effect sensor. The target is a Motorola Moto X 2^{nd} generation Android smartphone. The complete setup is shown in Fig. 1.

Apps considered: Existing studies show that different components in a phone exhibit different energy consumption behaviour. Based on the observations in [2], we consider the following components: (1) Foreground sceen, (2) Foreground CPU, (3) Foreground Network, (4) Background CPU, (5) Background Network. We consider one Android app each from the following categories, therefore, as follows:

(1) Foreground Screen – Image Gallery
(2) Foreground CPU + (1) – MX Video Player (offline video)
(3) Foreground Network + (2) – Youtube
(4) Background CPU – Google Play Music (offline audio playback)
(5) Background Network + (4) – Google Play Store (app download)

Apps (1), (2), (3), which are foreground apps, should ideally exhibit energy consumption in an increasing order, as should background apps (4) & (5), respectively.

Automated data collection: We use the Google developer provided Android app testing and automation tool *Monkeyrunner* for automating data collection from the aforementioned apps for long durations.

3 EVALUATION

We present our preliminary results in this section.

Validation of component-wise power consumption: We report the measured MF results for the 5 Android apps mentioned in the previous section, in Fig. 2. We observe that for the foreground apps, Image Gallery and MX Video show similar MF signatures. However, Youtube shows much higher values. For the background apps, similarly, the MF values are consistently higher as we move from Google Play Music to Google Play Store. This indicates that power consumption is indeed reflected in the MF patterns captured by our sensors.

Charging scenarios: Next, we take a look at the differences in MF values obtained, when subjected to different charging scenarios, i.e., (1) No charging, (2) AC adapter, (3) Wireless charger, and (4) Powerbank. For this experiment, we consider only the Youtube app. The observations are shown in Fig. 3. We find that the MF signatures are distinct for all the 4 charging scenarios considered.

Charging scenario and app combinations: We extend our experiments to analyzing the MF values obtained for different charging scenario and app combinations. We consider the (1) No charging, (2) AC adapter, and (3) Powerbank charging scenarios, and the (1) Facebook Messenger, (2) WhatsApp,

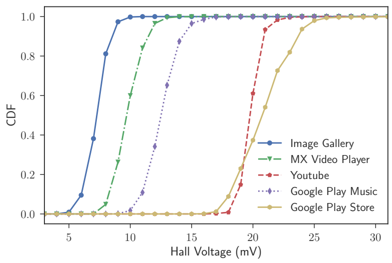

Figure 2: CDF of MF values for 5 apps to validate energy consumption

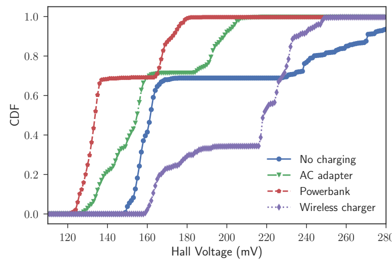

Figure 3: CDF of MF values for different charging scenarios

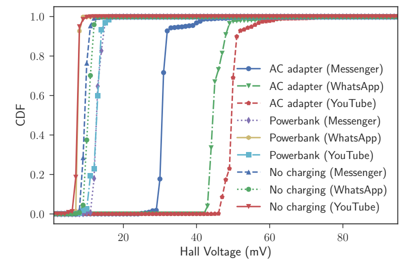

Figure 4: CDF of MF values for charging scenario and app combinations

and (3) Youtube apps. The CDFs for the different combinations are presented in Fig. 4. We observe once again that the MF signatures are different for the different cases. As an example, we see that WhatsApp shows the highest MF values consistently (and therefore highest energy consumption); however, we can still distinctly identify which charging scenario WhatsApp is being executed in.

Table 1: Results for 3 charging scenarios & 3 apps

Classifier	F1-Score	Accuracy (%)
SVM	0.86	85.38
Random Forest	0.86	85.79
Random Tree	0.85	85.30

Classification results: The classification results for 3 charging scenarios, and 3 Android applications are presented in Table 1. We consider 3 different classifiers for this experiment, i.e., (1) SVM, (2) Random Forest, and (3) Random Tree. We obtain $\sim$ 0.86 F1-Score, and $\sim$ 85% accuracy on average, in classifying charging scenario-app combinations. These results indicate the efficacy of our proposed mechanism Magneto, in discriminating among MF signatures, under varying app and charging conditions.

4 CONCLUSION AND FUTURE WORK

In this work, we presented a MF based mechanism *Magneto*, which learns magnetic signatures from mobile app usage, and leverages such learning to classify different Android apps under various charging scenarios, with $\sim$ 85% accuracy. We also show that MF values are representative of power consumption by different smartphone components.

Future directions: In this abstract, we presented preliminary results for our mechanism *Magneto*. However, a number of perspectives remain unaddressed, which we discuss here.

Effect of external factors on Hall sensor and magnetometer: During our experiments, we observed that the reading from our sensors are heavily dependent on the environment we subject those to. For example, temperature plays an important role – the higher the room temperature, higher is the range of values. In fact, we observed that factors such as how many ACs are on in the room where the experiment is being conducted, or what temperature they are set at, have bearing on the MF signatures of applications. We also observed that the number of humans, their mobility, and the arrangement of furniture in the room also affect the MF readings from our sensors. We plan to resolve these issues by carefully observing the ranges for each factor, and then categorizing those into distinct *environments*. We can then train use a *Multitask Learning* framework to address the variety in training conditions.

Effect of running multiple apps: We also need to address the effects of executing multiple apps on the same smartphone. We expect to receive superimposed signatures, which may be hard to dissect, and would therefore require detailed analysis.

Experiments in the wild: In this abstract, we presented observations from experiments in laboratory conditions. However, the observations may vary widely when subjected to in-the-wild testing. We plan to deploy a small sensor box in different areas of the university, to study such effects.

REFERENCES

[1] Shuo Chen, Rui Wang, XiaoFeng Wang, and Kehuan Zhang. 2010. Side-channel leaks in web applications: A reality today, a challenge tomorrow. In *Security and Privacy (SP), 2010 IEEE Symposium on*. IEEE, 191–206.

[2] Xiaomeng Chen, Ning Ding, Abhilash Jindal, Y Charlie Hu, Maruti Gupta, and Rath Vannithamby. 2015. Smartphone energy drain in the wild: Analysis and implications. *ACM SIGMETRICS Performance Evaluation Review* 43, 1 (2015), 151–164.

[3] Edward Ramsden. 2011. *Hall-effect sensors: theory and application*. Elsevier.

[4] Lin Yan, Yao Guo, Xiangqun Chen, and Hong Mei. 2015. A study on power side channels on mobile devices. In *Proceedings of the 7th Asia-Pacific Symposium on Internetware*. ACM, 30–38.