GhostTouch: Targeted Attacks on Touchscreens without Physical Touch

Kai Wang*
Zhejiang University

Richard Mitev*
Technical University of Darmstadt

Chen Yan
Zhejiang University

Xiaoyu Ji†
Zhejiang University

Ahmad-Reza Sadeghi
Technical University of Darmstadt

Wenyuan Xu
Zhejiang University

Abstract

Capacitive touchscreens have become the primary human-machine interface for personal devices such as smartphones and tablets. In this paper, we present GhostTouch, the first active contactless attack against capacitive touchscreens. GhostTouch uses electromagnetic interference (EMI) to inject fake touch points into a touchscreen without the need to physically touch it. By tuning the parameters of the electromagnetic signal and adjusting the antenna, we can inject two types of basic touch events, taps and swipes, into targeted locations of the touchscreen and control them to manipulate the underlying device. We successfully launch the GhostTouch attacks on nine smartphone models. We can inject targeted taps continuously with a standard deviation of as low as 14.6×19.2 pixels from the target area, a delay of less than $0.5s$ and a distance of up to $40mm$. We show the real-world impact of the GhostTouch attacks in a few proof-of-concept scenarios, including answering an eavesdropping phone call, pressing the button, swiping up to unlock, and entering a password. Finally, we discuss potential hardware and software countermeasures to mitigate the attack.

1 Introduction

Touchscreens allow users to interact with computers using their fingers and have become a trending alternative to mouses and keyboards. In particular, *capacitive* touchscreens provide multi-touch capabilities, long service life, and cost-effectiveness, and therefore, have been widely used on personal devices such as smartphones, tablets and watches, and even on safety-critical devices such as medical equipment [26] and spacecraft [13].

Reliable and accurate touch sensing is a critical requirement for touchscreens on all devices. However, the ability to measure small electric fields also makes capacitive touchscreens sensitive to environmental impacts such as electromagnetic interference (EMI) [10] and charger noise [23], which can induce fake touches that may greatly impair user experience and even cause unintended device behavior. For instance, there are numerous reports about unresponsive, self-tapping and malfunctioning touchscreens caused by EMI emitted through fluorescent lights [6] and faulty chargers [28, 41, 43].[1]

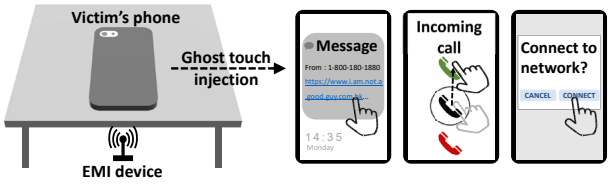


Figure 1: A GhostTouch attack scenario. The attacker uses an EMI device under a table to remotely attack the touchscreen of a smartphone face-down on the table. By injecting fake touches, the attacker can trick the smartphone to (1) click a malicious link containing malware, (2) connect a malicious network, and (3) answer an eavesdropping phone call.

At first glance, the impact of EMI on capacitive touchscreens seems to be largely unpredictable, and hence may not be exploited for targeted attack on the underlying device. Therefore, our motivation and focus in this paper is to address the research question of whether it is possible for an attacker to use EMI to inject controllable fake touches to a touchscreen without any physical contact and manipulate the underlying device in a predictable way. EMI attacks on devices have been studied before. However, as we elaborate on related work in Section 8, and to the best of our knowledge, there have been no published EMI attacks on capacitive touchscreens that can manipulate touch points on the touch screen without requiring any physical contact.

We propose GhostTouch, the first contactless EMI-attack against capacitive touchscreens. The core idea is to interfere with the capacitance measurement of touchscreens using electromagnetic signals, which are injected into the receiv-

*Kai and Richard are co-first authors.
†Xiaoyu Ji is the corresponding author.

[1] In one case a malfunctioning touchscreen even booked a thousand-dollar hotel room itself without the owner's awareness [33].

ing electrodes integrated into the touchscreen. As a result, an electromotive force is induced in the measuring circuit that affects the touch point detection.

To achieve our attack we had to overcome two technical challenges: 1) It is difficult to affect a touchscreen by EMI, since modern touchscreens and devices go through thorough electromagnetic compatibility (EMC) tests [31] and utilize anti-interference design such as shielding [35] and layout optimization [7] to avoid the influence of environmental interference. To address this challenge, we carefully design the transmitting antenna, signal frequency, and attack distance to improve the electromagnetic signal propagation gain, therefore, achieving an effective touch injection. 2) Even if we can inject touches, it is still difficult to create predictable touch events with the touchscreen specifics undisclosed and varying from device to device. We probe the screen to disclose the touchscreen specifics and adjust the parameters of the attack signal accordingly to inject predictable touch events, such as a tap, a swipe-up, or a swipe-down in targeted locations.

Overcomming these challenges, our attack can inject two types of basic touch events, *taps* and *swipes*, into a targeted location of the touchscreen without any physical contact. Most complex gestures can be achieved by the combination of these two basic interactions. By tuning the parameters of the electromagnetic signal and adjusting the antenna, we can control the location and pattern of the fake touches and achieve various touch behaviors including press and hold, swipe to select, slide to scroll, etc., depending on the device model.

We demonstrate the feasibility of this attack in the real world with the setup as shown in Figure 1. In places like a cafe, library, meeting room, or conference lobbies, people might place their smartphone face-down on the table[2]. An attacker may embed the attack equipment under the table and launch attacks remotely. For example, an attacker may impersonate the victim to answer a phone call which would eavesdrop the private conversation, or visit a malicious website.

Our main contributions are as follows:

- We propose `GhostTouch`, the first contactless attack against capacitive touchscreens that can inject taps and swipes into a targeted location of the touchscreen and control the fake touch behavior without any physical contact.
- We evaluate `GhostTouch` attacks successfully on touchscreens of 9 different smartphone models. Our attack can target any area on the touch screen. For example, on Nexus 5X we can inject taps continuously into an area as small as 36.3×175.8 pixels with an delay of less than 0.5 seconds and inject targeted swipes with a success rate of 62.5% and an average delay of 1.6 seconds.
- We demonstrate the real-world impact of `GhostTouch` with 4 practical attack scenarios: answering an eavesdropping phone call[3], pressing the button, swiping up to unlock, and entering a password.
- We suggest both hardware and software countermeasures to protect touchscreens from such attacks.

[2] A study [20] among 3246 participants shows that 54.37% of people would often or sometimes set their phones face-down on the table.

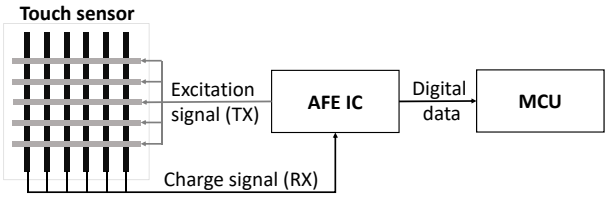


Figure 2: Typical system architecture of a capacitive touchscreen.

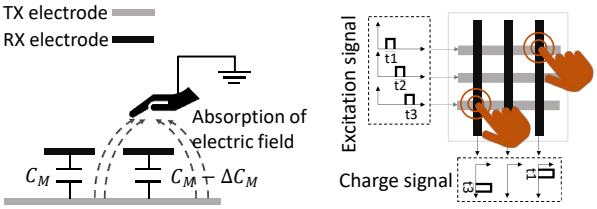
2 Background on Capacitive Touchscreens

A capacitive touchscreen is an input device normally layered on top of the display that detects human touches based on the capacitance variation. When a finger touches the screen, the capacitance at the touch point changes significantly as the charge stored in the screen gets drawn to the finger. By monitoring the capacitance variation at each point of the screen, a touchscreen can detect *touch points* and report *touch events*, e.g., tap, swipe, based on the timing and the locations of the detected touch points. For example, two consecutive touch points in aerial vicinity are recognized by the OS as a swipe, reported to the corresponding app as a swipe over two (touch) points. In the following, we introduce the most popular design of capacitive touchscreens used on smartphones, i.e., a system architecture supporting mutual capacitive sensing and scan driving method, which we consider in this paper.

System architecture: Figure 2 shows a typical system architecture of capacitive touchscreens [25], which includes a touch sensor, an analog front-end integrated circuit (AFE IC), and a micro controller unit (MCU). The touch sensor consists of a grid of transmitting (TX) and receiving (RX) electrodes made of transparent conductive materials, e.g., indium-tin-oxide (ITO). The AFE IC sends excitation signals to the TX electrodes and measures the charge signal from the RX electrodes. The charge signal is digitized and sent to the MCU, which processes the signal and detects touch events. This architecture is designed to support two efficient methods that enable multi-touch sensing, i.e., mutual capacitive sensing, which relates to how the capacitance variance at a single point is measured, and scan driving method, which is used in combination with mutual capacitive sensing to locate the touch points.

Mutual capacitive sensing: When an excitation signal is applied to a TX electrode, the electrode generates an electric

[3] Video demo: https://github.com/USSLab/GhostTouch



(a) Mutual capacitive sensing. (b) Scan driving method.

Figure 3: Illustration of mutual capacitive sensing and scan driving method.

(a) Electrostatic gun and antenna. (b) Generated pulse signal.

Figure 4: (a) The experiment setup of the feasibility study and (b) a pulse signal generated by the electrostatic gun. The pulse lasts for 186 ns.

field that creates a flow of electric charge to the air-gapped RX electrodes, which essentially forms a mutual capacitance C_M between the RX and TX electrodes at each intersection [27], as shown in Figure 3(a). When a finger touches the screen, it absorbs a part of the electric field and changes the mutual capacitance to $C_M - \Delta C_M$, where ΔC_M is caused by the charge drawn to the finger. In this case, the variance of capacitance changes the charge signal measured by the RX electrode, and a touch point at the intersection of RX and TX is detected. The excitation signal is normally a square wave with a frequency of 100 kHz to 500 kHz. Mutual capacitive sensing outperforms other methods such as self-capacitive as it can efficiently locate the touch point by exploiting TX-RX pairs.

Scan driving method: The scan driving method (SDM) [16, 17] is designed to locate the touch points on the screen by exciting all TX electrodes in turn, as shown in Figure 3(b). As only one TX is excited at a time, the touchscreen can locate a touch to a specific position by the row and column of the active TX-RX pair and can also support multi-touch detection. SDM involves several major parameters: the number of TX electrodes N, the time it takes to scan all TX electrodes T_p, and the time it takes to scan one TX electrode T_{p1}. Compared with other methods, the scan driving method has a simple structure, shorter sensing time, and lower signal-to-noise ratio (SNR) and has been adopted on most smartphones.

3 Feasibility of Injecting Touches with EMI

Capacitive touchscreens detect touches based on the charge signals on the RX electrodes and locate touches by scanning the TX electrodes. However, the RX and TX electrodes are essentially conductors that electromagnetic waves may couple on (i.e., convert to electrical signals) and interfere with the touch sensing. Therefore, we are motivated to explore the feasibility of injecting fake touch points into the touchscreens of various smartphones using electromagnetic interference (EMI) and analyze the distribution of the injected touch points.

Electromagnetic interference: Electromagnetic interference [21] appears when undesirable voltages or currents are present in the environment of a device. This can lead to malfunctioning or degradation of the performance. The voltages or currents may affect a device by conduction or radiation. In our case, we focus on electromagnetic interference by radiation, which can interfere with a device through electromagnetic coupling, a phenomenon that generates an electrical charge in the electrical wiring or circuits of a device. Maxwell's equation explains the principle of electromagnetic coupling: $\oint_{\partial\Sigma} \mathbf{E} \cdot \mathrm{d}\mathbf{l} = -\int_{\Sigma} \frac{\partial \mathbf{B}}{\partial t} \cdot \mathrm{d}\mathbf{A}$, where $\partial\Sigma$ is a closed contour, Σ is a surface bounded by $\partial\Sigma$, $\mathbf{E}$ is the electric field, $\mathbf{B}$ is the magnetic field, $\mathrm{d}\mathbf{l}$ is an infinitesimal vector of $\partial\Sigma$, and $\mathrm{d}\mathbf{A}$ is an infinitesimal vector of Σ. In a touchscreen, the measuring circuit can be considered as the closed contour $\partial\Sigma$. The changing magnetic field $\mathbf{B}$ of an electromagnetic radiation passing through the surface Σ of a touchscreen can induce an electromotive force as the left part of the equation, which may affect the capacitance measurement of touchscreens.

Experiment setup: The experiment setup is shown in Figure 4(a). We use an electrostatic gun [1] to generate a strong pulse signal, which is sent to an antenna we made using Dupont jumper wires and gets converted to strong electromagnetic interference. The electrostatic gun can generate short pulses with the waveform shown in Figure 4(b), and the pulse amplitude can vary from 1kV to 18kV. In this experiment, we set the amplitude to 10kV. We place a 5mm-thick acrylic board between the antenna and the phone's touchscreen, and inject EMI with the antenna at two types of positions: parallel to the vertical or horizontal edges of the phone. We experiment on 12 phone models and show the results in Table 1.

Results: Although the capacitive touchscreens of smartphones go through thorough electromagnetic compatibility tests and anti-interference design, 8 of the 12 tested smartphones are susceptible to EMI. We record and analyze the distribution of the injected points on the 8 phones. We observe two types of results regarding the density and distribution of the injected points. Among the 8 susceptible phones, two can only be injected with sparse fake points while six can be injected with dense fake points, indicating a greater susceptibility and a higher attack success rate. The injected points distribute along a horizontal line of the touchscreen on 3 phones and a vertical line on other three other phones, validating the possibility to inject fake points along both the

Table 1: Results of injecting fake touch points on 12 phones.

Phone model	Success	Injection Speed		Point Distribution	
		Sparse	Dense	Horizontal	Vertical
Nexus 5X	✓	×	✓	×	✓
Google Pixel 1	✓	×	✓	×	✓
OPPO K3	×	N/A	N/A	N/A	N/A
OPPO Reno	×	N/A	N/A	N/A	N/A
OPPO Reno 2	✓	×	✓	✓	×
OPPO Reno 3	×	N/A	N/A	N/A	N/A
OPPO Reno 3 Pro	✓	×	✓	✓	×
One Plus 8T	✓	✓	×	×	×
Huawei P10 Plus	✓	×	✓	✓	×
Huawei P40	✓	✓	×	×	×
Samsung S20 FE	✓	×	✓	×	✓
iPhone 7 Plus	×	N/A	N/A	N/A	N/A

horizontal and vertical direction of the touchscreen.

Observations of the touch point distribution: We show the distribution of the injected touch points with Google Pixel 1, Nexus 5X and Huawei P10 Plus as an example. The touch point data is recorded using the Android Debug Bridge (ADB). Figure 19 in Appendix shows a visual distribution of the injected points on the three phones, and Figure 20 in Appendix shows the cumulative distribution function (CDF) plots of the injected points along the horizontal (X-axis) and vertical (Y-axis) directions. The results show that more than half of the injected points are distributed nearly uniformly on a specific line of the touchscreen where the antenna is placed, either vertical (Google Pixel 1 and Nexus 5X) or horizontal (Huawei P10 Plus). The direction of fake point distribution is the same as the direction of the antenna. We believe the difference in distribution is due to the different touchscreen layouts, especially the RX electrodes. For example, the RX electrodes of Nexus 5X are vertical while the RX electrodes of Huawei P10 Plus are horizontal, which all correspond to the distribution of fake points on these phones.

Observations of the capacitance variation: We further explore the reason behind fake touch points based on the raw capacitance data of the touchscreen. Using ADB, we are able to record the capacitance data on the Huawei P10 Plus before and during the electromagnetic interference. We calculate the difference of the capacitance data to acquire the variation caused by the EMI and plot the result in Figure 5. The capacitance of a line in the middle of the screen decreases dramatically, which corresponds to the distribution of injected points on this phone in Figure 19(c) in Appendix.

Our feasibility study confirms that the touchscreens of various phone models are susceptible to EMI. Therefore, it is feasible for an attacker to inject fake touch points to the touchscreen of the victim's smartphone without any physical contact. However, as the next step, we will study methods to transform random electromagnetic interference to elaborate electromagnetic attacks that can inject controllable touch events and manipulate the smartphone in real-life scenarios.

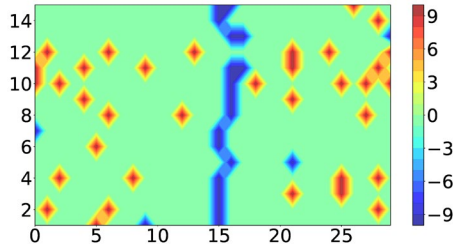


Figure 5: A visual illustration of the capacitance variations caused by EMI on the touchscreen of a Huawei P10 Plus. The plot corresponds to the screen in the landscape orientation. The capacitance variations in the middle (valuing between -3 to -9) conform to the results in Figure 19(c) in Appendix.

4 Threat Model

The attacker's goal is to manipulate the victim device by injecting fake touches to the touchscreen in a contactless manner. We make the following assumptions for the attack:

- **Victim device:** The victim device is equipped with a capacitive touchscreen, such as a smartphone or tablet. The device is unaltered before the attack and placed *face-down* on a surface (e.g., a table) during the attack.
- **Attacker's knowledge:** The adversary may know the victim's device model and acquire a device of the same model to study beforehand. Further, the adversary may acquire the victim's phone number via social engineering.
- **Attacker's capability:** The attacker can only attack the device by manipulating the touchscreen via electromagnetic signals. However, the adversary cannot physically touch the victim device or ask the user to perform any tasks.
- **Attack setup:** The attacker may hide the attack equipment under the table where the victim devices might be attached to a surface (e.g., under a table in a meeting room, or charging station). The attacker can control the attack equipment remotely.

5 Attack Design

In this section, we present `GhostTouch`, the first contactless attack against capacitive touchscreens of smartphones. Our goal is to inject controllable touch events, such as taps and swipes, into a targeted area of the touchscreen and use them to manipulate the device. To achieve this goal, we need to tackle the following technical challenges:

1) Effectively inject fake touch points: Although our study confirms that EMI injection is feasible, the fake points are injected using unpredictable signals after trial and error. To

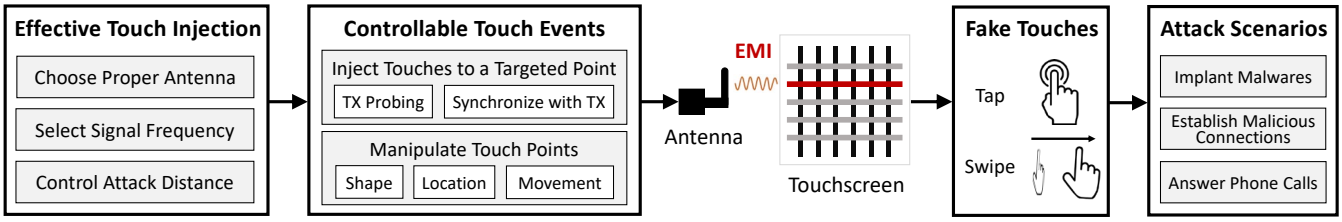


Figure 6: An illustration of the GhostTouch design. We first prepare the EMI signal for effective touch point injection and then design the signal for controllable touch events. We emit the crafted signal with an antenna to make the touchscreen mistakenly sense two types of basic touch events, i.e., taps and swipes, which can be used to construct more complex touch behaviors for various attack scenarios.

achieve a powerful attack, we need to understand the interference process and design electromagnetic signals for effective and efficient injection.

2) Create controllable touch events: The fake touch points in previous work and our feasibility study can only distribute randomly on the screen. To achieve controllable touch events such as taps and swipes, we need to constrict the fake touch points into a target area of the screen and adjust their positions as desired.

To address the first challenge, we study three main factors that affect the effectiveness and efficiency of touch point injection, i.e., the transmitting antenna, signal frequency, and attack distance. Our goal is to find the best options for these factors that in combination can achieve the maximum intensity of touch point injection in a reproducible and cost-effective way. To address the second challenge, we study methods to make the randomly distributed touch points repeatedly appear in a constricted area of the touchscreen. By adjusting the timing of the transmission, the transmitting period, and antenna positions, we can control the position of the injected touch points.

The design of GhostTouch is shown in Figure 6. In the first stage, we prepare the EMI signal for effective touch point injection by choosing a proper antenna, selecting signal frequency, and controlling the attack distance. In the second stage, we design the EMI signal for controllable touch events by probing the touchscreen, synchronization, and adjusting key signal parameters. After these stages, the crafted EMI signal is emitted by the antenna to attack the touchscreen of a smartphone. GhostTouch can induce two types of basic touch events, taps and swipes, which can be used to construct more complex touch behaviors for various attack scenarios. We introduce the details in the following.

5.1 Effective Touch Point Injection

In the first stage, we study the main factors that affect the effectiveness and efficiency of touch point injection, including the type and length of the transmitting antenna, the frequency of the EMI signal, and the distance between the transmitting antenna and touchscreen. We seek to find the optimal combination of these factors to increase the possibility of touch point injection. Though the transmitting power plays an important role for EMI, it is generally considered that the higher the power the stronger the interference, therefore we do not discuss the transmitting power in the attack design.

In our attack, the electromagnetic interference needs to satisfy two requirements: (1) the intensity of the induced electromotive force needs to be high enough to influence the touch sensing, and (2) the electromagnetic interference needs to affect only a part of the touchscreen so that the injected touch points can appear in a restricted area instead of all over the screen. To meet these requirements, we elaborate on our considerations in the transmitting antenna, signal frequency, and attack distance.

5.1.1 Transmitting Antenna

An electromagnetic field can be generated and received by an antenna. In our attack, the electrodes in a touchscreen essentially act as antennas that unintentionally pick up the electromagnetic interference. The RX electrodes are especially vulnerable because the induced electrical signals can directly affect the touch sensing. To maximize the efficiency of electromagnetic coupling, the antenna we use to emit the EMI needs to match the equivalent antennas in the touchscreen, including both the antenna type and length. There are many types of antenna, mainly including electric dipole (e.g., Hertzian antenna) antenna and magnetic dipole (e.g., small loop antenna) antenna [22]. The electrodes of a touchscreen can be regarded as electric dipole antennas, and the circuit formed by a TX electrode, an RX electrode and the AFE IC can be regarded as a magnetic dipole antenna. Thus, we can use both types of antennas to transmit the EMI. To make the antennas' length match on a similar magnitude, we measure the size of the touchscreen and make a rough estimation.

For example, the size of a Nexus 5X is 147×72.6 mm. We have verified that both our self-made electric dipole antenna using a Dupont jumper wire of 140mm and a 4mm-diameter tip antenna (with a total coil length of around 70mm) are effective in our attack as they match the equivalent antennas in the touchscreen.

5.1.2 Signal Frequency

The frequency of an electromagnetic signal determines the efficiency of it being transmitted or received by a given antenna. We can estimate the effective signal frequency based on the electrical length [36] of the targeted/selected antenna. Electrical length is defined as the ratio of the physical length of the antenna to the wavelength of the electromagnetic signal. Empirically, an antenna whose electrical length is less than 1/20 or 1/50 can be considered as electrically short, which means that it can barely emit EM signals of the desired wavelength (frequency). Higher antenna gain can generally be achieved with larger electrical length, e.g., when the ratio between the antenna's physical length and the signal's wavelength is 1/2 or 1/4. In our attack, assume the physical length of the antenna is l, then the signal frequency corresponding to a 1/50 electrical length is $c/50l$, where c is the speed of light. To have the signal being effectively transmitted and received, the signal frequency needs to be higher than $c/50l$ to ensure an electrical length higher than 1/50.

For example, consider the Nexus 5X we discussed earlier, to make the electromagnetic signal couple into the 147mm RX electrode, the signal's frequency needs to exceed 40.8MHz. To verify the estimated frequency, we conduct a frequency sweeping experiment starting from 4MHz using a Rigol DG5072 arbitrary waveform generator and the 140mm self-made antenna. The end frequency is 70MHz because it is the maximum frequency supported by the equipment. We set the signal amplitude to 20Vpp in the experiment. The results show that we can inject a significant amount of touches into the touchscreen of a Nexus 5X at a signal frequency of 46MHz, which conforms to our estimation. Here we need to note that we may be able to find other effective frequencies if we can scan above 70MHz with appropriate equipment.

5.1.3 Attack Distance

The energy of an EM signal is inversely proportional to the square of the transmitted distance [44]. In addition, the distance affects the spatial distribution of the electromagnetic field. There are mainly two types of electromagnetic fields, i.e., near field and far field, which are different in the energy distribution [36]. Near field stores the energy of the signal source and is mainly distributed near the source. Empirically, we can consider an electromagnetic field as a near field when the distance to the source is smaller than $2D^2/\lambda$, where D is the size of the antenna and λ is the wavelength. Since our attack requires the EM interference to affect a part of the screen and possess a high intensity, we keep the attack distance within the near field and as short as possible.

For example, with a 140mm antenna and a 46MHz signal frequency, a distance within 6mm to the antenna can be considered as the near field. We explore the impact of attack distance on a Google Pixel 1. The results in Figure 21 in Appendix show that as the attack distance increases from 5mm to 10mm, the injected touch points get less intense and less concentrated.

In this stage, we prepare the attack signal for effective touch point injection by studying the optimal combination of the transmitting antenna, signal frequency, and attack distance. Although we try to constrict the injected touch points into a restricted area by controlling the attack distance, the best effect we can achieve in this stage is to inject fake touches randomly along a targeted line as shown in Figure 21 in Appendix, which corresponds to the location of one or several neighboring RX electrodes close to the antenna in the touchscreen. This is because our EM signal is coupled into the RX electrodes when varying TX electrodes are driven. Note that the RX electrodes on different smartphones have different orientations, so a specific smartphone allows either vertical or horizontal excitations. In the next stage, we will study how to inject fake touch points into a smaller area, e.g., around a targeted point, and explore methods to create controllable touch events.

5.2 Creating Controllable Touch Events

Creating controllable touch events such as tapping on a specific button or swiping in a specific direction requires us to achieve a higher level of control over the injected touch points. Specifically, we need to inject touches to *a targeted point* on the screen instead of along a targeted line and be able to manipulate the injected touch points, such as their shape, location, and movement.

5.2.1 Injecting Touches to a Targeted Point

The core idea of injecting touches to a targeted point on the screen is to synchronize our interference signal with the TX scanning of touchscreen. Before elaborating on this, we probe the TX electrodes to understand how the excitation signals are sent.

Touchscreen probing: Not all smartphone manufacturer are publishing their devices touch sampling rate. Moreover, some devices dynamically change the sampling rate depending on the app displayed or other parameters (e.g., last touch point time). For these it is possible to find the sampling rate and the currently sensed position of the screen by probing the screen, as described in Appendix A.

Synchronization with TX scanning: We illustrate our idea in Figure 7. The excitation signal is sketched according to the probing results. We generate a short interfering signal with a duration T_d equal to or less than the time to scan one TX electrode T_{p1}. Suppose that the signal is coupled into an RX electrode at the time when the 2nd TX electrode is driven, then a fake touch point will appear at the intersection of the interfered RX electrode and the 2nd TX electrode. The touch point will not appear at other locations because there is no interference when other TX electrodes are driven. By setting

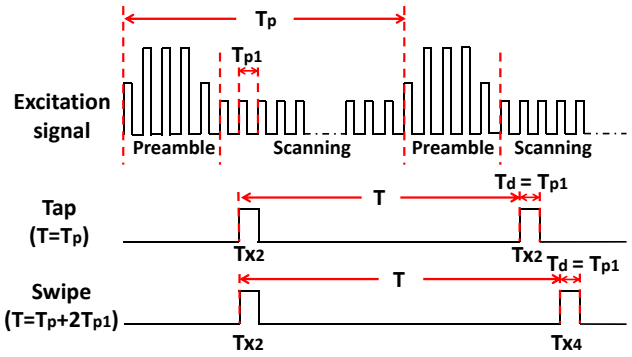


Figure 7: An illustration of synchronizing the interfering signal with the excitation signal of the touchscreen to achieve controllable taps and swipes.

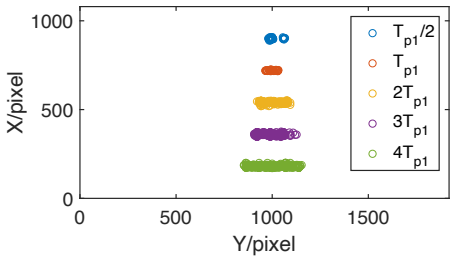


Figure 8: A visual distribution of the injected touch points on a Nexus 5X using five different transmitting durations. The touch points become more disperse as the duration increases.

the period of the interfering signal T equal to the scanning period T_p, i.e., synchronizing with the targeted TX electrode, we can make fake touch points repeatedly appear at the same point, which can be detected by the touchscreen as a single tap.

5.2.2 Manipulation of Touch Points

We can manipulate the touch points' shape, location, and movement by adjusting several waveform parameters such as the transmitting time, duration T_d, and period T.

Shape: We can control the shape by adjusting the duration T_d of the interfering signal, because as the duration increases, more TX electrodes are driven when the RX electrode is interfered, therefore making the touch points appear in a larger area. However, there is a trade-off when adjusting the duration. When the duration is too short, the intensity of the interference may be too small to affect the touchscreen. We demonstrate the ability to manipulate the shape with an experiment on Nexus 5X, where we set the duration T_d to $0.5T_{p1}$, T_{p1}, $2T_{p1}$, $3T_{p1}$, and $4T_{p1}$. Figure 8 shows the touch points' visual distribution on the screen. As we increase the duration, the injected touch points get more disperse in shape. By default, in `GhostTouch` we set the duration T_d to T_{p1}.

Location: The location of fake touch points can be modelled by the intersected RX-TX electrodes. We can inject fake touch points to any location on the screen by adjusting the timing of interference and the antenna's position. To interfere when a targeted TX electrode is driven, an attacker can either adjust the timing based on the feedback of existing locations or by prediction based on real-time touchscreen probing. To interfere with a targeted RX electrodes, the attacker can move the antenna near the RX or use an antenna array.

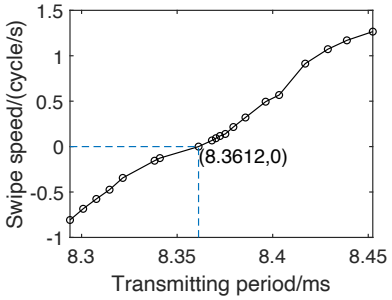
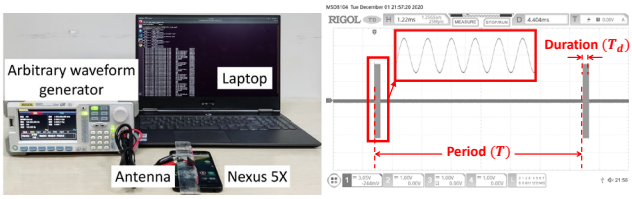


Figure 9: The speed of touch point movement changes while we adjust the transmitting period. It demonstrates the ability to move fake touches in arbitrary directions and with arbitrary speeds.

Movement: In some cases the attacker needs to move the injected touch points, e.g., to adjust the touch location or create swipes. We can easily achieve touch movement along both directions of the RX electrodes by setting the transmitting period T higher or lower than the scanning period T_p. The amount of deviation from T_p determines the speed of movement. To demonstrate the ability to move the injected points, we experiment on a Nexus 5X with varying transmitting period. We record the touch data using ADB and calculate the speed of movement. The results in Figure 9 show that we can move the fake touch points in arbitrary direction and speed.

5.2.3 Controllable Touch Events

With the above methods to generate and fine-tune the EMI signal, we are able to create controllable touch events. Specifically, in `GhostTouch` we focus on two types of basic touch events, i.e., taps and swipes. We validate injecting controllable tap, swipe-up, and swipe-down on a Nexus 5X using the experiment setup and interfering signal shown in Figure 10. The interfering signal is generated using a Rigol DG5072 arbitrary waveform generator [9] and a self-made antenna. For each type of touch event, we try 20 times and each trial lasts for 3 seconds. We are able to inject taps consistently into any area targeted on the touch screen even an area as small as 180×180 pixels in the middle of the screen with a success rate of 85%. Moreover, we can inject swipe-up and swipe-down into any part of a targeted line with success rates of both 90%. An attacker can use the injected taps and swipes to construct more complex touch behaviors for various attack scenarios, which we will show in the following.



(a) Experiment setup. (b) Interfering signal.

Figure 10: The experiment setup and the interfering signal used to validate GhostTouch attack on a Nexus 5X.

5.3 Attack Scenarios

We demonstrate the threat of GhostTouch in three practical attack scenarios that can be implemented by injecting taps and swipes.

(1) *Implant a malware.* Suppose the adversary knows the victim's phone number or messaging app account, and sends a message to the victim containing a malicious link. When the victim's phone displays a notification upon receiving the message, the adversary can use GhostTouch to tap the notification. After automatically opening the message app, the attacker then taps the malicious link to initiate a drive-by download of a malware.

(2) *Establish a malicious connection.* To establish a malicious connection, e.g., WiFi or Bluetooth, the adversary sends a request to the victim's phone or uses an NFC tag to trigger the connection, which will make the phone display a notification. The adversary can tap the notification to open the connection request window and then tap the "CONNECT" button to approve the request. After establishing the connection, the attacker can perform Man-in-the-Middle attacks or control the phone with a Bluetooth mouse.

(3) *Answer an eavesdropping phone call.* Suppose the adversary knows the victim's phone number, and calls the number and inject a swipe to answer the call on the victim's device. This enables the adversary to eavesdrop on the victim user. This attack will not raise the victim's attention when the phone is switched to silent mode, which many users would do when they are sleeping, or are at work [4] or a conference. The adversary may also prevent the phone from ringing by answering the call before the first ring.

6 GhostTouch System Evaluation

In this section, we discuss the implementation of the GhostTouch system and its evaluation.

6.1 Implementation

The GhostTouch system consists of two parts, a *touch injector* and a *phone locator*, as shown in Figure 11. The touch injector can inject touch events, e.g., a tap, a swipe, or multi-touch, into the touchscreen, and it includes a signal generator, an amplifier, an on/off switch, and a receiving antenna array. The on/off switch is used to select the correct antennas to emit the EMI signals such that it can inject touch events into the targeted RX lines. The phone locator can identify the position of the touchscreen. It consists of a sensing antenna array, a data acquisition device, and a location calculator.

Antenna Array. The antenna array is consisting of the transmitting and sensing antennas. As mentioned in Section 4, it will be placed in the appropriate location to facilitate signal emitting and sensing and attack the target device. Note that the transmitting antennas are dipole antennas because they are designed to inject touch points along the RX lines, and the sensing antennas are coil antennas because they are designed to receive signals radiating from the touchscreen.

Touch Injector. In our experiments, we implemented two types of touch injectors with different capabilities: a powerful full-fledged injector and a smaller portable one. As shown in Figure 12, the powerful injector utilizes an arbitrary waveform generator (Rigol DG5072), an amplifier (Mini-Circuits ZHL-100W-GAN+ [50]) and an on/off switch Time Relay.

The portable injector consists of a signal oscilloscope and a ChipSHOUTER [5] that integrates the amplifier and an antenna. The signal oscilloscope generates a square wave to drive the ChipSHOUTER to emit pulses of a broadband signal that covers the frequency bands proven to be effective in interfering with touchscreens. Due to the hardware limits of the ChipSHOUTER, the pulse width is limited to 80ns and 960ns, and an example pulse output of ChipSHOUTER is shown in Figure 22 in Appendix. Since the powerful injector has the flexibility of emitting different EMI signals, we utilize the powerful version for feasibility evaluation. The ChipSHOUTER injector is used to validate the attacks in real-world scenarios due to its small size and the fact that it represents a lower bound for our experiments.

Phone Locator. As mentioned above, the antenna array is part of the phone locator, and can be used in combination with both touch injectors. The sensing antennas use a data acquisition device of NI MyDAQ [32]. This device allows for measuring and analysing the radiated signals of the touchscreen and inferring the phone position relative to the sensing antenna. Note that we can reuse the hardware of the phone locator to assist attack synchronization, as described in Appendix A.

In the rest of this section, we will evaluate the single touch injection, multi-touch injection, touch injection scenarios in real world, and phone locator.

6.2 Single Touch Injection

We evaluate the performance of single touch injection, including two basic touch events, tap and swipe.

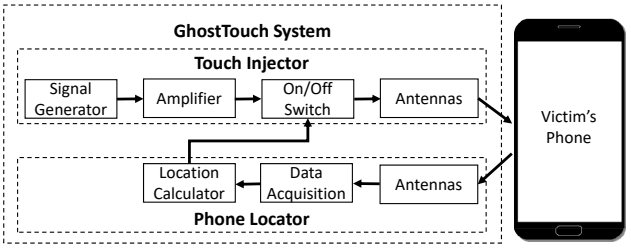
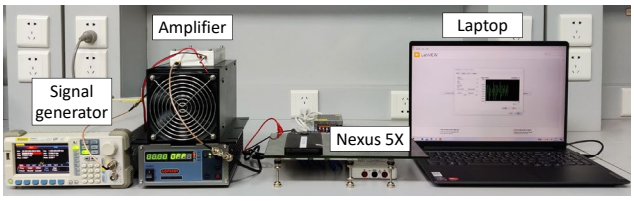
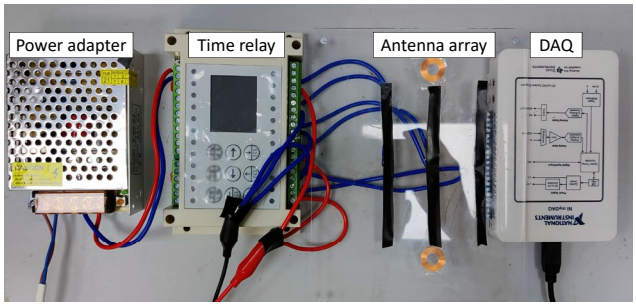


Figure 11: An illustration of the GhostTouch system. We build a touch injector to inject touch events into the touchscreen, and a phone locator to locate the victim's phone.



(a) System setup.



(b) Antenna array.

Figure 12: The high-power experiment setup of the GhostTouch system including the antenna array.

6.2.1 Experiment Methodology

Experiment setup: In this part, we use the powerful setup, as shown in Figure 12, and set the transmitting duration to be $3T_{p1}$, the transmitting period to be the same as the scanning period T_p (120Hz), and the signal frequency of the EMI to be 46MHz. The default distance between the antenna array and the screen is 5mm. The targeted smartphones (Nexus 5X by default) are connected to a laptop for touch injection recording.

Data collection: To quantify the attack results, we need to obtain the touch event data, e.g., the timestamps and locations of the injected points. We can acquire the data either from an Android application or by using Android Debug Bridge (ADB), a command-line tool that can communicate with Android devices.

Metrics: We evaluate the performance of GhostTouch from two perspectives: the similarity to real human touches and the attack capability. The injected touch points and real human touch points may vary mainly in two aspects, the shape and concentration of touch points, which can be quantified by the following metrics:

1. *Range*, which is the difference between the maximum and minimum of the injected points' X/Y coordinates. It describes the shape of the injected points.
2. *Standard deviation of the X/Y coordinates*, which describes the concentration of the injected points.

Metrics to evaluate the attack capability includes the injection speed, attack delay and success rate:

1. *Injection speed*, which is the number of injected points in unit time.
2. *Attack delay*, which is the time it takes to inject the first successful touch event since the attack starts.
3. *Success rate*, which is the proportion of attacks that successfully inject the targeted touch event.

6.2.2 Tap

We use the powerful experiment setup, to evaluate the performance of injecting targeted taps. We set the transmitting period to T_p and repeat the experiments 100 times with each time lasting for 3 seconds. Since the T_p is drifting over time, we measure it constantly and adjust the transmitting period. After recording the data of the injected points by ADB, we calculate the range of x and y coordinates, the standard deviation of the coordinates, and the injection speed. We show the samples of 'taps' in Figure 23(a) in Appendix.

Similarity between injected points and real touch events: The metrics of the injected points are shown in Figure 13. The mean of $range_x$ and $range_y$ are 36.3 pixels and 175.8 pixels, respectively, and the mean of std_x and std_y are 5.4 pixels and 44.0 pixels, respectively.

To compare our GhostTouch with real touch events, we recruited 30 volunteers of three professions (students, professors, administrative staff), including 8 females and 22 males aged between 20 and 50 to tap the 'Home' button on the Nexus 5X, using the thumb and forefinger 30 times each. Then we randomly select 1800 samples from the volunteers' taps and the taps injected by the attacker, respectively. The comparison between the GhostTouch and the real touch events are shown in Figure 14. Compared with human taps, the injected taps are distributed in a smaller range on the x-axis, and distributed in a larger range on the y-axis. However, the difference between the injected taps and user's taps is very small and hence not distinguishable from the source.

Attack capability: (1) Injection speed: According to Figure 13, the mean of the injecting speed is 45.38 point/s. Considering the maximum human touch speed is about 7 points/s,

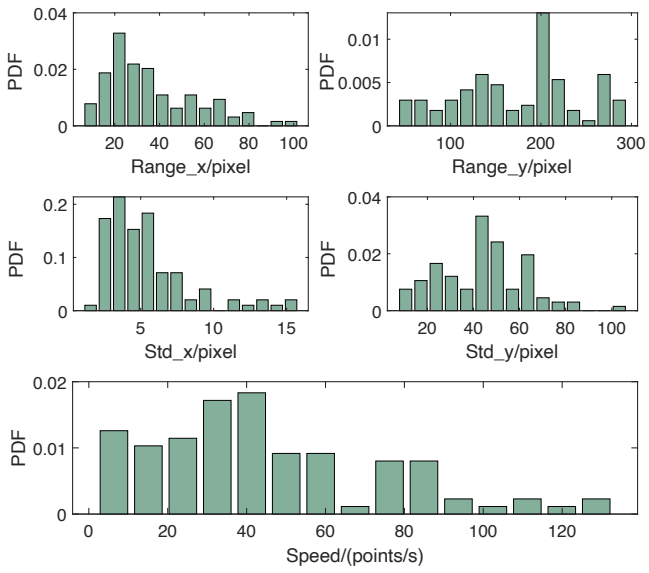


Figure 13: The performance of injecting taps in 100 trials, including the range of the x and y coordinates (top), the standard deviation of the coordinates (middle), and the injection speed (bottom).

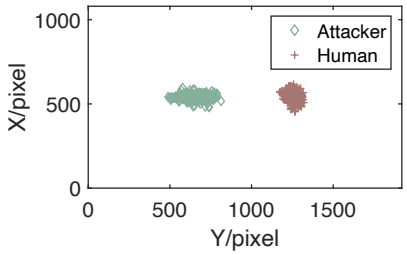


Figure 14: A comparison of the similarity between the taps from 30 volunteers and the ones injected by the attacker. Compared with real touch events, the taps injected by the attacker are distributed in a smaller range on the x-axis, and distributed in a larger range on the y-axis.

this injection speed can satisfy the attacker's requirements. (2) Consistency means how long the injected points will stay in one position, which is important because an adversary may need to tap the same point repeatedly for a few seconds. For example, we inject 'taps' into Nexus 5X's touchscreen for 15 seconds and the injected points stay within a small area for 15 seconds. Detailed results are reported in Appendix (Figure 24). (3) The attack delay is less than 0.5 seconds.

6.2.3 Swipe

We use the experiment setup as shown in Figure 12(a) for evaluation. By slightly changing the transmitting period T around T_p, we can 'swipe up' or 'swipe down'. For a sample

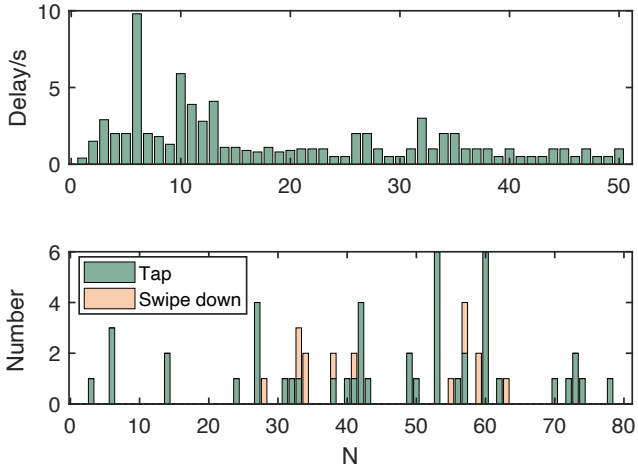


Figure 15: The performance of injecting a swipe-up in 80 trials, including the time delay until a successful swipe-up is injected (top) and the number of unintended touch events, e.g., taps or swipe-downs (bottom).

of injected swipes refer to Figure 23(b) in Appendix. Similar to the results shown in Figure 9, the direction and speed of the swipe are related to the difference between T and T_p.

We evaluate the attack delay and success rate of injecting swipes by setting the transmitting period T to $7.7ms$ and repeatedly trying to swipe up. It is counted as a success if there are no taps or swipes down until the first swipe up on Nexus 5X as our showcase. (1) To calculate the success rate, we measure the number of false events before the first successful swipe up, e.g, taps and swipes-down. The number of false events are 30 out of 80 times, which leads to the success rate of 62.5% (50/80). (2) Attack delay: It takes 1.6 seconds on average to inject a successful swipe up. Detailed results are shown in Figure 15.

6.3 Multi-touch Injection

Multi-touch gestures have become a popular input operation, they can either be multiple simultaneous touches or multiple swipes at different locations of the screen. The most robust way of realizing multi-touch injections is to inject touches or swipes into different RX lines at the same time. To inject multiple touches at the touchscreen, we utilize an antenna array, as shown in Figure 12(b). The on/off switch will choose multiple antennas over the right locations to emit the EMI signal, with each chosen antenna being coupled with the corresponding RX lines. Thus, the attacker can successfully inject taps along the targeted RX lines simultaneously and achieve multi-touch injection. We validate the feasibility of injecting three touch points using three antennas, and the performance results of multi-touch injection is similar to the attack capabilities and properties using one antenna (c.f. Section 6.2).

6.4 Touch Injecting Scenarios

Experiment Setup: To demonstrate the threat of injecting taps and swipes, we illustrate three attack scenarios conducted with the ChipSHOUTER touch injector setup and one scenario using the powerful setup with an antenna array. We use a ChipSHOUTER device to generate pulses by charging to $500V$ and discharging its capacitors. Although the shape of the pulses is fixed, we can adjust how often to emit a pulse (i.e., pulse periods) by adjusting the frequency of the square wave that drives the ChipSHOUTER. In our experiments, we set the square wave signal with 20% duty cycle when the transmitting period equals the scanning period T_p (120Hz), and pulse width is set to 350ns.

Answering the phone call: We inject swiping actions in the middle line on the Nexus 5X to answer the phone call. The tip of the ChipSHOUTER was positioned 5mm over the middle of the screen. When the phone is called, we transmit the EMI signal with a transmitting period of 130Hz to swipe up. As a result, we answer the phone call successfully in all the 10 tests and it takes about 4.1 seconds on average, with a max of 6 seconds and a min of 2 seconds.

Pressing the button: We inject a tap into a certain button on the screen to press this button. We implemented an Android app. It displays a button oriented on the middle of the X-axis and with $77dp$ distance to the right side of the screen, where normally "OK" or "Accept" buttons would be displayed. The button is sized at $36dp$ height and $80dp$ width. The app collects all taps not on the button and stops when the button is pressed for the first time. We evaluated this attack on the Nexus 5X using the ChipSHOUTER, and the tip was hovering 5mm over the bottom of the screen. With an injection frequency of 120Hz, it took 7.5 seconds for the injected touch points to press the button. 11.3 taps were injected wrongly until the next tap hit the button.

Swiping up to unlock: We inject swiping actions in the middle line on the Nexus 5X. For the Nexus 5X device, its lock screen has a "swipe up to unlock" mechanism. The tip of the ChipSHOUTER was positioned 5mm over the middle of the screen. After a proper pulse frequency for injecting swipes was found (130Hz), injecting a swipe to unlock 20 times took 8.5 seconds on average with a minimum time of 1s and a maximum of 20s.

Entering a password: Once the attacker acquires the information of the password either by shoulder surfing or social engineering, she can utilize the touch injector to unlock the phone. We first select the correct antennas that are close to the target areas and adjust the timing to emit EMI signals such that we can 'press' the desired numbers in the virtual keyboard. As test cases we picked two PIN codes 3699 and 9999 to test. We were able to enter the PIN codes successfully in about 20s and 1s, respectively. It is possible to enter any PIN or password with `GhostTouch`, yet complex passwords will require extra time to accomplish successful injection.

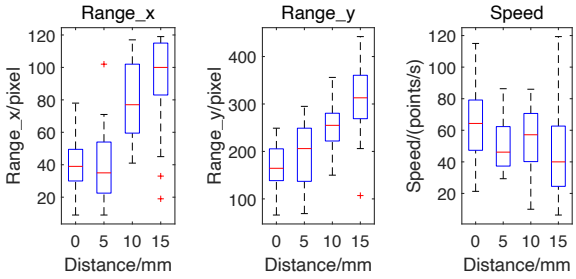


Figure 16: An illustration of the impact of the attack distance. The injection speed tends to decline as the distance increases.

6.5 Factors Affecting Touch Injection

We evaluate the factors that may affect the performance of touch injector, including attack distance, the phone model, ChipSHOUTER coil buzzing, and wireless charging.

Attack distance. We evaluate the impact of the distance between the transmitting antenna and the touchscreen using the setup shown in Figure 12(a), which can output a high-intensity EMI. The devices are attacked 40 times at a range of 0 to 15mm lasting 3 seconds each. Subsequently, the range of the injected points' x/y coordinate and the injecting speed is calculated. The results are presented using a box plot from Matlab, as shown in Figure 16. The central mark of each box indicates the median, and the bottom and top edges respectively indicate the 25th and 75th percentiles. The outliers are plotted individually using the '+' symbol, and the whiskers extend to the most extreme data points except for outliers.

According to the results, the injecting speed tends to decline as the distance increases, which is according to the attenuation of the EMI. We were capable to achieve a distance of up to 40mm.

Phone models. We evaluate the `GhostTouch` attacks on 11 phone models, using the portable setup with the ChipSHOUTER. We set the attack distance to 6mm for all phones for comparison. Broadly distributed touch points can be injected into 9 of these smartphones. We therefore explore whether we can realize the GhostTouch attack on these phones, and once successful we evaluate the performance by injecting taps for 4 seconds each. We calculate the injection speed and the standard deviation of the injected points' x/y coordinates. We record the direction of the swipe for each phone. According to the results, as shown in Table 2, we can inject touch points at chosen positions for 6 out of 11 phone models with GhostTouch attacks and therefore they are vulnerable to the attack scenarios described in Section 6.4. Since the injecting speed for Huawei Honor View 10 is low, we extend the experiment duration of injecting taps to 40 seconds and calculate the average result. For Galaxy S20 FE 5G and iPhone SE (2020), our approach can inject touch points successfully and perform malicious operations, but not always with high precision. Such a vulnerability is still dangerous,

Table 2: Attack performance on 11 phone models using the ChipSHOUTER. Nine phones are vulnerable to the attack, on six we can inject touch points precisely.

Phone model	Success	Inje. Speed	Direction	Std/pixel		Std/mm	
				X	Y	X	Y
Galaxy A10s	✓	3.5	Vertical	158.9	111.9	14.9	10.5
Huawei P30 Lite	✓	2.0	Vertical	182.0	189.0	11.1	11.6
Honor View 10	✓	0.3	Vertical	41.4	4.9	1.3	0.6
Huawei Mate 40 Pro	×	N/A	N/A	N/A	N/A	N/A	N/A
Galaxy S20 FE 5G	(✓)	2.8	Vertical	N/A	N/A	N/A	N/A
iPhone 12	×	N/A	N/A	N/A	N/A	N/A	N/A
iPhone SE (2020)	(✓)	1.0	Vertical	N/A	N/A	N/A	N/A
Nexus 5X	✓	8.2	Vertical	14.6	19.2	0.9	1.1
Redmi Note 9S	✓	2.5	Horizontal	73.3	210.2	4.7	13.5
Nokia 7.2	✓	8.7	Vertical	36.5	156.3	2.3	9.9
Redmi 8	(✓)	1.5	Horizontal	N/A	N/A	N/A	N/A

to illustrate, we managed to establish a malicious Bluetooth connection on iPhone SE (2020), with an average delay of 7.1 seconds.

ChipSHOUTER coil buzzing. The ChipSHOUTER while generating a wide range signal and due to its small form factor is emitting a high-pitched audible coil buzzing noise. We measure this buzzing noise using the Benetech GM1357 [12]. It is 44dB right next to the ChipSHOUTER, 42dB 20cm above the table under which the ChipSHOUTER is placed, and 38dB 30cm above the table. Note that the buzz of a refrigerator is about 40dB. While this noise is audible when in close proximity in a silent room, it is too faint to hear in a crowded place (e.g., conference hall, cafe).

Wireless charging. Our attack can be successfully launched in the same manner while the device is being charged. The device's touchscreen is still fully functional during wireless charging, as the smartphone components shield the touchscreen from the magnetic field. Further, wireless chargers usually operate at $130-175kHz$, insufficient to couple to an RX electrode.

6.6 Phone Locator

To inject touch points precisely, the attacker needs to know how the victim's phone is placed. We implement the phone locator as shown in Figure 12(b). It consists of sensing antennas and a NI MyDAQ. In practice we can place a matrix of sensing antennas to locate the phone positions with the following observation: a sensing antenna over the touchscreen can detect the radiated signals (e.g., the leaked signals of the TX excitation signal of the touchscreen at a frequency of 120Hz), while the ones away from the touchscreen will detect a weaker signal or none at all due to attenuation. Thus, the sensing antennas are used to receive the leaked signals, which are processed by the NI MyDAQ to deduce the phone location and orientation, with respect to the sensing antennas. After obtaining the phone position, we can infer the positions of the buttons based on the operating system and target application, e.g., Bluetooth connection accept dialog.

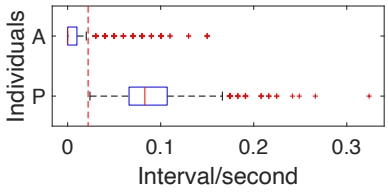


Figure 17: Touch interval of 30 volunteers (P) and a `GhostTouch` attacker (A). The touch events from a human and an attacker can be separated by setting a threshold, represented by the dotted line.

7 Potential Countermeasures

Our results showed that certain smartphones are less vulnerable to the `GhostTouch` attack, which could be due to better electromagnetic shielding or effective validation. Inspired by this, we propose three categories of countermeasures to mitigate the threat of `GhostTouch` attack:

Reinforcement: Manufacturers may reinforce the touchscreen to protect it from the threat of `GhostTouch` attack. First, adding electromagnetic shielding is effective to block EMI. Second, increasing the voltage of the excitation signal may increase the SNR, which will mitigate the influence of EMI. Third, a driving method based on a more sophisticated excitation signal waveform may be used to filter out injected current by EMI. Although these countermeasures could reduce the impact of EMI on the touch screen, they require modifications to the hardware of the touchscreen or lead to higher energy consumption. Thus, these countermeasures are not suitable for existing touchscreens.

Detection algorithm: The manufacturer can improve the detection algorithm of the touchscreen. For example, the `GhostTouch` attack can be detected utilizing the touch interval between pressing and lifting the finger. To explore the effectiveness of this method, we calculate the touch interval of a `GhostTouch` attacker and 30 volunteers as in Section 6.2.2. We randomly choose 2000 samples from the attacker and volunteers each and analyze the distribution of the touch interval using a box plot from Matlab. As shown in Figure 17, the upper adjacent of the attacker is lower than the lower adjacent of the volunteers. We can set a threshold, e.g., the red line in Figure 17, to identify whether the touch point belongs to a user or an attacker.

Moreover, the capacitance distribution shows a certain pattern when the touchscreen is under our `GhostTouch` attack opposed to being used by a human. Based on these facts, the manufacturer could detect abnormal touch points, reject them and warn the user.

Identity verification: Application permissions may be re-

stricted and identity verification needs to be conducted when executing high-risk actions. For example, conduct identity verification before connecting to a Bluetooth device or an unknown WiFi. Identity verification can be realized by requesting the user to verify his fingerprint, face or provide his PIN or password.

8 Related Work

In the following, we provide a summary of the existing attacks on touchscreens as well as attacks utilizing electromagnetic interference (EMI).

EMI attack: There have been several studies on EMI attacks over the last years. EMI attacks are used for Denial of Service (DoS), injection of false information into sensing circuits, or to glitch computations. Sabath et al. [37] launched a jamming attack using high-power EMI, which causes degradation or loss of the main function of critical electronic systems. Hayashi et al. [14] showed that electrical devices with cryptographic modules are vulnerable to electromagnetic interference. Schmidt et al. [38] showed that it is possible to induce faults into cryptographic systems using EMI and therefore breaking RSA. Dehbaoui et al. [8] showed that it is possible to inject faults into hardware and software implementations of AES using EMI. O'Flynn et al. [34] used EMI to force a hardware keystore to leak sensitive data by precisely manipulating packets sent over the USB stack. Kune et al. [24] investigated the susceptibility of analog sensors to EMI attacks and implemented the EMI attack against implantable cardiac electronic devices and consumer electronic devices containing microphones. Selvaraj et al. [39] presented an EMI attack which can cause bit flips or inject false actuation signals in embedded systems. Giechaskiel et al. [11] demonstrated the threat of EMI attack by injecting malicious commands into a smartphone. There have been other signal injection attacks on sensors [19, 42, 45, 48, 49] and defense mechanisms [46, 47]. Our attack `GhostTouch` takes another approach and utilizes the observation that EMI can induce current flow into a sensing circuit. We focus on capacitive touchscreen and their sensing mechanism to inject wrongly recognized touches into the touch panel. Capacitive touchscreens have more complex structures and mechanisms. Furthermore, capacitive touchscreens of smartphones have been tested for electromagnetic compatibility, and therefore, it is significantly more challenging to launch a fake touch injection attack on them.

Attacks on touchscreen: Research on the security of touchscreens can be divided into two categories, passive eavesdropping and active spoofing attacks. In passive eavesdropping, the adversary infers the input of a touchscreen using side-channel information. In prior work, Maggi et al. [29] took the image from surveillance as side channel information to get the keystrokes of a victim. Aviv et al. [3] leveraged smudge to get the graphical password. Hayashi et al. [15] leveraged the electromagnetic signal emanations of a tablet display to reconstruct the displayed screen image.

Active spoofing attacks may modify software or hardware like the work proposed by Schwartz et al. [40], modifying the touch display driver to inject false touch points. Attacks like the one described by Maruyama et al. [30] use EMI to attack the touchscreen controller. However, the attack needs the victim to touch the panel while the attack is active and thus could be easily perceived by the victim. Our attack does not require the victim to touch the panel. Moreover, they could not control the position of the injected touch points. Approximately, the injected points are uniformly scattered along a line which is the RX electrode touched by a finger.

In contrast our attack `GhostTouch` takes a Dupont jumper wire or a 4mm tip as the antenna. By changing the position of the antenna, the adversary could control which line the touch points are injected into. By shaping the EMI signal, the adversary could control which segment of this line the points are injected into.

9 Conclusion

In this paper, we introduced a novel attack coined `GhostTouch`, which targets the capacitive touchscreen used on many mobile devices such as smartphones or tablets. `GhostTouch` controls and shapes the near-field electromagnetic signal, and injects touch events into the targeted area on the touchscreen, without the need for physical touch or access to the victim's device. Consequently, the adversary can stealthily manipulate the victim's smartphone. Through the extensive experiments and evaluation, we demonstrate that our `GhostTouch` attack works for most widely-used smartphones. Moreover, we discuss possible countermeasures against the `GhostTouch` attack.

Acknowledgments

We thank the anonymous reviewers for their valuable comments. This work is supported by China NSFC Grant 61925109, 61941120, and 62071428, as well as, the German Research Foundation (DFG) within CRC 1119 CROSSING (S2 and P3), the Intel Private AI Center and Huawei Open Lab for Sustainable Security and Safety (openS3Lab).

References

[1] 3ctest EDS 20H. http://www.3ctest.cn/product/show/178.

[2] Adey64. Nexus 5 has a 120hz touch controller. https://forum.xda-developers.com/t/nexus-5-has-a-120hz-touch-controller.2559505/, 2013.

[3] Adam J Aviv, Katherine L Gibson, Evan Mossop, Matt Blaze, and Jonathan M Smith. Smudge attacks on smartphone touch screens. *Woot*, 10:1–7, 2010.

[4] Yung-Ju Chang and John C Tang. Investigating mobile users' ringer mode usage and attentiveness and responsiveness to communication. In *Proceedings of the 17th International Conference on Human-Computer Interaction with Mobile Devices and Services*, pages 6–15, 2015.

[5] ChipSHOUTER. https://www.newae.com/chipshouter.

[6] Dankgus. Touchscreen interference issue, confirmed–weird! https://www.surfaceforums.net/threads/touchscreen-interference-issue-confirmed-weird.14810/, 2015.

[7] Chris Daskalou. How conducted emi influences touch sensors. https://fieldscale.com/blog/emi-noise-touch-sensors/, 2020.

[8] Amine Dehbaoui, Jean-Max Dutertre, Bruno Robisson, and Assia Tria. Electromagnetic transient faults injection on a hardware and a software implementations of AES. In *Proceedings of 2012 Workshop on Fault Diagnosis and Tolerance in Cryptography*, pages 7–15, 2012.

[9] Rigol DG5072. https://www.batronix.com/shop/waveformgenerator/Rigol-DG5072.html.

[10] Embedded. Understanding electromagnetic interference sources in touchscreens. https://www.embedded.com/understanding-electromagnetic-interference-sources-in-touchscreens/, 2011.

[11] Ilias Giechaskiel, Youqian Zhang, and Kasper B Rasmussen. A framework for evaluating security in the presence of signal injection attacks. In *Proceedings of European Symposium on Research in Computer Security*, pages 512–532, 2019.

[12] Benetech GM1357. https://www.i-tech.com.au/benetech-gm1357-digital-sound-level-meter-gm-1357-55795.html.

[13] Chelsea Gohd. The touchscreen controls of spacex's crew dragon give astronauts a sci-fi way to fly in space. https://www.space.com/spacex-crew-dragon-touchscreen-astronaut-thoughts.html, 2020.

[14] Yu-ichi Hayashi, Naofumi Homma, Takeshi Sugawara, Takaaki Mizuki, Takafumi Aoki, and Hideaki Sone. Non-invasive EMI-based fault injection attack against cryptographic modules. In *Proceedings of 2011 IEEE International Symposium on Electromagnetic Compatibility*, pages 763–767, 2011.

[15] Yuichi Hayashi, Naofumi Homma, Mamoru Miura, Takafumi Aoki, and Hideaki Sone. 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*, pages 954–965, 2014.

[16] Steve Hotelling, Joshua A Strickon, and Brian Q Huppi. Multipoint touchscreen, February 16 2010. US Patent 7,663,607.

[17] Steven P Hotelling, Christoph H Krah, and Brian Quentin Huppi. Multipoint touch surface controller, October 2 2012. US Patent 8,279,180.

[18] IDC. Smartphone market share. https://www.idc.com/promo/smartphone-market-share/vendor, 2021.

[19] Xiaoyu Ji, Yushi Cheng, Yuepeng Zhang, Kai Wang, Chen Yan, Wenyuan Xu, and Kevin Fu. Poltergeist: Acoustic adversarial machine learning against cameras and computer vision. In *Proceedings of 2021 IEEE Symposium on Security and Privacy (SP)*, pages 160–175, 2021.

[20] Paul K. Do you ever set your phone face-down on the table? https://www.phonearena.com/news/Poll-Do-you-ever-set-your-phone-face-down-on-the-table_id88055, 2016.

[21] Bernhard E Keiser. *Principles of electromagnetic compatibility*. Artech House, 1987.

[22] Abdul Qadir Khan, Muhammad Riaz, and Anas Bilal. Various types of antenna with respect to their applications: a review. *International Journal of Multidisciplinary sciences and Engineering*, 7(3):1–8, 2016.

[23] Hans W Klein. Noise immunity of touchscreen devices. *Cypress Semiconductor Corporation, White Paper*, 2013.

[24] Denis Foo Kune, John Backes, Shane S Clark, Daniel Kramer, Matthew Reynolds, Kevin Fu, Yongdae Kim, and Wenyuan Xu. Ghost talk: Mitigating emi signal injection attacks against analog sensors. In *Proceedings of 2013 IEEE Symposium on Security and Privacy*, pages 145–159, 2013.

[25] O. Kwon, J. An, and S. Hong. Capacitive touch systems with styli for touch sensors: A review. *IEEE Sensors Journal*, 18(12):4832–4846, 2018.

[26] FOCUS LCDs. Touch screens for use in medical instrument displays. https://focuslcds.com/journals/touch-screens-for-use-in-medical-instrument-displays/, 2019.

[27] Jeffrey Lee, Matthew T Cole, Jackson Chi Sun Lai, and Arokia Nathan. An analysis of electrode patterns in capacitive touch screen panels. *Journal of display technology*, 10(5):362–366, 2014.

[28] Leelauer. Why does my touch screen go crazy while charging? https://forums.androidcentral.com/google-nexus-7-tablet-2012/497397-why-does-my-touch-screen-go-crazy-while-charging.html, 2019.

[29] Federico Maggi, Alberto Volpatto, Simone Gasparini, Giacomo Boracchi, and Stefano Zanero. A fast eavesdropping attack against touchscreens. In *Proceedings of 2011 7th International Conference on Information Assurance and Security (IAS)*, pages 320–325, 2011.

[30] Seita Maruyama, Satohiro Wakabayashi, and Tatsuya Mori. Tap'n Ghost: A Compilation of Novel Attack Techniques against Smartphone Touchscreens. In *Proceedings of 2019 IEEE Symposium on Security and Privacy*, pages 620–637, 2019.

[31] Manisha Mathur, JK Rai, and N Sridhar. Electromagnetic compatibility analysis of projected capacitive touch technology based panel computer for military application. *Journal of ElEctromagnEtic WavEs and applications*, 30(13):1689–1701, 2016.

[32] NI MyDAQ. https://www.ni.com/pdf/manuals/373060g.pdf.

[33] NBD. The cell phone being charged automatically booked a ten thousand yuan presidential suite and checked the chat history. http://www.nbd.com.cn/articles/2018-10-08/1260630.html, 2018.

[34] Colin O'Flynn. MIN () imum Failure: EMFI Attacks against USB Stacks. In *Proceedings of 13th USENIX Workshop on Offensive Technologies*, 2019.

[35] Electronic Products. Shielding touchscreens from emi. https://www.electronicproducts.com/shielding-touchscreens-from-emi/#, 2008.

[36] Jean-Michel Redouté and Michiel Steyaert. *EMC of analog integrated circuits*. Springer Science & Business Media, 2009.

[37] Frank Sabath. What can be learned from documented Intentional Electromagnetic Interference (IEMI) attacks? In *Proceedings of 2011 URSI General Assembly and Scientific Symposium*, pages 1–4, 2011.

[38] Jörn-Marc Schmidt and Michael Hutter. *Optical and EM Fault-Attacks on CRT-based RSA: Concrete Results*. Verlag der Technischen Universität Graz, 2007.

[39] Jayaprakash Selvaraj, Gökçen Yılmaz Dayanıklı, Neelam Prabhu Gaunkar, David Ware, Ryan M Gerdes, and Mani Mina. Electromagnetic induction attacks against embedded systems. In *Proceedings of the 2018 on Asia Conference on Computer and Communications Security*, pages 499–510, 2018.

[40] Omer Shwartz, Amir Cohen, Asaf Shabtai, and Yossi Oren. Shattered trust: When replacement smartphone components attack. In *Proceedings of 11th USENIX Workshop on Offensive Technologies (WOOT 17)*, 2017.

[41] Slane35. Touchscreen problems while charging. https://forum.xda-developers.com/showthread.php?t=1784773, 2012.

[42] Timothy Trippel, Ofir Weisse, Wenyuan Xu, Peter Honeyman, and Kevin Fu. Walnut: Waging doubt on the integrity of mems accelerometers with acoustic injection attacks. In *Proceedings of 2017 IEEE European Symposium on Security and Privacy (EuroS P)*, pages 3–18, 2017.

[43] User1950278. Glitchy touchscreen caused by charger [closed]. https://electronics.stackexchange.com/questions/77631/glitchy-touchscreen-caused-by-charger, 2013.

[44] Martin Weik. *Computer science and communications dictionary*. Springer Science & Business Media, 2000.

[45] Wenyuan Xu, Chen Yan, Weibin Jia, Xiaoyu Ji, and Jianhao Liu. Analyzing and enhancing the security of ultrasonic sensors for autonomous vehicles. *IEEE Internet of Things Journal*, 5(6):5015–5029, 2018.

[46] Zhijian Xu, Guoming Zhang, Xiaoyu Ji, and Wenyuan Xu. Evaluation and defense of light commands attacks against voice controllable systems in smart cars. *Noise & Vibration Worldwide*, 52(4-5):113–123, 2021.

[47] Chen Yan, Hocheol Shin, Connor Bolton, Wenyuan Xu, Yongdae Kim, and Kevin Fu. Sok: A minimalist approach to formalizing analog sensor security. In *Proceedings of 2020 IEEE Symposium on Security and Privacy (SP)*, pages 233–248, 2020.

[48] Chen Yan, Guoming Zhang, Xiaoyu Ji, Tianchen Zhang, Taimin Zhang, and Wenyuan Xu. The feasibility of injecting inaudible voice commands to voice assistants. *IEEE Transactions on Dependable and Secure Computing*, 18(3):1108–1124, 2021.

[49] Guoming Zhang, Chen Yan, Xiaoyu Ji, Tianchen Zhang, Taimin Zhang, and Wenyuan Xu. Dolphinattack: Inaudible voice commands. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, pages 103–117, 2017.

[50] Mini-Circuits ZHL-100W-GAN+. https://www.minicircuits.com/pdfs/ZHL-100W-GAN+.pdf.

Appendix

A Screen Refresh Rate Probing

Probing the excitation signals directly on the TX electrodes is difficult because the pins are insulated and hidden inside the touchscreen. To overcome this problem, we infer the excitation signals by measuring their electromagnetic emissions. From the probed signal, we can acquire the number of TX electrodes N, the time it takes to scan all TX electrodes T_p, and the time it takes to scan one TX electrode T_{p1}. For example, Figure 18 shows a waveform of the excitation signal we measured on a Nexus 5X. It shows that it takes approximately $T_p = 8.6$ms to finish scanning all TX electrodes, which consists of a preamble and a scanning segment. Within the scanning segment, there are 27 pulses, which corresponds to the $N = 27$ TX electrodes being scanned in turn, and it takes $T_{p1} = 0.27$ms to scan one TX electrode. The scanning period T_p is the reciprocal of the touch sampling rate of the touchscreen. Our measurement therefore suggests the touch sampling rate of the Nexus 5X to be around 120Hz [2].

By counting the pulses, it is possible to synchronize the emission of the electromagnetic signal to the RX electrode which is currently sensed. For that we used a MyDAQ for signal filtering and preamble extracting, it then outputs a signal to trigger the emission of our EMI signal. Hence, it is also possible to use the same antenna to receive passively before the attack.

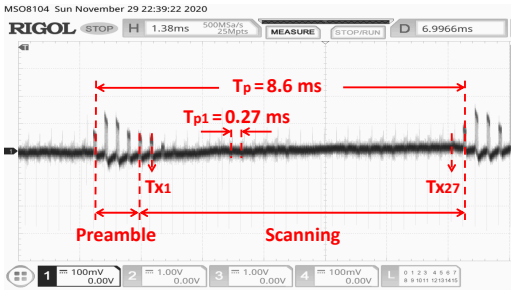
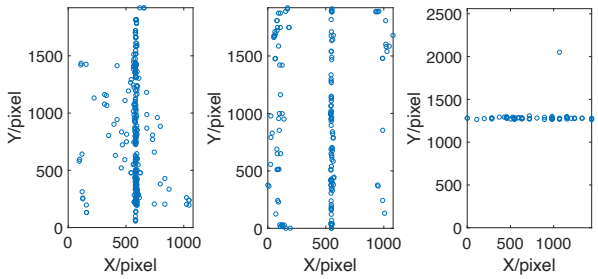


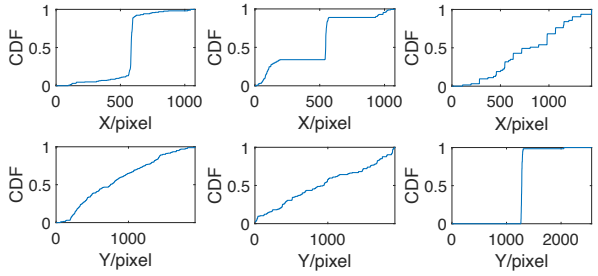
Figure 18: A waveform of the TX excitation signals on a Nexus 5X recorded by an oscilloscope. It shows that it takes approximately 8.6ms to scan all 27 TX electrodes and 0.27ms to scan one TX electrode.

B Feasibility



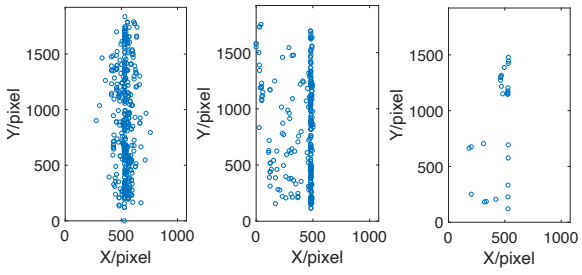
(a) Google Pixel 1. (b) Nexus 5X. (c) Huawei P10 Plus.

Figure 19: Visual distribution of the injected touch points on a Google Pixel 1, Nexus 5X and Huawei P10 Plus in the portrait orientation. The X and Y axes refer to the horizontal and vertical edges of the screen.



(a) Google Pixel 1. (b) Nexus 5X. (c) Huawei P10 Plus.

Figure 20: Cumulative distribution of the injected points along the X (1st row) and Y (2nd row) axes of the screen on Google Pixel 1, Nexus 5X, and Huawei P10 Plus. More than half of the injected points distribute on a specific line of the screen, either vertical or horizontal depending on the phone model.



(a) 5mm. (b) 7mm. (c) 10mm.

Figure 21: Visual distribution of the injected touch points on Google Pixel 1 at three attack distances. The injected touch points become less intense and less concentrated as the attack distance increases.

C ChipSHOUTER's Pulse

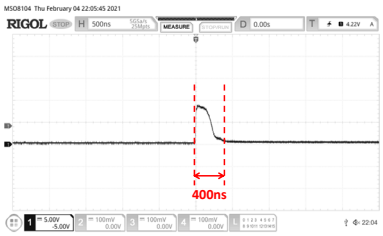
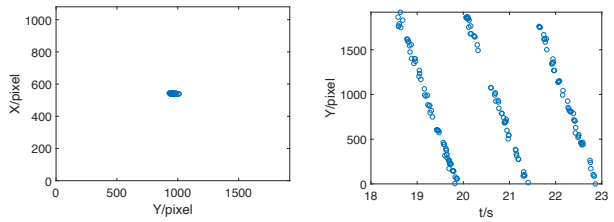


Figure 22: A waveform of the pulse generated by the ChipSHOUTER.

D Taps and Swipes



(a) Tap. (b) Swipe.

Figure 23: Illustrations of the injected taps on the screen and an injected swipe by drifting the touch points over time.

E Consistency of Taps

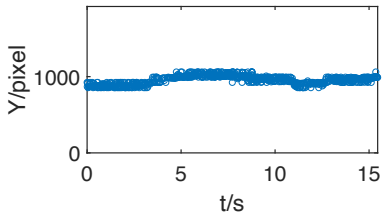
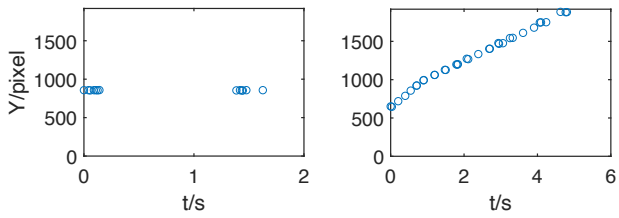


Figure 24: An illustration of the consistency of tap locations. The injected points stay in a small area for 15 seconds.

F Samples with an 8mm-thick Table



(a) Tap (b) Swipe

Figure 25: Tap and swipe injected beneath an 8mm-thick table.

G Accidental Touch Protection

Many smartphones have a "Mistouch Prevention Mode" also called "Pocket Mode", which turns off the touchscreen, prevents it from turning on or disables input in order to prevent accidental touches when the proximity sensor detects that the screen is blocked. This function may affect the practicality of the `GhostTouch` attack: 1) For phone models like, e.g., Google Pixel 1, Nexus 5X, MIX2, this function is only supported in calls. 2) For iPhones, the function turns off the screen during calls or prevents waking the screen on a notification. 3) For devices like Huawei P40, it would prevent touch detection when the proximity sensor is covered. Case 1) will not affect the `GhostTouch` attack. For case 2), it can still answer an eavesdropping call. For case 3), for Huawei phones this mode is turned off by default, which results in not many devices having this mode enabled. In addition, it is still possible to answer the phone call e.g., on Samsung Galaxy Note10, and the touchscreen can be activated by completing the check shown on the screen, e.g., by swiping twice or dragging the 'lock' icon. Considering the global market shares of Huawei and Apple smartphones are respectively 14.6% and 11.8% in the third quarter of 2020 [18], a significant proportion of smartphones are exposed to the threat of the `GhostTouch` attack.