2019 IEEE Security and Privacy Workshops (SPW)

SpyCon: Adaptation Based Spyware in Human-in-the-Loop IoT

Salma Elmalaki*, Bo-Jhang Ho*, Moustafa Alzantot*, Yasser Shoukry**, and Mani Srivastava*
*University of California, Los Angeles, {selmalaki,bojhang,malzantot,mbs}@ucla.edu
**University of Maryland, College Park, yshoukry@ece.umd.edu

***Abstract*—Personalized IoT adapt their behavior based on contextual information, such as user behavior and location. Unfortunately, the fact that personalized IoT adapt to user context opens a side-channel that leaks private information about the user. To that end, we start by studying the extent to which a malicious eavesdropper can monitor the actions taken by an IoT system and extract user's private information. In particular, we show two concrete instantiations (in the context of mobile phones and smart homes) of a new category of spyware which we refer to as Context-Aware Adaptation Based Spyware (SpyCon). Experimental evaluations show that the developed SpyCon can predict users' daily behavior with an accuracy of** 90.3%**. Being a new spyware with no known prior signature or behavior, traditional spyware detection that is based on code signature or system behavior are not adequate to detect SpyCon. We discuss possible detection and mitigation mechanisms that can hinder the effect of SpyCon.**

***Keywords*-spyware; privacy; IoT; human-in-the-loop;**

I. Introduction

Context-aware systems provide personalized services that are adaptive according to user context and surrounding environments. These pervasive systems have enabled a multitude of applications in several IoT sectors including smart cities, health care, and automotive systems. However, these enhanced capabilities come at the expense of privacy weaknesses [1]. As pictorially illustrated in Figure 1, a human-in-the-loop (HITL) IoT utilizes edge devices (e.g., mobile phones and wearables) to sense and infer human states. Such states are then used by the IoT to produce actions and adapt its behavior to match the human state. Resting on this observation, in this paper, we investigate how the coupling between human behaviors and decisions taken by IoT system can open a side channel, leaking sensitive information about users. In particular, this paper raises the following questions: (1) Can an eavesdropper who monitors the actions taken by the IoT be able to reverse engineer these actions in order to estimate human states and leak sensitive information? (2) Can we develop new software mechanisms that can detect and mitigate such privacy leaks? To explore the answers, we introduce a new type of spyware that exploits privacy leaks in context-aware adaptations which we call **SpyCon**.

A. Related Work

Context Monitoring Malware on Mobile Platforms: Mobile systems are becoming an integral component in multiple IoT systems due to their sensing capabilities [2]. Different side-channel attacks have been proposed, for example, using inertial sensors and touch screen to infer user input such as passwords [3; 4]. Besides, we witnessed how to exploit cellular signal strengths, air pressure, or power consumption for locations [5], gyroscope for eavesdropping conversations [6], system-level aggregate statistics for user's real world identity [7], and the state of shared memory for foreground apps, and even, `activity` transition sequences [8]. There is a trend that malicious apps are adapting to wearable devices [9]. For example, MoLe [10] exploits the wrist motion derived from smartwatches to infer keystroke inputs. These many examples show that "Your apps are watching you" [11] in a broad spectrum which a majority of users will never realize, and for sure "These aren't the droid you're looking for" [12].

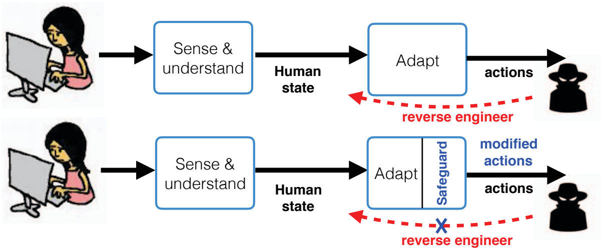


Figure 1: A cartoon that illustrates the flow of information in a personalized IoT. The IoT uses devices (typically on the edge) in order to sense and infer the human state. The IoT then adapts to the human state by applying some actions. (top) the proposed personalized IoT Spyware monitors the actions taken by the personalized IoT and reverse engineers it in order to estimate the human state (bottom) the proposed VindiCo framework which prevents modifies the IoT actions in a fashion that prevents any spyware from estimating the human state.

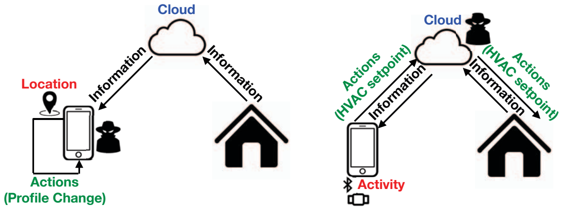
Contrary to the aforementioned side-channel attacks, we consider a spyware *which does not have access to sensor information* like inertial or gyroscope sensors, a spyware which can monitor only the actions that are triggered—by HITL IoT—based on changes in these sensory data.

Malware Detection Techniques: Several techniques have been proposed for malware detection and can be broadly categorized into two groups. (1) *Code signature-based approach* [13] detects stealthy behavior based on the code flow. (2) *Behavior-based approach* [14] performs information leakage detection in execution time, but tackling the issue from different layers of an operating system. DroidRanger [15] points out that an app can download binary payload at runtime, and hence code-signature based approach can not diagnose its intention but can raise a warning.

B. Paper Contribution

- We exploit a new side-channel attack vector arising from monitoring actions and decisions taken by IoT systems. We call this new set of attacks a *context-aware adaptation based spyware*, or in short, **SpyCon**.
- We show two concrete instantiation of SpyCon. The first instantiation targets mobile phones in which the SpyCon can maliciously infer user's behavior by monitoring the decisions taken by context-based apps. We assess the performance of the developed SpyCon through a one-

DOI 10.1109/SPW.2019.00039



(a) SpyCon at the Edge (b) SpyCon at the Cloud

Figure 2: SpyCon mounted at different places in HITL IoT.

month user study involving human subjects. The second instantiation targets smart homes, in which the SpyCon monitors HVAC activity reported to the cloud and use it to infer human activity. We assess the performance of the developed SpyCon through an industrial-level simulation engines simulating Heating, Ventilation, and Air Conditioning (HVAC) systems.

In the remainder of this paper, we show two examples of SpyCon, one targets IoT mobile/edge devices while the other one targets the IoT cloud. We refer to these two SpyCon as "SpyCon Edge" and "SpyCon Cloud", respectively. We then discuss a novel information-based detection engine that can be used to detect SpyCon.

II. SpyCon Edge: Popular Phone Manager Apps

A typical pipeline of an IoT application includes edge devices which generate sensor data, an IoT cloud which stores and forwards these data, and a mobile phone which consumes the data, computes the actions, and presents them to IoT users. Mobile phones are arguably a primary target for spyware due to their sensing capabilities and the integrations with IoT systems. In this scenario, we consider SpyCon is mounted on a mobile phone to leak sensitive information, depicted in Figure 2a. The SpyCon, for example, can reveal a user location such as next to the smart fridge (the sensitive information) if a notification reminder of a grocery list is observed (the adaption). Location-based phone settings management is one of the most popular context-aware applications[1]. Due to their capability to adapt to user context, apps like Llama [17], Tasker [16], and Locale [18] have gained more than 1 million downloads from Google Play Store. Moreover, these location-based apps are increasingly integrated as part of larger IoT systems. For example, Tasker can be combined with IBM Watson IoT platform to allow HITL IoT to take location-based decisions. Motivated by the popularity of these location-based context-aware apps, we choose user location as the sensitive data for which SpyCon is trying to leak.

A. *Spyware Description*

We designed a SpyCon that monitors changes in phone settings to demonstrate it is possible to leak user location, which is arguably the most sensitive type of user information [19]. The phone setting changes are triggered by the decisions taken by a location-based context-aware app as part of an IoT system. We start by making the following two important remarks:

- **No user permissions:** Unlike location information, many phone settings can be monitored without seeking user permissions. For example, SpyCon can easily get current screen brightness or alarm volume without user consent.
- **Ambiguity on setting changes:** Manual adjustment can make changes in phone settings through physical buttons. Although SpyCon can not discriminate *a priori* between the changes in the phone settings done by a location change or by manual adjustment from users, machine learning algorithms can be handy in discovering repetitive patterns in the data.

PS	Description	PS	Description
R	Ringer mode	P	Wallpaper
H	Touch sound	D	Dialpad sound
W	Enable WiFi	A	Alarm volume
I	Ringer volume	M	Media volume
T	Display timeout	B	Screen brightness
V	Vibration on touch	L	Screen locking sound

Table I: Phone settings (PS).

The operation of the designed SpyCon is divided into two phases:

- **Logging:** SpyCon monitors all the changes in phone settings and records a timestamped value upon a change is detected. A list of phone settings that we consider in our SpyCon is given in Table I.
- **Data Mining:** Once enough data is collected, SpyCon analyzes these data to discover repeated patterns and hence infers user's daily behavior. More details about the data mining algorithm are given in Section II-B4 after we discuss the user study setup.

B. *SpyCon User Study*

*1) **Shadow Logging Application**:* To understand how much information is leaked by context-aware apps like Tasker and Locale, we developed a shadow app that resembles the functionality of Tasker and Locale in order to collect the ground truth data. First, we ask users to enter the same *profiles* which they would provide in the context-aware apps (Tasker or Locale). To be more specific, users have to enter a fixed-radius circular geo-fence as a *context* trigger, as well as a set of *actions* (e.g., adjusting screen brightness or changing ringer mode to vibration) that would be activated when the user enters these geofences. The full phone settings we considered are listed in Table I. Secondly, in order to keep track of the golden output (ground truth) for later evaluation, the shadow app keeps and timestamps a record whenever the active profile is changed, implying that the user moves to a different location.

*2) **SpyCon Application**:* We developed a SpyCon whose only task is to log phone settings in the background *without any interaction* with all the other apps, including the context-aware app[2]. All the settings collected by the SpyCon app can be accessed *without permissions* in Android OS, including those listed in Table I.

[1] By the time this paper was written, context-aware phone settings management applications ranked 3rd in the Productivity category in the Android Developer Challenge [16].

[2] In the real world, this SpyCon can provide some functionality but collect data stealthily, which is a typical way a spyware hides its true intention.

However, it should be noted that any SpyCon app may benefit from knowing whether a context-aware adaption application is installed. This information can be retrieved through different ways such as the `getInstalledApplications()` API [3].

*3) **User Study**:* We implemented both apps mentioned above on Android running on Nexus 4 and Nexus 5. Seven participants are recruited in our user study, including four males and three females. Each participant carries our phone for four weeks. Users can choose the settings/profiles based on their personal preferences, and they are allowed to change the phone settings manually. Based on the data we collected during the user study, we explore what sensitive information can be mined maliciously as shown in the following experiment.

*4) **Experiment 1: Data Mining by Clustering**:* Revealing the semantics of the user location trace, or equivalently, the active profile sequence from phone settings is challenging since both the profile and the phone settings do not always have a one-to-one mapping. This is because (1) Users configure only a subset of the 12 settings listed in Table I in their profiles and hence it is not known a priori which subset of settings are used by the user. Furthermore, different profiles may include different phone settings to be changed. (2) Users can manually override the phone settings by, for instance, pressing the volume buttons or adjusting the brightness through the *Settings App*. Thus, we use a clustering technique to approach the user data mining problem, and in particular, we use k-means algorithm.

Deciding the number of clusters in the k-means algorithm is known to be hard in general and is usually application dependent. Since our SpyCon does not know how many profiles are defined by users, we brute-forcedly set k to be any value between 2 through 7 (selected based on the maximum number of profiles defined by our participants). Our algorithm returns the clustering result with the highest silhouette score.

C. *Critical Phone Settings*

Inspired by how most unsupervised machine learning algorithms work, we implement a greedy algorithm to find dominant phone settings. The algorithm procedures are provided below:

1) Initialize the selected feature set $S = \phi$.
2) We examine every other setting f not in S by performing k-means with feature set $S \cup \{f\}$. The silhouette score h_f is computed accordingly.
3) Denote $\hat{h}$ as the maximum h_f from the previous step. If $\hat{h}$ is larger than previous silhouette score, then $S = S \cup \{f\}$ and go back to step 2. Otherwise, the algorithm terminates.

D. *Privacy Implications*

The clustering result of one participant in our study is demonstrated in Figure 3. Figure 3a shows the actual user profile changes across the day (the golden output as explained in Section II-B1). Figure 3b shows the k-means clustering result (using an adaptive number of clusters) and demonstrates similar patterns with the golden output in Figure 3a. Our algorithm is able to capture subtle events, for example, learning that the user regularly went to a particular place (which turns out to be the child care) after leaving or before returning home, despite the portion of time this user spent in child care is short. Clustering result derived by dominant features from our feature selection algorithm is shown in Figure 3c. Figures 3b and 3c clearly indicate the ability of the developed SpyCon to reconstruct user context (switching profiles in this case) by just monitoring its side effect (changes in phone settings)[4].

UID	# clusters using all features				Dominant features
	base	2	3	2-7	
1	75.2	+18.9	+22.9	+19.1	+21.8 W,R,V
2	56	+17.2	+24	+18.3	+24.1 R,B,W
3	80.5	+12.9	+14.4	+13.6	+16.7 R
4	45.6	+37.3	+34.2	+35.6	+35.9 W,R,L
5	42	+24	+35.2	+24	+41.8 T,R,A
6	57.9	+4.4	+36	+4.4	+40.7 A,R,B,W
7	78	+15	+15.5	+15	+15.6 R
Avg.	**62.2**	**+18.5**	**+25.8**	**+18.2**	**+28.1**

Table II: Clustering accuracy (in percentage) of all users compared to the baseline accuracy (the accuracy that the SpyCon can have based on blind guesses) by applying k-means using the settings in Table I.

The overall accuracy of our clustering algorithm is reported in Table II. We define *baseline accuracy* by using blind guesses, that is, the SpyCon always reports a user is at home without observing any phone settings. The results in the rest of the columns are the additional information (the increase in accuracy) the SpyCon gains over the baseline accuracy if an inference is used based on a different number of clusters. The accuracy derived from dominant features is slightly higher because the feature selection algorithm excludes noisy features leading to a better result. We report dominant features for each user in the last column of Table II. We observed that the ringer mode is a dominant feature.

In summary, this study shows that the designed SpyCon can estimate and learn with an average accuracy of 90.3% the user behavior, in particular:

- Average commuting time between home and work.
- Average time spent at work and at home.
- Weekend behavior, such as if a specific place is frequently visited on Sundays and average time spent at home.

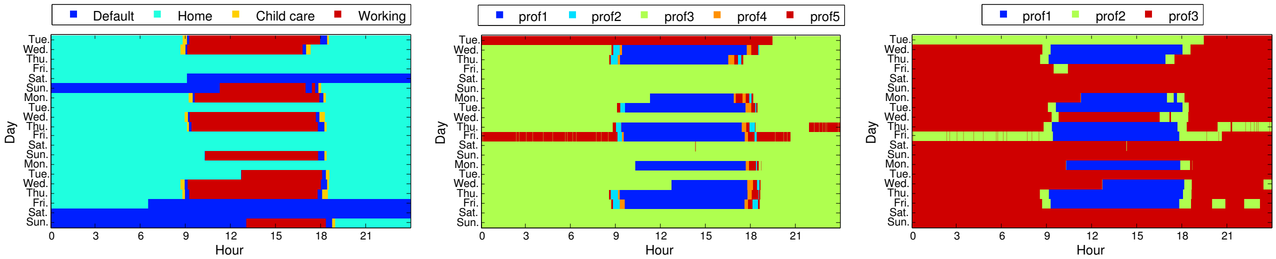
E. *Experiment 2: Detection Using Current Antivirus Apps*

After we had implemented this spyware app, we examined it using 5 well-known anti-virus applications [5]. None of them reported this app as malware. This follows from the fact that the proposed SpyCon does not have any suspicious code signature. This motivates the need to find a new detection technique that suits this kind of spyware.

[3] Even though Android may protect this API by adding a permission in the future, studies have shown that it is hard for most users to relate the side-channel privacy implications to the granted permissions in different apps [20].

[4] If the user specifies two profiles with the same settings, SpyCon will recognize them as the same profile. However, the incentive of the user to define the same settings for multiple profiles defies the idea of the context-aware app.

[5] These 5 anti-virus applications are AVG AntiVirus, Symantec Norton Security & AntiVirus, AVAST Mobile Security & Antivirus, McAfee Security & Power Booster, and Kaspersky Internet Security for Android.



(a) Golden output (b) Clustered by all features (c) Clustered by dominant features

Figure 3: One example of profile timeline from user #2.

Context-aware App	Context	Side-channel
Tasker [16]	location	phone settings
Locale [18]	location	phone settings
Silence [21]	calendar events	phone settings
RockMyRun [22]	biometrics	music played
HABU music [23]	mood	music played

Table III: Context-aware apps in the market and their side-channel.

F. *Experiment 3: Beyond Location SpyCon*

While the previous experiment aims at studying how the proposed SpyCon can leak the semantics of the user location, we further explore how acquiring side-channels can reveal other sensitive user information. To this end, we study several context-aware apps in the Android market and report the monitored *context* and the corresponding *actions* taken by these apps in Table III. Since other apps can observe these actions (even without asking for user permissions), these actions open a side-channel that leaks information about the user behavior. For example, if a SpyCon knows a priori the presence of Silence App [21] (an app which changes your phone settings based on the calendar events), it can reveal the timing or repetition of calendar events based on the side-channel of phone settings. Similarly, monitoring changes in the played music media[6] can leak information about the user biometrics (heart rate and running pace) and user mood whenever such context-aware apps are used. In general, the proposed SpyCon can take advantage of any pair of `get` and `set` methods that are in the Android framework APIs.

III. SpyCon Cloud: Smart HVAC System

Cloud servers continue to be one of the weakest points when it comes to data breaches [24]. We argue in this example, that even non-sensitive information collected on HITL IoT clouds can be used to leak sensitive user information. To that end, we choose the personalized smart HVAC system as an example of a HITL IoT application. The personalized HVAC incorporate the human's activity to change the HVAC set point to maximize his comfort level. In this scenario, as pictorially shown in Figure 2b, human's activity such as cooking, sleeping, sitting, etc. is used with the current temperature in the house to tune the HVAC set point. The activity of the human is a wealth of information that should be kept safe as it leaks the behavioral pattern of the human along with his house occupancy patterns. Although the HVAC set point is calculated on the phone and pushed to the cloud service that controls the smart thermostat, a SpyCon operating on the cloud could monitor the changes in temperature set points along with the current home temperature to infer the human's daily behavior.

A. *Spyware Description*

We conduct a simulation-based experiment using EnergyPlus [25], an industrial-level physics-based simulation engine, to model HVAC systems. We use the weather reports in Colorado-Denver during January 2018 [26]. The user activity is used to control the set point of the HVAC across the day to maximize the human thermal comfort. A SpyCon mounted in the cloud can monitor the following information to infer the user's daily behavior: (1) Changes in the HVAC set point triggered by the IoT, (2) current house temperature, (3) time of the day, and (4) day of the week.

B. *Experiment 4: Data Mining by Clustering*

Using the same procedure in Experiment 1a (Section II-B4), we simulated several humans independently in EnergyPlus. Due to space, we only show the ground-truth activity and the occupancy of one human (human #1) across time for a month in Figure 4a. The results of the clustering algorithm used by the SpyCon to infer the human's daily behavior and the home occupancy are shown in Figure 4b and 4c, respectively.

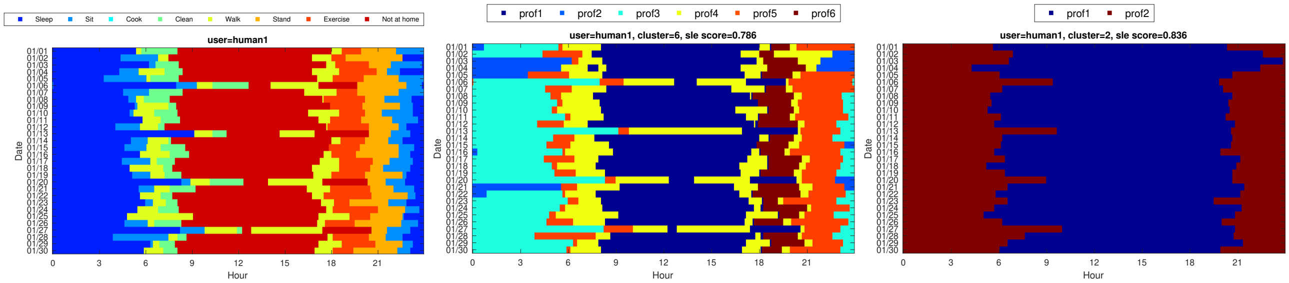
C. *Privacy Implications*

The results shown in Figure 4b suggest that SpyCon operating in the cloud can infer sensitive information like when the IoT user wakes up and goes to sleep (prof3), and when the user leaves the house and comes back (prof1). SpyCon Cloud is also able to detect the occurrence of a periodic user activity just after returning home from work which is not performed during the weekends (prof6). The accuracy of the clustering is listed in Table IV. By using two clusters to detect the occupancy (home/away), SpyCon Cloud achieves an accuracy of 75% for human #1 and 91.72% for human #2. To detect the daily behavioral patterns, we increase the amount of clusters and achieved an accuracy of 76.2% for human #1 and 65.4% for human #2.

IV. Detection and Mitigation Discussion

Sharing information in the context of HITL IoT systems may lead to privacy leaks stemming from the tight coupling between human behavior and actions produced by IoT (as shown in Experiment 1 and Experiment 4). SpyCon exploiting these privacy leaks cannot be detected by the

[6] While Android framework does not provide an API to directly retrieve which music is playing, our experiments show that a SpyCon can retrieve such information by reading the metadata associated with the currently active media.



(a) Golden output for human #1 activity (b) Clustered into 6 profiles (c) Clustered into 2 profiles

Figure 4: Activity timeline from simulated human #1.

UID	Number of Clusters (k-means)					
	2	4	5	6	7	8
1	75.65	73.47	72.59	76.71	85.31	73.06
2	91.72	70.08	70.14	73	63.23	50.52

Table IV: Clustering accuracy (in percentage) to detect the human's daily behavioral pattern for two simulated humans in a house using EnergyPlus simulator to simulate HVAC system. The clustering algorithm (k-means) uses features mentioned in Section III-A

current state-of-the-art signature-based and behavior-based detection techniques (as shown in Experiment 2). In this discussion, we propose a novel information-based detection engine and mitigation firewall.

The basic idea behind this engine is to keep track of the ability of *any* SpyCon to infer the human state through monitoring actions triggered by changes in these states. To this end, we draw on the literature of information theory and leverage *mutual information (MI)* to quantify the amount of correlation (or dependence) between two random variables. In our scenario, we use MI between *state* and *action* as a metric to measure how certain a SpyCon may infer the human state from observed actions. MI provides a theoretical bound on the inference capability of *any* learning algorithm. Generally speaking, the lower the MI between context and actions is, the smaller the accuracy *any* inference algorithm can get. Push into one extreme; if the MI is zero, then *no* algorithm can infer context from monitored actions.

Once the MI (and hence the correlation) between actions and human states are above a certain threshold, the mitigation firewall starts to *carefully* corrupt the information before being shared with other edge devices in the IoT system in an attempt to lower this correlation and prevent *any* inference algorithm from discovering patterns in the data. While completely blocking all side-channels may not be practical, our goal is to drastically reduce its bandwidth. This process needs to be done without any prior assumption on the type of inference algorithms used by SpyCon.

One way to reduce the correlation is to mask some action values (report zero value instead of actual value). As a result, at any given context change, the SpyCon can only observe partial action values after an adaptation occurs. The decision of masking depends on flipping a biased coin with a selected probability p, which serves as the parameter of mitigation effectiveness. As a preliminary evaluations, we examine the decrease in MI when mitigation is applied and how it affects the overall accuracy of detection in SpyCon.

A. *Mitigation of SpyCon Edge*

In Figure 5a, we show how the MI decreases across all the users with respect to different values of masking probability p. As expected, when the MI decreases, the ability to infer the context decreases as shown in Figure 5b.

B. *Mitigation of SpyCon Cloud*

We plot the mitigation results of human #1 in Figure 6. We applied the masking with different probabilities. We observe that the performance of the SpyCon Cloud in detecting the user profile is adversely affected by increasing the probability of masking. Similarly, as shown in Figure 6b, the SpyCon ability to detect home occupancy degraded severely as a consequence of applying the mitigation compared to Figure 4. This mitigation entails changing the HVAC set-point in order to hinder the ability of SpyCon to infer the human behavior which can affect the human thermal comfort. Accordingly, we compared the Prediction Mean Vote (PMV) values–as a measure for the human thermal comfort [27]–before and after mitigation in Figure 7. The PMV score ranges from -3 to 3 which is the range of thermal sensation from very cold (-3) to very hot (3). According to ISO standard ASHRAE 55 [27], a PMV in the range of -0.5 and 0.5 for an interior space is recommended to achieve thermal comfort. We used the Fanger model in EnergyPlus to estimate the PMV value [27]. In particular, as seen in Figure 7, a choice of masking probability equal 0.8 lead to user PMV in the range of -0.8 to 1.2 while achieving a degradation in SpyCon accuracy by 45%.

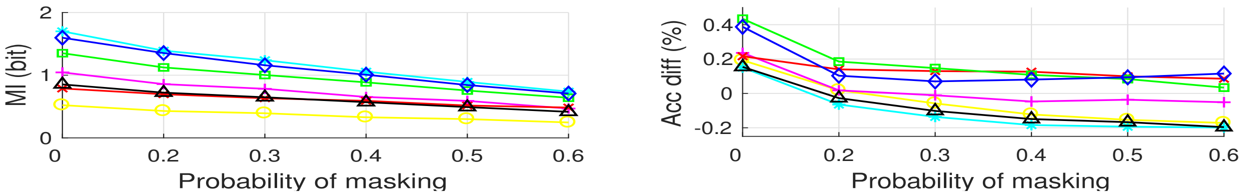
Acknowledgment

This research was supported in part by the National Science Foundation under award CNS-1705135, the Army Research Laboratory (ARL) under Cooperative Agreement W911NF-17-2-0196, and the King Abdullah University of Science and Technology (KAUST) through its Sensor Innovation research program. Salma Elmalaki is supported by Microsoft Research Fellowship. Any findings in this material are those of the author(s) and do not reflect the views of any of the above funding agencies.

References

[1] A.-R. Sadeghi, C. Wachsmann, and M. Waidner, "Security and privacy challenges in industrial internet of things," in *The 52nd ACM/EDAC/IEEE Design Automation Conference (DAC), 2015*. IEEE, 2015, pp. 1–6.

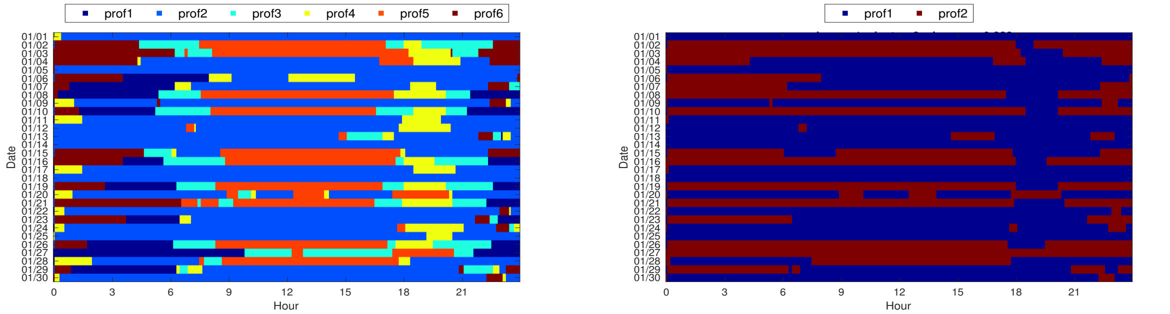
[2] K. Nahrstedt, H. Li, P. Nguyen, S. Chang, and L. Vu, "Internet of mobile things: Mobility-driven challenges, designs and implementations,"



(a) MI with masking

(b) Accuracy with masking

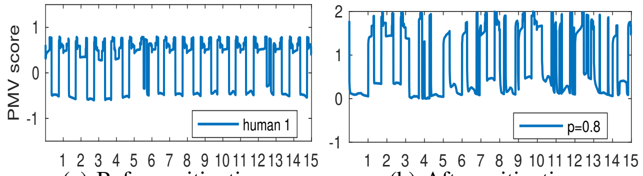
Figure 5: MI and accuracy after mitigation where each line represents one of the seven users.



(a) Profiles detection after masking mitigation *(p=0.8)*

(b) Occupancy detection after masking mitigation *(p=0.8)*

Figure 6: Activity and occupancy timeline of human #1 after mitigation.



(a) Before mitigation

(b) After mitigation

Figure 7: PMV score before and after mitigation across 15 days.

in *Internet-of-Things Design and Implementation (IoTDI), 2016 IEEE First International Conference on*. IEEE, 2016, pp. 25–36.

[3] E. Miluzzo, A. Varshavsky, S. Balakrishnan, and R. Roy Choudhury, "Tapprints: your finger taps have fingerprints," in *Proceedings of the 10th international conference on Mobile systems, applications, and services*. ACM, 2012, pp. 323–336.

[4] P. Marquardt, A. Verma, H. Carter, and P. Traynor, "(sp)iphone: decoding vibrations from nearby keyboards using mobile phone accelerometers," in *Proceedings of the 18th ACM conference on Computer and communications security*. ACM, 2011, pp. 551–562.

[5] Y. Michalevsky, A. Schulman, G. A. Veerapandian, D. Boneh, and G. Nakibly, "Powerspy: Location tracking using mobile device power analysis." in *USENIX Security*, 2015, pp. 785–800.

[6] Y. Michalevsky, D. Boneh, and G. Nakibly, "Gyrophone: Recognizing speech from gyroscope signals," in *23rd USENIX Security Symposium (USENIX Security 14)*, 2014, pp. 1053–1067.

[7] X. Zhou, S. Demetriou, D. He, M. Naveed, X. Pan, X. Wang, C. A. Gunter, and K. Nahrstedt, "Identity, location, disease and more: Inferring your secrets from android public resources," in *Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security*. ACM, 2013, pp. 1017–1028.

[8] Y. Nan, M. Yang, Z. Yang, S. Zhou, G. Gu, and X. Wang, "Uipicker: User-input privacy identification in mobile applications," in *24th USENIX Security Symposium (USENIX Security 15)*, Washington, D.C., Aug. 2015, pp. 993–1008.

[9] A. Raij, A. Ghosh, S. Kumar, and M. Srivastava, "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, pp. 11–20.

[10] H. Wang, T. T.-T. Lai, and R. Roy Choudhury, "Mole: Motion leaks through smartwatch sensors," in *Proceedings of the 21st Annual International Conference on Mobile Computing and Networking*. ACM, 2015, pp. 155–166.

[11] S. Thurm and Y. I. Kane, "Your apps are watching you," *WALL Street Journal*, Dec. 2010.

[12] P. Hornyack, S. Han, J. Jung, S. Schechter, and D. Wetherall, "These aren't the droids you're looking for: retrofitting android to protect data from imperious applications," in *Proceedings of the 18th ACM conference on Computer and communications security*. ACM, 2011, pp. 639–652.

[13] S. Arzt, S. Rasthofer, C. Fritz, E. Bodden, A. Bartel, J. Klein, Y. Le Traon, D. Octeau, and P. McDaniel, "Flowdroid: Precise context, flow, field, object-sensitive and lifecycle-aware taint analysis for android apps," *Acm Sigplan Notices*, vol. 49, no. 6, pp. 259–269, 2014.

[14] W. Enck, P. Gilbert, S. Han, V. Tendulkar, B.-G. Chun, L. P. Cox, J. Jung, P. McDaniel, and A. N. Sheth, "Taintdroid: an information-flow tracking system for realtime privacy monitoring on smartphones," *ACM Transactions on Computer Systems (TOCS)*, vol. 32, no. 2, p. 5, 2014.

[15] Y. Zhou, Z. Wang, W. Zhou, and X. Jiang, "Hey, you, get off of my market: Detecting malicious apps in official and alternative android markets." in *19th Annual Network Distributed System Security Symposium (NDSS)*, 2012.

[16] "Tasker app," https://play.google.com/store/apps/details?id=net.dinglisch.android.taskerm&hl=en, [Online; accessed 9-Mar-2016].

[17] "Llama - Location Profiles Application," https://play.google.com/store/apps/details?id=com.kebab.Llama&hl=en, [Online; accessed 9-Mar-2016].

[18] "Locale Application," https://play.google.com/store/apps/details?id=com.twofortyfouram.locale&hl=en, [Online; accessed 9-Mar-2016].

[19] "Geolocation privacy legislation," https://www.gps.gov/policy/legislation/gps-act/, online; accessed March 2018.

[20] A. P. Felt, E. Ha, S. Egelman, A. Haney, E. Chin, and D. Wagner, "Android permissions: User attention, comprehension, and behavior," in *Proceedings of the Eighth Symposium on Usable Privacy and Security*. ACM, 2012, p. 3.

[21] "Silence 2.0," http://downloads.tomsguide.com/Silence,0301-52850.html, online; accessed March 11, 2017.

[22] "Rock my run," www.rockmyrun.com, online; accessed March 11, 2017.

[23] "Habu music," https://play.google.com/store/apps/details?id=com.gravitymobile.habumusic&hl=en, online; accessed March 11, 2017.

[24] M. Almorsy, J. Grundy, and I. Müller, "An analysis of the cloud computing security problem," *arXiv preprint arXiv:1609.01107*, 2016.

[25] D. B. Crawley, L. K. Lawrie, C. O. Pedersen, and F. C. Winkelmann, "Energy plus: energy simulation program," *ASHRAE journal*, vol. 42, no. 4, pp. 49–56, 2000.

[26] "Weather data by location," https://energyplus.net/weather, online; accessed October 1, 2018.

[27] "Thermal environmental conditions for human occupancy," ASHRAE/ANSI Standard 55-2010 American Society of Heating, Refrigerating, and Air-Conditioning Engineers, 2010.