The Curious Case of the Curious Case: Detecting touchscreen events using a smartphone protective case

Tomer Gluck, Rami Puzis, Yossi Oren and Asaf Shabtai
Ben-Gurion University of the Negev
Beer-Sheva, Israel
gluckt@post.bgu.ac.il, {puzis,yos,shabtaia}@bgu.ac.il

***Abstract*—Security-conscious users are very careful with software they allow their phone to run. They are much less careful with the choices they make regarding accessories such as headphones or chargers and only few, if any, care about cyber security threats coming from the phone's protective case. We show how a malicious smartphone protective case can be used to detect and monitor the victim's interaction with the phone's touchscreen, opening the door to keylogger-like attacks, threatening the user's security and privacy. This feat is achieved by implementing a hidden capacitive sensing mechanism inside the case. Our attack is both sensitive enough to track the user's finger location across the screen, and simple and cheap enough to be mass-produced and deployed en masse. We discuss the theoretical principles behind this attack, present a preliminary proof-of-concept, and discuss potential countermeasures and mitigations.**

***Index Terms*—touchscreen leak, security, privacy, smartphone.**

1. Introduction

Personal mobile devices are widely popular and are now used to perform sensitive tasks such as banking, sending and receiving work-related emails, and even connecting remotely to servers. In each of these tasks the user enters private data into the device using the touchscreen, either for the purpose of authentication or as part of the data exchange. Adversaries who are interested in obtaining touch event information often use malicious software.

Many users take protective measures, such as using anti-malware or following *cautious behavior* (e.g., install applications from trusted sources, avoid unknown links), for protecting their private information. Organizations define strict security policy that is applied by installing dedicated software and/or hardware components on the employees phones. None of these measures, however, consider the phone protective case as a security risk. Thus, for higher-value targets, malicious adversaries may decide to make use of custom hardware devices, commonly called bugs or implants.

As discussed by Farshteindiker et al. in [1], a malicious implant has three main functional requirements: first, it must be able to *collect data* from its victim; next, it must be able to *exfiltrate* the data to the attacker; finally, the implant requires some sort of *power supply* to power its computation and communication functions. Farshteindiker et al. suggest a method by which an implant in close proximity to the phone can use the phone itself to exfiltrate data. This method is using a transducer that can be placed inside the protective case and communicate with a web-page through the gyroscope readings. Farshteindiker et al. also discuss several options for supplying power to such a device. Our paper focuses on the *data collection* functionality of the implant. In particular, we present a method for remotely detecting touch screen events using a malicious smartphone protective case, which is cheap and simple enough to be mass-produced.

In order to get the malicious protective case onto the user's phone, the attacker can use a social engineering attack in which the case is distributed or given as a gift or souvenir (e.g., at a trade show or scientific conference). For mass information theft, the implant can be injected into the supply chain using various supply-chain interdiction methods [2].

In this study, we focus on protective cases as an attack platform, because they are widely used, are almost always attached to the victim's phone (in contrast to chargers or headphones), and surround the phone on all sides (a fact that aids in detection of touch events). Using a case has the additional advantage of working on any device, regardless of its operating system or internal structure. Since the malicious case is completely separated from the phone's hardware and software, it does not require a dedicated application to run on the phone, and it work in a "plug and play" manner.

Contribution. We design and evaluate a capacitive sensor which can be implanted inside a phone's protective case and is capable of detecting the motion of a user's finger on a smartphone's touch screen with a high precision. We discuss the hardware and software requirements of our sensor and show how it can be used to recover a phone's secret unlock pattern. After the motion data is analyzed, it can be stored or transmitted to the attacker using one of the methods described in prior research [1].

2. Capacitive Sensing

The touchscreens of nearly all modern smartphones and tablets use capacitive sensing to detect touch on the screen. There are two main methods of capacitive sensing: surface capacitance and projected capacitance.

In the *surface capacitance* sensing method, four electrodes are placed on the four edges of the screen and they are connected to a conductive layer overlay the whole screen. When the user's finger touches the screen's conductive surface, a voltage drop measured by these electrodes changes relative to the finger's distance from the electrodes. In the *projected capacitance* sensing method, a grid of equidistant electrodes is placed under the screen glass. Each adjacent pair of electrodes then forms a capacitor with a fixed capacitance, which is measurable by applying voltage to one of the electrodes and measuring how long it takes for the voltage on the other electrode to reach a certain value. Placing a conductive object, such as a finger, in the magnetic field emitted by the electrodes, increases the amount of charge that can be stored in the electrodes and hence increases the capacitance. Consequently, the time required to charge the capacitor increases as the finger gets closer to the electrodes. In contrast to the surface capacitance method, this method requires an additional layer of signal processing and noise reduction to derive the finger's exact coordinates.

In our implementation, we use the projected capacitive sensing method due to its simplicity and ability to detect the finger without actual contact with the sensor.

3. Attack Description

The main idea behind the discussed attack is obtaining the touch position on a touchscreen using a simple looking protective case. In order for the sensor to be implementable in a typical phone protective case, it needs to be located on the edges of the phone. In addition to capacitive sensing, there are a few alternative methods that can be used for touch sensing using a sensor that circles the screen. Those methods include Infrared (IR) touch sensing or surface acoustic wave (SAW) touch sensing. Neither of these methods is suited for the task, since the IR sensor needs relatively large bezels to fit the LEDs, and the SAW touch will only work with very good contact between the sensor and the screen, something that a plastic case cannot provide.

As mentioned, our implementation uses the projected capacitive sensing method. Our setup consists of four electrodes, acting as receivers, and an additional electrode that serves as the transmitter. Not to be confused with the four electrodes of the *surface capacitance*, our implementation is not connected to conductive layer on the screen. As seen in Figure 1 (blue bars), the four electrodes are placed on the inner part of the case, one on each of the four sides. The transmitting electrode sits on the back of the phone. Traditional projected capacitive touchscreens use an array of electrodes that are placed very close to each other. This arrangement can only detect touch events in very short distances, usually covering just the thickness of the glass so it can detect touch. In our implementation, capacitance is used differently than regular projected capacitance; since it has a wider gap between electrodes, it can sense and measure a greater distance of the finger from the four electrodes. In the following demonstration we attempt to obtain a phone's unlock pattern, commonly used in Android smartphones.

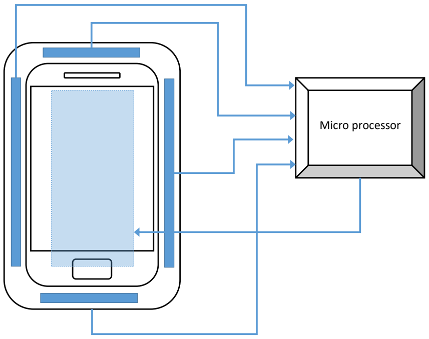


Figure 1: Electrode placement relative to the phone

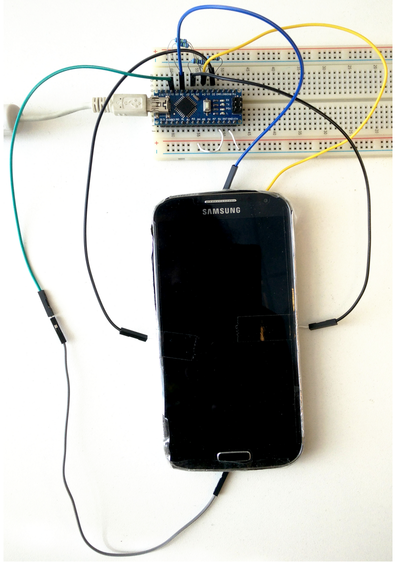


Figure 2: Experimental setup

3.1. Evaluation Setup

We designed and conducted an experiment to demonstrate and evaluate the effectiveness of the studied attack. The hardware setup of our experiment is presented in Figure 3. We concentrated on keeping the setup as cheap and simple as possible, both in terms of the choice of components and the choice of construction methods. We used a Samsung Galaxy S5, and all of the tests were conducted while the phone was turned on and placed flat on a desk, with one finger touching the screen.

The **five electrodes** composing the *touch sensor* were made of ordinary aluminum foil. Four electrodes, placed on the sides of the phone, are cut into a narrow rectangular shape (two 14cm long and two 7cm long) that runs the length of the sides, top and bottom of the phone. The electrode placed on the back of the phone is also a rectangle ($10cm \times 3cm$), sized so that it can sit at the center of the back of the phone approximately 2cm from the phone edges. All of the electrodes are taped to the phone, with an isolation layer between them and the phone frame. A striped wire is taped to each electrode to make a conductive connection on one end, and they are connected to the MCU (microcontroller unit) pins on the other end. Near the MCU, between the input and each of the outputs, we have connected a 1M ohm resistor (for a total of four resistors).

The **MCU** used is an Arduino Nano board based on an ATMEL-MEGA328p chip. It uses five ports to connect to the five electrodes. The sampling rate of the sensor was set to 200Hz. The MCU executes a code that uses the open-source capacitive sensor library [3]. After performing signal processing, the MCU sends the data to a PC using a serial connection over USB. After each trial we restarted the system to eliminate changes in the test environment.

In a real world implementation the electrodes will be attached to the protective case on the inner side covered with an isolation layer of the same color as the protective case itself. The electronic circuit implementing the sensor with the simple signal processing and RF power source can be located on phone cover. That include wires, electronic components, MCU and a battery all can be thin enough to fit in a phone case. The electromagnetic noise that this circuit will emit is expected to be negligible compared to the electromagnetic emission of the phone and thus, the device itself will not interfere with the touch sensor. The device can exfiltrate the collected data in a concealed way using the methods proposed in [1].

The *attacker's device* analyzes the data collected from the four sensing electrodes and converts it to X and Y coordinates on the two-dimensional screen surface. This data can then be further analyzed to extract keyboard presses, lock screen PINs, or patterns.

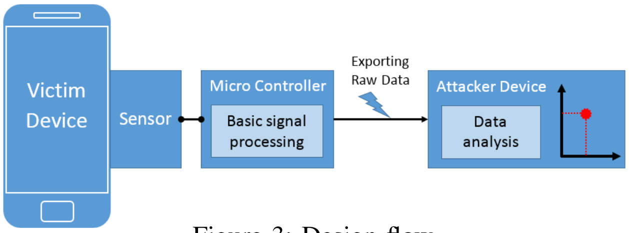


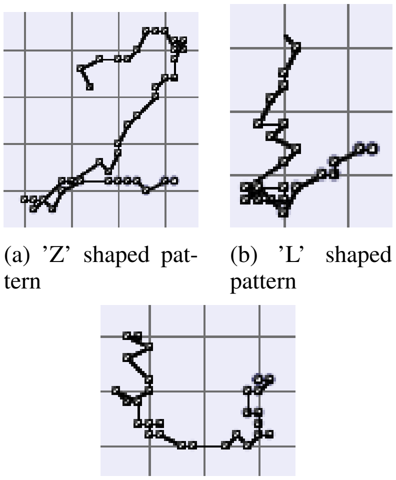
Figure 3: Design flow

3.2. Evaluation Results

During initial experiments we noticed an important, phenomenon. As the user's finger gets closer to the screen, the capacitance reading increases as expected. However, at the moment the finger touched the screen itself, a significant change in the reading is registered. This important finding allows us to differentiate between a finger hovering over the screen and an actual touch event. We suspect that this phenomenon is caused by the conductive layer located just below the glass of the screen, the same layer that is used for the native touchscreen capabilities of the phone. We note again that this measurement was obtained by sensors that surround the phone, but do not touch the user's finger or the screen's surface.

The three examples illustrated in Figure 4 show that the patterns drawn on the phone's screen can be reconstructed from measurements obtained by the external touch sensor.

We evaluated the implementation by drawing on the screen and capturing five different patterns (the letters L, O, Z, U and S); each pattern was drawn 30 times. We applied a simple pattern recognition algorithm[1] that calculates the path distance of a sample (i.e., captured pattern) and patterns provided by a preconfigured dictionary. As presented in Table 1 the naive algorithm is able to identify the letters with an average accuracy of 75%. Thus, it should be possible to reconstruct coarse touch events such as the phone's secret unlock pattern. Additional processing would be required before we are able to reconstruct fine-grained touch events such as keyboard typing.



(a) 'Z' shaped pattern

(b) 'L' shaped pattern

(c) 'U' shaped pattern

Figure 4: Sensor outputs for 3 patterns drawn on the screen

Letter	Success Rate
L	76.6%
O	73.3%
Z	63.6%
U	80.0%
S	93.3%

Table 1: Success rate for five patterns.

Environmental effects had an impact on our readings. When the user's hand was placed under the phone, our ability to obtain readings decreased. Although the changes in capacitance due to touch events were still noticeable in the raw sensor outputs, our test setup was unable to reliably convert the measurements into X-Y coordinates under these conditions. In a full attack setup, the attacker would probably have to use additional technics to eliminate the Parasitic capacitive coupling (e.g., adding a ground mesh), and apply some more signal processing to eliminate the effects of this activity on the final measurements. The readings were also affected when we connected a charger to the phone. In that particular case, the bottom electrode's sensitivity dropped, probably because the phone was connected to the charger's ground plane. However, all of the other electrodes continued obtaining standard readings, still making the attack possible.

[1] https://mccormick.cx/news/entries/gesture-recognition

4. Discussion

We demonstrate a technology that can be embedded in a simple looking phone protective case and can be used to detect the finger pressing position on a screen. This attack vector is different from other methods, as it requires minimal involvement of the attacker in the physical environment of the victim, and none in the software environment. This attack is cheap and easy enough to mass-produce. It is possible to implement the attack on a large scale by inserting the touch logging device in the supply chain of phone protective cases.

Currently, users or organizations don't think of a smartphone protective case as an item that poses a security risk, and most efforts to secure devices focus on protecting communication and software. Simple protective cases are routinely given as gifts and sold at stores and on the Internet. Most, if not all, users would not hesitate to put on a protective case, even if the source of the case was unknown or considered untrustworthy.

5. Countermeasures

In order to be successful, the attack vector relies on the ability to get the user to put the malicious smartphone protective case on his/her device. In order to achieve this feat the attacker participate in an official event and give the protective cases away as a souvenir or prize. Someone who purchases a protective case from a store faces the same risk, in the case of supply chain insertion.

A good way to eliminate this type of threat is to use a transparent protective case. Essential components of the attack mechanism cannot be hidden in a transparent case. In contrast, there are some phone protective case materials and features that are more suited to the attack mechanism; therefore, a case made of a stiff, thick, or opaque material is more likely to be used in this type of attack. A case with an external battery is an excellent candidate as they also have power source that can power the system.

In order to eliminate any suspicions about a given phone protective case, it is possible to use an x-ray scanner or metal detector or some other mechanism that can identify system components. Obviously, such tests are not relevant for a case with an external battery which contains metal and electric components.

It is also possible to produce phones that are less susceptible to this kind of attack. A non-conductive stylus such as those used with resistive touch screens can be effective against this attack. Phones screens that don't use the type of touch screen technologies that have a conductive layer near the screen (e.g., IR and SAW) will not result in an immediate change at the sensor reading when a finger touches the screen. The lack of the touch indication capability makes it more difficult to distinguish between a hovering finger and a touching finger. Although possibly helpful in the face of the discussed attack, it is unlikely that manufacturers will abandon this type of touchscreen.

Finally, the most effective countermeasure is user awareness; the user must be suspicious about any phone accessory coming from unknown or Untrustworthy sources.

6. Related Works

Previous work has shown that touchscreen events can be gathered using malicious software running on the smartphone. This is the case even if the attacker cannot directly access the phone's touchscreen API, and only has access to a subset of its other sensors [4]–[6]. The use of anti-virus software, strengthening the operating system, and simply exercising user caution can serve as effective defensive measures against such software-based attacks. In 2007, Sekiguchi demonstrated an approach for detecting touch screen events based on analysis of the electromagnetic noise emitted by the touchscreen [7]. This method requires expensive equipment, and is difficult for the attacker to carry out in an adversarial situation. In 2015, Ali et al. showed how the formation and direction of a user's hands can be detected by their effect on Wi-Fi signals [8]. Their attack assumed that the victim is positioned between two cooperating Wi-Fi transceivers, an assumption which is less practical in the context of a hardware implant. Low-tech measures such as shoulder surfing [9] can also be used to obtain touch screen data.

7. Conclusions and Future Work

We investigated the use of a malicious smartphone protective case as a side-channel to infer patterns drawn on a smartphone touchscreen. We observed that a case with dedicated sensors can determine the finger location on the touchscreen. We developed a proof-of-concept that uses the sensor readings and processing to infer the finger movement pattern on the screen. We have demonstrated that smartphone protective case must be considered a security threat as it may serve as a side channel from which confidential information can be leaked from a smartphone. In future work we intend to deeper investigate the sensor design. We will evaluate end-to-end attack that is able to detect soft keyboard events and leak the events to an external server and propose capacitive touch screen design that may be used to detect rogue capacitive sensors.

References

[1] A. G. B. Farshteindiker, N. Hasidim and Y. Oren, "How to phone home with someone else's phone: Information exfiltration using intentional sound noise on gyroscopic sensors," *USENIX Association*, 2016.

[2] J. Appelbaum, A. Gibson, C. Guarnieri, A. Müller-Maguhn, L. Poitras, M. Rosenbach, L. Ryge, H. Schmundt, and M. Sontheimer, "The digital arms race: NSA preps america for future battle," *Der Spiegel*, vol. 1, no. 17, Jan 2015.

[3] PaulStoffregen, "Capacitivesensor," web, 2014, github.com/PaulStoffregen/CapacitiveSensor.

[4] L. Cai, S. Machiraju, and H. Chen, "Defending against sensor-sniffing attacks on mobile phones," in *Proceedings of the 1st ACM SIGCOMM Workshop on Networking, Systems, and Applications for Mobile Handhelds, MobiHeld 2009*, 2009.

[5] L. Cai and H. Chen, "Touchlogger: Inferring keystrokes on touch screen from smartphone motion," in *6th USENIX Workshop on Hot Topics in Security, HotSec'11, San Francisco, CA, USA, August 9, 2011.*

[6] R. Schlegel, K. Zhang, X. Zhou, M. Intwala, A. Kapadia, and X. Wang, “Soundcomber: A stealthy and context-aware sound trojan for smartphones,” in *Proceedings of the Network and Distributed System Security Symposium (NDSS)*, 2011.

[7] H. Sekiguchi, “Novel information leakage threat for input operations on touch screen monitors caused by electromagnetic noise and its countermeasure method,” *Progress In Electromagnetics Research B*, vol. 36, no. 36, pp. 399–419, 2012.

[8] K. Ali, A. X. Liu, W. Wang, and M. Shahzad, “Keystroke recognition using wifi signals,” in *Proceedings of the 21st Annual International Conference on Mobile Computing and Networking, MobiCom 2015*, 2015.

[9] A. H. Lashkari, S. Farmand, O. B. Zakaria, and R. Saleh, “Shoulder surfing attack in graphical password authentication,” *CoRR*, vol. abs/0912.0951, 2009.