COMPUTERS & SECURITY 80 (2019) 273–290

Available online at www.sciencedirect.com

ScienceDirect

journal homepage: www.elsevier.com/locate/cose

Computers & Security

Find me a safe zone: A countermeasure for channel state information based attacks

Jie Zhang[a], *Zhanyong Tang*[a,*], *Meng Li*[a], *Dingyi Fang*[a], *Xiaojiang Chen*[a], *Zheng Wang*[b]

[a] *School of Information Science and Technology, Northwest University, China*
[b] *Metalab, School of Computing and Communications, Lancaster University, UK*

ARTICLE INFO

Article history:
Received 3 February 2018
Revised 20 September 2018
Accepted 22 September 2018
Available online 13 October 2018

Keywords:
Channel state information-based attacks
Sensing
Countermeasures
Security
Privacy protection
Gesture recognition

ABSTRACT

Recently, channel state information (CSI) is shown to be an effective side-channel to perform attacks in public environments. Prior work has demonstrated that by analyzing how the CSI measurements of the wireless signal are affected by the mobile user's finger movements or gestures, an attacker can recover the user's input with a high success rate. Furthermore, the setup of this new attack is trivial, where the adversary only needs to place one or two malicious wireless devices near the target user. It would be difficult for many users to identify the nearby malicious devices while they want to continue to use mobile applications in public places. This dilemma makes protection of CSI-based attacks an urgent need.

This article presents the first countermeasure for CSI-based attacks. Our key insight is that the success of any CSI-based attack requires high-quality CSI measurements; and we can significantly reduce the risk of information leakage by directing the user to a nearby location where the CSI readings are inherently noisy. To this end, we develop a regression based method to assess the risk of CSI-based attacks and exploit a well-established localization technique to identify potential malicious wireless devices. We then use this information to guide the user to a safe zone. We evaluate our approach by applying it to protect pattern lock and keystrokes in various indoor and outdoor environments. Experimental results show that our approach can effectively protect mobile users against CSI-based attacks.

1. Introduction

Portable mobile devices, such as smart phones and tablets, are widely used in public places – from indoor restaurants and shopping malls to outdoor bus stations. At the same time, many mobile services, including shopping and banking applications, use PIN- and text-based passwords or locking patterns for authentication and authorization; and people are increasingly relying on their mobile devices to perform activities like social networking and mobile payment. Given that a leakage of passwords or locking patterns could lead to a catastrophic consequence, there is an urgent need to protect using mobile devices in public places.

* Corresponding author.
E-mail addresses: jz@stumail.nwu.edu.cn (J. Zhang), zytang@nwu.edu.cn (Z. Tang), lijmeng@stumail.nwu.edu.cn (M. Li), dyf@nwu.edu.cn (D. Fang), xjchen@nwu.edu.cn (X. Chen), z.wang@lancaster.ac.uk (Z. Wang).
https://doi.org/10.1016/j.cose.2018.09.017
0167-4048/

In recent years, channel state information (CSI)[1] is shown to be an effective side-channel for uncovering users' input. CSI has been demonstrated to be useful in inferring PIN- and text-based passwords (Li et al., 2016) and locking patterns (Zhang et al., 2016). Launching a CSI-based attack is relatively simple. The setup of the attack only requires the adversary to place one or two wireless devices (which could be WiFi routers, mobile phones, tablets, or laptops etc.) next to the target user. By recording how the CSI measurements are affected by the user's finger movements or gestures (while typing in sensitive information), an attacker can then map the measured CSI readings to a pattern or keystroke. This is because the gesture or fingertip movement of each pattern or keystroke usually corresponds to a unique CSI pattern.

Compare with other side-channel attacks that exploit the acoustic signal (Liu et al., 2015; Zhu et al., 2014), video (Shukla et al., 2014; Yue et al., 2014), sensors (Raij et al., 2011; Wang et al., 2015a), oil residues (Aviv et al., 2010) or fingerprints (Zhang et al., 2012), CSI-based attacks have the advantages that the malicious devices[2] used to launch the attack can be placed further away from the target, and the attack does not require having physical access to the target device or seeing the content showing on the screen. This increases the success of CSI-based attacks. Furthermore, because wireless devices are very common in public places, users will find it difficult to identify malicious wireless devices to stay vigilant. For these reasons, we believe CSI-based attacks are a real security threat in public places.

In this article, we propose, for the first time, a countermeasure for CSI-based attacks. Our key insight is that any CSI-based attack relies on high-quality CSI measurements collected during the input of sensitive information; and if we can quantify the quality of the CSI measurements, and guide the user to a location where the CSI readings will be inherently noisy, the attack will unlikely to be successful. Our countermeasure detects the attack by monitoring the network traffics (Zhang et al., 2017). We define a signal to noise (SNR) metric of CSI readings from the attack's perspective, which is used by our countermeasure to quantify how likely a CSI-based attack will succeed in the current physical environment. Our approach uses sensors that are common on smartphones to identify potential malicious WiFi devices, as well as the walking direction and pace of the user. Such information is obtained by leveraging the well-established localization techniques developed by the wireless communication community. We then use the SNR to assess the risk of CSI-based attacks. If a potential CSI-based attack is detected, our countermeasure then suggests the user to add some body noise – e.g., shake the mobile devices or turn around when input sensitive information – or combines the risk assessment with the location information to guide the user to move away from the malicious devices, and to determine towards which direction the user should move, as well as when and where the user is safe from CSI-based attacks.

Our prototype system is implemented as a background service for Android. The system automatically detects when the user is entering sensitive information, by monitoring relevant system events. If a sensitive input is detected, the system will assess the risk of the surrounding environment and suggest the user to move to a safe location if the risk is considered to be high.

We evaluate our approach by applying it to protect Android pattern lock, which is commonly used by mobile payment systems for authentication and authorization. Experiments performed in various indoor environments show that our approach can successfully protect users against CSI-based attacks. This work is the first countermeasure for CSI-based attacks. We show that our approach is simple to implement and does not require specialized hardware, yet it can effectively protect users against CSI-based attacks.

2. Background

CSI is a metric that describes how the wireless signal transmission is affected by the surrounding physical environment, due to effects like reflections, diffractions and scattering (Halperin et al., 2010). In recent years, researchers have shown that CSI can be used as a side channel to reconstruct sensitive information with a high accuracy. The range of CSI-based attacks includes guessing PIN-based passwords (Zhang et al., 2016), keystrokes (Ali et al., 2015) and pattern lock (Li et al., 2016) for mobile devices, as well as lip-reading (Wang et al., 2014a). The underlying principle of CSI-based attacks is that the different fingertip or lip motions will cause a unique interference to the multi-path signals and can be reflected by the CSI which can then be used to infer sensitive information.

Implementing a CSI-based attack is relatively trivial. In a typical attacking scenario, the adversary only needs to place one or two standard wireless devices near the user (see Fig. 1). To launch the attack, the adversary needs to record the CSI values while the user is entering the sensitive information. Obtaining the CSI readings requires some hacking to the driver program of the wireless device, but there is already an open source CSI toolkit implementation available (Daniel et al., 2012). As public WiFi access points are now commonplace, users will find it difficult to identify malicious WiFi devices in public places. This makes CSI-based attacks a real threat that must be tackled.

CSI-based Attacks. There are in general two types of CSI based attacks, the in-band inference (IBI) and the out-band inference (OBI) modes. Using the former technique, the adversary needs to deploy only one malicious WiFi device as a public access point, but the attack itself requires the user's device to connect to the malicious device. This attack is illustrated in Fig. 1a. Using the latter technique, the adversary needs to deploy two malicious WiFi devices (Fig. 1c), but the attack does not require the user to connect to any of the malicious devices. However, the OBI attack is hard to implement because it is difficult to detect when to track the user's input. *This work solely focuses on protecting users against IBI attacks.*

[1] CSI is a property that describes how a wireless signal is affected by the surrounding environment and the movement of objects. See also Section 2.

[2] In this article, we refer the wireless devices used to launch the attack as malicious devices.

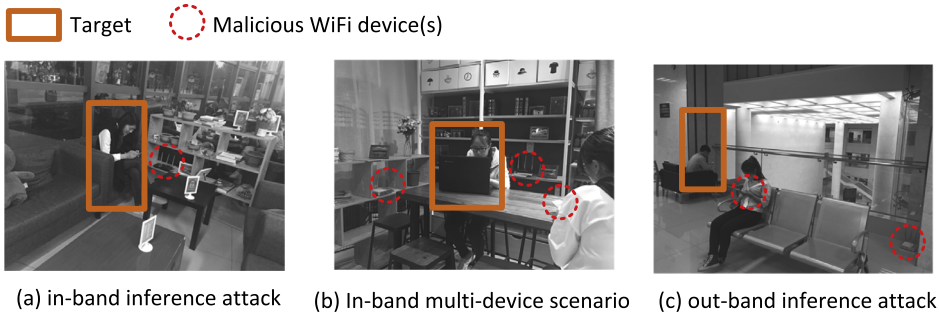


Fig. 1 – Example attacking scenarios which are common in our day to day life. To launch the attack, the adversary places a number of malicious WiFi devices (marked by a circle) near the target user (marked by a rectangle).

CSI-based Attack Setup. A classical CSI-based attack follows several steps. First, the adversary needs to identify when the user is entering sensitive information, i.e., the sensitive input window. Doing so is relatively straightforward for a screen lock, as this is often the first activity users perform when picking up their mobile devices. For other sensitive inputs, previous attacks (Li et al., 2016) show that is possible to do so by analyzing the meta data (such as the IP address that the application is connecting to) of mobile applications. Secondly, during the sensitive input window, the malicious device periodically sends ICMP Echo requests to the target device (for IBI attacks). Per the ICMP protocol, the target device replies to each Echo request a response packet. The reply is then used to calculate the CSI value[3]. Finally, at the end of the input window, the collected CSI values will be analyzed and matched against a pattern database to recover the user's input.

Data analysis. After collecting the CSI measurements, an attacker needs to preprocess the CSI measurements to remove the noises. The high dimensional raw CSI data are projected into a lower-dimensional space using techniques like the Principal Component Analysis. The attacker will then need to determine the start and the end points of the CSI data of interests (e.g., when the user started and finished entering sensitive information). After locating the interesting data sequences, an algorithm is then applied to reconstruct the user's input from the data.

Example. To illustrate how CSI can be used to uncover sensitive information, we have performed an experiment where a user drawed three Android locking patterns (Fig. 2a) on a mobile device that is within 0.5 meters to two WiFi endpoints. The Android locking patterns are chosen according to the strategies mentioned in Uellenbeck et al. (2013) and Ye et al. (2017). In the experiment, we collected the CSI values while the user was drawing each pattern. For each pattern, we repeated the process five times. Fig. 2b shows the CSI amplitude for each pattern. As can be seen from the diagram, each locking pattern corresponds to a unique CSI structure which is relatively consistent across multiple inputs of the same pattern. An attacker can infer the user's inputs by matching the CSI measurements against a knowledge database of inputs to CSI mappings.

[3] The sender needs to ping the receiver at a high rate to the CSI values can be captured at a high resolution.

3. Threat model

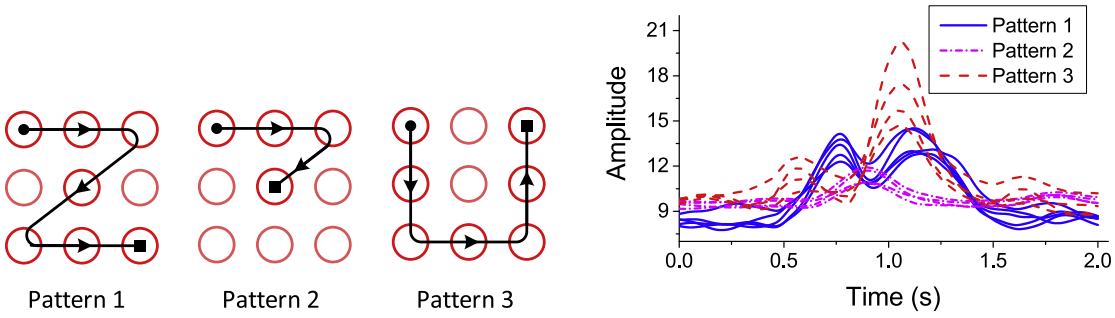
In our threat model, we assume an adversary wants to access some sensitive information (e.g. passwords, PINs, pattern lock or keystrokes) when a user is using a mobile device. We assume the adversary can identify when the user starts and finishes entering sensitive information, which is already achieved in Li et al. (2016). To launch the attack, the adversary needs to be able to place at least one (for IBI attacks) or two (for OBI attacks) malicious wireless devices in a place where a target is likely to remain relatively stationary and within e.g. 0.5 to 5 meters to the device. Fig. 1 illustrates some of the many possible attacking scenarios in our day to day life. The attack can take place in an indoor environment. To implement the attack, the adversary can use any wireless device (e.g. wireless routers, laptops, or smartphones, which are universal in public places but are not trusted, and can be seen as malicious devices) that supports the ICMP protocol and can report CSI readings. Nearly all wireless devices that can run Linux can be used for this attack; so the setup of the attack is trivial.

We consider scenarios where the attacker might leverage more than two malicious public WiFi devices to launch the attack. We also consider scenarios where the attacker can dynamically change the WiFi signal strength to confuse the protection scheme. However, we assume that the adversary does not have physical access to the user's mobile phone. Furthermore, we assume that the attacker can neither change the hardware and software components of the target device, nor install malicious software on the device. It is to note that to protect against internal device tampering or other side channel attacks falls out of the scope of this work.

4. Motivation

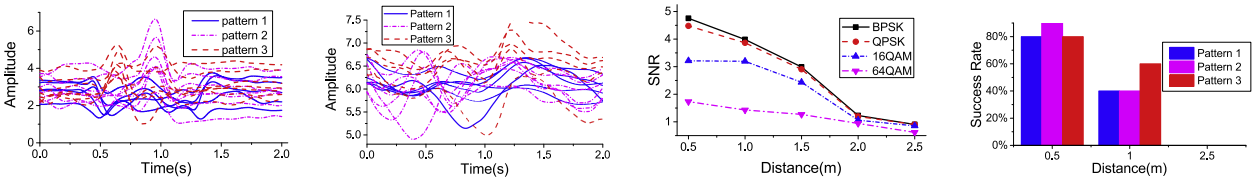
Our key insight to protect against CSI-based attacks is that the success of the CSI-based attack strongly depends on the quality of the CSI measurements, but the quality of the measurements decreases as the target device moves further away from the malicious WiFi devices.

Consider Fig. 3a and 3b, which show the resulted CSI measurements for the three patterns presented in Fig. 2a when the target device is within 1 and 2.5 m, respectively from the malicious WiFi devices. Note that we applied the method



(a) Example locking patterns (b) CSI measurements

Fig. 2 – The working mechanism of CSI based attacks. In this example, the target device is within 0.5 m to a pair of WiFi endpoints used to collect CSI readings. Each of the three patterns (a) is drawn five times on a mobile device and the collected CSI measurements are shown in (b). It is observed that each pattern corresponds to a unique CSI amplitude structure which is consistent across multiple inputs. By matching the collected CSI values to a pattern, an attacker can reconstruct the user's inputs.



(a) CSI readings within a distance of 1 meter (b) CSI readings within a distance of 2.5 meters (c) Average SNR values of pattern 1 for different distances (d) Accuracy of Figure 1(c) under different distances

Fig. 3 – CSI measurements for the three patterns shown in Fig. 2a when the distance from the target to the malicious devices is 1 m (a) and 2.5 m (b). The further away the target to the malicious devices, the more noisy the CSI measurements and the lower the success rate are (d).

presented in Zeng et al. (2016) to reduce the noise of the raw CSI data.

If we compare the CSI measurements taken when the target device is within 0.5 m to the malicious devices (Fig. 2b) with Fig. 3a and 3 b we can observe visually that the CSI measurements taken from a distance of 1 and 2.5 m tend to be more noisy and the CSI differences between patterns are increasingly insignificant as the target device moves further away from the malicious WiFi devices. This is because the further away the user from the malicious devices is, the stronger negative impact (i.e. noise) of the multipath (i.e. presence of many reflected paths from the surrounding objects) will have to the CSI measurements.

To quantify the noise of the CSI measurements, we calculated the average signal to noise ratio (SNR) of pattern 1 under different distances. The results are given in Fig. 3c. This figure illustrates the SNRof four wireless modulation schemes, namely BPSK, QPSK, 16QAM and 64QAM, drops as the distance from the target device to the malicious device increases[4]. It is observed that the noise of the CSI measurements significantly increases as the distance to the malicious device increases, leading to a drop in the SNR when the distance changes from 0.5m to 2.5m. As can be seen from Fig. 3d, the increased noise of the CSI also reduces the efficiency of the attack. When the distance is less than 0.5m, the attack can successfully recover all three patterns; however, it fails to infer any of the patterns when the distance increases to 2.5m.

This example shows that the user's location has a significant impact on the success of CSI-based attacks. The attack is unlikely to succeed if the user stays in an area where the attacker cannot obtain good quality CSI measurements. Finding an appropriate safe zone is, however, non-trivial. On the one hand, we do not want to move the user further away than necessary, as doing this might affect the normal wireless communication quality and have a negative impact on the user experience. On the other hand, we cannot put the user too close to the malicious WiFi devices, which will be risky. This means that our approach needs to adapt to different physical environments and attacking scenarios. In the rest of this paper, we will describe how to build such an adaptive system based sensor-based location estimation.

5. Overview of our approach

Fig. 4 depicts our 4-step approach that gives the users some countermeasures to protect the user against CSI-based attacks. Our approach is simple to implement and it does not

[4] In this paper, we use the BPSK modulation scheme in evaluation because it shares many commonalities among four schemes.

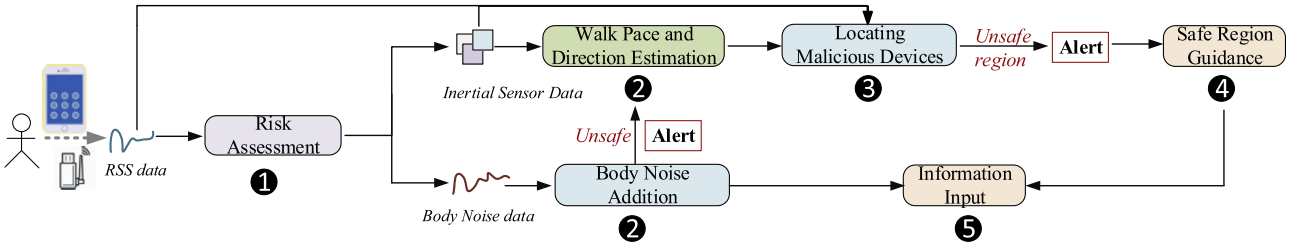


Fig. 4 – Overview of our approach. Our system constantly evaluates the risk of CSI-based attacks based on the user's current location. If the risk is assessed to be high, the system can give the user two countermeasures, and one is adding body noise, and another is finding a safe zone. If the risk assessment is still high after adding body noise, it uses mobile sensor information to estimate the user's walking speedup and direction as well as to locate potential malicious devices. Using these estimations, the system will suggest the user to walk several steps to find a safe zone.

require any specialized hardware devices. This means that our software system can be ported to most commercial mobile systems. As we will show in Section 6, this simple yet powerful technique can effectively protect users against CSI-based attack.

Our prototype system is implemented as a background service running for the Android operating system. It automatically detects specific events that are related to the input of sensitive information, including touching the screen lock, entering texts in a password box etc. Once a registered event is detected, the system tries to guide the user to a safe location, using the following four steps:

①*Risk Assessment*. The success of any CSI-based attack requires having relatively stable WiFi signals. To assess the risk of CSI-based attacks, our system works as follows. The system constantly monitors the network traffics to detect potential CSI-based attacks. It then periodically checks the threat level to suggest actions to be taken by the user. The evaluation is based on the SNR, detailed in Section 6.1. If the threat level is high, our system will suggest the user to either add some body noises or find a location where the attack is much less likely to succeed by using data collected from mobile sensors in the scene.

②*Countermeasure Selection*. There are two major countermeasures to defeat CSI-based attack, and one is adding body noise and another is finding a safe zone, described as follows.

Body Noise Addition. The user can add body noise to defeat the CSI-based attack, such as the user shakes the devices when he/she inputs the sensitive information or the user just turns around and enter one digit/character at a time when facing different directions. If the risk assessment is still high after adding body noise, then the system will suggest the user to walk several steps to find a safe zone, described as follows.

Walking Pace and Direction Estimation. To help the user to find a safe zone, we will need to locate the malicious devices and identify the walking direction of the user. In this step, our system collects the acceleration and orientation information from the accelerometer and orientation sensors that are available on most mobile devices. It uses the accelerometer data to estimate the user's walking distance and pace, and uses the orientation data to identify the user's walking direction and angle. This is described in Section 6.3.1.

③*Locating Malicious Devices*. In addition to the user's walking pace and direction, we also need to know the position of all possible malicious WiFi devices[5]. This is essential for circumstances where the adversary deploys multiple malicious WiFi devices, because the system needs to find a region that is safe for all malicious devices. To locate the malicious devices, we use the distance and angle information given by step 2. The detailed implementation of this step is given at Section 6.3.2.

④*Safe Region Guidance*. After locating the locations of all possible malicious devices, our system then directs the user to move to a safe location. Our system dynamically evaluates the risk of the current location and suggests if the user needs to move further and if so towards which direction. Our system constantly assesses the risk and will repeat steps 2 to 4 if the risk becomes high due to the change of situations, e.g. when the adversary changes the antennae or signal of the wireless devices. This step is discussed at Section 6.3.3.

6. Implementation details

6.1. Risk assessment

This work targets IBI attacks where an attacker leverages the public wireless devices as transmitters to launch the attack. Prior work of CSI-based attacks require the wireless devices to exchange ICMP packets at a high data rate of around 2000 packets per second. The high frequent data exchange ensures that the attacker can obtain the user's input with a high resolution. Our work leverages this pattern to detect potential CSI-based attacks by monitoring the number of ICMP packets sent within an observation window (Zhang et al., 2017).

Our software system works by firstly assessing how likely a CSI-based attack can be successfully launched when a mobile device is used in a public place. The system runs as an Android background service. It automatically detects operating system events that are registered with sensitive information inputs. In our current implementation, we detect two types of events: (1) mobile payment using pattern passwords, (2) mobile

[5] To launch the attack, the wireless devices must be used as access points. Our prototype treats all wireless devices as potential malicious devices.

payment using digital passwords[6]. If any of these events are detected, the risk assessment process will be triggered. It is to note that the set of supported events can be easily extended. Furthermore, our system can also be manually launched by the user.

Recall that the success of a CSI-based attack requires to have stable wireless signals to collect clean CSI measurements (Section 4). Intuitively, the noisier the wireless environment is, the more unlikely the attack will succeed. To measure the quality of the wireless signal, we use SNR which measures the signal strength (signal to noise ratio) of the communication link from the end-user's perspective. The SNR is calculated using the following formula:

$$\text{SNR} = \frac{csi_m - csi_c}{noise} \tag{1}$$

where csi_c is the CSI measurement prior to the user interacts with the device, csi_m is the averaged CSI reading during the interactive window, and *noise* is the background noise due to the multipath of wireless signals.

To calculate the SNR, our system takes a CSI reading every 100 ms; the latest CSI measurement (csi_c) prior to the interactive window is used as a clean reference of CSI measurement. The difference between csi_m and csi_c indicates how the CSI measurement is affected by the user's finger or gesture motion. In other words, the difference indicates the quality of the CSI measurement from the attacker's perspective. Here, the SNR given by Eq. (2) reflects how well does the measurement capture the subtle changes to the wireless signal when the user is interacting with the target device. Therefore, the stronger the SNR, the cleaner the CSI measurement can capture the user's input and the more likely the attack can succeed.

To understand how does the SNR values correlate to the success rate of the CSI-based attack, we perform a set of experiments to launch the attack where the target device is within a various range of distances to the malicious devices; we then record the SNR readings from the user's device and the success rate. The experiments were evaluated using Android pattern lock in the scenario shown in Fig. 3a. The distance between the target and the malicious devices ranges from 0.5 m to 5 m. To collect the data, we asked our participants to repeatedly draw a screen lock five times. We reproduced the attack presented at Li et al. (2016) to crack the pattern and record the success rate.

Fig. 5 plots the correlation of SNR readings to the success rate of the attack, evaluated in the scenario shown in Fig. 3. This diagram shows that the success rate of the attack strongly correlates to the SNR. The stronger the SNR, the higher the success rate will be. This is because a stronger SNR value gives a cleaner CSI measurement which helps the attack.

To quantify the relation between the SNR values and the success rate of the attack, we use regression to fit a set of data points collected in our experiments. Each pair of the data points consists of an SNR measurement and the success rate of the attack for a number of patterns. We evaluated a range of linear and non-linear regression models and use the root mean square error (RMSE) to evaluate the performance of each model. We choose a piecewise regression model, which combines a linear regression and a univariate cubic model, because it gives the best RMSE (0.0845 in our case). The resulted regression model is:

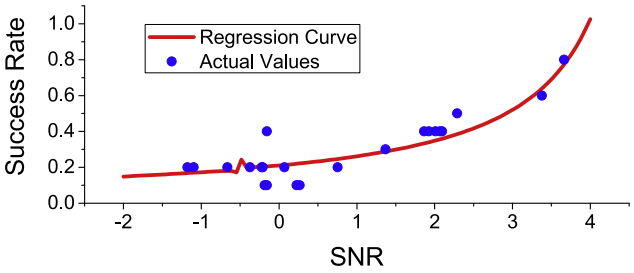


Fig. 5 – SNR values and the success rate. The success rate and SNR can be fitted using a regression model.

$$r = (p_1 \times SNR + p_2)/((SNR)^3 + q_1 * (SNR)^2 + q_2 * SNR + q3) \tag{2}$$

where r is the success rate, and the coefficients, $p_1 = 1081$, $p_2 = 543.9$, $q_1 = -1033$, $q_2 = 4657$ and $q3 = 2582$, are automatically determined by the regression algorithm. It is to note that we also collected data from all other environmental settings shown in Fig. 3. We found that Eq. (2) can accurately describe the relation between SNR and the success rate r.

6.2. *Body noise addition*

When the threat level is considered to be high (i.e. greater than a threshold), our system will issue a warning for the user to take appropriate countermeasures. Our goal is to allow users to continue interacting with the mobile devices if the interaction is safe.

As we all know, the body movements of the user also have an effect on CSI measurements, especially when the surrounding environments are more complex , the negative effect is more obvious. Therefore, adding body noises when the user enters the PIN or Pattern Lock looks like possible to defend against CSI-based attacks. In this section, we investigate the effect of body noises on the success rate of CSI-based attacks. It is known that there are many methods for the user to add body noises, for example, the user can shake the devices or just turn around when entering one digit/character at a time with different movement directions. We carry out a survey about the body noises addition countermeasures, however, the results show that only 13% of the participants are willing to add body noises when they input sensitive information, and among the 13% participants, only 37% can correctly input the sensitive information when they are requested to shake the mobile devices or add other body movements.

Unfortunately, even the user would like to choose adding body movements to make noises, maybe it cannot defend against CSI-based attacks. As can be seen from Fig. 6 that even when the user adds some body noises when entering the

[6] The principles for attacking digital- and pattern-based passwords are similar. In this paper, we use the pattern-based passwords as an example to demonstrate the feasibility of the system.

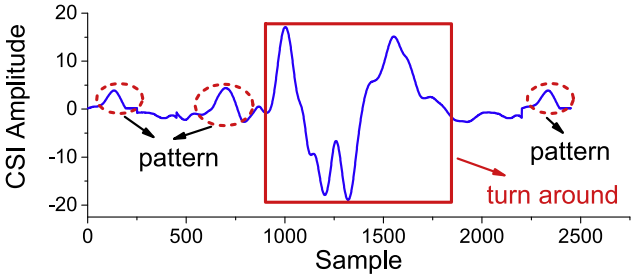


Fig. 6 – The CSI waveforms when add additional body noises. The user first enters two patterns and then turns around, and continues to enter the third pattern. .

PIN or lock pattern, the attackers sometimes can also crack and steal these sensitive information. This is because the CSI measurements of device shaking and significant body movements are different from that of pattern locks and they have their own unique characteristics. These characters make the attackers easily remove the body noises added from the collected hybrid measurements. Thus, only adding some simple body noises sometimes doesn't work and we should choose other more effective methods. According to the prior survey results of participants, about 87% of volunteers are willing to choose to walk several steps to a safe region, obviously this way of interacting will be more friendly and acceptable than adding body movements in public.

6.3. *Safe region guidance*

In order to guide the user to enter a safe zone, firstly the system should estimate the parameters such like steps and step size when the user is walking. Secondly we can use these parameters to locate all potential malicious devices and calculate the SNR values. Finally combing the relationship between SNR values and success rate, the system will guide the user to a safe zone.

6.3.1. *Walking pace and direction estimation*

To find where and decide which direction the user should move, our system tries to locate all the potential malicious WiFi devices. In this work, the collected data of mobile phone sensors are used to estimate the walk length and directions, furthermore the received signal strength (RSS) of the WiFi signal is used to locate potentially malicious WiFi devices.

Considering the scenario depicted in Fig. 7a as an example, the system will automatically estimate the walk steps and walk length of the user when potentially malicious WiFi devices are detected. The details are shown in Algorithm 1.

We apply a weighted moving average method to remove the high-frequency noise and the inherent power frequency noise of the sensors. The high-frequency noise is introduced by the unconscious vibration of hand muscles during walking.

Then the resultant acceleration is calculated as Eq. (3) to estimate the walk steps.

$$R = \sqrt{X_{ac}{}^2 + Y_{ac}{}^2 + Z_{ac}{}^2} \tag{3}$$

Algorithm 1 Walkingpace and direction estimation.

Input:
$T[]$: Three axises acceleration data
$R[]$: Orientation data
Δm: Magnitude threshold for estimating steps
Δt: Time threshold for estimating steps
Output:
S_{num}: Walk steps during the user's walk
$SL[]$: Walk length for each step
$SD[]$: Direction for each step
1: **for** axises acceleration data $T[]$ **do**
2: $S_{num} = 0$; $L[] = []$; $D[] = []$;
3: *resultant acceleration* $r[] \leftarrow calculatereac(T[])$
4: **end for**
5: *magnitude* $p[] \leftarrow findpeaks(r[])$
6: *time* $t[] \leftarrow getpeakstime(p[])$
7: *peaks' number* $p_{num} \leftarrow getpeaksnumber(p[])$
8: **for** $i = 1 : p_{num}$ **do**
9: **if** $checksteps(p[], t[], \Delta m, \Delta t)$ **then**
10: $S_{num}++ \leftarrow getstepsnum(S_{num})$
11: *steps' time* $st[] \leftarrow updatestepstime(t[])$
12: **end if**
13: **end for**
14: **for** $j = 1 : S_{num}$ **do**
15: *each step's length* $SL[] \leftarrow calculatelength(st[], r[])$
16: *each step's direction* $SD[] \leftarrow getdirection(st[], R[])$
17: **end for**

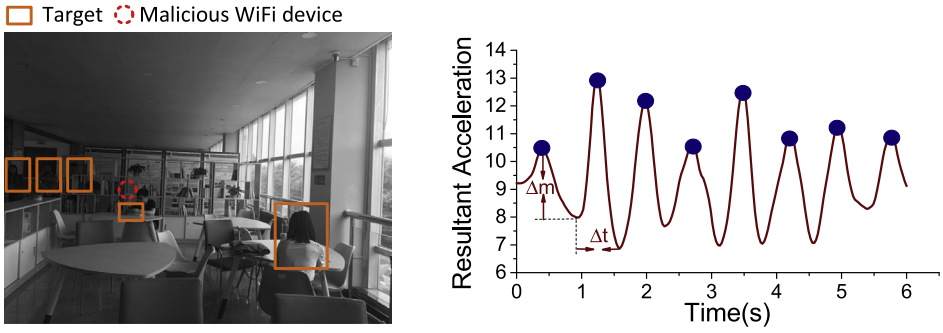
Where X_{ac}, Y_{ac} and Z_{ac} separately represent the noise-removed acceleration sensor's data of x-axis,y-axis and z-axis (line 3), and the results are shown in Fig. 7b.

When using the resultant acceleration to estimate walk steps, a "peak to peak" method can be utilized to detect the steps (Zeng et al., 2016). However, not all the detected peaks will be useful for step estimation, magnitude threshold Δm and time threshold Δt (Li et al., 2012) are used to filter out false peaks. When the magnitude difference between peaks and neighbor valley is larger than Δm and the duration time of two neighbor valleys is larger than Δt, the peaks can be stored and used to estimate the walk steps.

After estimating the walk steps, the system will calculate each step length using Weinberg's approach (Weinberg, 2002) and then use PDR algorithm (Lachapelle et al., 2006) to calculate the coordinates of each step according to the orientation data and each step length. The coordinates of each step can be regarded as known reference nodes in locating malicious WiFi devices.

6.3.2. *Locating malicious devices*

Note that prior work has proposed that the success rate of CSI-based attacks could be increased if the attacker use multiple WiFi devices (Abdelnasser et al., 2015; Wang et al., 2015b; Wang et al., 2014b). So the system needs to consider an extreme case that all surrounding WiFi devices may be evil although it is unlikely to happen in practice. It is challenging to find a safe zone in such complex situations. We suppose that the user's starting point is in an unsafe zone according to the relationship between SNR values and success rate of CSI-based attack, if the user just wants to input sensitive



(a) An example of public scenario (b) Threshold to estimate walk step

Fig. 7 – Walk pace and direction estimation. (a) gives an example in public scenarios. There are two users in the zone and other three users are entering the zone. (b) shows the results of resultant acceleration after noise removal.

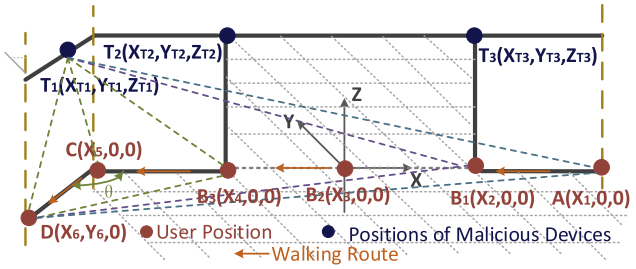


Fig. 8 – Target AP localization based on the movements of the user. The user first walks along a straight line and then make a turn. The initial position of the user is position A, B_1 is the user's first step, B_2 is the user's second step, and so on. There exists three malicious devices in the place and the positions of three malicious devices are separately T_1, T_2, and T_3.

digital passwords to pay bills, then the system should quickly advise the user to choose one direction to move to a safe zone. To make some specific recommendations such like the user's moving directions and steps, first the system should obtain the positions of malicious devices, then combines the calculated SNR values and positions to estimate candidate directions. In our system, the positions of possible malicious WiFi devices are mainly used to reduce the number of candidate directions, in other words, the system needs to get the most appropriate moving direction to avoid pure trial-and-error process. Thus, locating the positions of malicious devices is very essential.

RSS propagation model is used to locate malicious WiFi devices without any prior knowledge of the places. Unlike traditional localization methods using training process to obtain the unknown parameters, in this paper, inspired by Xu et al. (2014), the user's walk information can be used as reference nodes to solve the unknown parameters problem. In detail, as shown in Fig. 8, the start coordinate of the user's walk can be seen as (X_1, 0, 0), the positions of the following steps can be calculated using the method described in Section 6.3.1. The details of locating the malicious devices are shown in Algorithm 2.

Algorithm 2 Locating malicious devices.

Input:
S_{num}: Walk steps during the user's walk
$SL[]$: Walk length for each step
$SD[]$: Direction for each step
$RSS[]$: RSSvalues during the user's walk
$st[]$: Each step's time
N_{min} and N_{max}: Constraints of path-loss exponent
$search_{step}$: Search step of path-loss exponent
Output:
(X_i, Y_i, Z_i): Positions of all possible malicious devices

```
for i = 1 : S_num do
    coordinates KC[] ← getcoordinate(S_num, SL[], SD[])
    each step's RSSvalues rss[] ← getrss(S_num, st[], RSS[])
    conversion parameter P[] ← getparameter(rss[])
    conversion matrix A[] ← getmatrix(P[], KC[])
    conversion matrix B[] ← getmatrix(P[], KC[])
end for
for j = 1 : S_num − 1 do
    for n_j = N_min : search_step : N_max do
        first ratio1 r1[] ← getratio(P[], n_j)
        targets' coordinates C[] ← getcoordinate(A[], B[], n_j)
    end for
end for
second ratio r2[] ← getratio(KC[], C[])
if compare(r1[], r2[]) then
    optimal path-loss exponent n_opt ← findoptiaml(r1[], r2[])
    return optimal positions P_op[] ← findoptimal(n_opt, P[])
end if
if checkwrongdata(P[], rss[], KC[]) then
    Position P*[] ← calculatemean(P[])
else if
    Position P'[] ← deletewrongdata(P[])
end if
```

As analyzed in Xu et al. (2014), several known coordinates for reference nodes need to be obtained in advance. In our system, the positions during the user's walk can be seen as reference nodes. Also taking the scenario in Fig. 7a as an example, the walking pace $SL[]$, S_{num} and the direction $SD[]$ of the user first need to be mapped into reference nodes and their

positions can be calculated into the positions relative to the start position of the user's walk (line 2). As shown in Fig. 8, the user walks five steps and the first fourth steps are a straight line. If we assume that the coordinates of the start point of the walk is $(X_1, 0, 0)$, then the coordinates of the following steps are estimated using PDR algorithm (Lachapelle et al., 2006). Then the system will separate the RSS values *RSS*[] during the user's walk into the values *rss*[] for the reference nodes (line 3).

After obtaining the positions of reference nodes, the system will use a ratio approach to eliminate the transmitting power uncertainty, and then combines a search method for path-loss exponent to determine the positions of malicious WiFi devices. To be specific, in an anonymous environment, when the RSS values of reference nodes are obtained, a ratio can be calculated through the following equation:

$$\hat{d_1}/\hat{d_i} = (P_i/P_1)^{1/n_j} \tag{4}$$

Where $P_i = 10^{rss_i/10}$, and rss_i is the RSS values of each step, d_i is the distance between the $i-th$ walk step (see ith reference nodes in Fig. 8) and the malicious WiFi devices.

If there are multiple reference nodes and the path-loss parameter is known, the position of malicious device can be obtained using the following matrix:

$$A\theta = B \tag{5}$$

Where

$$A = \begin{bmatrix} 2(P_2^{\frac{2}{n}}x_2 - P_1^{\frac{2}{n}}x_1) & 2(P_3^{\frac{2}{n}}x_3 - P_1^{\frac{2}{n}}x_1) & \vdots & 2(P_m^{\frac{2}{n}}x_m - P_1^{\frac{2}{n}}x_1) \\ 2(P_2^{\frac{2}{n}}y_2 - P_1^{\frac{2}{n}}y_1) & 2(P_3^{\frac{2}{n}}y_3 - P_1^{\frac{2}{n}}y_1) & \vdots & 2(P_m^{\frac{2}{n}}y_m - P_1^{\frac{2}{n}}y_1) \\ 2(P_2^{\frac{2}{n}}z_2 - P_1^{\frac{2}{n}}z_1) & 2(P_3^{\frac{2}{n}}z_3 - P_1^{\frac{2}{n}}z_1) & \vdots & 2(P_m^{\frac{2}{n}}z_m - P_1^{\frac{2}{n}}z_1) \\ P_1^{\frac{2}{n}} - P_2^{\frac{2}{n}} & P_1^{\frac{2}{n}} - P_3^{\frac{2}{n}} & \vdots & P_1^{\frac{2}{n}} - P_m^{\frac{2}{n}} \end{bmatrix}^T$$

$$B = \begin{bmatrix} P_2^{\frac{2}{n}}(x_2^2 + y_2^2 + z_2^2) - P_1^{\frac{2}{n}}(x_1^2 + y_1^2 + z_1^2) \\ P_3^{\frac{2}{n}}(x_3^2 + y_3^2 + z_3^2) - P_1^{\frac{2}{n}}(x_1^2 + y_1^2 + z_1^2) \\ \cdots \\ P_m^{\frac{2}{n}}(x_m^2 + y_m^2 + z_m^2) - P_1^{\frac{2}{n}}(x_1^2 + y_1^2 + z_1^2) \end{bmatrix} \theta = \begin{bmatrix} X \\ Y \\ Z \\ S \end{bmatrix}$$

and (x_i, y_i, z_i) is the position of the $i-th$ walk step, $S = \sqrt{X^2 + Y^2 + Z^2}$. (X, Y, Z) is the position of malicious device.

After obtaining the coordinates of reference nodes, the system will calculate the conversion parameter *P*[] using the RSS values *rss*[] for each reference node (line 4); calculate the conversion matrix *A*[] and *B*[] using the conversion parameter *P*[] and known coordinates *KC*[] of reference nodes (line 5 and line 6).

For a given path-loss exponent n_j, we then calculate the first ratio *r*1[] using Eq. (4) (line 10) and the second ratio *r*2[] using the estimated positions *C*[] and known coordinates *KC*[] (line 14), and then will find an optimal path-loss exponent n_{opt} which the first ratio can best match the second ratio (line 16) and the optimal path-loss exponent n_{opt} will represent a optimal position P_{op}[], which is the position of malicious device (line 17).

6.3.3. *Safe region guidance*

Our system dynamically evaluates the risk of the current location and gives the user guidance, and the details are shown in Algorithm 3. Our system constantly assesses the risk and will

Algorithm 3 Decision making and safe region guidance.

Input:
P[]: Positions of malicious devices
*M*1[]: Regression model of SNRvalues of gestures and success rate
*R*1[]: Relationship between SNRvalues of gait and gestures
CM[]: CSImeasurements for gait
P_m: User's current positions

Output:
D_m: Towards direction
L_m: Walk Length

1: *SNRvalues of gait* $S_g \leftarrow getSNR(CM[])$
2: *SNRvalues of gestures* $S_h \leftarrow getSNR(R1[], S_g)$
3: **for** S_h **do**
4: *success rate ra*[] $\leftarrow getrate(S_h, M1[])$
5: **if** *checkrate*(*ra*[]) **then**
6: *Decision Making* **struct** *DM*1[] $\leftarrow decide(ra[])$
7: *number of malicious devices* $P_{num} \leftarrow getnumber(P[])$
8: **if** $P_{num} == 1$ **then**
9: *Direction* $D_m \leftarrow getoppsitedirection(P_m, P[])$
10: **else if** $P_{num} > 1$ **then**
11: *Malicious direction D*[] $\leftarrow getdirection(P_m, P[])$
12: *Direction* $D_m \leftarrow getoppsitedirection(D[])$
13: **end if**
14: *Length* $L_m \leftarrow getlength(S_g, R1[], M1[])$
15: **else if** !*checkrate*(*ra*[]) **then**
16: *Decision Making* **struct** *DM*2[] $\leftarrow decide(ra[])$
17: *Direction* $D_m = 0$; *Length* $L_m = 0$
18: **end if**
19: **end for**
20: **for** D_m **do**
21: **if** *checkdirection*(D_m, *R*[]) **then**
22: *Direction* $D_m \leftarrow continuedirection(D_m, P_m, P[])$
23: **else if** !*checkdirection*(D_m, *R*[]) **then**
24: *Direction* $D_m \leftarrow updatedirection(P_m, P[])$
25: **end if**
26: **end for**

repeat steps 1 to 3 if the risk becomes high due to the change of situations, e.g. when the adversary changes the antennae directions or signal strength of the wireless devices.

Prior works show that the multiple wireless devices can improve the attack success rate (Abdelnasser et al., 2015; Wang et al., 2015b; Wang et al., 2014b). Thus, when guiding the user to a safe region, first the system needs to detect how many malicious devices in the public places. Take Fig. 10a for example, there are two malicious WiFi devices and the user stands in the position *P*, which is the risk region for malicious WiFi device 1 and the safe region for malicious WiFi device 2. Thus, the user needs to walk towards the common opposite direction of the two malicious WiFi devices, such as directions *T*1 and *T*6.

Recall that the system decides whether the user is in a safe zone according to the relationship between SNR values of gestures and success rate. However, the SNR values of gestures

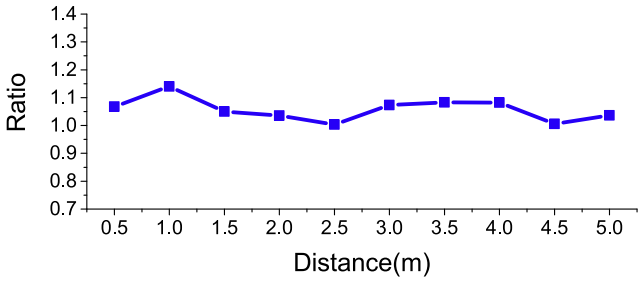


Fig. 9 – Relationship between SNR values of human gaits and gestures. The experiments are done under different distances and record the SNR values of human gaits and gestures. The ratio is calculated using the SNR values of human gaits divided by the SNR values of human gestures.

can be obtained only when the user draws gestures, which is inconvenient for the user. Based on that, we do an experiment to measure the relationship between the human gaits and gestures, Fig. 9 shows the results. We observe that the SNR ratio of human gaits and gestures keeps around 1 in different distances. Thus, the system can leverage the SNR values of human gaits to estimate the success rate to make a decision.

Our system then guides the user to a safe zone using the relationship of SNR values between gaits and gestures, as shown in Fig. 9. Note that the user just needs to walk several steps to be in a safe zone, as shown in Fig. 10b, the user just needs to walk several steps towards P_1 or P_2 to a safe zone.

In order to give the user a real-time guidance, the system will monitor the SNR values of human gaits in real time. If the user does not follow the current guidance, the system will change another guide line for the user.

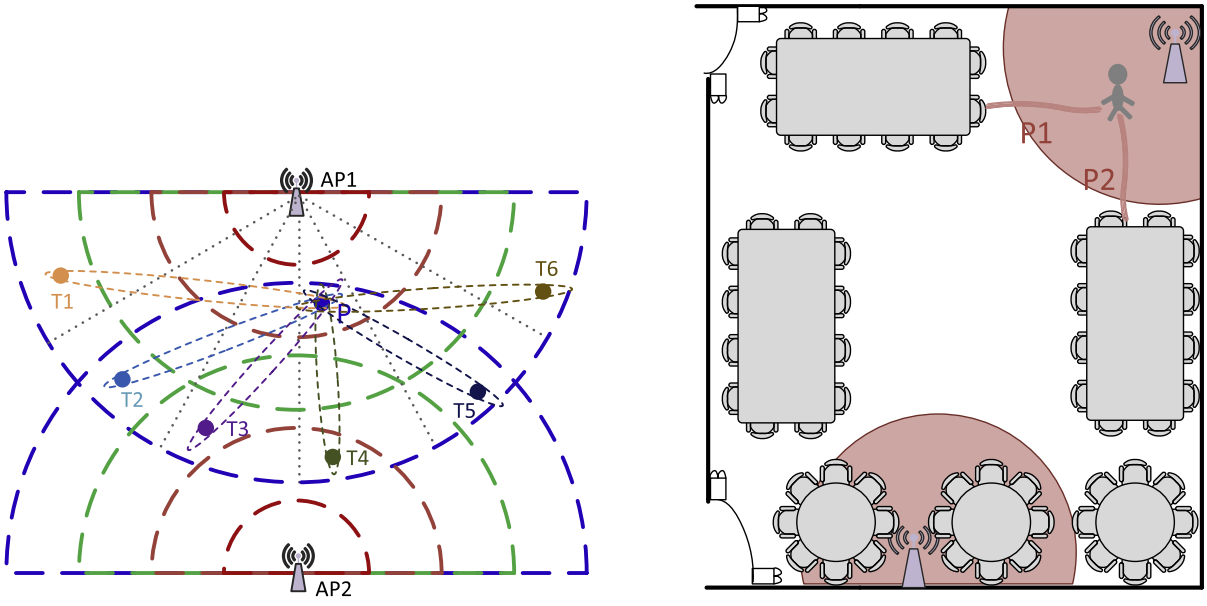
7. Experimental Setup

Scenarios We evaluate our approach in various indoor environments, described as follows. The first scene is an entrance hall of a building, where the multipath of WiFi signals is weak. The second scene is an indoor classroom with a size of 9 m × 12 m, where the multipath effects are more severe. The last scene is a typical indoor environment with a size of 3.5 m × 7 m, where has furniture and the strongest multipath effects.

Attack Setup. We implemented the attack described in Li et al. (2016) and Zhang et al. (2016). We used two wireless devices: a laptop with Intel 5300 NIC and a TP-Link WR 1043ND WiFi router. The laptop is used as a receiver to collect the CSI measurements and the WiFi router is used as a transmitter. The attack is evaluated in an 802.11 n wireless environment. The wireless devices are placed at 0.5 meter to 5 meters away from the user. To measure the CSI, the wireless receiver and transmitter exchange ICMP packets at a data rate of 1000 packets per second. This data rate is used in prior work (Wu et al., 2015).

Target devices. Our target devices are Android smartphones. We tested our approach on a Xiaomi MI4 phone and a Samsung Galaxy S7 phone. It is to note that our system makes use of the acceleration sensor which is a standard configuration for modern smartphones.

Use cases. We evaluate our approach by applying it to protect Android pattern locks and PINs. We used 15 graphical patterns, which are often used as login passwords of banking system and payment passwords of Alipay or Wechat, as shown in



(a) Multiple malicious WiFi devices (b) Guidance in real scenario

Fig. 10 – Safe region guidance. In (a), there are two malicious devices in the public place, the user stands in the position P, only the directions T1, T2, T5, T6 will lead the user to a safe zone. (b) gives a case of the guidance. The user first is in a risk zone, and the system will give several alternative directions for the user.

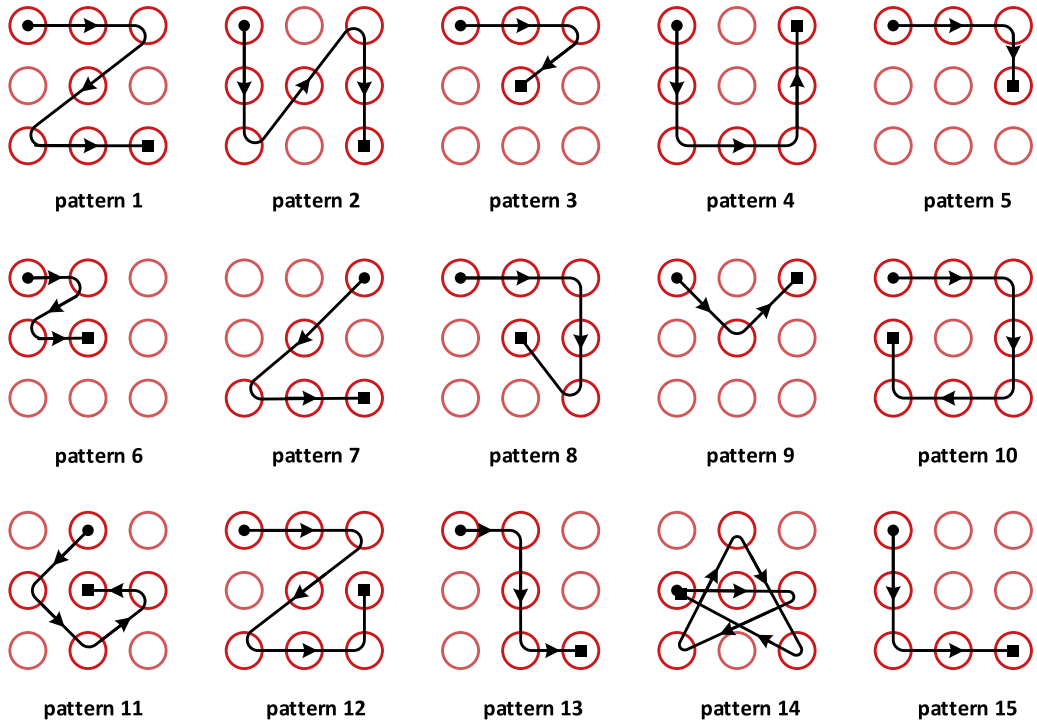


Fig. 11 – The Android locking patterns used in the evaluation.

Fig. 11. As mentioned in Uellenbeck et al. (2013), there are several typical strategies used by the participants, such as people often picked the top left corner as a starting point and prefer straight lines, the graphical patterns used in the experiments are chosen according to the strategies and they were also used in prior work (Ye et al., 2017; Zhang et al., 2017). The structure of these patterns ranges from simple to complex. We use the Android 3×3 native pattern grid. We also test our approach on PINs using a Xiaomi MI4 where the key number from 0 to 9 is separately pressed.

Participants. We recruited four users to participate in our experiments with the approval of the research ethics board (REB) of the host institution. Each participant was given the opportunity to practice a pattern or password several times, so that they could draw the pattern at their natural speed. On average, this practice session took 10 trials per user per pattern/password. When entering the information, some participants sat, while others stood, some hold the device by hands, while others placed it on a table. Each case was evaluated on two target devices for given distances. For each setting, we replayed the attacking process five times and reported the average success rate.

Prototype Countermeasure. Our prototype system is implemented as an operating system background service for the Android system. We test it on the latest Android 7.1 Nougat operating system. Since our software does not rely on specialized hardware, it can be ported to other mobile operating systems including IOS and Windows 10 Mobile.

8. Experimental results

In this section, we first reproduce the CSI-based attack and demonstrate the relationship between the success rate and the SNR values. We then analyze the localization results in different environments. Finally, we use a case study to demonstrate the effectiveness of our approach in a real world scenario.

8.1. *Impact of SNR values*

Result 1: *The quality of the SNR strongly correlates to the success rate of the CSI-based attack.*

In this experiment, we use the CSI measurements of patterns and PIN-based passwords collected from four participants to calculate the SNR. This experiment is carried out in three scenarios described in Section 7. The attack uses the method discussed in Li et al. (2016) to recover the passwords.

Fig. 12 shows that the success rate becomes higher with the increase of SNR values. This is expected high SNR values lead to high quality CSI measurements, which demonstrates that SNR is strongly correlated to the success rate.

8.2. *Evaluation of localization results in different environments*

Result 2: *Our localization method is robust to different environments.*

In this experiment, we test three scenarios to demonstrate the robustness of our localization method. Fig. 13a shows the results of three scenarios described in Section 7. As can be

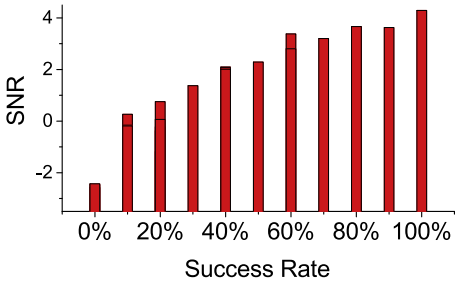


Fig. 12 – The impact of SNR on success rate and user network experience. The experiments is done under different distances.

seen from the diagram, the error increases as the mutipath effect becomes stronger. In the environment with the strongest multipath effect, the library, the error is less than 1 m. After analyzing the localization results, we observe that multipath has a negative impact on the localization accuracy and stronger multipath will lead to higher localization error. Note that the localization error is not significant in our problem, as the largest error is just only two steps for ordinary users, and the system can let the user walk two more steps while guiding to the safe zone.

8.3. *Impact of multiple malicious devices on the localization results*

Result 3: *The number of malicious WiFi devices has no impact on the localization accuracy.*

In this experiment, we investigate whether the number of malicious WiFi devices has a significant impact on the localization accuracy. To answer this question, we separately deploy one, two, and three WiFi devices in different environments to do localization.

Fig. 13b shows the localization results when an attacker deploys multiple malicious devices. Similar to the localization results when there is just one malicious device, our approach can also achieve a high accuracy for locating multiple malicious devices. After having a close look at Fig. 13b, we found that the number of malicious devices has little impact on the localization error. This is because the RSS values are measured in parallel across wireless devices without interference.

8.4. *Detecting changing power*

Result 4: *The attacker may dynamically change the WiFi signal strength to confuse the protection scheme but our approach can detect over 90% of the changing power when the attacker changes the wireless power of the malicious devices.*

Recall that the success rate requires high-quality CSI measurements, the attacker may dynamically change the power of malicious devices to confuse the protection scheme. Therefore, we will need to detect the change of power. However, the user cannot obtain the malicious devices' power directly. Instead, we use RSS values to detect the changing power. In this experiment, RSS values were recorded on both 2.4 GHz and 5GHz under different power conditions. The experiments are done using TL-WDR7500 router from various distances.

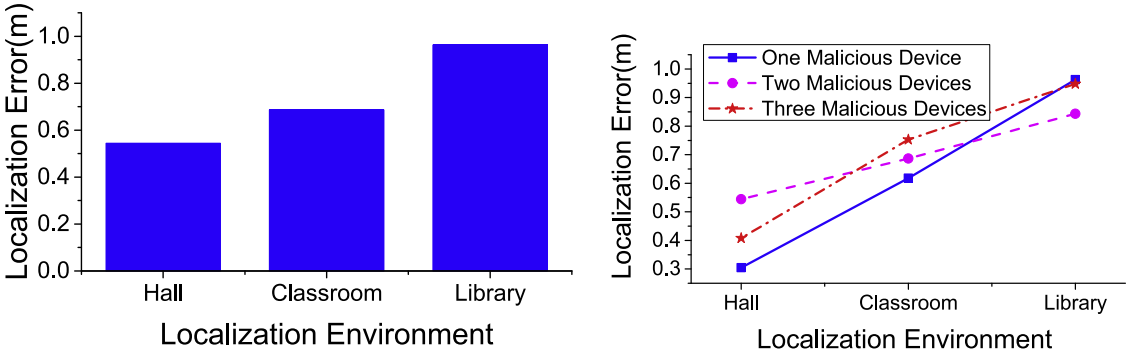
Fig. 14a shows that when the attacker changes the power, there will appear a sharp rise or fall for RSS values, and then the value will settle down. When someone walks by, the RSS values will fluctuate in a range. Thus, RSS values can be used to detect the changing power.

Fig. 14b shows the detection results of changing power on 2.4 GHz and 5 GHz. For the scenario where the power is changed to a lower value, we can detect 90% of the cases for 2.4 Ghz and 83.3% of the cases for 5 Ghz. For the scenario where the power is boosted to a higher value, we can detect 90% of the cases for 2.4 Ghz and 100% of the cases for 5 Ghz. After guiding the user to a safe region, the system will detect the RSS values in real-time and when power changing is detected, the risk will be reassessed.

8.5. *Safety after taking countermeasures*

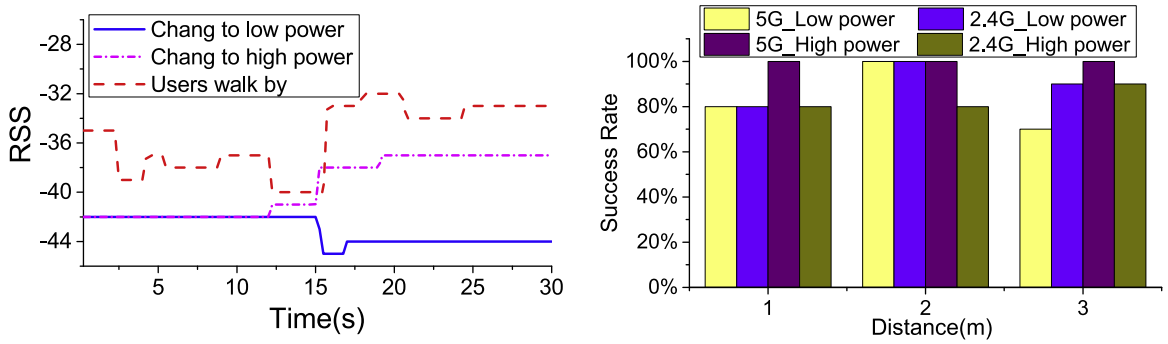
Result 5: *The CSI-based attack is unlikely to succeed after taking protection countermeasures.*

For the body noise addition countermeasures, the CSI waveforms after taking the countermeasures are shown in Fig. 15. We can see from Fig. 15a that the CSI-based attack



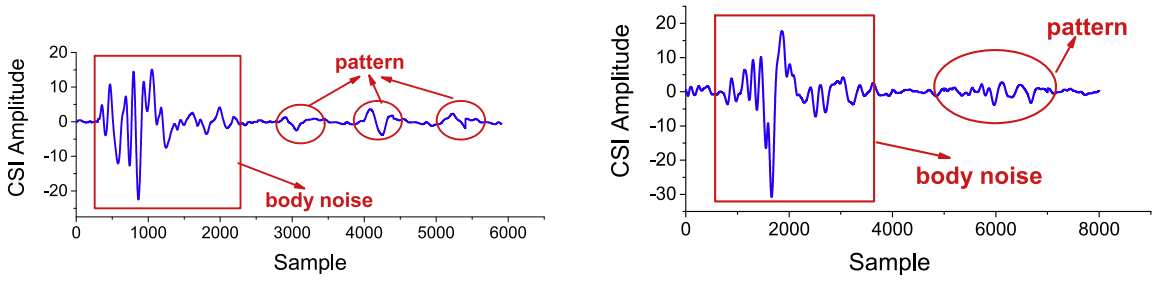
(a) Different environments (b) Multiple malicious devices

Fig. 13 – Localization error for multiple malicious devices in different environment. In (a), there are one malicious device in the hall, classroom and library. In (b), there are separately one, two, three malicious devices in hall, classroom and library.



(a) Power change vs user's walk by (b) Accuracy of detection

Fig. 14 – Detection on sudden power change. (a) shows the difference of RSS values between the user's walk and power change, (b) shows the accuracy of power change detection under different frequency channels.



(a) Succeeds when adding additional body noises (b) Fails when adding additional body noises

Fig. 15 – The comparison of the CSI measurements when adding additional body noises. In the experiments, the user enters three patterns. In (a), the CSI-based attack succeeds when adding body noises. In (b), the attack fails when adding body noises.

can still succeed after adding body noise, and that is because although the CSI waveforms of body noise can confuse the attacker, however, when the positions and directions before and after adding body noise don't change too much, an experienced attacker can still distinguish the CSI waveforms of body noise from CSI waveforms of patterns because of the difference between them. On the contrary, when they change too much, the CSI waveforms of patterns will be completely different from the attacker's prior measurements, and the attacker will not decode the pattern successfully using the CSI measurements, as shown in Fig. 15b. Thus, adding body noise countermeasures sometimes doesn't work. Fig. 16a shows the results of success times after adding body noise. The results show sometimes, even when the user adds body noise, the CSI-based attack can still succeed.

For safe region guidance, because the safe region is the place where CSI measurements are noisy. In order to demonstrate the safety in a safe zone, we asked our participants to input 15 graphical unlock patterns on Android smart phones in the safe region, and each pattern is drawn five times and we collect the CSI measurements to recover the patterns. Fig. 16b shows the results of the success times of CSI-based attack in safe region. We can observe that the success times of 15 unlock patterns are all 0, thus, the system can protect the user's sensitive information effectively.

8.6. *Impact on the user's network experience*

Result 6: *Our protection methodology has little impact on the user's network experience.*

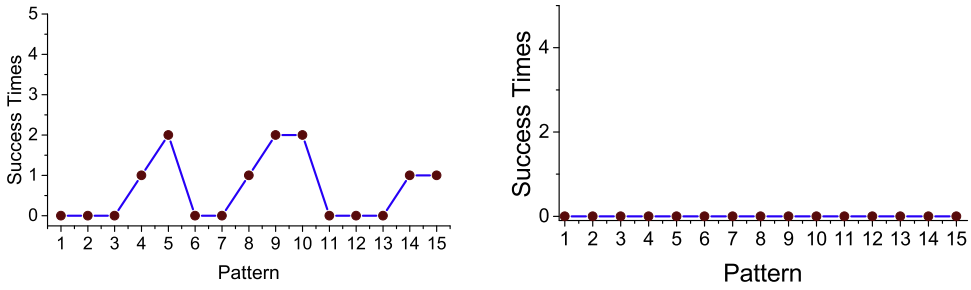
We know that the further distance will lead to bad network experience. In this experiment, we would like to know whether the safe region will have a negative impact on the user network experience. To do so, we evaluate our approach in different distances. We record the packet loss rate and delay, which are two important factors that can affect the quality of wireless communication.

Fig. 17 shows that the packet loss rate is below 1% and the delay is less than 5 ms when the distance between the user and the wireless device is from 0.5 m to 5 m. According to experience, when the packet loss rate is less than 8% and the delay is less than 200 ms, the wireless communication quality will not affect the user's network experience. Thus, the user's network experience will not be affected in safe region.

8.7. *Case Studies*

Result 7: *The case studies for different participants in different scenarios confirm the effectiveness of our system.*

In this experiment, we asked our participants to draw 15 unlock patterns five times separately in a risk region and a safe region in four scenarios, described in Section 7.



(a) The success times of adding body noise (b) The success times of safe region guidance

Fig. 16 – The comparison of success times of CSI-based attack after taking different countermeasures. (a) shows the success times of CSI-based attack when adding body noises and (b) shows the success times when the user is in a safe zone. Each kind of experiment is done five times.

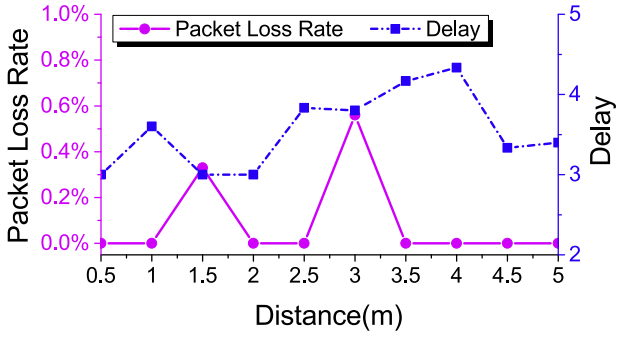


Fig. 17 – User network experience.

To simulate the real-world attack scenarios, the participant is not required to sit at the certain positions or input the sensitive information at the certain time. Instead, we collect CSI measurements when the user takes out the device to input the sensitive information.

Fig. 20a shows the successful times of recovering the patterns in the risk region for four participants in the first scenario. We observe that the maximum is five while the minimum is 2. This suggests that the attacker can successfully obtain the user's sensitive information when the user is in the risk region. Fig. 18 shows the metrics of precision, recall and accuracy when the four participants are in the risk zone.

Fig. 20b shows the successful times of recovering the patterns in the safe region in four scenarios. We observe that the successful times in four scenarios are almost 0, expect the pattern 4 and pattern 10 in scenario 1, and pattern 14 in scenario 3. That is because the attack may occur during the sudden change of SNR values. Before the sensitive input window, the SNR values are detected to be low because of the surrounding noise and the user will input the sensitive information. However, at the start of the input window, the SNR values suddenly increase and the attacker launches the attack at this moment. That may lead to the success of the attack. However, in actual scenarios, it is hard for the attacker to grasp the moment. Fig. 19 shows the metrics of precision, recall and accuracy when the four participants are in the safe zone.

Fig. 20c how often a CSI-based attack can be successfully launched when the attacker uses multiple APs. We observe that almost all the patterns can be recognized by the attacker as multiple APs can improve the accuracy. When the attack uses multiple APs to launch the CSI-based attack, there may not exist a nearby safe zone. In such scenarios, our system would suggest the user not to enter sensitive information in the public place. This is also discussed in Section 10.

In this experiment, we also investigate the success rate with noisy CSI measurements. We try to answer the question: Will the object movements in surrounding environments affect the success rate of CSI-based attacks. To answer this question, we ask another participant to walk near the target user when the victim starts entering the sensitive information. In this experiment, the user is 0.5m far away from the malicious WiFi device.

Fig. 21a shows the results of clean and noisy CSI measurements collected during the time the user enters the sensitive information. We observe that the clean CSI measurements only contain the CSI information that comes from the user's finger motions; but the noisy CSI measurements contain not only the finger's information, but also the CSI information that comes from the surrounding objects. We discover that the impact of moving surroundings on CSI measurements is greater than that of the finger motions. Another obvious difference between the clean and noisy CSI measurements is the CSI amplitude, which is used to recover the sensitive information. However, it is difficult for the attacker to extract the CSI measurements of finger motions from noisy CSI measurements. Therefore, the attacker will be unable to recover the sensitive information using the noisy CSI measurements and the CSI-based attack will not succeed.

Our prototype system can be implemented as a background service for Android operating system, and it can run with Alipay Pay and Wechat Pay, as shown in Fig. 21b and 21 c. The system monitors specific system events to detect sensitive inputs. When a sensitive event is detected, the system will be called to give the user a warning and guide the user to a safe region. The system will give the user a direction to walk and detect whether the user is walking according to guidance lines, when the user does not follow the guidance, the system will give the user a reminder and choose another direction for the user.

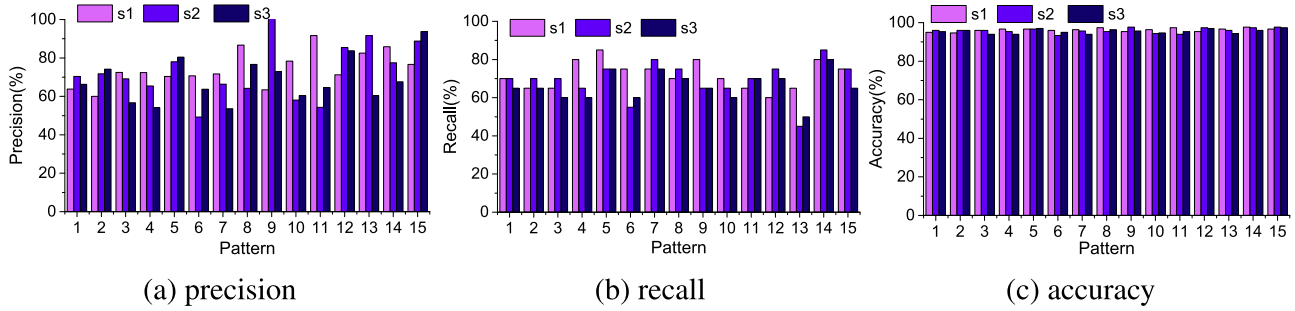


Fig. 18 – The metrics of CSI-based attack in a risk zone. (a) shows precision for four participants in risk zone in three scenarios, (b) shows recall for four participants in risk zone in three scenarios and (c) shows accuracy for four participants in risk zone in three scenarios.

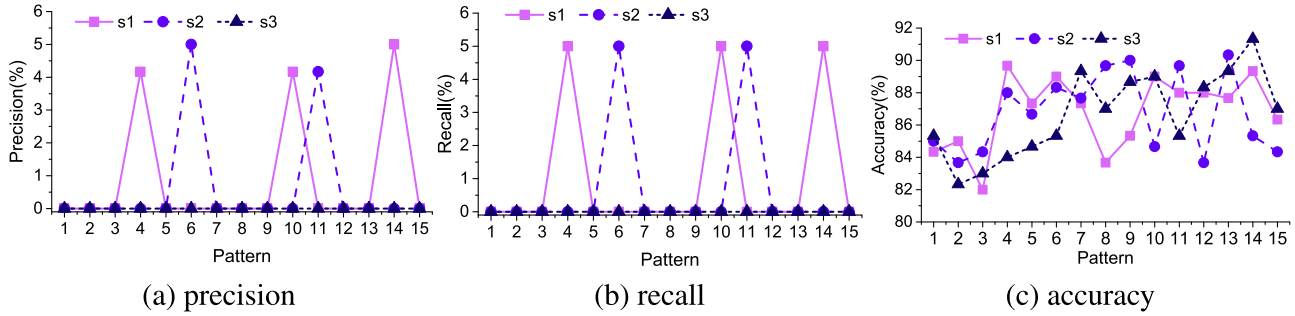


Fig. 19 – The metrics of CSI-based attack in a safe zone.(a) shows precision for four participants in safe zone in three scenarios, (b) shows recall for four participants in safe zone in three scenarios and (c) shows accuracy for four participants in safe zone in three scenarios.

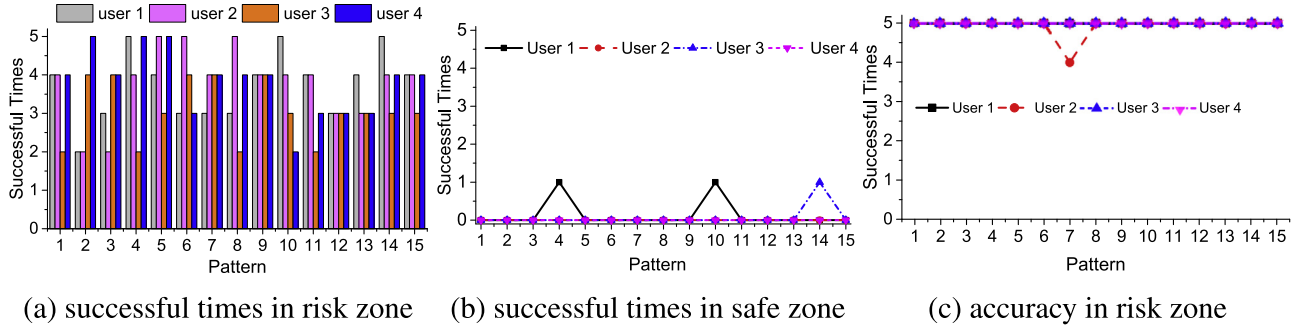


Fig. 20 – The successful times of CSI-based attack for four participants in a risk zone and a safe zone in the first scenario.(a) shows the successful times of the attack when four participants is in a risk zone, (b) shows the successful times of the attack when four participants is in a safe zone, (c) shows accuracy for four participants in risk zone using multiple APs.

8.8. *User's acceptability study*

We would like to evaluate whether users are willing to use our system and whether users feel comfortable while using our system to protect sensitive information. We recruited 20 participants in this study. All the participants were asked to participate and none was compensated for the experiments.

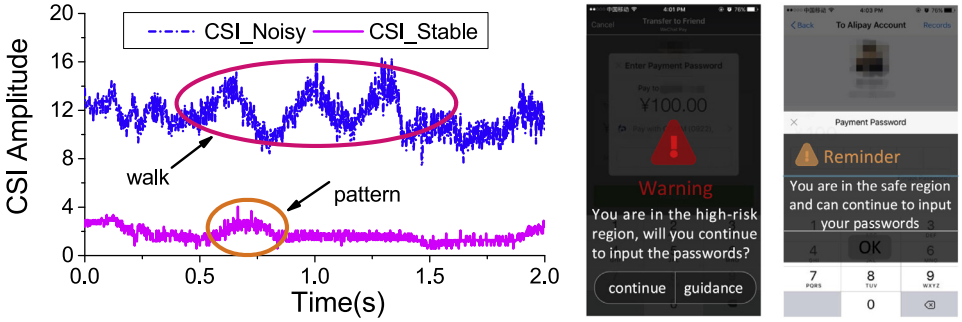
The study is conducted as follows: we first show the participants detailed information about CSI-based attacks and our system:

1. What CSI-based attack does and how the wireless signals recognize the users' sensitive information.

2. What the channel state information looks like and how it is collected and stored.

3. What our system does and how it can defeat CSI-based attacks.

Then the participants were asked two questions. First, we asked them whether they are willing to use the system to protect their sensitive information and most of the participants give a positive answer. We also asked the participants to choose one countermeasure to protect sensitive information, a safe zone or adding body noise. All of the participants choose the safe zone. The above results indicate that the system has a high acceptance.



(a) Clean CSI measurements and noisy CSI measurements (b) Wechat Pay (c) Alipay Pay

Fig. 21 – Case studies. (a) shows the case of the CSI measurements when there exists moving surroundings or not, and different lines show the CSI measurements across stable environments and noisy environments. (b) and (c) gives two cases of system interaction, and the systems can run across sensitive information-related applications.

9. Related work

Our work lies at the intersection between gesture recognition using wireless signals and the indoor localization using AP and mobile smart devices.

CSI -based Gesture Recovery. Wang et al. (2017) study how CSI can be leverage to detect fall. Xi et al. (2014) and Yang et al. (2013) demonstrate the capability of CSI to detect the person's number and positions. Melgarejo et al. (2014) leverage directional antenna to recognize the fine-grained gestures. Inspired by that, Ali et al. (2015) achieve keystroke detection and Wang et al. (2014a) achieve lip-reading. Li et al. (2016) crack the digital passwords of Alipay if the user connects to the attacker's malicious WiFi devices, and Zhang et al. (2016) demonstrate graphical unlock passwords of smart devices can be obtained even the user does not connect to any malicious WiFi devices.

RSS -based indoor localization. Rai et al. (2012) demonstrate that RSS-localization based on fingerprints needs to be trained for every new space and each time when there is a significant change in a given place, thus, the method is not proper in this paper. Based on that, Goswami et al. (2011) propose a calibration method, however, the method need to control over the APs and have prior knowledge of their locations. Chintalapudi et al. (2010) improve the calibration method for RF modeling using AP and inertial sensors of smart devices with some known positions. In this paper, we seek for a localization method that doesn't need any prior knowledge about the place and can locate the malicious WiFi devices accurately.

CSI -related attacks. Jiang et al. (2013) leverage the unique characteristics of CSI to verify the authenticity of MFs, and propose a WiFi management frame source authentication system. Wang et al. (2018) propose a novel Sybil attack detection based on CSI, and the detection algorithm can tell whether the static devices are Sybil attackers. Tung et al. (2014) evaluate sniffing attack (enables concurrently transmitting malicious clients to eavesdrop other ongoing transmissions) and power attack (enables malicious clients to enhance their own capacity at the expense of others') and they propose a novel CSI feedback system to prevent CSI forging without requiring any modification at the client side.

10. Discussion

Naturally there is room for further work and improvements. We discuss a few points here.

Hidden WiFi hotspots An attacker may use hidden WiFi hotspots (i.e., the SSIDs of the wireless devices are not public available) to launch the attack. Our current implementation does not detect hidden WiFi hotspots. However, they can be discovered using the the method described in Fazal et al. (2010). Once these hidden WiFi hotspots are detected, our approach remains applicable.

Multiple malicious WiFi devices An attacker can also increase the chance for successfully launching the attack by using multiple malicious devices. In this scenario, it is possible that there exists no safe region in an indoor environment. If this happens, our system will suggest the user to use body movements to introduce some artificial noises or not to enter sensitive information at all.

Other identity verification mechanism Our goal is provide countermeasures for CSI-based attacks, which is useful to obtain the sensitive information when the target user enters graphical passwords or digital passwords. That is because the success of CSI-based attack lies in the fact that the mobile user's finger movements or gestures will affect the CSI measurements of the wireless signal, an attacker can recover the user's input with a high success rate by analyzing the affected CSI measurements. For other identity verification mechanisms, such as face recognition and fingerprints, the CSI-based attack does not work.

11. Conclusion

This paper has presented a novel countermeasure for CSI-based attacks. We exploit the observation that the success of the attack requires having a clean CSI measurement. We define a signal to noise metric to measure the quality of the CSI readings from the attacker's perspective, and use this metric to quantify how likely a CSI-based can be successfully

launched. Given the user's current location and the surrounding environment, our scheme evaluates the risk of CSI-based attacks. If the risk is considered to be high, it then directs the user to move to a safe location. We evaluate our approach by applying it to protect Android pattern lock and keystrokes. Our evaluation is conducted in various typical indoor environments. Experimental results show that the proposed countermeasure can successfully protect users against CSI-based attacks in public places.

Acknowledgment

This work was partially supported by projects of the National Natural Science Foundation of China (Nos. 61672427, 61672428); the International Cooperation Foundation of Shaanxi Province, China (No. 2017KW-008); the Service Special Foundation of Shaanxi Province Department of Education (No. 16JF028); the Research Project of Shaanxi Province Department of Education (No. 15JK1734); the Key Research Project of Shaanxi Province of China (No. 2017GY-191); the Research Project of CCF-NSFOCUS Kunpeng Science Foundation; the China Scholarship Council (201806970007); the U.K. Engineering and Physical Sciences Research Council under Grants EP/M01567X/1 (SANDeRs) and EP/M015793/1 (DIVIDEND); and a Royal Society International Collaboration Grant (IE161012).

REFERENCES

Abdelnasser H, Youssef M, Harras KA. Wigest: a ubiquitous wifi-based gesture recognition system. In: Proceedings of the IEEE conference on computer communications (INFOCOM), 2015. IEEE; 2015. p. 1472–80.

Ali K, Liu AX, Wang W, Shahzad M. Keystroke recognition using wifi signals. In: Proceedings of the international conference on mobile computing and networking; 2015. p. 90–102.

Aviv AJ, Gibson KL, Mossop E, Blaze M, Smith JM. Smudge attacks on smartphone touch screens. Woot 2010;10:1–7.

Chintalapudi K, Padmanabha Iyer A, Padmanabhan VN. Indoor localization without the pain. In: Proceedings of the international conference on mobile computing and networking, MOBICOM 2010, Chicago, Illinois, Usa, September; 2010. p. 173–84.

Daniel H, Anmol S, David Wetherall WH. Linux 802.11n CSI tool. https://github.com/dhalperi/linux-80211n-csitool; 2012.

Fazal L, Kappes M, Krishnakumar AS, Ganu SN, Krishnan P. Detection of hidden wireless routers. 2010. US Patent 7,840,698.

Goswami A, Ortiz LE, Das SR. WIGEM: a learning-based approach for indoor localization. In: Proceedings of the seventh conference on emerging networking experiments and technologies. ACM; 2011. p. 3.

Halperin D, Hu W, Sheth A, Wetherall D. 802.11 with multiple antennas for dummies. SIGCOMM Comput Commun Rev 2010;40(1):19–25.

Jiang Z, Zhao J, Li XY, Han J, Xi W. Rejecting the attack: Source authentication for wi-fi management frames using CSI information. Proc. IEEE Infocom 2013;12(11):2544–52.

Lachapelle G, Godha S, Cannon ME. Performance of integrated HSGPS-IMU technology for pedestrian navigation under signal masking2006.

Li F, Zhao C, Ding G, Gong J, Liu C, Zhao F. A reliable and accurate indoor localization method using phone inertial sensors. In: Proceedings of the 2012 ACM conference on ubiquitous computing. ACM; 2012. p. 421–30.

Li M, Meng Y, Liu J, Zhu H, Liang X, Liu Y, Ruan N. 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; 2016. p. 1068–79.

Liu J, Wang Y, Kar G, Chen Y, Yang J, Gruteser M. Snooping keystrokes with mm-level audio ranging on a single phone. In: Proceedings of the 21st annual international conference on mobile computing and networking. ACM; 2015. p. 142–54.

Melgarejo P, Zhang X, Ramanathan P, Chu D. Leveraging directional antenna capabilities for fine-grained gesture recognition. In: Proceedings of the 2014 ACM international joint conference on pervasive and ubiquitous computing. ACM; 2014. p. 541–51.

Rai A, Chintalapudi KK, Padmanabhan VN, Sen R. Zee: zero-effort crowdsourcing for indoor localization. In: Proceedings of the 18th annual international conference on mobile computing and networking. ACM; 2012. p. 293–304.

Raij A, Ghosh A, Kumar S, Srivastava M. Privacy risks emerging from the adoption of innocuous wearable sensors in the mobile environment. In: Proceedings of the SIGCHI conference on human factors in computing systems. ACM; 2011. p. 11–20.

Shukla D, Kumar R, Serwadda A, Phoha VV. Beware, your hands reveal your secrets!. In: Proceedings of the 2014 ACM SIGSAC conference on computer and communications security. ACM; 2014. p. 904–17.

Tung YC, Han S, Chen D, Shin KG. Vulnerability and protection of channel state information in multiuser mimo networks 2014;22(3):775–86.

Uellenbeck S, Wolf C, Holz T. Quantifying the security of graphical passwords:the case of android unlock patterns. In: Proceedings of the ACM SIGSAC conference on computer & communications security; 2013. p. 161–72.

Wang C, Zhu L, Gong L, Zhao Z, Yang L, Liu Z, Cheng X. Accurate sybil attack detection based on fine-grained physical channel information. Sensors 2018.

Wang G, Zou Y, Zhou Z, Wu K, Ni LM. We can hear you with wi-fi!. In: Proceedings of the ACM international conference on mobile computing and networking; 2014a. p. 593–604.

Wang H, Lai TTT, Roy Choudhury R. Mole: motion leaks through smartwatch sensors. In: Proceedings of the 21st annual international conference on mobile computing and networking. ACM; 2015a. p. 155–66.

Wang W, Liu AX, Shahzad M, Ling K, Lu S. Understanding and modeling of wifi signal based human activity recognition. In: Proceedings of the 21st annual international conference on mobile computing and networking. ACM; 2015b. p. 65–76.

Wang Y, Liu J, Chen Y, Gruteser M, Yang J, Liu H. E-eyes: device-free location-oriented activity identification using fine-grained wifi signatures. In: Proceedings of the 20th annual international conference on mobile computing and networking. ACM; 2014b. p. 617–28.

Wang Y, Wu K, Ni LM. Wifall: device-free fall detection by wireless networks. IEEE Trans. Mob. Comput. 2017;16(2):581–94.

Weinberg H. Using the adxl202 in pedometer and personal navigation applications. Analog Devices AN-602 Application Note 2002;2(2):1–6.

Wu C, Yang Z, Zhou Z, Qian K. Phaseu:real-time los identification with wifi. In: Proceedings of the computer communications; 2015. p. 2038–46.

Xi W, Zhao J, Li XY, Zhao K, Tang S, Liu X, Jiang Z. Electronic frog eye: counting crowd using wifi. In: Proceedings of the Infocom, 2014 IEEE; 2014. p. 361–9.

Xu Y, Zhou J, Zhang P. Rss-based source localization when path-loss model parameters are unknown. IEEE Commun. Lett. 2014;18(6):1055–8.

Yang Z, Zhou Z, Liu Y. From RSSI to CSI: Indoor localization via channel response. ACM Comput. Surv. (CSUR) 2013;46(2):25.

Ye G, Tang Z, Fang D, Chen X, Kim KI, Taylor B, Wang Z. Cracking android pattern lock in five attempts. Proceedings of the network and distributed system security symposium, 2017.
Yue Q, Ling Z, Fu X, Liu B, Ren K, Zhao W. Blind recognition of touched keys on mobile devices. In: Proceedings of the 2014 ACM SIGSAC conference on computer and communications security. ACM; 2014. p. 1403–14.
Zeng Y, Pathak PH, Mohapatra P. Wiwho: Wifi-based person identification in smart spaces. In: Proceedings of the ACM/IEEE international conference on information processing in sensor networks; 2016. p. 1–12.
Zhang J, Tang Z, Li R, Chen X, Gong XQ, Fang D, Wang Z. Protect sensitive information against channel state information based attacks. Proceedings of the IEEE international conference on computational science and engineering, 2017.
Zhang j, Zheng X, Tang Z. Privacy leakage in mobile sensing: your unlock passwords can be leaked through wireless hotspot functionality. Mob. Inf. Syst. 2016;2016(2):1–14.
Zhang Y, Xia P, Luo J, Ling Z, Liu B, Fu X. Fingerprint attack against touch-enabled devices. In: Proceedings of the ACM workshop on security and privacy in smartphones and mobile devices; 2012. p. 57–68.
Zhu T, Ma Q, Zhang S, Liu Y. Context-free attacks using keyboard acoustic emanations. In: Proceedings of the 2014 ACM SIGSAC conference on computer and communications security. ACM; 2014. p. 453–64.

Jie Zhang is currently working toward the Ph.D. degree in Northwest University. Her research interests include gesture recognition based wireless signals and wireless security. Her address is School of Information Science and Technology, Northwest University, Xi'an, 710127, P.R. China and email address is jz@stumail.nwu.edu.cn.

Zhanyong Tang received his Ph.D. degree in Computer Software and Theory from Northwest University, Xi'an, China in 2014. Now, he is an associate professor in the School of Information Science and Technology, Northwest University. His research interests include network and information security, software security and protection, localization, and Wireless Sensor Network. His address is School of Information Science and Technology, Northwest University, Xi'an, 710127, P.R. China and email address is zytang@nwu.edu.cn.

Meng Li received the BE degree in Software Engineering from Northwest University, Xi'an, China in 2017. She is currently working toward the MS degree in Software Engineering. Her research interests include gesture recognition based wireless signals. Her address is School of Information Science and Technology, Northwest University, Xi'an, 710127, P.R. China and email address is lijmeng@stumail.nwu.edu.cn.

Dingyi Fang received his Ph.D. degree in Computer Application Technology in 1983 from Northwestern Polytechnical University, Xi'an, China. His current research interests include mobile computing and distributed computing systems, network and information security, and wireless sensor networks. His address is School of Information Science and Technology, Northwest University, Xi'an, 710127, P.R. China and email address is dyf@nwu.edu.cn.

Xiaojiang Chen received his Ph.D. degree in Computer Software and Theory in 2007 from Northwest University, Xi'an, China. His current research interests include network and software security, software architecture, localization, and Internet of Things. His address is School of Information Science and Technology, Northwest University, Xi'an, 710127, P.R. China and email address is xjchen@nwu.edu.cn.

Zheng Wang received his Ph.D. degree in Computer Science in 2011 from the University of Edinburgh. Academic advisor: Professor Michael O'Boyle. His current research focus is in the areas of parallel compilers, runtime systems and the application of machine learning to tackle the challenging optimization problems within these areas. His address is C43, Infolab21 Lancaster University Lancaster LA1 4WA United Kingdom and email address is z.wang@lancaster.ac.uk.