Periscope: A Keystroke Inference Attack Using Human Coupled Electromagnetic Emanations

Wenqiang Jin*
wqjin@hnu.edu.cn
Hunan University

Srinivasan Murali
srinivasan.murali@mavs.uta.edu
The University of Texas at Arlington

Huadi Zhu
huadi.zhu@mavs.uta.edu
The University of Texas at Arlington

Ming Li
ming.li@uta.edu
The University of Texas at Arlington

ABSTRACT

This study presents Periscope, a novel side-channel attack that exploits human-coupled electromagnetic (EM) emanations from touchscreens to infer sensitive inputs on a mobile device. Periscope is motivated by the observation that finger movement over the touchscreen leads to time-varying coupling between these two. Consequently, it impacts the screen's EM emanations that can be picked up by a remote sensory device. We intend to map between EM measurements and finger movements to recover the inputs. As the significant technical contribution of this work, we build an analytic model that outputs finger movement trajectories based on given EM readings. Our approach does not need a large amount of labeled dataset for offline model training, but instead a couple of samples to parameterize the user-specific analytic model. We implement Periscope with simple electronic components and conduct a suite of experiments to validate this attack's impact. Experimental results show that Periscope achieves a recovery rate over 6-digit PINs of 56.2% from a distance of 90 cm. Periscope is robust against environment dynamics and can well adapt to different device models and setting contexts.

CCS CONCEPTS

• **Security and privacy → Mobile and wireless security**.

KEYWORDS

Keystroke inference attack; human coupled electromagnetic emanations; analytic model

ACM Reference Format:
Wenqiang Jin, Srinivasan Murali, Huadi Zhu, and Ming Li. 2021. Periscope: A Keystroke Inference Attack Using Human Coupled Electromagnetic Emanations. In *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security (CCS '21), November 15–19, 2021, Virtual Event, Republic of Korea.* ACM, New York, NY, USA, 15 pages. https://doi.org/10.1145/3460120.3484549

*The work was done when the author was a Ph.D. student at The University of Texas at Arlington.

CCS '21, November 15–19, 2021, Virtual Event, Republic of Korea.

ACM ISBN 978-1-4503-8454-4/21/11.
https://doi.org/10.1145/3460120.3484549

1 INTRODUCTION

Mobile devices, such as smartphones and tablets, have penetrated into everyday life. People typically enter sensitive inputs to mobile devices with virtual keyboards, including bank card number, security code, and digit PIN. Prior research has shown that these keystrokes can be inferred from onboard motion sensor readings [9, 30, 33, 36, 39, 57], acoustic signals at microphones [7, 18, 29, 32, 46, 62], video recordings [5, 6, 11, 42, 45, 54–56], and radio signals captured by surrounding wireless infrastructures [1, 16, 27, 28, 61]. To access these side channels, most existing works have to impose strong assumptions over attacker's capabilities or attacking scenarios. For example, motion sensor based attacks require the pre-installation of certain malware to victim's device to access sensor readings. Video based attacks rely on the line-of-sight (LoS) view of the typing process or object of interest that reflects typing motions. Radio signal based attacks analyze reflected signals to characterize environment disturbance caused by finger movements to learn which key is pressed. It cannot tolerate any background context changes, as otherwise, the subtle signal fluctuations introduced by finger movements are easily buried under larger-scale signal variations caused by environment dynamics. All the above restrictions render many existing keystroke inference attacks impractical in real-world scenarios.

In this work, we present an attack that leverages electromagnetic (EM) emanations leaked from device's touchscreens to snoop keystrokes. While the EM emanations have been explored for keystroke inference attacks [14, 51, 53], previous efforts have been focused on physical keyboards. When a key is pressed, the keyboard sends a packet of information known as a *scan code* to the computer. The scan code is bound to a physical button on the keyboard. The information leakage threat exists because part of the internal circuit acts as an antenna and radiates unintentional encoded information in EM waves. The attacker can easily reproduce each keystroke by relating it to its unique EM wave pattern. For virtual keyboards on mobile devices, their working principle is quite different. The way to recognize a keystroke does not rely on the scan code, but rather the current changes in the electrode grid. (Details will be covered in Section 4.1.) Thus, the fingerprinting EM leakage from a specific physical button no longer exists.

For the first time, our attack analyzes touchscreen's EM emanations under the *human coupling effect*. As suggested by [43, 58], a human body can be treated as a conductor with low impedance (a few $k\Omega$). When a user's finger approaches the screen, it generates a radiative coupling with the touchscreen's circuit. A portion of

electric charges are extracted from the electrode grid to the finger through the coupling capacitance. As the finger moves over a screen to enter inputs, it changes the coupling capacitance. Consequently, it influences the touchscreen's EM emanations, which can be detected by a remotely located eavesdropper. Our attack is built on this phenomenon to map EM emanation fluctuations with finger movements for performing keystrokes. Compared with state-of-the-art inference attacks, our scheme is more practical to execute from the following aspects. First, it eavesdrops keystrokes in a non-invasive way. Hence, it avoids the requirement to infect the victim device in advance of the attack. Second, as EM emanations can easily penetrate through obstacles, no LoS view is needed to launch the attack. Third, since our attack relies on direct EM radiations from touchscreens rather than reflected signals, it is robust against environment dynamics. We name our proposed attack as Periscope as it can observe and disclose victim's keystrokes covertly without a LoS view.

Despite these promising features, harnessing EM emanations for keystroke inference still faces a significant challenge, that is, to establish a relationship between observed EM emanations and a specific key press. A straightforward solution is to build a machine learning model that maps between these two. For this purpose, the attacker first needs collect labeled dataset of a reasonable size and train the model properly. During the attack phase, unknown EM emanations are fed into the trained model as inputs, with the output as which key was most likely pressed. In fact, this approach is adopted in most existing acoustic and radio signal based inference attacks [1, 18, 27, 28, 61]. However, training significantly hinders the attack deployment. As users' typing behaviors are distinct, user-dependent inference models are preferred to capture this uniqueness. It requires either access to the victim's device for some time or possession of her labeled dataset.

To avoid the training hurdle, we aim to develop an analytic model that characterizes the relation between EM emanations and keystrokes. To facilitate the analysis, we divide the continuous EM readings of entering the entire PIN into several segments, each associated with one key pair. By looking into the equivalent circuits of the touchscreen with finger coupling, we first derive the closed-from expression between realtime EM readings and instant finger-screen distances. Nonetheless, the latter may not directly reflect specific keystrokes. To fill the gap, we further estimate the finger movement speed and direction of entering one key pair. With these parameters, time-dependent finger-screen distances are equivalently transformed to a 3D finger movement trajectory, which are further cast to two 2D planes. The projected trajectories reveal finger movement lengths for entering one key pair in both horizontal and vertical directions on the screen. After such projection and transformation, we establish an explicit relation between EM readings and finger movements. Meanwhile, we notice that different key pairs may share an identical finger movement trace. To alleviate the inference ambiguity, we propose to explore the inter-dependency between consecutive key pairs to narrow down possible keystrokes. We model the entire PIN entering process as a Hidden Markov Model (HMM), with the recovered finger movement traces as observations, whereas the exact key pairs as hidden states. Finally, HMM outputs a list of PINs ranked based on their probability of being the target PIN.

To evaluate the proposed Periscope, we build a prototype with an Arduino board [3] and a conductive wire, with the total cost around $10. Extensive experiments show that our Periscope achieves a recovery rate over 6-digit PINs of 56.2% at a distance of 90 cm. Tests also show that Periscope is robust against environment dynamics and transparent to attacker displacement. Besides, it stays effective for a diverse set of devices and environment context. We summarize the contributions of this paper as follows.

- We investigate a novel side-channel attack to eavesdrop user's digit inputs on mobile devices by analyzing human-coupled EM emanations from touchscreens. While EM emanation based inference attacks have been studied on physical keyboards before, they are inapplicable to virtual keyboards due to their distinctive working principles.
- By analyzing touchscreen circuits under the human coupling effect, a closed-form expression is derived to characterize the relation between EM readings and finger movements. With the analytic model, keystrokes can be easily recovered from EM readings without training hurdles.
- We develop a prototype and demonstrate the severity of the threat. It outperforms state-of-the-art inference attacks in terms of setup practicability with much fewer deployment restrictions. Besides, the total cost of the prototype is as low as $10.

2 RELATED WORK

Existing keystroke inference attacks that exploit side-channel information can be broadly classified into the following categories.

Motion sensor based attacks. Efforts have been made on inferring user's keystrokes from data generated by on-board motion sensors. Early works [9] and [39] utilize mobile device's accelerometer readings to infer victim's passwords. By further involving gyroscope, [36] and [57] are able to increase the attack success rate. In this line of research, some recent works [30, 33] show that the similar idea can be applied to wearables to snoop victim's inputs. However, these attacks cannot succeed unless the victim device is pre-installed with certain malware to acquire motion sensor data, which limits their applicability.

Acoustic signal based attacks. Genkin et al. [20, 21] are among the first to study acoustic cryptanalysis that exploits sounds emitted by computers or other devices to reveal sensitive information. Some keypads such as ATM inputs and door keypads provide an audio feedback to the user for each button pressed. Such audio feedback is observable from a fair distance. Prior works [7, 18] quantify the delays between feedback pulses to reconstruct the keystrokes. This type of attack is susceptible to acoustic background noise. Besides, not all keypads emit audio feedback. Another line of research infers user inputs by employing acoustic ranging techniques. They utilize microphones to locate finger taps and thus the corresponding buttons on a screen [29, 46, 62]. It is not easy to derive an analytic model that characterizes finger movement trajectory with respect to audio sound. Researchers [4, 52, 63] have to resort to machine learning techniques and train classifiers to reconstruct the keystrokes so far. Tedious data sample collection and offline training process are unavoidable.

Video based attacks. Empowered by advanced computer vision techniques, video based attacks have been investigated for a while. Its idea is to use cameras to record the typing process or an object that reflects typing motion and then identify inputs by analyzing the recorded video. Prior works have demonstrated the feasibility of launching inference attacks by recording hand movement [45, 56], eye movement [11, 54, 55], tablet backside motion [47], reflections from nearby objects (e.g., glasses and plastic bottle) [5, 6, 42]. In these attacks, cameras should have a LoS view for object of interest; otherwise, keystroke activities cannot be detected. Besides, this type of attack does not work under poor lighting conditions.

Radio signal based attacks. Emerging research efforts have been made on eavesdropping keystrokes from radio signals due to the wide deployment of wireless infrastructures (e.g., WiFi and cellular towers). In particular, prior works [1, 16, 27, 61] reveal victim's keystrokes via the WiFi channel state information (CSI). Ling et al. [28] recovered the typed PIN on an ATM by analyzing the reflected cellular signals. As they rely on wireless infrastructures to launch the attack, the signal strength is relatively strong. Hence, the attacking distance is up to several meters. On the other hand, as radio signals are highly susceptible to environmental dynamics, these attacks cannot tolerate any changes in the environment other than the victim's hand or finger movement. As Periscope utilizes direct EM radiations from the device's touchscreen rather than reflected signals from the target, it is robust against environment dynamics. Besides, no extra wireless infrastructure is needed to launch the attack. Like acoustic signal based attacks, due to the complexity of formulating the relationship between observed wireless disturbances and specific key presses, radio signal based attacks typically resort to machine learning techniques too. Recently, Fang et al. [16] proposed a training-free keystroke inference attack by leveraging structures of dictionary words. They built a prototype with USRP, with a total cost around several thousand dollars.

EM emanation based attacks. EM radiations unintentionally leak from electronic devices. It has been investigated as a side channel to and victim's keystrokes on physical keyboards [14, 51, 53]. Notably, each key is associated with a unique scan code. Once it is pressed, the PC recognizes the key by reading the imported information through the data cable. The attack is based on the observation that the encoded keystroke information is radiated to the open air the form of EM emanations as it is transmitted over the cable. The working principle of soft keyboards is different. A keystroke is recognized by locating the touched position on a screen surface from current changes. Therefore, the existing eavesdropping method toward physical keyboards is inapplicable here. For the first time, Periscope examines the EM radiation changes caused by human coupling effects when a finger performs keystrokes. We then build a mapping relation between EM emanations and finger movement trajectory which serves as the foundation of our attack.

EM emanations have also been exploited to infer displayed information on a device's screen [24, 31, 49], profile device memory usages [12], and identify the model of LCD monitors [37], which are parallel to what we study here.

A comprehensive comparison with related works is summarized in Table 4 in the Appendix.

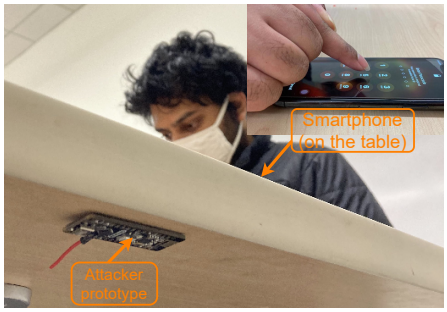


Figure 1: Eavesdrop EM emanations using an attack device.

3 ADVERSARY MODEL

Attack scenario. The attack scenario is considered as that an adversary seeks to infer a victim's secret PIN by eavesdropping her keystrokes on a mobile device. As shown in Figure 1, the victim places her device on a table and types on a soft numeric keyboard on the screen. Such scenarios are prevalent in daily life, such as in a library or a cafe where users unlock their smartphones by entering digit PINs. A similar setting is considered in prior works [27, 34, 47]. We plan to investigate in our future work a more complicated scenario that an attack is launched when a victim user holds the mobile device. The attacker is assumed in physical proximity to the victim. It is well concealed, e.g., placed underneath a table or in a bush nearby [34]. We focus on soft numeric keyboards with a classic layout, though the attack can target other layouts just as easily.

What an attacker cannot do. Unlike many prior keystroke inference attacks, the attacker does not necessarily have a LoS view of victim's keyboard or any other object of interest, such as hand movement, eye movement, and tablet backside motion. We do not assume the existence of any covert channel that reveals victim's onboard sensor readings to the attacker either. Also, there is no ideal environment, static or quiet, to launch attacks. The victim can make free body movements during the typing process; other people may walk by or talk in the background. Besides, it is unlikely for an attacker to collect large amounts of data samples from a specific victim to train an individual keystroke inference model properly before the attack. The above settings render most of the existing keystroke inference attacks infeasible.

What an attacker can do. The attack is able to figure out which mobile device a victim is using and thus its numeric keyboard layout. In practice, the attacker can investigate the MAC address of the victim's WiFi traffics to obtain the device manufacturer information by looking up prefixes of MAC addresses [13]. As mentioned in [22, 38, 64], the victim's DNS responses contain its device name. With the information above, many mobile devices can be fingerprinted. Prior work [22, 38] provide technique details on setting up a free WiFi access point to access the victim's MAC address and DNS responses for device fingerprinting unnoticeably.

4 PRELIMINARIES

4.1 How Do Touchscreens Work?

The majority of current mobile devices, such as smartphones and tablets, are equipped with touchscreens. While there are various sensing touch technologies, mutual capacitive sensing has been

the most prominent due to its high sensitivity, energy efficiency, and low manufacturing cost [40]. We thus focus on this type of touch-sensing devices here.

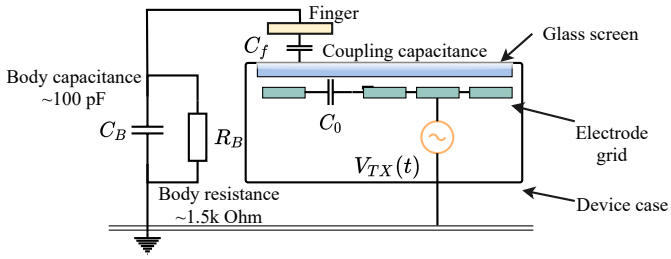


Figure 2: The composition of a mutual capacitive touchscreen.

As shown in Figure 2, a capacitive touchscreen consists of a grid of transmitter (TX) and receiver (RX) electrodes, which are mutually coupled with a capacitance of C_0. TX electrodes are driven by an alternating voltage signal $V_{TX}(t)$, which creates an alternating current flow from TX to RX electrodes. When a finger touches the screen, it extracts some electric charges from the electrode grid to the human body through a coupling capacitance C_f. The touchscreen controller monitors the changes in the current that flows into RX electrodes and reports the change as a touch event to the system OS. Meanwhile, it locates the current change in the electrode grid as the touched position on the screen. The input is then recognized accordingly.

4.2 Touchscreen EM Emanations and Measurements

The alternating currents between touchscreen's TX and RX electrodes generate time-variant EM fields that continuously emit EM radiations to the open space. Periscope intends to map the radiation to user's typing inputs.

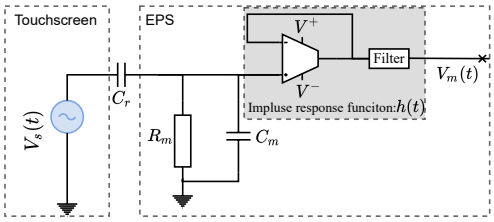


Figure 3: The circuit for touchscreen's EM emanation measurement.

Figure 3 depicts an equivalent circuit of using an electric potential sensor (EPS) to measure touchscreen's EM emanations. An EPS typically consists of a capacitor C_m, a resistance R_m, a voltage amplifier, and a low-pass filter. By placing the eavesdropper, i.e., EPS, within the EM field of victim's touchscreen, these two will be remotely coupled via a small capacitance C_r. Denote by $V_s(t)$ the time-variant voltage that drives EM emanations from the touchscreen. The captured EM emanation at EPS, measured in electric potential changes $V_m(t)$, is expressed as

$$V_m(t) = V_s(t) \cdot \frac{1/(\frac{1}{R_m} + j2\pi f C_m)}{\frac{1}{j2\pi f C_r} + 1/(\frac{1}{R_m} + j2\pi f C_m)} \cdot h(t). \quad (1)$$

Here $h(t)$ denotes the joint impulse response of the amplifier and the low-pass filter. f stands for the frequency of the driving voltage $V_{TX}(t)$. Among the parameters in (1), C_m, R_m, and $h(t)$ are fixed values. C_r depends on the attacker-victim distance. It can be treated as a fixed value too under a specific eavesdropping event. Now $V_m(t)$ is determined by $V_s(t)$. As demonstrated next, $V_s(t)$ is impacted by finger movement. Hence, we establish a connection between EM readings and finger movement. To validate this claim, we show in Figure 4 the spectrogram of EM readings $V_m(t)$ when a user enters a 6-digit PIN. There are 6 bars with intense magnitude, each representing the tap of one key. We also notice that the majority frequency components are scattered at the lower end of the spectrum band, below 60 Hz. It indicates that EM emanations can be easily captured by cheap EPS with a fair sampling rate.

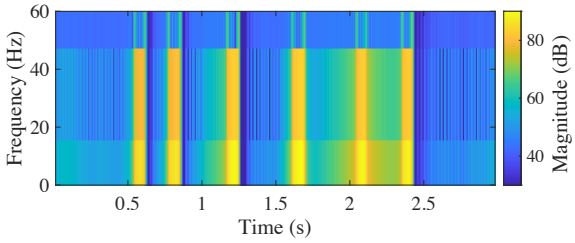


Figure 4: Spectrogram of EM emanation measurement $V_m(t)$.

4.3 Impact of Finger Coupling

The driving voltage of touchscreen EM emanation $V_s(t)$ is influenced by finger coupling that is modeled next.

Figure 5(a) is an equivalent circuit for a mutual capacitive touchscreen when no touching. R_{TX} (R_{RX}) represents the resistor at the TX (RX) electrode. Recall that C_0 is the TX-RX coupling capacitance. The equivalent circuit is transformed to Figure 5(b) when touching. As a finger moves close to the screen, they become remotely coupled via capacitance C_f. As shown in Figure 5(c), the finger extracts some electric charges through coupling to the human body (characterized in C_B and R_B). We call the above phenomenon as *finger/human coupling effect*. When a finger is coupled to the screen, $V_s(t)$ is expressed as

$$V_s(t) = V_{TX}(t) \cdot \frac{R_{TX}}{R_{TX} + 1/j4\pi f C_0 + Z(t)} \quad (2)$$

where $Z(t)$ denotes the equivalent time-variant impedance of the right-half circuit of Figure 5(b)

$$Z(t) = 1/(\frac{1}{1/j2\pi f C_f(t) + 1/(j2\pi f C_B + 1/R_B)} + \frac{1}{1/j4\pi f C_0 + R_{RX}}). \quad (3)$$

Let $z(t)$ be the instant finger-screen distance. According to [10], $C_f(t)$ can be expressed as

$$C_f(t) = \frac{\epsilon_0 \epsilon_r A}{z(t)}, \quad (4)$$

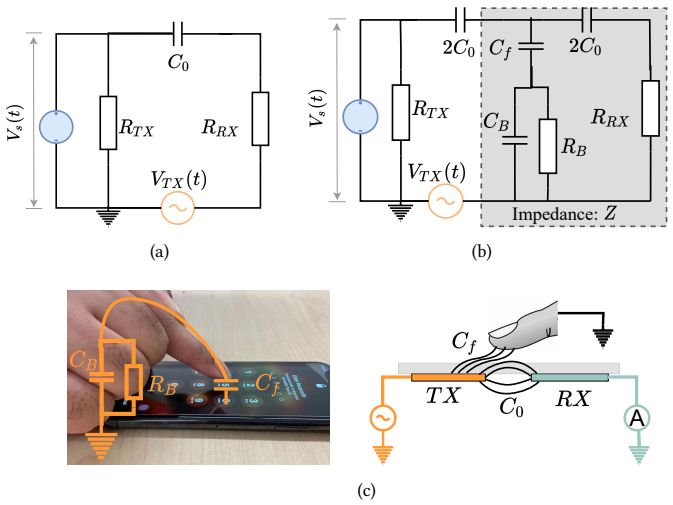


Figure 5: Illustration of finger coupling effect. (a) Equivalent circuit without finger touches. (b) Equivalent circuit with finger touches. (c) Screen-finger coupling.

where ϵ_0 and ϵ_r are dielectric permeability coefficients. A is the overlap area between the fingertip and the screen. As ϵ_0, ϵ_r and A are fixed values in one keystroke, $C_f(t)$ is negatively correlated with $z(t)$. Together with (2) and (3), we have the following relation $z\downarrow, C_f\uparrow, Z\downarrow, V_s\uparrow$. In short, the touchscreen emits stronger EM emanations when the finger moves closer to it and vice versa. According to (1), V_m has a positive correlation with V_s. We thus have $z\downarrow, C_f\uparrow, Z\downarrow, V_s\uparrow, V_m\uparrow$. This relationship chain indicates that the finger coupling effect reveals a side channel to monitor finger movements: remote EM emanation measurements $V_m(t)$ reflect finger's realtime distance to the screen $z(t)$ when performing keystrokes.

4.4 How to Calculate $z(t)$ from $V_m(t)$?

While the above analysis exhibits a negative correlation between $z(t)$ and $V_m(t)$, we seek to further quantify this relationship, i.e., how to calculate $z(t)$ from $V_m(t)$ exactly? Essentially, our goal is to derive a closed-form expression of $z(t)$ as a function of $V_m(t)$ via (1)-(4). Nonetheless, this task is nontrivial.

Some parameters in (1)-(4), such as R_m, C_m, C_r, and $h(t)$, are not readily available. For example, C_r is determined by the placement of the victim device and EPS. To resolve this issue, our trick here is to utilize multiple measurements that can cancel out the unknown parameters during the calculation.

As a note, EPS measures $V_m(t)$ in its amplitude, denoted as $|V_m(t)|$. Let $|V_m(t)|^*$ be the maximum value of $|V_m(t)|$. It is obtained the moment that a finger touches the screen. $|V_s(t)|$ and $|V_s(t)|^*$ are defined similarly. We have

$$\frac{|V_m(t)|}{|V_m(t)|^*} \overset{①}{=} \frac{|V_s(t)|}{|V_s(t)|^*} \overset{②}{=} \frac{|R_{TX}+1/j4\pi fC_0+Z(t)|^*}{|R_{TX}+1/j4\pi fC_0+Z(t)|} \tag{5}$$

where ① and ② are due to (1) and (2), respectively. As suggested by [19, 26], C_f and C_0 are generally very small, around 2 pF (2×10^{-12} F). Thus, their equivalent impedance is much larger than the body resistance R_B (around 1.5 $k\Omega$ [50]), the body capacitance C_B (around 100 pF), as well as the resistance of electrodes R_{TX} and R_{RX} (around 160Ω [26]). Then, (5) is rewritten as[1]

$$\frac{|V_m(t)|}{|V_m(t)|^*} \simeq \frac{|1/j4\pi fC_0+Z(t)|^*}{|1/j4\pi fC_0+Z(t)|}, \tag{6}$$

Similarly, $Z(t)$ is approximated as

$$Z(t) \simeq 1/(\frac{1}{1/j2\pi fC_f(t)}+\frac{1}{1/j4\pi fC_0}) = 1/(j2\pi fC_f(t)+j4\pi fC_0). \tag{7}$$

Let the maximum finger coupling capacitance be C_f^*. In practice, manufacturers tend to set the TX-RX coupling capacitance C_0 approximate to C_f^* [26]. We thus have $C_f^* \simeq C_0$, by which C_f^* is achieved under the minimum finger-screen distance $z_{\min}$. From (4), we have $\frac{C_f(t)}{C_f^*} = \frac{z_{\min}}{z(t)}$ which leads to

$$C_f(t) = C_f^*\frac{z_{\min}}{z(t)} \simeq C_0\frac{z_{\min}}{z(t)}. \tag{8}$$

Combining (6) - (8), we have

$$\frac{|V_m(t)|}{|V_m(t)|^*} \simeq \frac{|1/j4\pi fC_0+Z(t)|^*}{|1/j4\pi fC_0+Z(t)|} = \frac{\frac{z_{\min}}{z(t)}+2}{\frac{z_{\min}}{z(t)}+4}\cdot\frac{5}{3}. \tag{9}$$

and thus

$$z(t) = 1/(\frac{2\gamma_1}{1-\frac{|V_m(t)|}{|V_m(t)|^*}\frac{3}{5}} - 4 + \gamma_2)\times z_{\min}. \tag{10}$$

$z_{\min}$ is essentially the thickness of touchscreen's covering glass. It can be determined once the device manufacture information is figured out. For example, $z_{\min}$ equals to 0.6 mm for iPhone SE2 [23, 25, 48]. $|V_m(t)|^*$ is the maximum measurement the EPS captured. It can be treated as a known value. γ_1 and γ_2 are coefficients that compensate the approximation errors of $z(t)$ in (2)-(8). They can be estimated via offline calibration.

So far, we are able to express $z(t)$ into a function of $V_m(t)$. Given an instant EM emanation measurement, the corresponding finger-screen distance can be obtained following (10). More importantly, no training phase is needed. Unlike many wireless signal based inference attacks, our analytic model is transparent from underlying signal propagation channel conditions, as they have been incorporated into $|V_m(t)|$ and $|V_m(t)|^*$. Their impact is canceled with each other during the calculation. Still, the attacker cannot infer victim's typing inputs from $z(t)$ directly, unless it has the full knowledge of the finger movement trajectory. We present how to derive the latter from $z(t)$ in Section 6.

5 MEASUREMENT STUDY

The objective of this section is to validate the analytic result of Section 4 and investigate the feasibility of leveraging human-coupled EM emanations to launch keystroke inference attacks.

We build our prototype using an Arduino nano board [3] as a microcontroller unit (MCU) and a conductive wire as an antenna. These two are connected via Arduino's analog input pin shown in Figure 6. The antenna senses the electric potential changes caused by touchscreen EM emanations. The system samples received signals with an analog-to-digital (A/D) converter at a rate of 4000

[1]Given two complex values $a+jb$ and $c+jd$, if $b >> a$ then $|a+jb+c+jd| \simeq |jb+c+jd|$ since $\sqrt{(a+c)^2+(b+d)^2} \simeq \sqrt{(c)^2+(b+d)^2}$.

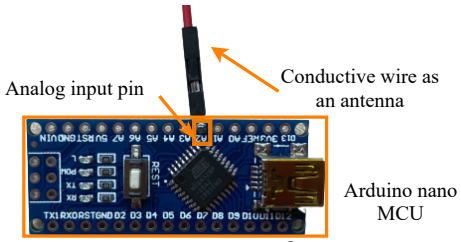


Figure 6: Prototype of Periscope.

samples/sec. Recall that the frequency of touchscreen EM emanations is bounded within 60 Hz according to the spectrogram analysis in Section 4.2. Therefore, the prototype's sampling rate is more than enough to capture signal variances in EM emanations. The entire prototype costs less than $10, which renders the attack easily accessible and widely deployable.

To validate the analytic model for $z(t)$ derived in Section 4.4, Figure 7 compares it with the ground truth measurement. It is observed that the former generally complies with the latter. Meanwhile, the approximation operations involved in the derivation process do introduce some marginal discrepancies between these two. We plan to investigate its impact on the attack performance in experimental evaluations.

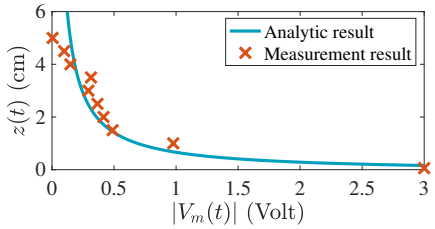


Figure 7: Estimation of $z(t)$ from $|V_m(t)|$

Figure 8 shows EM measurements when entering a 6-digit PIN. EM emanation variations reflect finger interactions with the screen. We find that the signal experiences a sharp increase when the finger moves towards the screen. It then decays quickly the moment a physical contact takes place. This is because the finger draws some electric charges from the screen. With reduced electric charges, the EM radiation from the screen drops accordingly. Later on, as the finger leaves the screen for the next key, the EM amplitude keeps decreasing until the finger is de-coupled from the screen. This observation coincides with the analytic result derived previously.

Figure 9 shows EM measurements by entering three different key pairs "42", "46", and "43". It is observed that their EM readings are distinct to each other. For example, "42" is associated with the shortest time duration between two consecutive EM amplitude peaks, as a finger moves in the shortest path to enter this key pair among the three. We further evaluate the similarity of EM emanations among ten key pairs originated from "4" in Figure 10. Normalized DTW distance is employed. A small value represents a high similarity between two pairs, while a larger one means they are barely correlated. We find that except for the diagonal, i.e., a key pair and itself, DTW distances between EM readings from any two different pairs are relatively large.

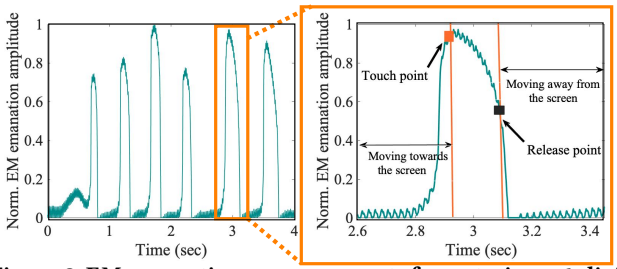


Figure 8: EM emanation measurements for entering a 6-digit PIN.

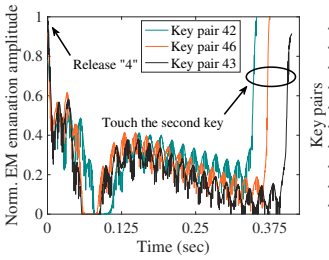


Figure 9: EM measurements of entering different key pairs.

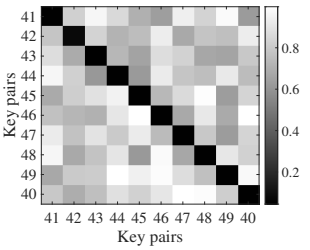


Figure 10: Similarity matrix for different key pairs using normalized DTW distance.

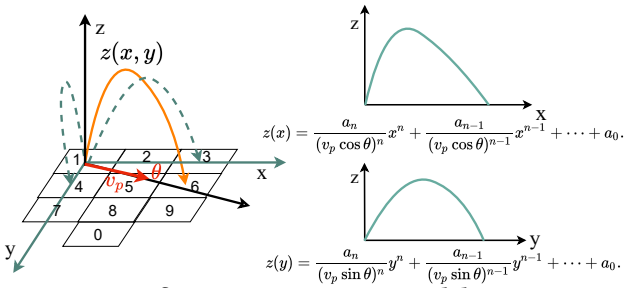


Figure 11: 3D finger movement trace and decomposition.

Based on the above observation, we propose to recognize individual key pairs from their EM measurements first and then the whole PIN.

6 DESIGN RATIONALE

Our design first separates the received continuous EM emanations into multiple segments, each representing signals from one key pair. Recall that we are able to map an instant EM reading to the associated finger-screen distance. Then we apply some transformations to convert time-dependent finger-screen distance to finger movement traces, which finally recover key pairs and thus the PIN.

Decomposition of 3D finger movement trace. As shown in Figure 11, the finger movement for entering one key pair can be characterized by a 3D trace. By treating the first keystroke as the origin point, we set up a 3D coordinate system, where the x-y plane is where the screen resides and z-axis is vertical to the screen. For any 3D finger movement trace, denoted as $z(x, y)$, let its projection on the x-z plane and y-z plane be $z(x)$ and $z(y)$, respectively. If we know the intersection between $z(x)$ ($z(y)$) and x-direction (y-direction) of the keyboard, the key pair is recovered. For this purpose, we further divide the x-direction, denoted

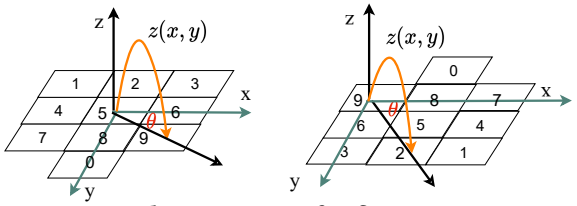


Figure 12: Coordinate systems for finger movement traces from arbitrary key pairs.

as L_x, of the keyboard into 3 units; each represents one key. Similarly, the y-direction, denoted as L_y, is divided into 4 units. Under this setting, key pair "16", for example, can be represented as $L_x = 2$ units, $L_y = 1$ unit. Our task now becomes how to determine L_x and L_y of a specific key pair from its EM emanation readings.

Relation between L_x (L_y) and EM readings. Denote by θ the angle between 3D trace $z(x, y)$ and $z(x)$, its projection on the $x - z$ plane. Let $x(t)$ ($y(t)$) be the finger's instant position at time t cast on the x-axis (y-axis). Then we have

$$x(t) = v_p t \cos\theta, \; y(t) = v_p t \sin\theta, \tag{11}$$

where v_p is the finger movement speed[2]. Besides, the time-series finger-screen distance $z(t)$ of one key pair can be approximated with a high dimensional polynomial

$$z(t) = a_n t^n + a_{n-1} t^{n-1} + \cdots + a_i t^i + \cdots + a_0. \tag{12}$$

The $n+1$ coefficients $a_0, \cdots, a_n$ can be determined by solving a linear equation system with n+1 samples: $(t_1, z(t_1)), \cdots, (t_{n+1}, z(t_{n+1}))$, where $z(t)$ can be calculated from $V_m(t)$ following (10). Combining (11) and (12), $z(x)$ is expressed as

$$z(x) = \frac{a_n}{(v_p \cos\theta)^n} x^n + \frac{a_{n-1}}{(v_p \cos\theta)^{n-1}} x^{n-1} + \cdots + a_0. \tag{13}$$

Similarly, $z(y)$ can be expressed as

$$z(y) = \frac{a_n}{(v_p \sin\theta)^n} y^n + \frac{a_{n-1}}{(v_p \sin\theta)^{n-1}} y^{n-1} + \cdots + a_0. \tag{14}$$

With $z(x)$ ($z(y)$), by examining its intersection with the x-axis (y-axis), we can easily obtain L_x (L_y). To be specific, solve x by setting $z(x) = 0$. L_x is the unit that x falls into. L_y is obtained similarly. The above calculation relies on the knowledge of v_p and θ. We will discuss in Section 7.2 how to derive these two critical parameters.

To sum up, for each key pair, the attacker collects at least $n + 1$ samples of EM readings. Their corresponding finger-screen distances are calculated following (10). Then a polynomial of n-degree that characterizes time-series finger-screen distance $z(t)$ is constructed. With the knowledge of victim's finger movement speed v_p and direction θ, $z(t)$ is converted to $z(x)$ and $z(y)$. Their intersections with x-axis and y-axis are L_x and L_y, respectively. From the above analysis, we can tell that time-series $z(t)$ bears the information of time interval for entering one key pair. To alleviate the impact of EM variances to our analytic model, we resort to $n + 1$ samples $(t_1, z(t_1)), \cdots, (t_{n+1}, z(t_{n+1}))$ to derive $z(t)$ first.

Discussions. Note that a given pair of L_x and L_y may not uniquely identify a specific key pair, but a set of key pair candidates.

[2]Soft keyboards are generally small in size. For most users, the entry of a PIN can be performed smoothly within a short time. Thus, it is practical to assume a constant finger movement speed for each user. Speeds from different users are not necessarily the same though.

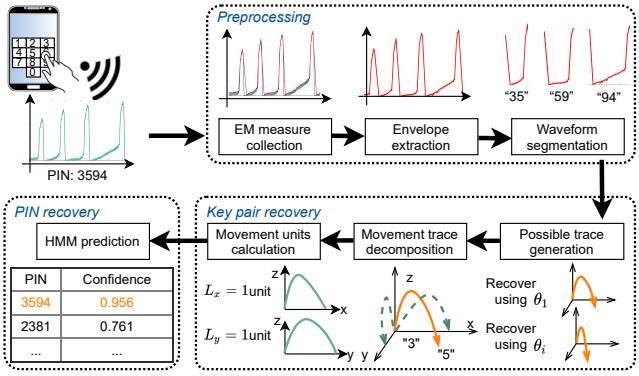


Figure 13: The system design of Periscope.

For example when $L_x = 2$ units and $L_y = 1$ units, satisfying key pairs include "61", "16", "34", "43", "67", "76", "49", and "94". To alleviate the inference ambiguity, we propose to model transitions between key pairs into a HMM to eliminate impossible combinations of key pairs. Details will be elaborated in Section 7.3.

In the above analysis, we use the key pair "16" to illustrate how to model a finger movement trace shown in Figure 11. A coordinate system, with "1" as the origin point, is set up. In fact, our method is applicable to arbitrary key pairs. Figure 12 demonstrates the cases of two other key pairs "59" and "92". Their origin points become "5" and "9", separately. In either case, we have the trajectory exist in the first quadrant of the coordinate system and thus $\theta \in [0, \pi/2]$ to facilitate our analysis.

7 DESIGN DETAILS OF PERISCOPE

The system overview of Periscope is given in Figure 13. It consists of three main components: *preprocessing*, *key pair recovery*, and *PIN recovery*.

7.1 Preprocessing

The goal is to extract clean signal segments for individual key pairs from continuous raw EM emanation readings.

Envelope extraction. As shown in Figure 14, raw EM emanation readings are mixed with oscillating signals, which add small-scale variations to the envelope. Essentially, the envelop signal is caused by finger coupling effect and thus contains useful information regarding finger movements. The oscillating signals, on the other hand, are produced by touchscreen's alternating driving voltage and useless for the attack. To extract the envelope, the extrema sampling based algorithm is employed [35, 59]. Specially, a sliding time window Δt is applied over the raw reading. The local maximal value within this window, $\max V_m(t')$ ($t' \in [t, t + \Delta t]$), is deemed as the filtered output for Δt.

Waveform segmentation. The purpose of this step is to segment the signal for each key pair out of a continuous waveform. We first identify critical time instances associated with finger release/touch events. For finger touch, it appears at EM reading peaks. We thus apply the classic peak detection algorithm [8] over the envelope signal to identify such events. Once the finger leaves the screen, the discharging coupling capacitance causes a sudden drop

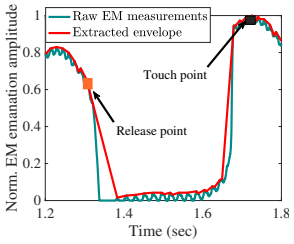


Figure 14: Envelope extraction and waveform segmentation.

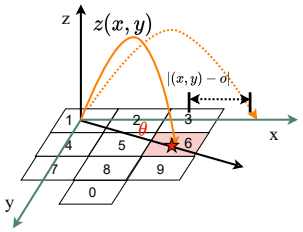


Figure 15: Estimation of θ.

in EM readings as shown in Figure 14. Hence, the finger release event is identified by locating the maximum derivative along the EM signal envelope between two consecutive peaks. Upon identifying the above critical events, the EM signal of one key pair is the waveform segment between the finger release (of the first key) and the next finger touch (of the second key).

7.2 Key Pair Recovery

Section 6 presents how to recover a key pair, recognized via L_x, L_y, from EM readings. As discussed, the attacker should be aware of the victim's finger movement speed v_p and direction θ. In the following, we focus on the estimation of these two parameters.

Estimation of θ. Let Θ be the set of possible directions of finger movement for entering a key pair. To estimate θ, our idea is compare among all the possible candidates in Θ and figure out the one that produces the highest estimation confidence level.

As discussed in Section 6, we take the first press of a key pair as the origin and set up a 3D coordinate system. Following the steps, the coordinate of the second press (x, y) is derived by solving $z(x) = 0$ and $z(y) = 0$. L_x and L_y are obtained accordingly. Let o be the geometry center of the key identified by L_x and L_y. A user typically taps the center of a key to enter an input. If θ is the correct direction, the derived (x, y) should be close to a key's center. Otherwise, (x, y) tends to deviate from the center, as illustrated in Figure 15. We then define the confidence level under θ as

$$l = 1 - \frac{|(x, y) - o|}{\sum_{\theta \in \Theta} |(x, y) - o|}. \quad (15)$$

l is a value between $[0, 1]$. It approximates 1 if (x, y) is close to a key center. Finally, finger movement direction is deemed as the one that produces the maximum confidence level among all the candidates, $\theta = \arg\max_{\theta \in \Theta} l$.

Generally speaking, most digit keyboards are asymmetric to the diagonal, e.g., the inter-key spaces in the x-axis and y-axis are distinct. Hence, multiple decomposition candidates are less likely to share the same probability for a given key pair. Symmetric keyboards, on the other hand, will impact the success inference rate, as the attacker cannot distinguish between x-/y-movements.

Estimation of v_p. As examined in prior works [2, 17], finger movement speed for typing is deemed consistent for each individual. We propose to estimate it by eliciting victims to enter some digits in their devices and estimating the speed from the collected samples. Specifically, the attacker can set up free WiFi. Once a victim is connected, the access point requires user approval by displaying a dialog box and asking the victim to enter designated numbers as a confirmation message [1]. An alternative approach is to set up a Text Captchas that asks the victim to input the chosen numbers [27]. Following the same key pair segmentation approach, we first separate victim entered number sequence into a series of key pairs. Then the time duration for entering one key pair is known. Since the exact key pair is known, so is the inter-key distance. The finger movement speed is estimated by dividing the distance by the time duration. We set v_p as the median value of measured speeds of all key pairs in one number sequence. To improve the estimation accuracy, the attacker can have the victim enter more than one sequence. According to our experiment result, three such digit sequences are sufficient to deliver satisfactory estimation. Periscope only needs a couple of user-specific samples to determine victim's typing speed. Some prior works apply user-agnostic models for typing inference. They typically employ sophisticated deep learning models that require large amounts of labeled training samples from various users to avoid overfitting.

7.3 PIN Recovery

So far, the attacker is able to infer L_x and L_y of a given EM waveform segment. As discussed, a pair of L_x and L_y can be mapped to multiple key pairs. We propose to leverage the interdependence of consecutive key pairs to resolve the inference ambiguity. For example, given $L_x = 2$ units and $L_y = 1$ units for the first waveform segment, satisfying key pairs include "61", "16", "34", "43", "67", "76", "49", and "94". Given $L_x = 2$ units and $L_y = 2$ units for the second waveform segment, satisfying key pairs include "19", "91", "37", and "73". Considering interdependence, the existing candidates for the first key pair "16", "34", "76", and "94" can be eliminated immediately, as none of them ends with "1", "9", "3", or "7", the first digit of the second key pair. Hence, viable candidates for the first key pair are narrowed down to "61", "43", "67", and "49", by 50%. As more key pairs are considered, this side information can be propagated back-and-forth to further reduce the ambiguity. We propose to model such interdependence between consecutive key pairs for PIN recovery using HMM.

We model the keystroke process as HMM characterized by $\lambda = (N, M, A, B, \pi)$. In the HMM, N is the number of hidden states. We treat key pairs as hidden states. As there are 100 possible key pairs, i.e., from "00" to "99", we have $N = 100$. The parameter M represents the number of possible observations for hidden states, i.e., L_x and L_y. As there are three and four possible values of L_x and L_y, respectively, we have $M = 3 \times 4 - 1 = 11$. A, with the size of $N \times N$, stands for the transition probability matrix, with each element denoting the transition probability from one hidden state to another. The observation probability matrix B, of the size $N \times M$, gives the possibility that a given observation can be observed in a hidden state. The initial state distribution vector π represents the belief about which state the HMM is in when our scheme is called for the first time.

To build the HMM, we need to determine parameters A, B, and π. The transition probability matrix A can be predefined by the natural continuity of the typing process. For example, if we assume equal probability of typing any keys, the hidden state "61" has a chance of 0.1 to transfer to each hidden state "1x", while the chance

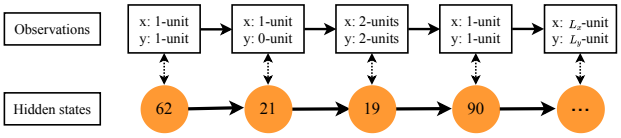


Figure 16: The state and transitions of HMM.

to other states is 0. B is obtained by evaluating the probability of a given key pair that generates certain observations. It actually reflects the accuracy of our proposed key recovery scheme. Due to random errors occurred in EM measurements, our scheme may generate false observations other than the ground truth at a certain probability. We propose to run our scheme offline ahead of the attack to derive B. In our design, we employ a uniform distribution for the initial state distribution π.

Given the observation sequence of key pairs $O = O_1O_2 \cdots O_S$, the PIN recovery problem is to find optimal hidden sequence $Q = Q_1Q_2 \cdots Q_S]$ to maximize $P(Q|O, \lambda)$. This problem can be solved by the Viterbi algorithm [41], a commonly adopted approach for HMM. In addition to search the most likely PIN, we also calculate the probability of all possible PINs generated by the HMM. The attacker can thus sort them according to their probabilities and form a list of candidates to infer the target PIN with multiple trials.

8 EXPERIMENTAL EVALUATIONS

The experiments are conducted using our prototype described in Section 5. It is built on a commercialized Arduino board that follows the FCC regulations and passively collects EM emanations. Hence, no risk is posed to human health. The collected data are anonymized and properly stored locally from potential leakage. The entire research has been approved by IRB.

The goal is to evaluate the performance of our proposed attack Periscope under different settings. A wide spectrum of impact factors are examined, such as system parameters, attack distances, environmental contexts, devices, and keyboard layouts. A comprehensive comparison is also made with existing schemes. A total of 20 volunteers, 12 males and 8 females between 22 to 28 years old, are recruited for the experiments. Before each experiment, detailed instructions regarding experimental procedures are provided. We design an App that mimics the UI that allows users to unlock the screen via digit PINs. During the experiment, each volunteer is asked to enter 60 randomly generated PINs into smartphones.

8.1 Key Pair Recovery Accuracy

As the basis of our attack, we first examine the accuracy of key pair recovery. Figure 17 shows the success rate over all the 100 possible key pairs from "00" to "99". Each row represents the first key, whereas each column represents the second key. It is observed that key pairs with longer inter-key distances tend to have better recovery accuracy. For example, the recovery rate of "01" is 94%; it becomes 89.5% for "08". Besides, we find that the success rate is not perfectly symmetric with respect to key pairs. In other words, the success rates of "ab" and "ba" are not exactly the same. This is because users may exhibit different typing behaviors when entering the same pair of keys but with reverse orders.

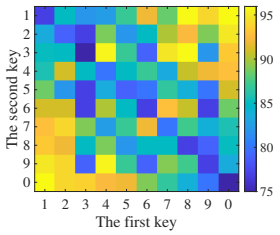


Figure 17: Key pair recovery accuracy.

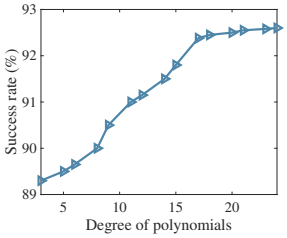


Figure 18: Impact of the order of polynomials.

Impact of degree of polynomials. To establish the relation between EM readings and L_x (L_y), we employ an n-degree polynomial to characterize the time-dependent finger-screen distance $z(t)$. Figure 18 shows key pair recovery accuracy with respect to the degree n. The success rate experiences a slight increase by adopting a higher degree polynomial. For example, the success rate is 89.6% when $n = 6$ and then raised to 91.2% when $n = 12$. It indicates that a polynomial with a higher degree can nicely tract the finger movement trace. Once n surpasses 17, such benefit becomes negligible. At the same time, a polynomial of higher degree incurs larger computation overhead in solving $z(x) = 0$ and $z(y) = 0$. To strike a balance between accuracy and efficiency, we set $n = 17$ by default.

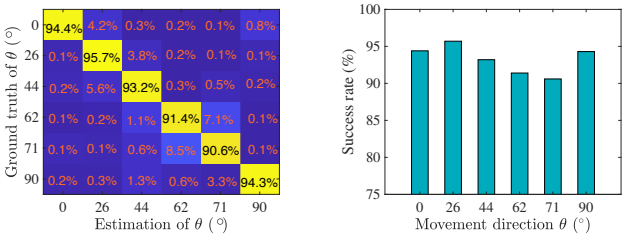


Figure 19: Recovery accuracy of θ. (a) Confusion matrix. (b) Recovery success rate of key pairs with different θ's.

Estimation of θ. The estimation of finger movement direction θ is critical to key pair recovery. Figure 19(a) shows the confusion matrix of θ estimation. A Google Pixel phone is adopted in the experiment. The rows represent all possible finger movement directions as the ground truth, whereas columns represent estimated results. As discussed in Section 6, $\theta \in [0, \pi/2]$. The figure easily tells whether our scheme causes any confusion between classes. The average recognition accuracy is 93.3%. We observe that most of the errors come from adjacent directions. For example, when the ground truth is 71°, the chance it recognized as 62° is 8.5%, which is the highest among all the cases. We also notice in Figure 19(b) that 62° and 71° are associated with relatively lower success rate, at 91.4% and 90.6% respectively, compared with other directions. This is because they are separated by a small margin of 9°.

Estimation of v_p. It is difficult to measure user's finger movement speed directly. To approximate the ground truth, we divide the distance between two touch points for entering a key pair by its

time duration. The distance can be readily computed from coordinates of the two touches, accessible from smartphone API. Table 1 exhibits the estimation error over v_p from three randomly selected volunteers. We find that the error decreases as the user is asked to enter more digit sequences in advance of the attack. Take volunteer 1 as an example, the estimation error is 5.12 cm/s with one digit sequence and drops to 1.39 cm/s under five digit sequences. It meets our expectation; the estimation becomes more robust to variations introduced by an individual sample. We further evaluate in Figure 20 the impact of number of digit sequences to key pair recovery. The recovery success rate quickly increases to 85% under three digit sequences. Beyond that, the growth becomes incremental. To trade between practicality and accuracy, we suggest having victims enter three digit sequences in advance of the attack.

No. of Seq.	V 1	V 2	V 3
1	5.12	4.43	3.24
2	2.72	3.0	2.16
3	1.96	2.46	1.74
4	1.49	2.27	1.25
5	1.39	1.97	0.82

Table 1: Estimation error of v_p (cm/s).

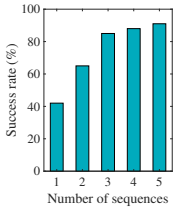


Figure 20: Key pair recovery performance.

8.2 PIN Recovery Accuracy

We now examine the recovery performance over an entire PIN that consists of multiple key pairs.

Table 2: PIN recovery success rate with top-10 candidates.

PIN length	3-digit	4-digit	6-digit	8-digit
Success rate	71.7%	61.7%	43.3%	35%

Impact of PIN length. In this experiment, volunteers are asked to input PINs with lengths varying from 3 to 8 digits. Table 2 shows the recovery success rate with top-10 candidates. As a note, our scheme can produce a list of candidate PINs. If the list of K candidate PINs contain the target PIN entered by the victim, then the correct PIN is deemed among the top-K candidates. This metric reflects the recovery accuracy and has been widely adopted in prior works. We find that the highest success rate 71.7% is achieved for 3-digit PINs. It decreases as PIN length grows, since successive correct inferences of all key pairs are needed to recover the entire PIN. We find the success rate is 43.3% with 6-digit PINs, the mostly commonly PIN length adopted by mobile devices nowadays. Our attack does pose a real threat to these devices.

Impact of the number of candidates. We further study how many candidates are needed to succeed in inferring the target PIN. In the experiments, we sort candidates generated by the HMM model according to their probability of being the target PIN in a descending order and select the top-K candidates to evaluate the recovery accuracy. In Figure 21, we give the PIN inference success rate under top-K candidates, where K ranges from 1 to 100. The result is encouraging. It is shown that, given top-1 candidate, the recovery accuracy is 18.3% for 6-digit PINs. That is, our attack

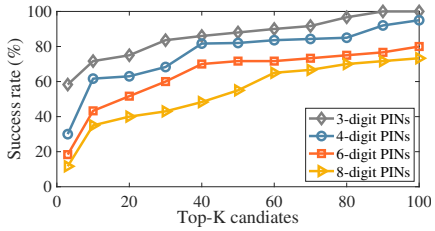
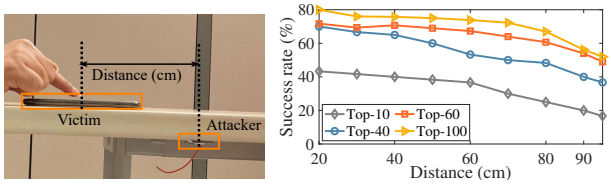


Figure 21: PIN recovery success rate with top-K candidates.

can correctly hit a victim's 6-digit PIN at a probability of 18.3% in one shot. The rate can be significantly improved if given top-10 candidates or top-20 candidates, which corresponds to 43.3% and 51.7%, respectively. As shown in the figure, if given top-40 candidates, the success rate reaches almost 70% for 6-digit PINs. As a comparison, WindTalker [27], a well-cited radio signal based inference attack, delivers a similar performance with more than 60 candidates, not to mention that WindTalker needs a large number of labeled training data samples.

8.3 Performance Under Different Settings

Impact of victim-attacker distances. In practice, a victim device may be placed at different distances away from the attacker. It is thus necessary to examine the impact of this factor to the attack accuracy. In the experiments, we set the distance from 20 cm to 95 cm. All are carried out with 6-digit PINs. As shown in Figure 22, the recovery success rate exhibits negative correlation with the distance. This is because a longer distance leads to weaker EM emanation receptions at the attacker. As a result, it becomes challenging to precisely recover finger movement traces from the EM readings. Still, the attacker can successfully disclose a target PIN at a probability of 20% even 90 cm away from the victim with top-10 candidates. It is worth mentioning that the victim and the attacker reside at two sides of a wood table, a non-LoS scenario shown in Figure 22(a). We anticipate an even higher success rate under a LoS scenario. Besides, we only deploy one prototype in the experiment. In practice, many of them can be used. Such settings can potentially further enhance the performance.



(a) (b)

Figure 22: Impact of victim-attacker distance. (a) Positions of the victim and the attacker. (b) PIN recovery success rate.

Impact of victim-attacker relative direction. We also examine if the relative direction between the victim and the attacker impacts the recovery accuracy. In the first set of experiments, we fix their distance at 20 cm and place the attacker at different directions

to the victim as shown in Figure 23(a). Figure 23(b) shows the PIN recovery success rate at these positions. We find that the accuracy is almost the same for all the cases. In the second set of experiments, the attacker's and victim's positions are fixed while varying the smartphone orientation. Again, no apparent difference in recovery accuracy is observed. Hence, the performance of Periscope is independent of victim-attacker relative direction. This is not the case for radio signal based attacks. Essentially, the attacker needs a LoS view over the victim; otherwise, the signal variance caused by multi-path propagation renders the signal hard to tract. In addition, the video-based attack also imposes stringent requirement over the recording angle. For example, Eyetell [11] experiences about 70% accuracy degradation when the two parties have a 10° displacement angle.

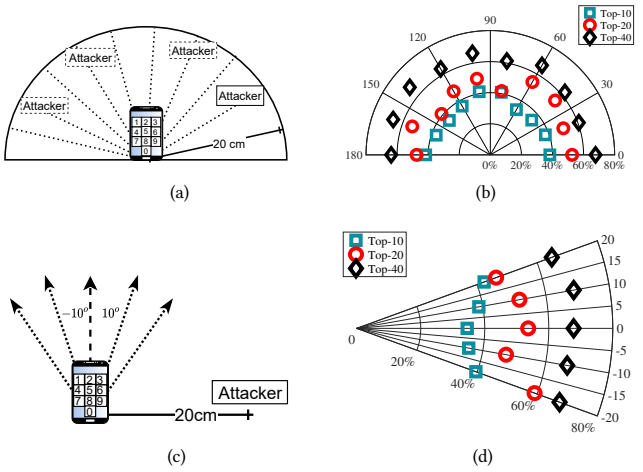


Figure 23: Impact of victim-attacker relative direction. (a) Test scenarios for the first set of experiments. (b) PIN recovery success rate. (c) Test scenarios for the second set of experiments. (d) PIN recovery success rate.

Impact of keyboard layouts. This part evaluates the impact of keyboard layouts on the attack performance. Two layouts, denoted as L1 and L2, are illustrated in Figure 24(b) and 24(c). Their key size is the same, whereas the inter-key distance is different. Figure 24(a) compares their PIN recovery accuracy. We notice that L1 exhibits a higher success rate than L2. This is partly because a larger keyboard leads to more distinct θ's. As a result, finger movement direction can be recognized at higher accuracy which leads to higher overall inference accuracy.

Impact of target diversity. People may have distinct typing behaviors during PIN inputs. Hence, it is critical to find out if Periscope is susceptible to this factor. Table 3 shows the PIN recovery accuracy across seven volunteers. While each individual exhibits a slightly different success rate, the overall performance is relatively consistent, with the average success rate all above 40% with top-10 candidates. It means our analytic model is capable of handling target diversity. As discussed, most acoustic and radio signal based attacks need to train user-specific models to accommodate diverse typing behaviors and is thus less practical for broad deployment.

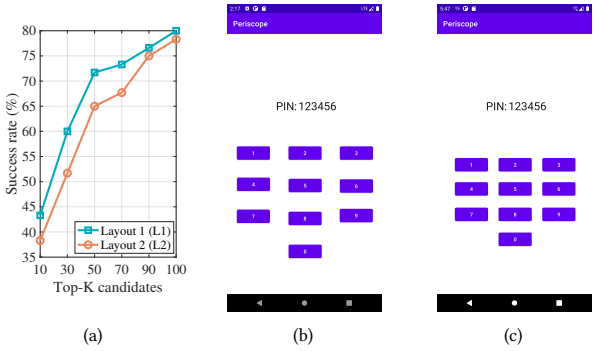


Figure 24: Impact of keyboard layouts. (a) PIN recovery success rate. (b) Layout 1. (c) Layout 2.

Table 3: PIN recovery success rate over different victims.

Index	1	2	3	4	5	6	7
Top-10	52.7%	46.7%	42.5%	50.1%	40.2%	40.6%	45.6%
Top-40	68.9%	72.1%	69.5%	74.1%	63.4%	67.8%	62.2%

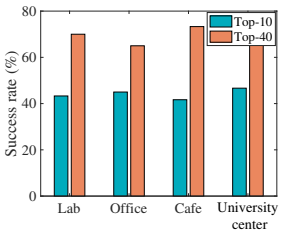


Figure 25: Impact of environmental context.

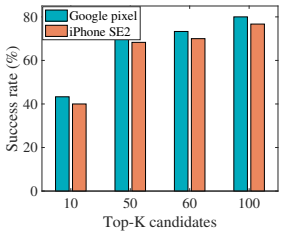


Figure 26: Impact of device diversity.

Impact of environmental context. We further evaluate the attack performance in four different environments, including lab, office, coffee shop, and university center. Figure 25 exhibits a promising performance in all the environments, no matter whether it is quiet or noisy, static or dynamic. In contrast, most acoustic based attacks can only succeed in quiet places, whereas radio signal based attacks do not work with dynamic backgrounds. In fact, public places tend to be noisy and dynamic.

Impact of different devices. To demonstrate the usability of Periscope, we also experimented on two smartphones, an iPhone SE2 with a 4.7-inch touchscreen and a Google Pixel phone with a 6-inch touchscreen. Figure 26 compares their PIN recovery accuracy. The performance is similar for both devices. It means our attack works for a diverse set of devices as long as they are equipped with a multi-capacitance touchscreen. We also notice that the success rate on Google pixel is slightly higher than that on iPhone SE2. This is attributed to the larger screen size of the former. Finger movement traces are more distinct on a larger screen.

We choose iPhoneSE2 and GooglePixel to represent iOS and Android smartphones, respectively. Periscope is effective for devices with capacitive touchscreens regardless of their materials/types. The screen materials/types could contribute to the finger-touchscreen capacitance (C_f). As discussed in Section 4.4, such a factor does not

impact the analytic model of $z(t)$. Hence, screen size dominants the influence from different screens—finger movement traces are more distinct on a larger screen.

8.4 Comparison with Other Schemes

In this part, we present the performance comparison with prior keystroke inference attacks on digit PINs. For the sake of fairness, we directly utilize the experimental results from these works. Three schemes WindTalker [27], SpiderMon [28], and the attack proposed by Liu et al. [30] are considered. Specifically, WindTalker measures the fluctuations of WiFi channels caused by victim's typing motions. SpiderMon utilizes variations of multi-path LTE signals to infer victim's inputs. Liu et al. [30] analyzed the motion status of smartwatches to launch the attack.

Figure 27(a) compares their success rates of recovering 6-digit PINs. It is observed that the performance of Periscope is similar to WindTalker and Liu et al. with top-10 candidates, while SpiderMon is the best. All their success rates reach 80% with top-100 candidates. Note that all other three schemes need a training phase. Large amounts of training samples should be collected from the target victim in advance of the attack. To do this, Liu et al. even require the pre-installation of malware on the victim's smartwatch for sample collection. These restrictions make them hardly practical in real-world scenarios. In contrast, Periscope only requires a couple of samples to parameterize its analytic model and is non-intrusive.

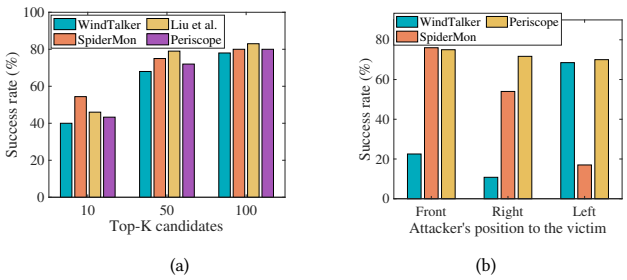


Figure 27: Performance comparison with other schemes. (a) Recovery rate of 6-digit PINs. (b) Impact of attacker's position.

Figure 27(b) compares their performance under the impact of victim-attacker relative direction. We find that both SpiderMon and WindTalker experience significant performance variance by placing the attacker in different directions with respect to the victim. In contrast, the success rate of Periscope is relatively stable regardless of the displacement. This is because the former two extract wireless channel disturbances to monitor finger movements. They heavily rely on LoS propagation channels as they provide the most tractable signals. These channels can be easily blocked by the victim's body if positioning the attacker to the left/right of the victim. On the other hand, as discussed in Section 4.4, the analytic model of Periscope is irrelevant of surrounding environmental conditions. Its performance is thus independent of victim-attacker relative direction.

The main advantage of Periscope lies in its practicality. Specifically, it does not require the pre-installation of malware on the victim's device, nor large amounts of labeled training samples. It is also robust to environmental dynamics. As indicated by the results above, Periscope's accuracy performance is as good as prior inference attacks.

9 DISCUSSIONS

9.1 Limitations and Future Work

Extending attack distance. Results in Section 8.3 show that the PIN recovery rate with top-10 candidates drops to 20% when the attack distance is beyond 90 cm. The threat can be more severe if the attack can be successfully performed remotely. The primary reason of the confined distance here is the weak signal strength of EM leakage. Besides, the EM field decays quickly over distance. Note that our prototype is built with simple electronic pieces, including an Arduino nano board and a conductive wire. Neither advanced transceiver module nor sophisticated signal processing unit is utilized. As our future work, we plan to build a more powerful prototype with dedicated components that can pick up useful signals from noisy and weak EM measurements so as to extend attack distance.

Recognizing letters. Our discussion has been focused on soft numeric keyboards. We plan to extend Periscope to recognize letter inputs. The challenge is to distinguish subtle EM emanations from more diverse combinations of key pairs, as the number of keys will almost be tripled. We propose to employ multiple eavesdroppers and explore their collaboration to launch attacks. *Sensor fusion* techniques [15, 33] will be applied. It combines EM readings from disparate sources such that the resulting information has less ambiguity than would be possible when these sources were used individually. It is expected that the aggregated EM measurement will provide more fine-granular recognition of finger movements.

9.2 Defense Solutions

Periscope explores human-coupled EM emanations to recover victim's inputs on soft keyboards. An intuitive defense solution is thus to adopt shuffled keyboards. This idea has been proposed before [44]; the system adopts a new randomly generated keyboard layout each time a user intends to enter a credential. Although attackers can still derive finger movement traces, they can be hardly mapped to specific keystrokes without the knowledge of keyboard layout. While leaving the key inference almost impossible, as pointed by [60], this idea sacrifices the authentication usability. Extra effort is incurred to the user in searching for keys on a shuffled keyboard. More input errors might also be introduced thereby.

In a more practical way, users may intentionally disrupt their typing behaviors, for instance, adding random pauses between keystrokes and/or adopting variant typing speeds when entering different key pairs. For both cases, attackers will tend to make mistakes in transforming time-dependent finger-screen distances to 3D finger movement traces. For the former, the trace length will appear much longer than the ground truth. For the latter, as a user adopts a dynamic speed, it is impossible for an attacker to generate a meaningful finger movement trace with v_p, a constant finger movement speed that is estimated in advance the attack. As a result, the derived L_x and L_y become error-prone in both cases. The attacker is less likely to accurately recover individual key pairs, let alone the whole PIN.

It is also possible to apply electromagnetic interference (EMI) shielding on touchscreens. This technique has been widely employed on many electronic devices; it refers to the shielding of radio waves so that radiations cannot penetrate the shield. In our case, it can serve as a barrier that prevents EM emanation leakage, or at least reduces the radiation strength. Nonetheless, this approach may be expensive and require hardware modifications, including the introduction of new EMI materials and touchscreen circuit redesign. Another alternative is to intentionally obfuscate the EM emanations emitted by the touchscreen, so that the trajectories of EM readings are not recognizable. A straightforward approach is to add well-calibrated noise to the touchscreen driving signal $V_{TX}(t)$. Then attacker's EM measurements $V_m(t)$ become polluted. Since the attacker is unaware of the injected noise pattern, it is hard to tell if observed EM variations are incurred by finger movements or intentionally injected noise.

10 CONCLUSION

In this paper, we present Periscope, a new eavesdropping attack that leverages human-coupled EM emanations from touchscreens to infer victims' typing inputs at a remote distance. We implemented the proposed attack with a prototype that costs less than $10. Its effectiveness is evaluated from various aspects. Periscope exhibits promising recovery accuracy over a distance up to 90 cm. It can well adapt to diverse device models and setting contexts. Compared with prior works, our approach is built on an analytic model that characterizes the relationship between EM measurements and finger movement traces. Therefore, it avoids the collection of large amounts of labeled data samples in advance of the attack. In summary, we believe that Periscope outperforms state-of-the-art keystroke inference attacks, especially in terms of practicality. Meanwhile, it can be further extended with longer attack distance and inference over letter inputs, which are deemed as our future work.

ACKNOWLEDGMENTS

We sincerely thank the anonymous reviewers for their insightful comments and valuable suggestions. The work is partially supported by NSF CNS-1943509.

REFERENCES

[1] Kamran Ali, Alex X Liu, Wei Wang, and Muhammad Shahzad. 2015. Keystroke recognition using wifi signals. In *Proceedings of the Annual International Conference on Mobile computing and networking*.
[2] Suleyman AlShowarah. 2017. The Effectiveness of Dynamic Features of Finger Based Gestures on Smartphones' Touchscreens for User Identification. *International Journal of Interactive Mobile Technologies (iJIM)* 11, 1 (2017), 133–142.
[3] Arduino. 2020. Arduino nano. (2020). https://www.arduino.cc
[4] Dmitri Asonov and Rakesh Agrawal. 2004. Keyboard acoustic emanations. In *Proceedings of the IEEE Symposium on Security and Privacy*.
[5] Michael Backes, Tongbo Chen, Markus Duermuth, Hendrik PA Lensch, and Martin Welk. Tempest in a teapot: Compromising reflections revisited. In *Proceedings of the IEEE Symposium on Security and Privacy*.
[6] Michael Backes, Markus Dürmuth, and Dominique Unruh. 2008. Compromising reflections-or-how to read LCD monitors around the corner. In *Proceedings of the IEEE Symposium on Security and Privacy*.
[7] Yigael Berger, Avishai Wool, and Arie Yeredor. 2006. Dictionary attacks using keyboard acoustic emanations. In *Proceedings of the ACM SIGSAC Conference on Computer and Communications Security*.
[8] Eli Billauer. 2020. Peak detection algorithm. (2020). http://billauer.co.il/peakdet.html
[9] Liang Cai and Hao Chen. 2011. TouchLogger: Inferring Keystrokes on Touch Screen from Smartphone Motion. In *Proceedings of the USENIX Summit on Hot Topics in Security*.
[10] Jen-Shih Chang, Arnold J Kelly, and Joseph M Crowley. 1995. *Handbook of electrostatic processes*. CRC Press.
[11] Yimin Chen, Tao Li, Rui Zhang, Yanchao Zhang, and Terri Hedgpeth. 2018. Eyetell: Video-assisted touchscreen keystroke inference from eye movements. In *Proceedings of the IEEE Symposium on Security and Privacy*.
[12] Moumita Dey, Alireza Nazari, Alenka Zajic, and Milos Prvulovic. Emprof: Memory profiling via em-emanation in iot and hand-held devices. In *Proceedings of the IEEE/ACM International Symposium on Microarchitecture*.
[13] Dnschecker.org. 2020. MAC Address Lookup. (2020). https://dnschecker.org/mac-lookup.php
[14] YuLei Du, YingHua Lu, and JinLing Zhang. 2013. Novel method to detect and recover the keystrokes of ps/2 keyboard. *Progress In Electromagnetics Research* 41 (2013), 151–161.
[15] Wilfried Elmenreich. 2002. An introduction to sensor fusion. *Vienna University of Technology, Austria* 502 (2002), 1–28.
[16] Song Fang, Ian Markwood, Yao Liu, Shangqing Zhao, Zhuo Lu, and Haojin Zhu. 2018. No training hurdles: Fast training-agnostic attacks to infer your typing. In *Proceedings of the ACM SIGSAC Conference on Computer and Communications Security*.
[17] Tao Feng, Ziyi Liu, Kyeong-An Kwon, Weidong Shi, Bogdan Carbunar, Yifei Jiang, and Nhung Nguyen. 2012. Continuous mobile authentication using touchscreen gestures. In *Proceedings of the IEEE Conference on Technologies for Homeland Security*.
[18] Denis Foo Kune and Yongdae Kim. 2010. Timing attacks on pin input devices. In *Proceedings of the ACM SIGSAC Conference on Computer and Communications Security*.
[19] Fujitsu. 2021. Capacitive Touch Sensors: application fields, technology overview and implementation example. (2021). https://www.fujitsu.com/downloads/MICRO/fme/articles/fujitsu-whitepaper-capacitive-touch-sensors.pdf
[20] Daniel Genkin, Adi Shamir, and Eran Tromer. 2014. RSA key extraction via low-bandwidth acoustic cryptanalysis. In *Proceedings of the Cryptology Conference*.
[21] Daniel Genkin, Adi Shamir, and Eran Tromer. 2017. Acoustic cryptanalysis. *Journal of Cryptology* 30, 2 (2017), 392–443.
[22] Chris Hoffman. 2020. How to See Who's Connected to Your Wi-Fi Network. (2020). hhttps://www.howtogeek.com/204057/how-to-see-who%E2%80%99s-connected-to-your-wi-fi-network/
[23] Leander Kahney. 2019. Your iPhone could be 'unbreakable,' if it were just 1 mm thicker. (2019). https://www.cultofmac.com/624356/your-iphone-could-be-unbreakable-if-it-were-just-1mm-thicker/
[24] Markus Guenther Kuhn. 2002. *Compromising emanations: eavesdropping risks of computer displays*. Ph.D. Dissertation. Citeseer.
[25] Kyuwon Kyoung and Reiji Hattori. 2014. Electromagnetic field analysis of capacitive touch panels. *Journal of Information Display* 15, 3 (2014), 145–155.
[26] Chang-Ju Lee, Jong Kang Park, Canxing Piao, Han-Eol Seo, Jaehyuk Choi, and Jung-Hoon Chun. 2018. Mutual capacitive sensing touch screen controller for ultrathin display with extended signal passband using negative capacitance. *Sensors* 18, 11 (2018), 3637.
[27] Mengyuan Li, Yan Meng, Junyi Liu, Haojin Zhu, Xiaohui Liang, Yao Liu, and Na Ruan. 2016. When CSI meets public WiFi: Inferring your mobile phone password via WiFi signals. In *Proceedings of the ACM SIGSAC Conference on Computer and Communications Security*.
[28] Kang Ling, Yuntang Liu, Ke Sun, Wei Wang, Lei Xie, and Qing Gu. 2020. SpiderMon: Towards Using Cell Towers as Illuminating Sources for Keystroke Monitoring. In *Proceedings of the IEEE Conference on Computer Communications*.
[29] Jian Liu, Yan Wang, Gorkem Kar, Yingying Chen, Jie Yang, and Marco Gruteser. 2015. Snooping keystrokes with mm-level audio ranging on a single phone. In *Proceedings of the International Conference on Mobile Computing and Networking*.
[30] Xiangyu Liu, Zhe Zhou, Wenrui Diao, Zhou Li, and Kehuan Zhang. 2015. When good becomes evil: Keystroke inference with smartwatch. In *Proceedings of the ACM SIGSAC Conference on Computer and Communications Security*.
[31] Zhuoran Liu, Niels Samwel, Léo Weissbart, Zhengyu Zhao, Dirk Lauret, Lejla Batina, and Martha Larson. 2020. Screen Gleaning: A Screen Reading TEMPEST Attack on Mobile Devices Exploiting an Electromagnetic Side Channel. In *Proceedings of the Network and Distributed System Security Symposium*.
[32] Li Lu, Jiadi Yu, Yingying Chen, Yanmin Zhu, Xiangyu Xu, Guangtao Xue, and Minglu Li. 2019. KeyListener: Inferring keystrokes on QWERTY keyboard of touch screen through acoustic signals. In *Proceedings of the IEEE Conference on Computer Communications*.
[33] Anindya Maiti, Oscar Armbruster, Murtuza Jadliwala, and Jibo He. 2016. Smartwatch-based keystroke inference attacks and context-aware protection mechanisms. In *Proceedings of the ACM on Asia Conference on Computer and Communications Security*.
[34] Seita Maruyama, Satohiro Wakabayashi, and Tatsuya Mori. 2019. Tap'n Ghost: A Compilation of Novel Attack Techniques against Smartphone Touchscreens. In *Proceedings of the IEEE Symposium on Security and Privacy*.
[35] Mathworks. 2021. Envelope Extraction. (2021). https://www.mathworks.com/help/signal/ug/envelope-extraction-using-the-analytic-signal.html

[36] Emiliano Miluzzo, Alexander Varshavsky, Suhrid Balakrishnan, and Romit Roy Choudhury. 2012. Tapprints: your finger taps have fingerprints. In *Proceedings of the International Conference on Mobile Systems, Applications, and Services*.

[37] Fan Mo, Ying-Hua Lu, Jin-Ling Zhang, Qiang Cui, and Sihai Qiu. 2013. A support vector machine for identification of monitors based on their unintended electromagnetic emanation. *Progress In Electromagnetics Research* 30 (2013), 211–224.

[38] Nirsoft. 2020. Who is connected sniffer. (2020). https://www.nirsoft.net/utils/who_is_connected_sniffer.html

[39] Emmanuel Owusu, Jun Han, Sauvik Das, Adrian Perrig, and Joy Zhang. 2012. Accessory: password inference using accelerometers on smartphones. In *Proceedings of the Workshop on Mobile Computing Systems & Applications*.

[40] Mickael Pruvost, Wilbert J Smit, Cécile Monteux, Philippe Poulin, and Annie Colin. 2019. Polymeric foams for flexible and highly sensitive low-pressure capacitive sensors. *npj Flexible Electronics* 3, 1 (2019), 1–6.

[41] Lawrence R Rabiner. 1989. A tutorial on hidden Markov models and selected applications in speech recognition. *Proc. IEEE* 77, 2 (1989), 257–286.

[42] Rahul Raguram, Andrew M White, Dibyendusekhar Goswami, Fabian Monrose, and Jan-Michael Frahm. 2011. iSpy: automatic reconstruction of typed input from compromising reflections. In *Proceedings of the ACM SIGSAC Conference on Computer and Communications Security*.

[43] J Patrick Reilly. 2012. *Applied bioelectricity: from electrical stimulation to electropathology*. Springer Science & Business Media.

[44] Daniel Schneider, Alexander Otte, Travis Gesslein, Philipp Gagel, Bastian Kuth, Mohamad Shahm Damlakhi, Oliver Dietz, Eyal Ofek, Michel Pahud, Per Ola Kristensson, et al. 2019. Reconviguration: Reconfiguring physical keyboards in virtual reality. *IEEE transactions on visualization and computer graphics* 25, 11 (2019), 3190–3201.

[45] Diksha Shukla, Rajesh Kumar, Abdul Serwadda, and Vir V Phoha. 2014. Beware, your hands reveal your secrets!. In *Proceedings of the ACM SIGSAC Conference on Computer and Communications Security*.

[46] Ilia Shumailov, Laurent Simon, Jeff Yan, and Ross Anderson. 2019. Hearing your touch: A new acoustic side channel on smartphones. *arXiv preprint arXiv:1903.11137* (2019).

[47] Jingchao Sun, Xiaocong Jin, Yimin Chen, Jinxue Zhang, Yanchao Zhang, and Rui Zhang. 2016. VISIBLE: Video-Assisted Keystroke Inference from Tablet Backside Motion.. In *Proceedings of the Network and Distributed System Security Symposium*.

[48] Unite4buy. 2021. iPhone SE specification. (2021). https://unite4buy.com/Apple-iPhone-SE-2020-3-128Gb-buy/

[49] Wim Van Eck. 1985. Electromagnetic radiation from video display units: An eavesdropping risk? *Computers & Security* 4, 4 (1985), 269–286.

[50] Tam Vu, Akash Baid, Simon Gao, Marco Gruteser, Richard Howard, Janne Lindqvist, Predrag Spasojevic, and Jeffrey Walling. 2013. Capacitive touch communication: A technique to input data through devices' touch screen. *IEEE Transactions on Mobile Computing* 13, 1 (2013), 4–19.

[51] Martin Vuagnoux and Sylvain Pasini. 2009. Compromising Electromagnetic Emanations of Wired and Wireless Keyboards.. In *Proceedings of the USENIX Security Symposium*.

[52] Junjue Wang, Kaichen Zhao, Xinyu Zhang, and Chunyi Peng. 2014. Ubiquitous keyboard for small mobile devices: harnessing multipath fading for fine-grained keystroke localization. In *Proceedings of the International Conference on Mobile Systems, Applications, and Services*.

[53] Litao Wang and Bin Yu. 2011. Analysis and measurement on the electromagnetic compromising emanations of computer keyboards. In *Proceedings of the International Conference on Computational Intelligence and Security*.

[54] Yao Wang, Wandong Cai, Tao Gu, and Wei Shao. 2020. Your Eyes Reveal Your Secrets: An Eye Movement Based Password Inference on Smartphone. *IEEE Transactions on Mobile Computing* 19, 11 (2020), 2714–2730.

[55] Yao Wang, Wandong Cai, Tao Gu, Wei Shao, Ibrahim Khalil, and Xianghua Xu. 2018. GazeRevealer: Inferring password using smartphone front camera. In *Proceedings of the International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services*.

[56] Yi Xu, Jared Heinly, Andrew M White, Fabian Monrose, and Jan-Michael Frahm. 2013. Seeing double: Reconstructing obscured typed input from repeated compromising reflections. In *Proceedings of the ACM SIGSAC Conference on Computer and Communications Security*.

[57] Zhi Xu, Kun Bai, and Sencun Zhu. 2012. Taplogger: Inferring user inputs on smartphone touchscreens using on-board motion sensors. In *Proceedings of the ACM conference on Security and Privacy in Wireless and Mobile Networks*.

[58] Zhenyu Yan, Qun Song, Rui Tan, Yang Li, and Adams Wai Kin Kong. 2019. Towards touch-to-access device authentication using induced body electric potentials. In *Proceedings of the International Conference on Mobile Computing and Networking*.

[59] Yanli Yang. 2017. A signal theoretic approach for envelope analysis of real-valued signals. *IEEE Access* 5 (2017), 5623–5630.

[60] Qinggang Yue, Zhen Ling, Xinwen Fu, Benyuan Liu, Kui Ren, and Wei Zhao. 2014. Blind recognition of touched keys on mobile devices. In *Proceedings of the ACM SIGSAC Conference on Computer and Communications Security*.

[61] Jie Zhang, Xiaolong Zheng, Zhanyong Tang, Tianzhang Xing, Xiaojiang Chen, Dingyi Fang, Rong Li, Xiaoqing Gong, and Feng Chen. 2016. Privacy leakage in mobile sensing: Your unlock passwords can be leaked through wireless hotspot functionality. *Mobile Information Systems* 2016 (2016).

[62] Tong Zhu, Qiang Ma, Shanfeng Zhang, and Yunhao Liu. 2014. Context-free attacks using keyboard acoustic emanations. In *Proceedings of the ACM SIGSAC Conference on Computer and Communications Security*.

[63] Li Zhuang, Feng Zhou, and J Doug Tygar. 2009. Keyboard acoustic emanations revisited. *Transactions on Information and System Security* 13, 1 (2009), 1–26.

[64] Anya Zhukova. 2020. How To See Who Is Connected To My WiFi. (2020). https://helpdeskgeek.com/how-to/determine-computers-connected-to-wireless-network/

A APPENDIX

Table 4: Comparison with state-of-the-art inference attacks.

Signal	System	Keyboard type	Pre-install malware	Environment adaptability	Hardware costs	Attack distance	LoS required	Training required	Setup difficulty
Motion status	TouchLogger [9]	Soft	Yes	-	Low	None	-	Yes	High
	Taplogger [57]	Soft	Yes	-	Low	None	-	Yes	High
	Mait et al. [33]	Physical	Yes	-	Low	None	-	Yes	High
	Liu et al. [30]	Physical	Yes	-	Low	None	-	Yes	High
	Accessory [39]	Soft	Yes	-	Low	None	-	Yes	High
	Tapprints [36]	Soft	Yes	-	Low	None	-	Yes	High
Audio	Berger et al. [7]	Physical	No	Low	Medium	Medium	No	No	Medium
	Kune et al. [18]	Physical & Soft	No	Low	Medium	Medium	No	Yes	Medium
	Zhu et al. [62]	Physical	No	Low	Medium	Short	Yes	No	High
	KeyListener [32]	Soft	No	Low	Medium	Short	Yes	No	High
	Liu et al. [29]	Physical	No	Low	Medium	Short	Yes	No	High
	Ubik [52]	Physical	No	Low	Medium	Short	Yes	Yes	High
	Agrawal et al. [4]	Physical	No	Low	Medium	Medium	Yes	Yes	Medium
	Zhuang et al. [63]	Physical	No	Low	Medium	Medium	Yes	Yes	Medium
	Shumailov et al. [46]	Soft	Yes	Low	Low	None	Yes	No	High
Video	Shukla et al. [45]	Soft	No	Low	Medium	Long	Yes	Yes	High
	Seeing double [56]	Soft	No	Low	High	Long	Yes	Yes	High
	GazeRevealer [55]	Soft	Yes	Low	Low	Medium	Yes	Yes	High
	EyeTell [11]	Soft	No	Low	High	Medium	Yes	Yes	High
	Wang et al. [54]	Soft	Yes	Low	Low	Medium	Yes	Yes	High
	VISIBLE [47]	Soft	No	Low	High	Medium	Yes	Yes	High
	Backes et al. [5]	-	No	Low	Medium	Long	Yes	No	Medium
	iSpy [42]	Soft	No	Low	Medium	Long	Yes	Yes	Medium
	Backes et al. [6]	-	No	Low	Medium	Long	Yes	No	Medium
RF	Li et al. [27]	Soft	No	Low	Medium	Long	No	Yes	Medium
	Ali et al. [1]	Physical	No	Low	Medium	Medium	No	Yes	Medium
	Zhang et al. [61]	Soft	No	Low	Medium	Long	No	Yes	Medium
	Fang et al. [16]	Physical	No	Low	High	Long	No	No	Medium
	SpiderMon [28]	Physical	No	Low	Medium	Long	No	Yes	Medium
EM	Li et al. [51]	Physical	No	High	High	Long	No	No	Medium
	Wang et al. [53]	Physical	No	High	High	Long	No	No	Medium
	Du et al. [14]	Physical	No	High	High	Long	No	No	Medium
Human -coupled EM	Periscope	Soft	No	High	Low	Medium	No	No	Low