iLock: Immediate and Automatic Locking of Mobile Devices against Data Theft

Tao Li
Arizona State University
tli@asu.edu

Yimin Chen
Arizona State University
ymchen@asu.edu

Jingchao Sun
Arizona State University
jcsun@asu.edu

Xiaocong Jin
Arizona State University
xcjin@asu.edu

Yanchao Zhang
Arizona State University
yczhang@asu.edu

ABSTRACT

Mobile device losses and thefts are skyrocketing. The sensitive data hosted on a lost/stolen device are fully exposed to the adversary. Although password-based authentication mechanisms are available on mobile devices, many users reportedly do not use them, and a device may be lost/stolen while in the unlocked mode. This paper presents the design and evaluation of iLock, a secure and usable defense against data theft on a lost/stolen mobile device. iLock automatically, quickly, and accurately recognizes the user's physical separation from his/her device by detecting and analyzing the changes in wireless signals. Once significant physical separation is detected, the device is immediately locked to prevent data theft. iLock relies on acoustic signals and requires at least one speaker and one microphone that are available on most COTS (commodity-off-the-shelf) mobile devices. Extensive experiments on Samsung Galaxy S5 show that iLock can lock the device with negligible false positives and negatives.

CCS Concepts

•Human-centered computing → Mobile devices; •Security and privacy → *Mobile and wireless security;*

Keywords

Device locking, FMCW, audio ranging, smartphone security

1. INTRODUCTION

The human society is in a wireless and mobile era. According to the Cisco Virtual Networking Index [2], 497 million mobile devices (mainly tablets, smartphones, and laptops) were added in 2014, and the number of global mobile devices in 2014 reached 7.4 billion and will reach 11.5 billion by 2019 at a CAGR of 9%. People are using mobile devices in every aspect of life, including work, education, voice/video communications, Internet browsing, web transactions, online banking, reading, multimedia playing, etc.

Mobile device losses/thefts are skyrocketing and posing severe threats to data security. According to a 2012 Kensington study [1], one laptop is stolen every 53 seconds; 70 million smartphones are lost each year, with only 7% recovered; and 4.3% of company-issued smartphones are lost/stolen every year. The true cost of a lost/stolen mobile device goes far beyond the device cost due to the lost productivity, the loss of intellectual property, data breaches, and legal fees.

The most common defense against device losses/thefts is to set a password on the mobile device. Unfortunately, the 2015 Kaspersky Lab survey [4] shows that 31% of smartphones and 41% of tablets are not password-protected. In addition, the time window for a password-protected device going from the unlocked mode to the locked mode may be long enough for a capable attacker to access all the sensitive information on the lost/stolen device. For example, the auto-lock options on iPad 2 include 2 min, 5 min, 10 min, 15 min, and NEVER. Many users choose a longer time period or even NEVER for convenience. If an unlocked device is lost/stolen, the user's sensitive information is fully accessible to whoever possesses the device.

Continuous authentication aims to continuously verify the identity of the user using a mobile device and is naturally a candidate defense against device losses/thefts. This line of work aims to verify the behavioral biometrics of the user exhibited in his keystrokes [18], finger touches on the screen [14], or app usage [12]. In addition to their relatively high false positives and negatives, these approaches often require a relatively long time window to collect sufficient data for capturing the behavioral biometrics. The attacker, however, may quickly access the user's private data and then completely wipe out the device for reinstallation, rather than using the device for an extended period of time.

In this paper, we present iLock, a secure and usable defense against device losses/thefts. iLock immediately and automatically locks a mobile device once it leaves the vicinity of its user. The key motivation behind iLock is that the departure of a user from his device causes the physical environment to change and thus noticeable changes in nearby wireless signals. So we can let the mobile device automatically, quickly, and accurately recognize its physical separation from its owner by detecting and analyzing the changes in wireless signals. Once significant physical separation from its user is detected, the device can immediately and automatically lock itself. iLock cannot help retrieve a

CCS '16, October 24–28, 2016, Vienna, Austria.

DOI: http://dx.doi.org/10.1145/2976749.2978294

lost/stolen device, but it can help prevent data theft. Specially, after iLock locks the device, the user can use various apps such as Find My Phone to track the device, remotely disable it, and even completely erase it.

iLock relies on acoustic signals and requires at least one speaker and one microphone that are available on most COTS mobile devices, such as smartphones, tablets, laptops, and all-in-one PCs. Once a user-defined vulnerable context (e.g., out of home) is automatically detected, the speaker keeps transmitting high-frequency acoustic signals inaudible to human ears. The signals are reflected by the user's body and finally reach the microphone after some delay. The device can then estimate its distance from the user based on the received signals and automatically lock itself once the distance estimation exceeds a user-defined threshold.

How could the user-device distance be estimated? One may simply let the speaker transmit an acoustic signal, which reaches the microphone via the speaker-user-microphone path. After computing the time-of-flight (ToF) as the difference between signal transmission and reception time, the device can estimate the user-device distance as $c \times \text{ToF}/2$, where c denotes the speed of sound about 340 m/s. This seemingly simple method unfortunately does not work because of very coarse-grained timestamps on mobile devices, which can be due to many reasons such as various delays between the application and physical layers [23]. For example, an error of 0.01 s may cause a distance-measurement error about 1.7 m which is obviously not acceptable for device locking.

iLock adopts a technique called FMCW (frequency modulated carrier wave) [16] to avoid computing the ToF directly based on inaccurate timestamps on mobile devices. FMCW transforms the time differences to frequency shifts between transmitted and received signals. With FMCW, the speaker changes the acoustic signal frequency linearly. The device computes Δf, the frequency difference between the signal transmitted at the speaker and the signal received by the microphone at the same time. Since the slope of the linear FMCW function is known, the ToF is roughly $\frac{\Delta f}{\text{slope}}$, and the user-device distance can still be estimated as $c * \text{ToF}/2$.

Implementing FMCW-based iLock on COTS mobile devices faces two critical challenges. First, the device must compute the frequency drift Δf as the frequency difference between the signals simultaneously transmitted at the speaker and arriving at the microphone. This seemingly simple requirement is difficult to fulfill on COTS mobile devices because the timestamps obtained from the OS are highly inaccurate. Second, the signal arriving at the microphone is actually a linear combination of multi-path signals coming from the direct speaker-microphone path, the speaker-user-microphone path, and other paths involving many other physical objects. The device thus should be able to separate the signal from the speaker-user-microphone path from other multi-path signals.

Our contributions in this paper are summarized as follows.

- We design iLock, the first system to immediately and automatically lock a COTS mobile device once its physical separation from its owner is significant. iLock can effectively thwart data theft on a lost/stolen mobile device without any user involvement.

- We propose a novel method to implement iLock based on the FMCW technique, which is applicable to almost all COTS mobile devices with at least one speaker and one microphone.

- We implement iLock and conduct extensive experiments on Samsung Galaxy S5 against various attackers. Our evaluation results show that iLock can immediately lock the device with negligible false positives and negatives.

The rest of the paper is organized as follows. Section 2 introduces the adversary model and our design goals. Section 3 details the iLock design. Section 4 presents the experimental evaluations. Section 5 discusses the energy consumption of iLock and other possible solutions. Section 6 briefs the related work. Section 7 concludes this paper.

2. ADVERSARY MODEL AND DESIGN GOALS

Adversary Model. We assume that the mobile device to protect is unlocked. This can be because the auto-lock option is disabled or has not taken effect if a long time window (e.g., 5 min) is chosen. The attacker possesses the device and tries to access sensitive information stored there. We consider three types of attackers according to their initial distance from the device relative to the (legitimate) user.

- Type-I attacker: This kind of attackers find the device the legitimate user accidentally lost in public places such as streets, restrooms, coffee shops, and subways. Type-I attackers are initially much farther away from the device than the user.

- Type-II attacker: Such attackers are still farther away from the device than the user, but the distance difference is very small. For example, the attacker can be a thief trying to steal the device from the user on a crowded bus/subway, and the attacker may also be a malicious coworker who just sat with the user for a meeting and saw the user leave without taking the device on the conference table.

- Type-III attacker: These attackers are closer to the device than the user. For example, the user may accidently put the device closer to the malicious coworker on the conference table and leave the meeting without taking the device.

Since iLock relies on acoustic signal transmissions and receptions, one may think about defeating iLock by letting the attacker jam the acoustic channel. Such jamming attacks are very easy to detect and mitigate. So we focus on dealing with the three types of attackers above.

Design Goals. iLock cannot help retrieve a lost/stolen device, but it can help prevent data theft on a lost/stolen device. We have the following design goals.

- iLock should be *device-free* and does not rely on any auxiliary device. It should also be applicable to most COTS mobile devices.

- iLock should *immediately* lock the device once the user-device distance exceeds a pre-defined threshold to minimize the time opportunity for data theft.

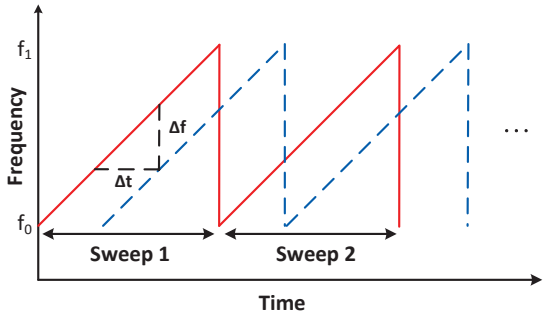


Figure 1: FMCW illustration. The frequency of the transmitted signal (red solid line) repeatedly increases from f_0 to f_1. After a time delay Δt, the signal arrives at the receiver (blue dashed line). The frequency shift Δf can be extracted by performing FFT over each sweep.

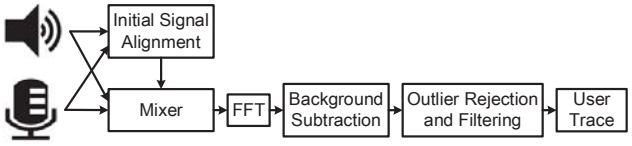


Figure 2: The system framework of iLock.

- iLock should be *automatic* and *user-friendly*. It should not require any explicit interaction between the user and device. Nor does the user's device-use habit need to be changed.
- iLock should be very *accurate* in detecting the user-device distance, which can translate into very low false positives and negatives for triggering device locking.

3. ILOCK

This section details the iLock design. We start by introducing FMCW in Section 3.1. Then we discuss how to defend against Type-I, Type-II, and Type-III attackers in Sections 3.2, 3.3, and 3.4, respectively.

3.1 Frequency-Modulated Carrier Waves

Fig. 1 gives a high-level overview of FMCW, and we refer the reader to [16] for a more detailed illustration. FMCW operations proceed in rounds. In each round referred to as a sweep, the transmitter linearly increases the transmission frequency from f_0 to f_1, where f_0 and f_1 are predetermined minimum and maximum frequencies. Each signal arrives at the receiver after some delay Δt (the so-called ToF). The transmitted and received signal frequencies for each sweep are depicted by red solid and blue dashed lines in Fig. 1, respectively. According to Fig. 1, it is clear that $\Delta t = \frac{\triangle f}{f_1 - f_0} T_{\text{sweep}}$, where T_{sweep} is the duration of each sweep. Finally, we can estimate the signal-travel distance $d = c\Delta t$, where c is the signal propagation speed.

3.2 Defeating Type-I Attackers: When Attackers Are Initially Farawy

iLock relies on FMCW to dynamically estimate the user-device distance and automatically locks the device once the user-defined safe distance is exceeded. iLock uses acoustic signals so that it can work on most COTS mobile devices with standard build-in microphones and speakers. Thus c is the speed of sound of about 340 m/s. The minimum FMCW frequency f_0 is set to be sufficiently high (e.g., 18 kHz) so that the signal is almost inaudible to human ears, and the maximum FMCW frequency f_1 can be set to half the highest sampling frequency of the microphone. For example, most COTS smartphones support the sampling frequency up to 44.1 kHz, so we can set f_1 equal to 22 kHz. T_{sweep} is a design parameter dictating the tradeoff between maximum detection range and frequency drift resolution, which becomes clear shortly.

The implementation of FMCW-based iLock on COTS mobile devices faces two critical challenges. **First**, the device must compute the frequency drift Δf as the frequency difference between the signals simultaneously transmitted at the speaker and arriving at the microphone, as shown in Fig. 1. To do so, the transmitted and received signals for the same sweep should be properly aligned. This seemingly simple goal is difficult to achieve on COTS mobile devices because the timestamps obtained from the OS are highly inaccurate in contrast to the short sweep duration. Specifically, there are many reasons for the skew between the sending timestamp and actual signal-emission time [23]. For example, the transmission instructions have to be transferred from the application layer to the physical layer, which may be delayed by many system events such as system interrupts. Similar reasons can also account for the skew between the receiving timestamp got from the OS and the actual receiving time by the microphone circuit. More accurate time measurements can be obtained from the kernel, but this option is not feasible on mobile devices. **Second**, the signal arriving at the microphone is actually a linear combination of multi-path signals coming from the direct path between the speaker and microphone, the speaker-user-microphone path, and other paths involving many other physical objects. The device thus should be able to separate the signal from the speaker-user-microphone path from other signals.

Below we illustrate how iLock tackles these two challenges with the system diagram in Fig. 2. We assume Type-I attackers in this section such that the signals are reflected by only one human object (the user him/herself).

The Signal Alignment module is designed to deal with the first challenge. Specifically, the speaker transmits acoustic signals with the frequencies sweeping from f_0 to f_1, which arrive at the microphone after some delay. In ideal situations with accurate timestamps and static signal propagation environments, the time gap between transmitted and received signal vectors for the sweep that can be obtained from the transmitted and received timestamps should be constant, as shown in Fig. 1. Such gaps, however, may vary a lot across each sweep mainly due to inaccurate timestamps.

Our design leverages the observation that the physical distance between the speaker and microphone is fixed and usually very short relative to the user-device distance,[1] so the signals arriving from the direct speaker-microphone path dominate other multi-path components. If the sweep duration is so short that signal propagation environments are approximately static, the time gap between transmitted and received signal vectors on the direct path should be constant across each sweep regardless of inaccurate timestamps. Let

[1]For example, the distances of the speaker to two microphones on a Samsung Galaxy S5 are 4.5 cm and 12.3 cm, respectively.

$\sin(f_{\text{tx}}t)$ and $\sin(f_{\text{rx}}t)$ denote the transmitted and received signals at the same timestamp, respectively. The Signal Alignment module computes $\sin(f_{\text{tx}}t)\sin(f_{\text{rx}}t) = \frac{1}{2}(\cos[(f_{\text{tx}} - f_{\text{rx}})t] - \cos[(f_{\text{tx}} + f_{\text{rx}})t])$ and then uses a low-pass filter to get $\cos[(f_{\text{tx}} - f_{\text{rx}})t]$. Then we advance the received signal vector by an offset k to minimize the frequency difference $f_{\text{tx}} - f_{\text{rx}}$. If the microphone only receives the signals from the direct speaker-microphone path, there can be an almost perfect overlap between the transmitted and received signal vectors after the shifting with $f_{\text{tx}} - f_{\text{rx}} \approx 0$. Due to the presence of the user and other physical objects, the transmitted and received signals cannot overlap each other. Finally, the transmitted signals correspond to the red solid line in Fig. 1, and the advanced received signals correspond to the blue dashed line in Fig. 1.

Then the Mixer module is invoked to compute $\cos[(f_{\text{tx}} - f_{\text{rx}})t]$ in the same way as in the Alignment module for the transmitted and received signals at the same instant in the same sweep. Different physical objects lead to different reflection paths, each corresponding to a different time shift. So the FFT module is subsequently used in each sweep to extract these different frequency shifts. Since each frequency shift corresponds to a different ToF measurement and thus a different signal-travel distance, we plot the received signal powers at different distances in Fig. 3a, which are obtained from a microphone on a Samsung Galaxy S5 with $f_0 = 18$ kHz, $f_1 = 22$ kHz, and $T_{\text{sweep}} = 20$ ms. There are many horizontal strips with each corresponding to a different path the signal traveled from the speaker to microphone. Some strips are not stable with time, as user movements change the multi-path propagation environment. The strips around distance zero are the brightest, corresponding to the direct speaker-microphone path.

We then use the Background Substraction module to highlight the effect of user movements. Specifically, the physical objects other than the user (e.g., doors and walls) can be assumed to be static relative to user movements, which generally holds given the very short duration to detect user movements and then lock the device. Therefore, the reflection paths due to these static objects are static across the sweeps, so we can easily remove their effects via subtraction. Fig. 3b shows the substraction result, where the signal power decreases as the distance increases.

Next, we use the Kalman filter in the Outlier Rejection and Filtering module to smooth out the data. Fig. 3c shows the user's movement trace before and after outlier rejection and filtering. In this experiment, the user initially sits on the chair with the smartphone on the table. Then he stands up and turns around to move away from the table and thus his smartphone. As we can see, his distance to the smartphone decreases when he stands up (around 2,000 ms) and increases when he moves away (after 2,000 ms).

When should the device be locked? In everyday life, the device is often placed within the arm's reach, so the user can set a threshold δ_1 about the arm length when installing iLock. We also define another distance threshold δ_2, beyond which the user can hardly put his device. iLock immediately and automatically locks the device when the user-device distance starts below δ_1 and then exceeds δ_2. We set $\delta_1 = 60$ cm and $\delta_2 = 1$ m in the experiments, and the user can freely adjust them in practice.

How accurate are the distance measurements in iLock? The resolution of distance measurements relies on that of ToF measurements which further depends on that of frequency measurements. The minimum frequency drift in iLock equals $1/T_{\text{sweep}}$ (i.e., the size of one FFT bin), which translates into a ToF resolution of $\frac{1/T_{\text{sweep}} * T_{\text{sweep}}}{f_1 - f_0}$. So the user-device distance resolution can be derived as $\frac{c}{2(f_1 - f_0)}$, for which we assume that the user-device distance is half of the speaker-user-microphone path length. With $f_1 = 22$ kHz, $f_0 = 18$ kHz, and $c = 340$ m/s, the user-device distance resolution is about 4.25 cm, which is sufficient to detect the user's significant departure from the device.

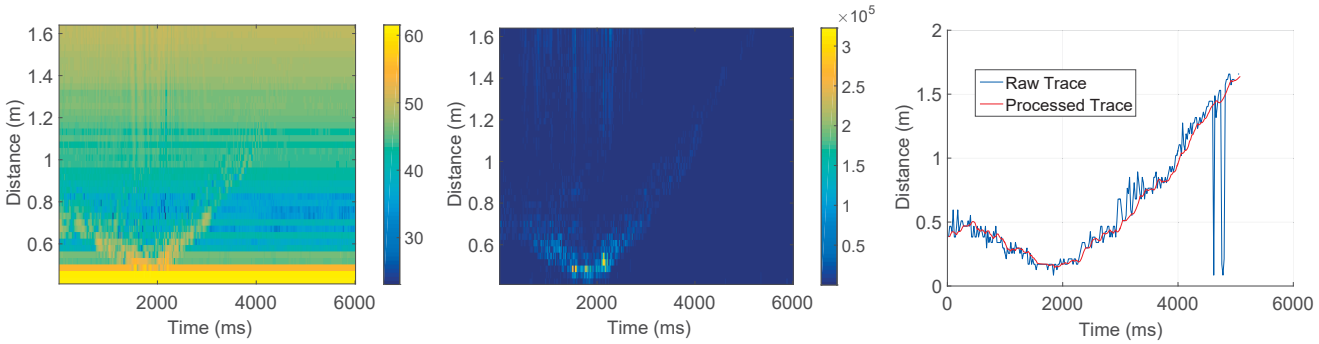
The maximum detection range for the user-device distance depends on both the sweep duration and also the speaker volume. Considering the sweep duration alone, we can compute the maximum user-device distance as $cT_{\text{sweep}}/2$, which equals 3.4 m if $T_{\text{sweep}} = 20$ ms. The speaker volume corresponds to transmission power and thus distance: the larger the speaker volume, the larger the transmission power consumption, the larger the detectable user-device distance, and vice versa. In our experiments, the 71% volume level leads to a maximum detection range at about 1.5 m.

Another issue worth mentioning is the impact of initial signal alignment on distance measurements. The net effect of initial signal alignment is to virtually place the speaker and microphone together. So each subsequent microphone-object-speaker distance measurement is actually $d' = d - d_{\text{sm}}$, where d is the actual signal travel distance, and d_{sm} means the distance between the speaker and microphone. For most portable mobile devices, d_{sm} is relatively small in contrast to user movements and can be safely ignored. For larger mobile devices such as laptops and all-in-one PCs, d_{sm} can be easily estimated and then used to obtain d.

3.3 Defeating Type-II Attackers: When Attackers Get Closer

The basic iLock design in Section 3.2 assumes that the attacker is initially faraway from the device, so only the movement of the user him/herself needs to be tracked. In this section, we discuss how to defeat Type-II attackers which are initially also close to the device but still at a greater distance than the user-device distance. There are many such scenarios in daily life. For example, the user leaves a conference room without taking his/her device on the table, where malicious coworkers or conference attendees try to access sensitive data on the user's device. The device may also slip out of the user's pocket or suitcase on public transport tools and be picked up by malicious passengers nearby. The existence of multiple persons (including the target user) nearby causes the target device to detect multiple movement traces. So the essential challenge is to identify the movement trace associated with the legitimate user, based on which to make salient device-locking decisions.

To begin with, we consider a common scenario that only one person near the device moves away from it. Even if other persons do not move, they may still have minor body movements which can be detected by the device. Since the target user is assumed to be initially closer to his/her device than other persons, his/her movement trace can be easily singled out based on the initial closer distance measurement. Fig. 4 shows an exemplary scenario where the target user leaves but the attacker stays, and Fig. 5 corresponds to the case that the attacker leaves but the user stays. It is very clear that the target user's movement trace can be easily



(a) Spectrogram (b) Spectrogram after Background Subtraction (c) User Leaving Trace

Figure 3: Single user tracking with FMCW. Figure (a) plots the spectrogram after we take FFT on each sweep. Figure (b) eliminates static multipath by subtracting the power of a previous sweep from the current sweep. Figure (c) illustrates the user's moving traces before and after outlier rejection and filtering.

identified, based on which the device can determine whether to lock itself according to the same rules in 3.2.

There can be ambiguity if the user-device distance is not much smaller than the attacker-device distance, especially when there are more than two persons near the device who may leave or stay with the device around the same time. For example, multiple passengers (including/excluding the target user) may exit at the same bus stop. As a result, there can be multiple movement traces corresponding to leaving persons and also multiple ones for staying persons. Leaving traces are easier to be distinguished from staying traces because the latter correspond to relative stable and smaller distances. But the leaving traces themselves may intersect, so may the staying traces themselves. The limited resources on COTS mobile devices make it impossible to accurately identify the movement trace for each individual person. Fortunately, our goal is to preserve data security in the case of device thefts/losses, so it makes more sense to weigh false positives over false negatives. Under the assumption that the target user is initially closer to the device than other persons nearby, we can take an aggressive approach as follows. We first construct a set of candidate leaving traces from the distance measurements. For example, if two persons leave the device with their leaving traces intersecting each other, we can construct four candidate leaving traces. Among the candidate traces satisfying the locking condition (i.e., starting below δ_1 and exceeding δ_2), we select the one whose minimum distance measurement is the smallest, denoted by d_{L}. Similarly, we construct a set of candidate staying traces, from which to select the one whose minimum distance measurement is the smallest, denoted by d_{S}. Let ω denote the maximum possible distance measurement error. As long as $d_{\mathrm{L}} \leq d_{\mathrm{S}} + 2\omega$, iLock associates the leaving trace with the target user and immediately locks the device.

3.4 Defeating Type-III Attackers: When Attackers Are Closer than the User

Now we illustrate how iLock withstands a Type-III attacker, the strongest one who is even closer to the device than its legitimate user (e.g. two scenarios in Fig. 6). Such attack scenarios are not unusual. For example, the user sits very close to the attacker in a conference room and acciden-

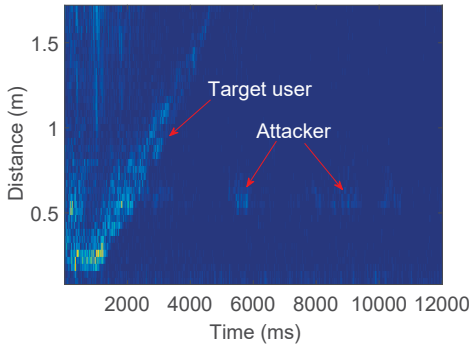


Figure 4: The scenario where the user leaves and attacker stays. The user departs from about 0.2m from the device. The attacker stays at 0.5m from the device with small movements. In this case, the device should be locked.

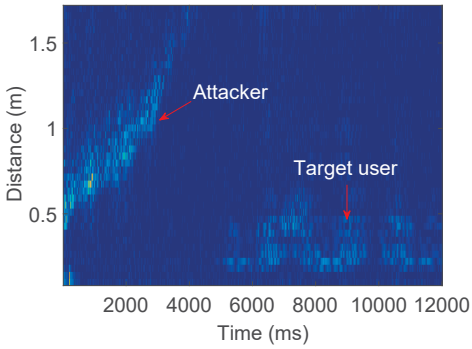


Figure 5: The scenario where the user stays and attacker leaves. The attacker departs from about 0.5m from the device. The user stays at 0.3m from the device with small movements. In this case, the device should not be locked.

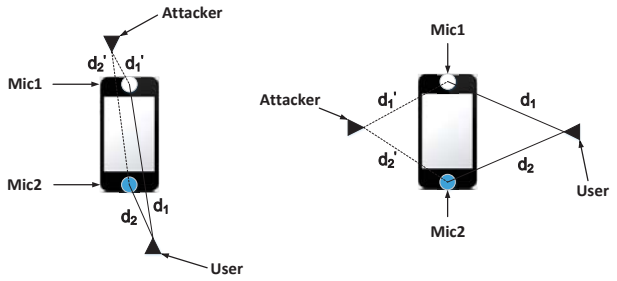


Figure 6: Two scenarios in which the attacker is closer to the device than the target user. In the left scenario, the attacker and target user are easier to be differentiated, because they are closer to Mic1 and Mic2, respectively. In the right scenario, the attacker and target user are difficult to be differentiated, because each of them have same distance to both microphones.

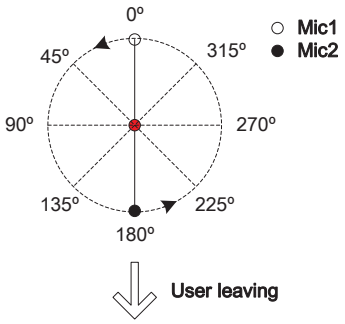


Figure 7: Mic1 is at the top of the phone, and Mic2 is at the bottom. The red center of the circle corresponds to the center of the phone. The phone is rotated around the center with an interval of 45° counterclockwise. We assume the user's leaving direction is fixed.

tally puts the device closer to the attacker. The previous defenses against Type-I and Type-II attackers thus fail.

The fact that more and more COTS mobile devices have two or more microphones enables possible defenses against Type-III attackers. For example, Fig. 6 shows dual microphones on one smartphone, where Mic2 at the bottom is mainly used for voice recording, and Mic1 at the top is designed for noise cancellation. Such dual-microphone configurations are very typical on current smartphones. The left sub-figure in Fig. 6 depicts a scenario where the user and attacker are closer to Mic2 and Mic1, respectively. In this scenario, the user's significant departure from the device can still be identified based on the distance measurements at the two microphones, in which case the device can be immediately locked. In contrast, the right sub-figure in Fig. 6 corresponds to a scenario in which the attacker and target user have similar distance to both microphones. The system will also lock the device immediately to ensure strong data security when there is an ambiguity in the right scenario.

Relying on dual microphones, our solution applies to Type-III attackers with arbitrary locations with regard to the microphones and the user. We additionally assume that the relative orientation changes between the device and user before user movements can be automatically estimated with high precision through existing techniques. For example, the latest result we are aware of [30] can reach a precision of 5° based on IMU sensors. Since the initial relative orientation when the user is using the device is known (i.e., either landscape or portrait mode), we can calculate the final relative orientation when the user stop using the device. As a result, we just need to compare the orientation of candidate leaving user measured by two microphones with the orientation of target user calculated by IMU sensors. We also notice that the relative user-device orientation is approximately fixed, as a normal user typically walks along a straight line with a short distance from the device instead of in a zigzag fashion.

Our solution uses the distance measurements at Mic1 and Mic2 in a cohesive way. Specifically, every moving physical object near the device can lead to a speaker-object-microphone distance measurement at both Mic1 and Mic2 according to the FMCW technique. Let $d_1(t)$ and $d_2(t)$ denote the distance measurements of Mic1 and Mic2 at time t, respectively. Note that COTS devices allow these two measurements to be perfectly aligned in time, i.e., with the same sampling clock. Consecutive distance measurements of the same object at the same microphone lead to a movement trace, either staying or leaving. Since Mic1 and Mic2 are very close to each other on the device in contrast to the user-device distance, they produce highly correlated movement traces for the same object. Assume that iLock finds two such correlated traces, so the next step is to determine whether these leaving traces should be associated with the user and triggers device lock if so. However, the distance measurement isn't accurate and stable enough to discover the orientation of candidate leaving trace, so we introduce a new metric as follows,

$$\eta(t) = \begin{cases} -1 & \text{if } d_1(t) - d_2(t) > \delta_{\text{dual}}, \\ 0 & \text{if } |d_1(t) - d_2(t)| \leq \delta_{\text{dual}}, \\ 1 & \text{if } d_2(t) - d_1(t) > \delta_{\text{dual}}, \end{cases}$$

where δ_{dual} is a system threshold and set to the theoretical distance resolution of 4.25 cm. We proceed to compute $\hat{\eta} = \frac{1}{N}\sum_{t=1}^{N}\eta(t)$, where N denotes the number of distance measurements. Obviously, $\hat{\eta}$ always belongs to $[-1, 1]$. When $\hat{\eta}$ is closer to 1 (-1), the object is closer to Mic1 (Mic2). If $\hat{\eta}$ is closer to 0, the object is about the same distance from Mic1 and Mic2.

We conjecture that $\hat{\eta}$ is closely tied to the device-object orientation and confirm it by experiments on a Samsung Galaxy S5. As shown in Fig.7, we fix the user's moving direction and evaluate $\hat{\eta}$ in eight different orientations (45° separation) by rotating the phone around its fixed center. 20 experiments are done for each orientation, and the distribution of $\hat{\eta}$ is shown in Fig. 8. We can observe that the data for symmetric orientations with regard to the vertical axis (e.g., 225° vs. 135°) overlap. So do the data for adjacent orientations (e.g., 45° vs. 90°). This observation is anticipated due to distance measurement errors and also because $\hat{\eta}$ relates to only relative distance measurements. But there is a clear distinction between the data for orientations far apart (e.g., 0° vs. 180° and 45° vs. 225°).

The above observation can be explored as follows. First, we obtain a more fine-grained $\hat{\eta}$-orientation distribution than that in Fig. 8, which can be device-specific. The obtainment of this distribution is a one-time process and can be done when the user installs and enrolls into iLock. Once two correlated leaving traces are detected, iLock computes $\hat{\eta}$ as

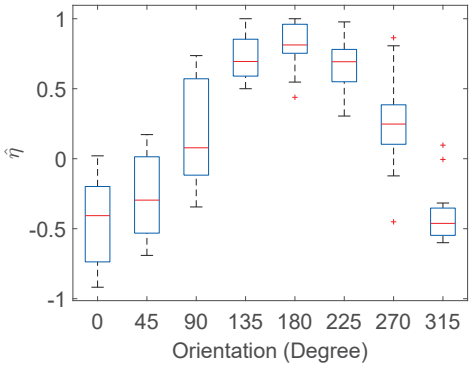


Figure 8: We calculate $\hat{\eta}$ in eight different orientations as illustrated in Fig. 7. The red line is the median, and the bottom and top edges of the box indicate the 25 and 75 percentiles, respectively. The whiskers extend to the most extreme data points not considered as outliers, and the outliers are plotted individually using the '+' symbol.

above, based on which to find the most probable orientation $\hat{\eta}$ corresponds to. If the likelihoods for multiple orientations are sufficiently close, all of them are candidate orientations. Recall that the initial user-device orientation can be precisely obtained beforehand, and the user normally works in the same orientation within a short distance where iLock targets. If any candidate orientation is within a predefined threshold from the initial user-device orientation, the leaving traces are determined to be associated with the legitimate user, so iLock immediately locks the device.

4. IMPLEMENTATION AND EVALUATION

We implement iLock and obtain similar evaluation results in several COTS Android devices such as Samsung Galaxy S5 and Xiaomi Redmi 2. For lack of space, only the experimental data on Samsung Galaxy S5 are reported in this paper. The Samsung Galaxy S5 phone has a Quad-core 2.5 GHz Krait 400 CPU, 2 GB RAM, and a 5.1-inch display. There are also two microphones, Mic1 at the top and Mic2 at the bottom. The speaker-Mic1, speaker-Mic2, and Mic1-Mic2 distances are 4.5 cm, 12.3 cm, and 14 cm, respectively. By default, the FMCW frequencies range from $f_0 = 18$ kHz to $f_1 = 22$ kHz; the sweep duration is $T_{\text{sweep}} = 20$ ms; and the speaker volume is 71%. One experiment is done in the university library, and all the others are done in a typical $12' \times 24'$ research office with desks, cabinets, computers, and six students. Unless specifically noticed, our experiment below is done on a table of 72cm height in our office with the orientation 0°; and the user stands up, turns around, and walks away with normal speed about 1.51 steps/second. Below we report the performance of iLock against Type-I, Type-II, and Type-III attackers, respectively.

4.1 Evaluation with Type-I Attackers

Recall that Type-I attackers are far away from the device when the user moves away. iLock in this scenario just needs to recognize the movement trace of the user alone and then locks the device if the trace starts below the near-distance threshold δ_1 and exceeds the far-distance threshold δ_2. The experiments are conducted in a $12' \times 24'$ office with six PhD students. We set $\delta_1 = 0.6$ m (a typical arm's reach) and $\delta_2 = 1$ m beyond which a typical user does not put the device. In our experiments, a male user uses the phone for a while and then leaves it unlocked on the table, in which case iLock is automatically activated. Note that the triggering events for iLock can be automatically detected by many existing methods, e.g., through detecting when the user stops touching/holding the unlocked phone via inertial gyroscope and accelerometer sensors.

False Negatives. We first evaluate the false-negative rate of iLock through 400 experiments. In each experiment, the user puts his phone in a random position and an arbitrary orientation within δ_1. The user leaves the device in his usual way. As soon as the user-phone distance exceeds 1 m (i.e., δ_2), iLock theoretically should lock the phone. The results are quite encouraging. Specifically, the phone is successfully locked 395 times, which lead to a locking rate (true-positive rate) of 98.75% or a false-negative rate of 1.25%.

False Positives. We then evaluate the false-positive rate of iLock. In this experiment, we put the unlocked phone randomly on the desk just besides the user (within δ_1). Instead of leaving the desk and phone, the user performs regular minor movements such as typing, writing, drinking, rotating his head/shoulder, and swinging back-and-forth. Zero false device locking occurs in the entire 15 minutes, indicating an extremely low false-positive rate in practice.

Impact of Phone Orientations. The next experiment is to investigate the effect of phone orientations. We change the phone's relative orientation to the user by rotating it according to Fig. 7. For each orientation, the user moves away from the phone 50 times in his own way, for which each movement starts from a random position within δ_1 and goes beyond δ_2 from the phone.

Fig. 9 illustrates the maximum detection ranges of Mic1 and Mic2 for different phone orientations. When the phone orientation is around 0° (180°), Mic2 (Mic1) yields a larger maximum detection range due to the closer distance between the user and Mic2 (Mic1). On the Samsung Galaxy S5, Mic2 is the master microphone, and Mic1 is designed for noise cancellation. So we can see that the average maximum detection range of Mic2 is larger than that of Mic1. Finally, combing the distance measurements from Mic1 and Mic2, iLock can always detect the user movement up to 1.4 m for any orientation.

Fig. 10 plots the true-positive rates for each orientation based on Mic1, Mic2, and their combination Mic1+Mic2. As expected, the peak performance for using Mic1 alone and Mic2 alone occur around 180° and 0° orientations, respectively. In addition, Mic2 shows better performance overall due to its higher capability as the master microphone. Finally, if we lock the phone as long as either one microphone detects a leaving trace, the true-positive rate is always above 90% regardless of initial phone orientations.

Impact of Initial Phone Positions. We also evaluate the impact of initial phone positions. In this experiment, the initial phone-user distance changes from 10 cm to 20 cm, 30 cm, 40 cm, and 50 cm, and the phone orientation is fixed to 0°. Fig. 11 and Fig. 12 show the maximum detection ranges and true-positive rates, respectively. We can see that the true-positive rate with Mic2 alone or Mic2 and Mic1 together can yield very high true-positive rates up to 100% for all distance settings. So initial phone positions have very little impact on iLock.

Impact of Departing Gestures. The user may leave the

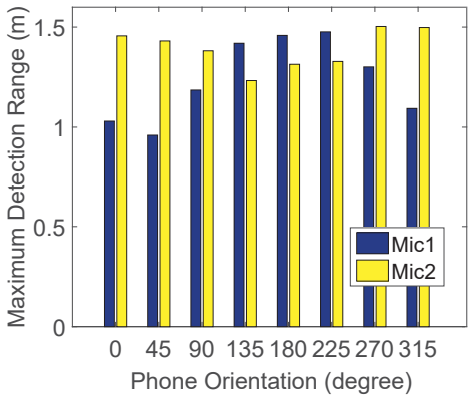


Figure 9: Maximum detection range vs. orientations.

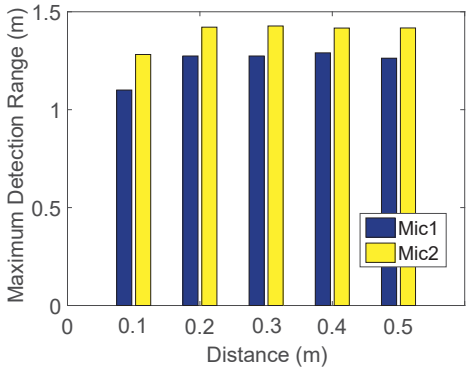


Figure 11: Maximum detection range vs. phone-user distance.

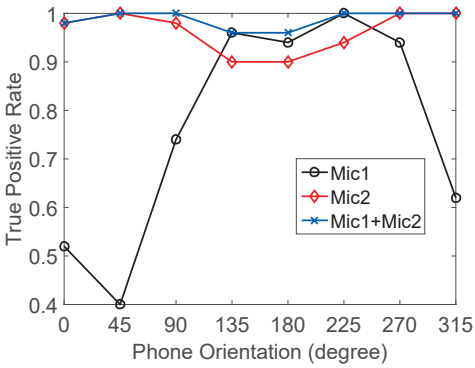


Figure 10: True-positive rates vs. orientations.

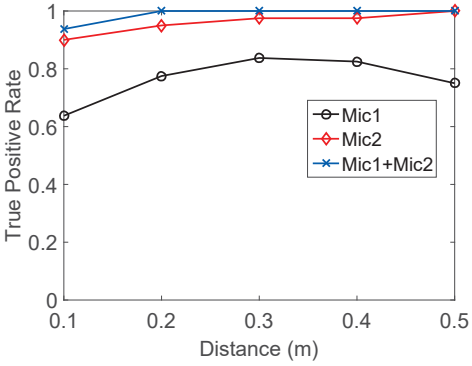


Figure 12: True-positive rate vs. phone-user distance.

device with different gestures. Intuitively speaking, the departing gesture should not affect the detection performance, as iLock only measures the user-device distance. We confirm this intuition by experimenting three common gestures. In the first gesture which is the default in our experiments, the user stands up, turns around, and walks away. In the second gesture, the user initially stands facing the phone and then steps back to leave. In the final gesture, the user rotates the chair, stands up, and then moves away. Each gesture is performed 20 times, and the average maximum detection ranges and true-positive rates are shown in Fig. 13. We can see that Mic2 and Mic1+Mic2 produce very high and stable true-positive rates for all three gestures.

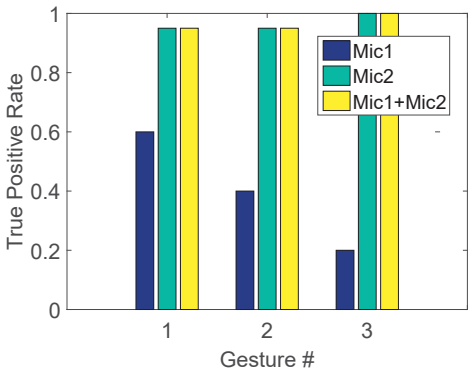
Impact of Departing Speeds. To evaluate the impact of moving speeds, we let the user perform the second gesture above with slow, normal, and fast speeds, corresponding to about 1.15, 1.51, and 2.0 steps/second, respectively. In this experiment, the user leaves 20 times for each speed setting, while the phone is initially 20 cm away at the 0° relative orientation. As we can see from Fig. 14, the performance of iLock becomes non-satisfactory when the user steps back at 2.0 steps/second. The main reason is that the fast speed reduces the time span for the same distance range, which in turn reduces the number of distance measurements given that the microphones have the constant sampling frequency. Fortunately, a normal user does not step back as fast as 2.0 steps/second. So the true performance of iLock is more reflected under the relatively slow and normal speeds.

Impact of Vertical Positions. The phone's vertical position may be different in various scenarios. For example, we tend to leave the phone on the desk around 70 cm high while in an office, on a chair about 40 cm high while on a subway, and the bar table about 100 cm high while in a bar. Fig. 15 shows the performance of iLock under different heights: 36 cm, 72 cm, 92 cm. For each height, the user moves away with the second gesture above for 20 times. We can see that different heights have very little impact on the true-positive rates of iLock.

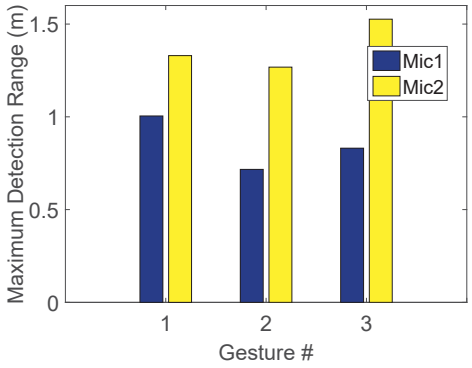
Impact of Speaker Volumes. iLock detects the leaving movement by signal reflections, so the signal strength can potentially affect its performance. We conduct the experiment under three volume levels corresponding to three signal strengths: low (26%), medium (52%), and large (71%). From Fig. 16, it is of no surprise to see that the performance via Mic2 alone or Mic1+Mic2 are quite high for medium and high volume settings.

Impact of Different Users. We also ask six PhD students to use iLock. Each student leaves in his own way for 20 times with the gesture and speed he likes. As shown in Fig. 17, iLock achieves a true-positive rate of 85% for student 2, 95% for student 5, and 100% for the rest. It is worth noting that student 2 walks much faster than others in the experiments, leading to the similar observation as in Fig. 14

Impact of Experimental Environments. We finally evaluate iLock in the lobby of the university library. The lobby is about 32,000 square feet and contains many tables, sofas and public desktop computers. During our experiment, there is a lot of noise from the vending machines, public



(a) True-positive rate



(b) Maximum-detection range

Figure 13: Performance of three leaving gestures.

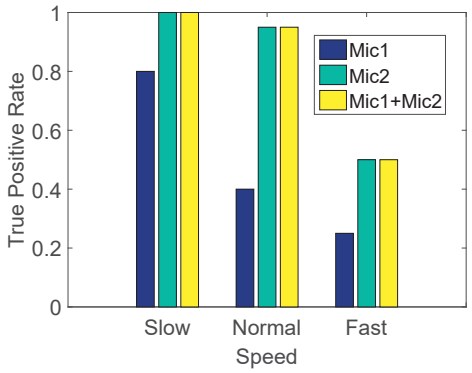


Figure 14: True-positive rate vs. leaving speeds.

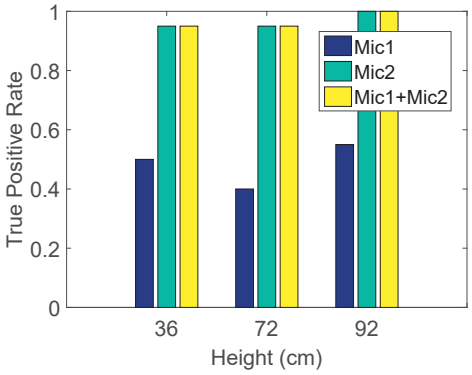


Figure 15: True-positive rates vs. phone heights.

computers, and student talks. In addition, the students walk around without our control, but we make sure that they are at least 1 m from the phone. The user puts the phone randomly on a table and leaves it 20 times with a normal speed under gesture 2. We obtain a true-positive rate of almost 100% by using Mic2 alone or Mic1+Mic2. So iLock can work very well in noisy and uncontrolled environments.

4.2 Evaluation with Type-II Attackers

We also evaluate iLock against Type-II attackers who get closer to but are still farther away from the device than the legitimate user. With the presence of Type-II attackers, iLock can detect multiple movement traces and needs to decide which trace is associated with the user. For this experiment, we use the Precision and Recall metrics defined as follows,

$$\texttt{Precision} = \frac{\#\texttt{TP}}{\#\texttt{TP} + \#\texttt{FP}} \text{ and } \texttt{Recall} = \frac{\#\texttt{TP}}{\#\texttt{TP} + \#\texttt{FN}}, \quad (1)$$

where #TP is the number of user departures correctly associated with the user, #FP is the number of other users' departures incorrectly associated with the user, and #FN refers to the number of user departures not associated with the user by mistake.

The experiment involves the user and one attacker, and their distance difference to the device varies from 20 cm to 30 cm, 40 cm, 50 cm, and 60 cm. For each distance difference, the user leaves 20 times while the attacker stays, and then the attacker leaves 20 times while the user stays. The Precision and Recall results based on Mic1+Mic2 are shown in Fig. 18. We can see that precision is always above 95%, corresponding to very low false-alarm rates. In contrast, the recall increases from 80% to 95% when the distance difference becomes larger, as larger distance difference makes it easier to distinguish the user's trace from the attacker's.

4.3 Evaluation with Type-III Attackers

Now we report the performance of iLock against Type-III attackers. This experiment involves the user and one attacker who is always closer to the phone than the user. As shown in Fig. 20, we use five representative scenarios in which the user and attacker are in different positions and orientations relative to the phone. In each scenario, the user leaves the device 20 times while the attacker stays, and then the attacker leaves 20 times while the user stays. In addition, the initial orientation of the device relative to the user can be accurately estimated with existing techniques [30]. Once two highly correlated leaving traces are detected, the metric $\hat{\eta}$ is computed according to the description in Section 3.4. Then we find the most probable orientation for $\hat{\eta}$ based on a fine-grained $\hat{\eta}$-orientation distribution, which we obtain beforehand for the Samsung Galaxy S5. Next, we compare the discovered orientation with the device's initial orientation relative to the user. Note that, in Fig. 8, $\hat{\eta}$ distributions of adjacent orientations overlap with each other, so we associate the traces discovered in nearby orientations to the target user to improve true positive rate. For example, if the device's initial orientation relative to the user is 180°, the leaving traces discovered between [135°, 225°] will be associated to the target user and the system locks the device

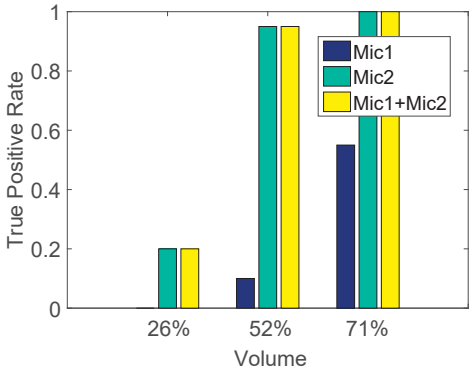


Figure 16: True-positive rates vs. different volumes.

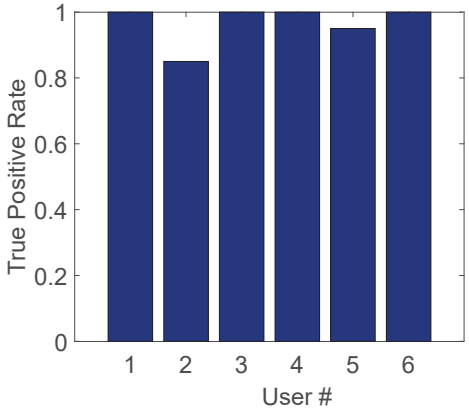


Figure 17: True-positive rates vs. different users.

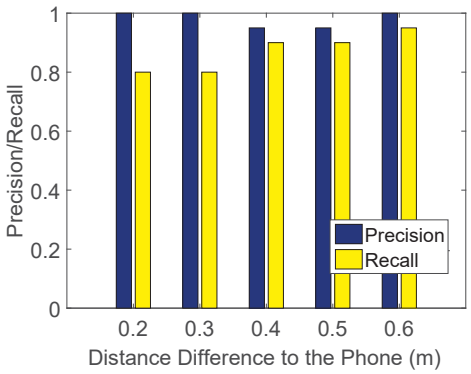


Figure 18: Precision and Recall with a Type-II attacker.

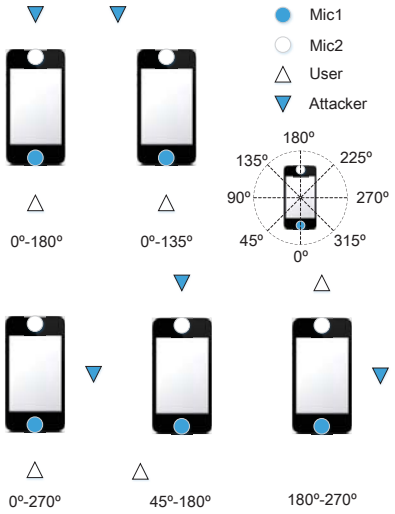


Figure 19: Representative scenarios with Type-III attackers, where x-y corresponds to the user's orientation x and attacker's orientation y in the shown orientation graph.

immediately to ensure data security. Users can devise their own mechanism to balance Precision and Recall.

As we can see in Fig. 20, the Precision and Recall results are overall quite acceptable for all five scenarios. The worst performance is observed when there is a small orientation difference between the user and attacker relative to the phone (i.e., 0°-270° and 180°-270°). This result is expected, as the smaller orientation difference makes it harder to distinguish the user's movement from the attacker's.

5. DISCUSSION

5.1 Energy Consumption

iLock incurs additional energy consumption on a mobile device in two main aspects. First, iLock needs to transmit high-frequency modulated acoustic signals and also record the signals reflected by physical objects. It is shown [28] that such acoustic transmitting and recording on Samsung Galaxy S5 may incur an energy consumption of about 800mW with Monsoon Power Monitor. Secondly, iLock consumes energy in data processing such as filtering, FFT, and mixing. In practice, iLock does not need to be activated all the time. In particular, iLock can only be activated when the device enters a vulnerable context. One such context is when the user stops using the device while the screen is still unlocked, and it is can be easily detected by exploring inertial sensors such as touchscreen, gyroscope, and accelerometer. Also note that many users spend most of the time in a safe zone such as home and office. Sophisticated localization techniques allow the device to accurately determine whether it is in a predefined safe zone. iLock is only activated when the device is out of the safe zone. So the energy consumption of iLock is quite amenable in contrast to its potentially huge benefits.

5.2 Other Potential Solutions

We also investigate and experiment other potential solutions. The most intuitive alternative is to directly analyze the received signals which can be perturbed by leaving movements. In the experiment, we indeed find some potential signal patterns for specific leaving gestures. So one may think about training a classifier to detect a user's leaving gesture. However, different users have different gestures, so every user who wants to use the system has to train a classifier, a time-consuming and clumsy process. In addition, even the same user may leave the device in a different way in different scenarios. As a result, it is almost impossible to train a classifier that can differentiate all possible gestures of the same user. So we give up this method.

Another candidate approach is to rely on the Doppler effect caused by user movements. In particular, the speaker

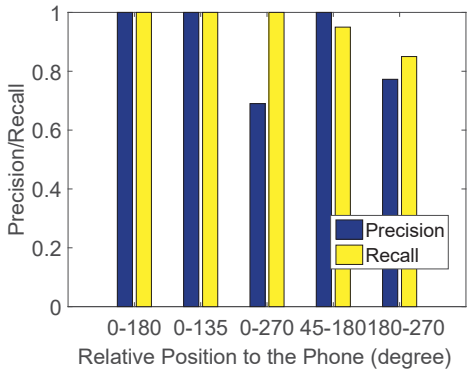


Figure 20: Precision and Recall with a Type-III attacker.

transmits acoustic signals with a fixed high frequency f_0, and the microphone records the reflected signals with frequency f_r. It follows that $f_r = \frac{c-v_r}{c-v_s} f_0$, where v_s is the speed of the reflection object (user), and c is the speed of sound. Since the receiver is stationary, $v_r = 0$. Then we can do an integration over v_s to get the distance the user moves. The Doppler shift, however, is very sensitive and can be induced by any body movement. Also, the frequency shifts by different body movements at different distances to the device are mixed together. As a result, we can hardly extract the user's movement pattern based on the Doppler effect and give up this idea as well.

Finally, one may think about implementing iLock based on WiFi or Bluetooth signals rather than acoustic signals. There are two primary reasons for not doing so. First, WiFi and Bluetooth interfaces are often very busy and occupied for data communications, while the speaker and microphone have much more idle time. Second, WiFi and Bluetooth signals propagate in the speed of light and have much higher requirement for time/frequency measurement accuracy, which is not attainable on COTS mobile devices. This is also the reason why existing FMCW implementations on WiFi signals use complicated and customized hardware not available on COTS mobile devices.

6. RELATED WORK

There are three ways to prevent the attackers' illegal access to mobile devices and the sensitive data therein. The first one is one-time authentication that authenticates users when they try to unlock and use the device. The second one is to authenticate users continuously when they are using the device. The third one is to lock the device immediately once the current user has left. We will analyze advantages and disadvantages of each method in what follows.

There are significant research and practice related to one-time authentication. Typically, one-time authentication schemes can be classified into three categories: *Something-You-Know*, *Someone-You-Are*, and *Something-You-Have*. In the *Something-You-Know* paradigm, users are asked to input a simple PIN, an alphanumeric password, or a gesture/graphical password. This method is vulnerable to shoulder-surfing attacks. The *Something-You-Have* paradigm requires auxiliary hardware (e.g. Signet Ring [29]) which is possessed only by the legitimate user. We note that the non-COTS hardware is a potential obstacle for the wide adoption of this paradigm. A growing body of work follows the *Someone-You-Are* paradigm [3, 7, 27]. This approach relies on physiological or behavioral biometrics which are unique to each person. Common physical features consist of fingerprints, facial features, retina patterns, etc. Physiological authentication methods may be vulnerable to spoofing attacks [3]. Behavioral biometrics may include keystroke patterns [17, 19], touching gestures [24, 25], gaits [10, 13], etc. As said, a significant number of mobile users do not password-protect their devices, not to mentioning adopting more advanced one-time authentication techniques. In addition, the time window for a password-protected device going from the unlocked mode to the locked mode may be long enough for a capable attacker to access all the sensitive information on the lost/stolen device. If an unlocked device is missing or stolen, the user's sensitive information is completely exposed.

Continuous authentication can complement one-time authentication by continuously authenticating the current user. In this way, after the attacker uses the device for a while, the device can detect the unauthorized user and log out. In [18], the user needs to wear a bracket with a built-in accelerometer, a gyroscope, and a radio. When using a desktop computer(typing the keyboard and using the mouse), the bracket records and sends the movement data to the computer. The computer checks whether the input to the computer matches the data from the bracket. A recent paper [11] points out attacks on the technique in [18]. The technique in [9] continuously authenticates users based on behavioral biometrics with 30 features. The equal error rates drop to 2%-3% with 11 to 12 strokes. Similar techniques based on behavioral biometrics are also presented in [8, 26]. We note that continuous authentication can only detect the attacker after he has used the device for a while. As a result, the attacker still has a good chance to obtain the victim's sensitive data before being logged out. In addition, if the attacker just watches content (e.g. photos and messages) on the screen and does not use the device, he would not be detected by continuous authentication methods at all.

Our method falls into the last category that the device locks itself immediately when the user leaves. If our method is combined with one-time and continuous authentication mechanisms, the attacker can hardly get any opportunity to access the user's sensitive data even if he possesses the missing mobile device. Our work is the first in this category to the best of our knowledge.

iLock is also related to recent work on object tracking and ranging. In particular, FMCW is used in WiTrack [6] for RF-based indoor localization and achieves the positioning accuracy of centimeter. WiTrack 2.0 [5] uses more antennas to support multi-user localization based on FMCW. Their methods are based on WiFi signals and customized transceivers that are not available on COTS mobile devices. In addition, the techniques in [21,22] use FMCW with audio signals to track the chest motion and finger movement, respectively. Finally, the work in [15, 20, 23] work on acoustic ranging between devices. iLock differs from these work in the research problem and also system implementation.

7. CONCLUSION

In this paper, we presented the design and evaluation of iLock, a secure and usable defense against data theft on a lost/stolen mobile device. iLock automatically, quickly, and accurately detects the user's physical separation from his/her device. Once significant physical separation is de-

tected, iLock immediately locks the device to thwart data theft. Relying on acoustic signals, iLock can be deployed on most COTS mobile devices with standard built-in microphones and speakers. Extensive experiments on Samsung Galaxy S5 confirmed the high efficacy of iLock with negligible false positives and negatives.

Acknowledgement

The authors would like to thank the anonymous reviewers for their constructive comments and helpful advice. This work was partially supported by the US National Science Foundation under grants CNS-1320906, CNS-1421999, and CNS-1514381.

8. REFERENCES

[1] http://www.channelpronetwork.com/article/mobile-device-security-startling-statistics-data-loss-and-data-breachesl.

[2] Cisco visual networking index global mobile data traffic forecast update 2014-2019. http://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/white_paper_c11-520862.html.

[3] http://gizmodo.com/hackers-iphone-5s-fingerprint-security-is-notsecure-1367817697.

[4] Kaspersky lab survey. http://www.kaspersky.com/about/news/virus/2015/Quarter-of-Users-Do-Not-Understand-the-Risks-of-Mobile-Cyberthreats.

[5] F. Adib, Z. Kabelac, and D. Katabi. Multi-person localization via rf body reflections. In *USENIX NSDI'15*, Oakland, CA, May 2015.

[6] F. Adib, Z. Kabelac, D. Katabi, and R. Miller. 3d tracking via body radio reflections. In *USENIX NSDI'14*, Seattle, WA, 2014.

[7] Y. Chen, J. Sun, R. Zhang, and Y. Zhang. Your song your way: Rhythm-based two-factor authentication for multi-touch mobile devices. In *IEEE INFOCOM'15*, Hong Kong, China, 2015.

[8] T. Feng, Z. Liu, K.-A. Kwon, W. Shi, B. Carbunar, Y. Jiang, and N. Nguyen. Continuous mobile authentication using touchscreen gestures. In *IEEE HST'12*, Waltham, MA, 2012.

[9] M. Frank, R. Biedert, E.-D. Ma, I. Martinovic, and D. Song. Touchalytics: On the applicability of touchscreen input as a behavioral biometric for continuous authentication. *IEEE Transactions on Information Forensics and Security*, 8(1):136–148, 2013.

[10] D. Gafurov, K. Helkala, and T. Søndrol. Biometric gait authentication using accelerometer sensor. *Journal of Computers*, 1(7):51–59, 2006.

[11] O. Huhta, P. Shrestha, S. Udar, M. Juuti, N. Saxena, and N. Asokan. Pitfalls in designing zero-effort deauthentication: Opportunistic human observation attacks. In *NDSS'16*, San Diego, CA, Feb. 2015.

[12] H. Khan, A. Atwater, and U. Hengartner. Itus: An implicit authentication framework for android. In *ACM Mobicom'14*, Maui, Hawaii, Sept. 2014.

[13] J. Kwapisz, G. Weiss, and S. Moore. Cell phone-based biometric identification. In *IEEE BTAS'10*, Washington DC, Sep. 2010.

[14] L. Li, X. Zhao, and G. Xue. Unobservable re-authentication for smartphones. In *NDSS'13*, San Diego, USA, Feb. 2013.

[15] K. Liu, X. Liu, and X. Li. Guoguo: Enabling fine-grained indoor localization via smartphone. In *ACM MobiSys'13*, Taipei, Taiwan, Jun. 2013.

[16] B. Mahafza. *Radar Systems Analysis and Design Using MATLAB Third Edition*. CRC press, 2013.

[17] E. Maiorana, P. Campisi, N. González-Carballo, and A. Neri. Keystroke dynamics authentication for mobile phones. In *ACM SAC'11*, TaiChung, Taiwan, Mar. 2011.

[18] S. Mare, A. Markham, C. Cornelius, R. Peterson, and D. Kotz. Zebra: Zero-effort bilateral recurring authentication. In *IEEE S&P'14*, San Jose, CA, May 2014.

[19] F. Monrose, M. Reiter, and S. Wetzel. Password hardening based on keystroke dynamics. *International Journal of Information Security*, 1(2):69–83, 2002.

[20] R. Nandakumar, K. Chintalapudi, and V. Padmanabhan. Centaur: locating devices in an office environment. In *ACM MobiCom'12*, Istanbul, Turkey, 2012.

[21] R. Nandakumar, S. Gollakota, and N. Watson. Contactless sleep apnea detection on smartphones. In *ACM MobiSys'15*, Florence, Italy, May 2015.

[22] R. Nandakumar, V. Iyer, D. Tan, and S. Gollakota. Fingerio: Using active sonar for fine-grained finger tracking. In *ACM CHI'16*, San Jose, CA, May 2016.

[23] C. Peng, G. Shen, Y. Zhang, Y. Li, and K. Tan. Beepbeep: a high accuracy acoustic ranging system using cots mobile devices. In *ACM SenSys'07*, Sydney, Australia, Nov. 2007.

[24] N. Sae-Bae, K. Ahmed, K. Isbister, and N. Memon. Biometric-rich gestures: a novel approach to authentication on multi-touch devices. In *ACM CHI'12*, Austin, TX, May 2012.

[25] M. Shahzad, A. Liu, and A. Samuel. Secure unlocking of mobile touch screen devices by simple gestures: you can see it but you can not do it. In *ACM MobiCom'13*, Miami, FL, Sep. 2013.

[26] W. Shi, F. Yang, Y. Jiang, F. Yang, and Y. Xiong. Senguard: Passive user identification on smartphones using multiple sensors. In *IEEE WiMob'11*, Shanghai, China, 2011.

[27] J. Sun, R. Zhang, J. Zhang, and Y. Zhang. Touchin: Sightless two-factor authentication on multi-touch mobile devices. In *IEEE CNS'14*, San Francisco, CA, Oct. 2014.

[28] Y.-C. Tung and K. Shin. Echotag: accurate infrastructure-free indoor location tagging with smartphones. In *ACM MobiCom'15*, Paris, France, Sep. 2015.

[29] T. Vu, A. Baid, S. Gao, M. Gruteser, R. Howard, J. Lindqvist, P. Spasojevic, and J. Walling. Distinguishing users with capacitive touch communication. In *ACM MobiCom'12*, Istanbul, Turkey, Aug. 2012.

[30] P. Zhou, M. Li, and G. Shen. Use it free: Instantly knowing your phone attitude. In *ACM MobiCom'14*, Maui, Hawaii, Sep. 2014.