POSTER: TOUCHFLOOD: A Novel Class of Attacks against Capacitive Touchscreens

Seita Maruyama
Waseda University, Japan
maruyama@nsl.cs.waseda.ac.jp,

Satohiro Wakabayashi
Waseda University, Japan
wakabayashi@goto.info.waseda.ac.jp

Tatsuya Mori
Waseda University, Japan
mori@nsl.cs.waseda.ac.jp

ABSTRACT

We present a novel class of attacks against capacitive touchscreens, which are common in devices such as smartphones and tablet computers. The attack we named TOUCHFLOOD aims to scatter touch events, alternating the selection of buttons on a screen. The key idea of TOUCHFLOOD is to intentionally cause a malfunction by injecting intentional noise signals from an external source. This paper describes the attack as well as the experimental results that clarify the conditions for successful attacks. The demo videos of the experiments using a smartphone are available at https://goo.gl/56G79e.

CCS CONCEPTS

• **Security and privacy → Hardware attacks and countermeasures**; **Mobile platform security**;

KEYWORDS

Touchscreen, Attack, Smartphone

1 INTRODUCTION

The majority of the current mobile devices, such as smartphones and tablets, are equipped with touchscreens. While there are various technologies for sensing touch, mutual capacitive sensors are widely used in smartphones as they have high resolution and multi-touch support [1]. This work introduces a novel class of attacks against capacitive touchscreens, named *TOUCHFLOOD*. The attack aims to alter the selection of a button on a screen; that is, when a victim thinks that she/he touches the cancel button, the attack can scatter the recognized touched position and make the operating system recognize another button, such as OK, as having been touched, which may lead to security threats, such as installing malware. There have been many studies on the side-channel attacks that target touchscreens (LCD's) [2, 4]. To the best of our knowledge, while these attacks passively steal data from the touchscreens, our TOUCHFLOOD is the first attack that *actively* irradiates signals toward touchscreens to cause targeted malfunctions. In the paper, we present the basic mechanism of TOUCHFLOOD and reveal the conditions that are needed to establish the attack.

2 DESCRIPTIONS OF THE TOUCHFLOOD ATTACK

A mutual capacitive touchscreen controller consists of the grid of the transmitter (TX) electrodes and receiver (RX) electrodes, which are mutually coupled. These TX/RX electrodes are used for sensing touch events. As the human body has a capacitance, it can act as a capacitor. When a finger approaches the screen's surface, it extracts an electric charge from the touchscreen through mutual capacitance. Thus, the touchscreen controller can detect touches by measuring the changes in electric current that flows into the RX electrodes; the current changes are caused by the changes in capacitance between the TX and RX electrodes. The pair of TX and RX electrodes for which the changes are detected is used to locate the area of touch.

It is known that a touchscreen controller in a smartphone can malfunction due to noise signals leaked from the smartphone's battery charger or screen [3]. touchscreen controller manufacturing companies have developed countermeasures against the electromagnetic interference (EMI) caused by noise signals, which are relatively weak. However, when a stronger noise signal is intentionally applied to a touchscreen controller, false touch events can be generated. As some hobbyists have reported [7], it is known that false touch events occur when a smartphone is brought close to a toy plasma ball, which is powered by an oscillator and a high voltage transformer circuit, producing a large alternating voltage, typically around 2–5 kV and around 30 kHz [6].

The key idea of TOUCHFLOOD is to cause an intentional malfunction by injecting intentional noise signals externally. We found that producing large alternating voltages at a specific frequency near a touchscreen can cause a malfunction through capacitive coupling with the RX electrodes. Injecting the intentional noise forces to change the current flow of the RX electrodes, and the touchscreen controller incorrectly recognizes the changes of current flow as the changes of capacitance, which will be detected as pseudo touch events.

Threat model As our prototype uses a thin copper sheet, it can be hidden inside a common object such as a table. By installing a maliciously programmed NFC tag under the tabletop, an attacker can take control of the particular UI components on a smartphone [5]. Given these setups, a victim will encounter an unexpected dialogue box that asks whether s/he permits the request, e.g., installing a malicious application. Although the victim aims to cancel the request, TOUCHFLOOD will alter the selection, letting the smartphone install the malware.

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).

CCS '17, October 30-November 3, 2017, Dallas, TX, USA
© 2017 Copyright held by the owner/author(s).
ACM ISBN 978-1-4503-4946-8/17/10.
https://doi.org/10.1145/3133956.3138829

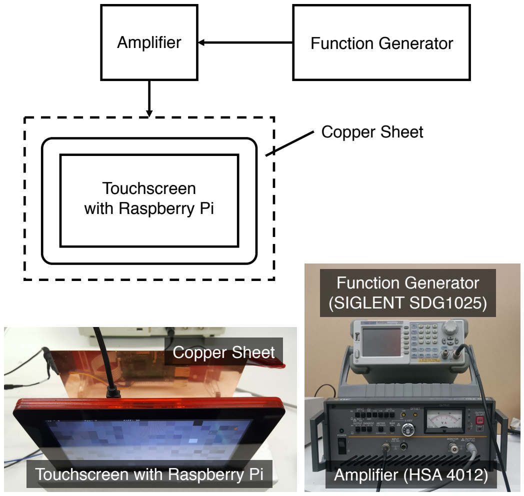
Amplifier
Function Generator
Copper Sheet
Touchscreen with Raspberry Pi
Function Generator (SIGLENT SDG1025)
Copper Sheet
Touchscreen with Raspberry Pi
Amplifier (HSA 4012)

Figure 1: Experimental setup.

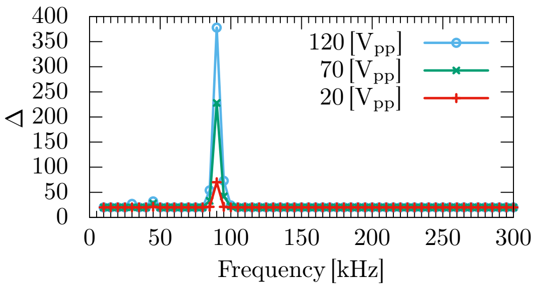

Figure 2: The effect of different frequencies on touchscreen.

3 EXPERIMENTS

To study the conditions that can cause the "false touches," we conduct several experiments using the touchscreen controller that provides raw data collected from the capacitive sensors.

3.1 Experimental setup

Figure 1 shows our experimental setup. Our objective is to measure the effect of noise signals on the behavior of touchscreens. For this experiment, we use the Raspberry Pi 7-inch Touchscreen Display. As an intentional noise signal, we use the sine-wave signal generated by a function generator. We set a copper sheet parallel to the touchscreen controller. This copper sheet is used to create a capacitive coupling with the capacitive sensors. The distance between the sheet and controller was set to 7 cm. We note that the attack can be applied from the rear side of a touchscreen controller, i.e., the rear side of a smartphone.

3.2 Effect of the frequencies and voltage values

We generate sine-wave noise signals with different frequencies and voltage values. We record raw capacitance values and touch events using the software we developed. Since the touchscreen has 264 capacitance sensors, which consists of a 12 × 22 matrix, we can obtain 264-dimensional time-series data. This setup enables us to analyze the spatial patterns of the generated touch events.

To measure the interference intensity on the touchscreen, we introduce a metric, Δ, defined as follows.

$$\begin{aligned} \delta_i &= x_i - \bar{x_i} \\ \Delta &= \max_i(\delta_i) - \min_i(\delta_i), \end{aligned}$$

where x_i ($i \in \{1, \dots, 264\}$) is a measured value for each sensor and $\bar{x_i}$ ($i \in \{1, \dots, 264\}$) is a measured value for each sensor when noise is not injected, respectively. We note that x_i is a variable of time; our capacitance logger sampled the raw values at the rate of 7 times per second. In contrast, $\bar{x_i}$ was set as a static value, which was collected when no signal was injected. If no noise signal is applied, Δ becomes roughly 20 when there are no touch events on the screen and Δ becomes greater than 250 when a finger touches the screen. Thus, the metric Δ can measure the impact of noise interference.

We measured Δ, applying noise signal to the copper sheet with three different voltages (20 Vpp, 70Vpp, and 120Vpp) and frequencies, ranging from 5 kHz to 300 kHz. Figure 2 shows the results. We first notice that there are clear peaks at the frequency of 90 kHz. This result indicates that there is a characteristic frequency of noise that can affect the touch controller. This frequency differs for different models of touchscreen controllers. So, specifying the model of the target is crucial to the success of the attack. To this end, a device fingerprinting technique can be used. We also notice that the effect of noise becomes larger with higher voltages in the signals. As we shall show in the next section, we need to apply a higher voltage to cause false touch events.

3.3 Spatial distribution of the false touch events

We now study the positions of the touch events caused by the noise signals. In this experiment, nothing touches the screen. Using our monitoring software, we record touch positions for 30 seconds with the sampling rate of two samples per second. The touchscreen has an 800×480 resolution and supports a 10-point multi-touch. The touchscreen controller is capable of reporting up to 10 positions per sample. Note that the touch events are collected from the outputs of the touchscreen controller, not from an operating system.

We used three different voltages (20 Vpp, 70 Vpp, and 120 Vpp) and the following two representative frequencies: 60 kHz as a frequency not affecting Δ and 90 kHz as a frequency affecting Δ the most. As expected, the touchscreen did not report any touch events with the 60 kHz frequency. In the following, we omit the results for the 60 kHz frequency. Figure 3 shows the results for the 90 kHz frequency. First, we notice that the touchscreen controller does not recognize touch events when the voltage is set to 20 Vpp. We also see that higher voltage signals cause false touch events more frequently. Second, we see intrinsic spatial patterns of touch events, which spread out on the screen linearly. We also see that many touch events are focused on the top or bottom edges of the screen panel. These observations indicate that even if an attacker waits for a long time, it seems unlikely that a false touch can be fired at

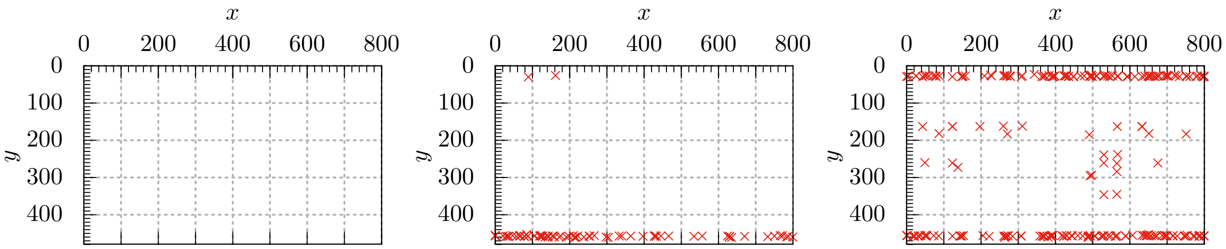

Figure 3: Coordinates of the touch points reported by the touchscreen controller. The injected signals had three different voltage values. The frequency was set to 90 kHz. Left: 20 Vpp, Center: 70 Vpp, Right: 120 Vpp

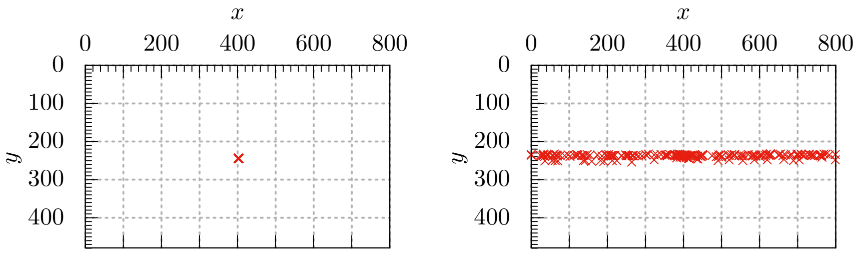

Figure 4: Coordinates of the touch points reported by the touchscreen controller. While the experiment a finger keeps touching the point centered on the screen. Left: no signal is applied. Right: a signal with 20 Vpp and 90 kHz is applied.

target coordinates with a high probability, given the skewed spatial distribution.

3.4 Limiting the dispersion with a real touch event

After several trials, we found that touching a screen can fix the skewed spatial distribution of false touches. Although not conclusive due to the "black box" nature of the touchscreen controllers, we conjecture that touching the screen with a finger stabilizes the area of capacitive coupling. The positive feature of this phenomenon is that while touching the screen makes the distribution focus on a certain area, it still keeps scattering the touch events; thus, false touch events can be created in a more predictable way.

We repeated the similar experiments, but added a finger touch this time. Figure 4 shows the experiment results. Under the low voltage signal of 20 Vpp, the false touch events occur only if a finger touches the screen. More importantly, we can see that the positions of the false touches are centered on the line where the true touch point is located. These are desirable characteristics because usually, GUI buttons are aligned in a row, e.g., CONNECT/CANCEL, YES/NO, or OK/CANCEL. Assuming that the touch events are uniformly scattered along a line, an attacker can expect that a touch event will be scattered on a wrong button with a probability of 1/2. We note that screen orientation also matters. If a screen is in portrait mode, scattered touch events along the vertical line may not produce a touch event on the targeted button. By making use of the device's fingerprinting techniques, an attacker can obtain the information about the model, as well as the current screen orientation; this information will be used to check whether or not TOUCHFLOOD attack is effective.

4 CONCLUSION

We introduced a novel class of attacks against capacitive touchscreens. Using an off-the-shelf touchscreen display, we presented the actual conditions needed for the successful attacks. Future works include an in-depth understanding of the mechanism of the attack, evaluation of the attack using smartphones and tablet computers, and the end-to-end attack combined with techniques that trigger a malicious touch event. We also need to develop effective countermeasures against the attack.

REFERENCES

[1] Li Du. 2016. An Overview of Mobile Capacitive Touch Technologies Trends. *arXiv preprint arXiv:1612.08227* (2016).

[2] Y. Hayashi, N. Homma, M. Miura, T. Aoki, and H. Sone. 2014. A Threat for Tablet PCs in Public Space: Remote Visualization of Screen Images Using EM Emanation. In *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security (CCS '14)*. 954–965.

[3] Hans W. Klein. 2013. Noise Immunity of Touchscreen Devices. http://www.cypress.com/file/120641/download. (2013).

[4] Federico Maggi, Simone Gasparini, and Giacomo Boracchi. 2011. A fast eavesdropping attack against touchscreens. In *Information Assurance and Security (IAS), 2011 7th International Conference on.* IEEE, 320–325.

[5] Seita Maruyama, Satohiro Wakabayashi, and Tatsuya Mori. 2017. Trojan of Things: Embedding Malicious NFC Tags into Common Objects. *CoRR* abs/1702.07124 (2017). http://arxiv.org/abs/1702.07124

[6] University of Oxford Department of Physics. 2012. Plasma ball. http://www2.physics.ox.ac.uk/accelerate/resources/demonstrations/plasma-ball. (2012).

[7] soomiq. 2014. Iphone goes Crazy Out of Control near Plasma ball. https://www.youtube.com/watch?v=bD_lv22T6Xo. (2014).