Security after Login: Identity Change Detection on Smartphones Using Sensor Fusion

Tao Feng, Xi Zhao, Nick DeSalvo, Zhimin Gao, Xi Wang and Weidong Shi
Computer Science Department, University of Houston
Email: tfeng3@cs.uh.edu

***Abstract*—Coinciding with the surge in popularity and adoption of mobile devices and the ever-expanding capabilities of these devices, the amount of sensitive information accessed and stored has increased exponentially. Inasmuch, these advancements have, and continue to demand great efforts from researchers and the industry alike in terms of improving security therein. Existing technologies either conduct user identity verification via a login stage or request authentication every time the user accesses a sensitive app. We propose, in this paper, an IdentityTracker. This framework is dedicated to tracking the user's identity, performing app-level access control management. Continuous and implicit tracking of the user's identity is accomplished through monitoring fingerprint authentication logs as well as detecting events when the phone has left the user's hand. This approach leverages multiple onboard sensors. We conducted two user-studies acquiring smartphone users' usage statistics to investigate security and usability needs of our solution. To monitor these subtle gestures in real-world uncontrolled environments, multi-session data collection has been conducted to iteratively improve system performance. The evaluation results have demonstrated the feasibility of IdentityTracker.**

I. Introduction

In response to the growth in popularity of smartphones, doors have opened offering newer and enhanced capabilities in terms of the computing power. These advancements have allowed previously impractical applications be implemented which were once constrained by resources. A recent study has confirmed that placing calls is now only the fifth-most frequent use of smartphones having been replaced by other uses such as browsing and applications [1]. While these now possible uses bring a multitude of convience and an overall richer experience, they also introduce new privacy and security issues in relation to sensitive information stored and accessed therein.

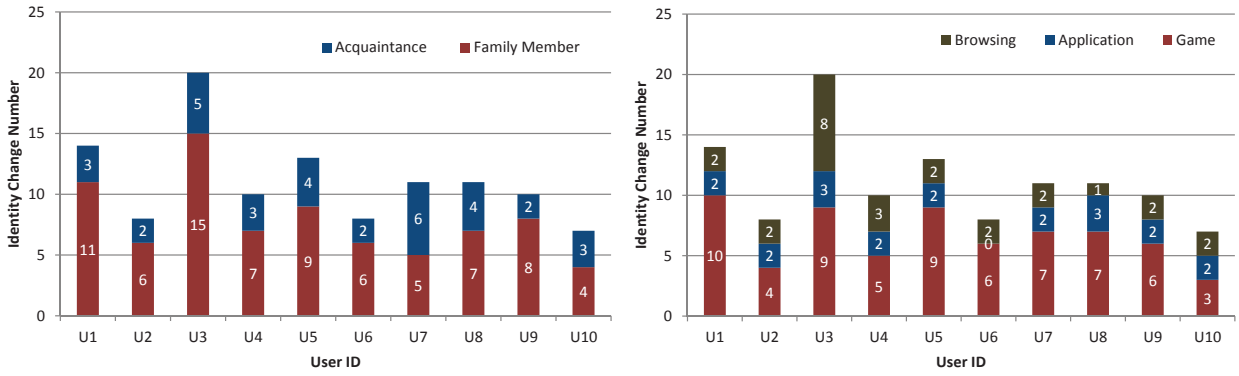
Previous mobile user authentication technology, such as password and swipe pattern are easy to be compromised due to their low information entropy. Swiping pattern is also vulnerable to simple attacks such as smudge attacks [2]. To deal with problems, Apple delivered iPhone 5s and iPhone 6, which employs fingerprint based identity verification to promote the security and privacy level of smartphones. Although the design of combine fingerprint sensor with home button did improve the protection in the login stage, this new identity verification system still have some flaws in practical. One key issue is that it cannot detect user's identity in the post login stage, so it is hard for it to handle privacy and usability problems in phone sharing scenarios, which is researched as a new topic for recent years [5], [7].

Phone Sharing Scenario 1: Children like to play their parents' smartphones. Normally, parents log in their devices and hand to their children. However, the mobile device cannot recognize the current user identity is already transferred from parents to children, and will react to the children as it reacts to their parents. This definitely is not a case the parents desired. The parents may permit their children to access and play games or other entertainment applications, but they may not happy if their children mistakenly perform online shopping or delete emails.

Phone Sharing Scenario 2: Sometimes smartphone users' family members or your acquaintances may need to borrow their smartphone for purpose like making a phone call, take a photo, or using some applications. The phone owners are willing to lend their device for these usages. However, the owners would not like them to go beyond those usages and approach to their privacy and sensitive information.

To exhibit potential enhancements in security, we have implemented and installed an application on ten users' smartphones and continued to track the smartphone usage for one week following. Results have offered statistics in relation to who the guest users are as well as why the guest was allowed to borrow the device as shown in Fig. 1. We find it very likely that the guest user may accidentally perform some operations the owner may not intend. Although there are some existing approaches that employ app-level access control (*i.e.,* Applock) to secure applications accessing or containing sensitive information, they introduce a heavy burden upon the owner themselves in terms of the constant explicit authentications required. To prove the inefficiency of said current systems, we implement a background service on smartphone devices to log the authentication events and its current user's identity. Here we consider authentications at the login stage and further authentication events. We differentiate between user necessary and unnecessary authentication events where: a) The user is either verified as the owner or b) The user is verified as not the owner and take note this action has aided in preventing unauthorized accesses. In the applock settings we may only lock apps that contain sensitive personal information, *i.e.* email, message, and bank application. Generated statistics of the ten-user one-week authentication data are displayed in

978-1-4799-1737-2/15/$31.00 ©2015 IEEE



(a) Lend to whom (b) Why hey borrowed

Fig. 1. Times owners lend their devices to others in one week, to whom and the reason

Fig. 2. As we can see from Fig. 2, for the majority of the users, 70% percent or greater of authentication instances are unnecessary and insomuch may be removed to enhance usability.

Apple delivered and Samsung both delivered smartphones with fingerprint sensing based identity verification to promote the security and privacy therein. These newly added sensors could aid in identifying the user's identity accurately and unlike password-based mechanisms it only belongs to the owner and cannot be shared with others. The inherent characteristics of fingerprint sensors unlock new possibilities in identity tracking solutions. In this paper, we propose IdentityTracker, a framework dedicated to tracking users' identities while performing app-level access control management. IdentityTracker performs continuous and implicit identity tracking of user identities by monitoring fingerprint authentication logs and detecting leaving-hand-events. This is all accomplished by leveraging multiple onboard sensors while concurrently managing app-level access control based on the detected identity and set applock policy.

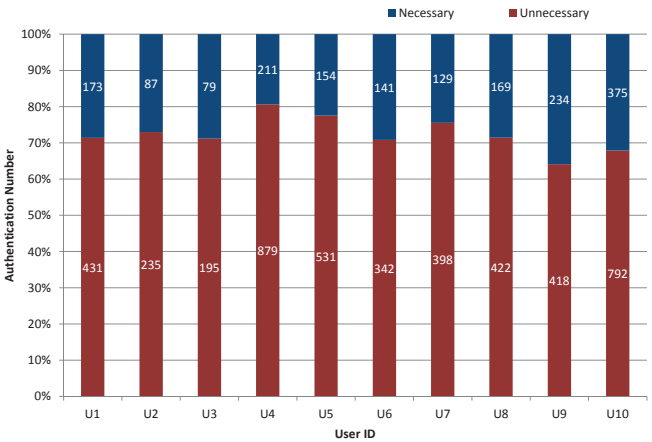


Fig. 2. Unnecessary authentication

II. Related Work

The research and process in relation to IdentityTracker draws from implicit and continuous identity authentication and activity recognition.

Implicit and Continuous Identity Authentication. Our process as described in this paper aims to monitor users' identity changes under uncontrolled environments by detecting device-leaving-hand events. Inasmuch, the process is performed in an implicit manner during regular smartphone usage. Several implicit identity sensing approaches have been proposed in the past that leverage the sensors on mobile diveices such as the accelerometer [10], GPS [11], touchscreen [12], [13], and microphone [8]. However, unlike previous works, we do not directly leverage the sensor readings and perform user authentication based on this. In our method, first the fingerprint is retrieved to identify the user's identity. We then subsequently monitor the device to detect if it has left the user's hand and in effect changed user identity.

Activity Recognition. Some existing works have explored user activity inference methods with accelerometer sensors [4], [6], [3]. In [9], Lu *et. al.,* proposed a continuous sensing engine for activity recognition on mobile platforms, which can robustly detect five common physical activities: stationary, walking, cycling, running, and in a vehicle (i.e., car, bus). Yang *et. al.,* [14] also completed research on activity recognition by exploiting the accelerometer data. Different from the aforementioned existing works, the goal of this paper is not to detect a long term and stable motion but a short term subtle gesture that takes place in a very short time frame.

III. System Framework

In this section, we first present the system overview of IdentityTracker and then discuss the details of the components and processes in our framework.

A. System Overview

Fig. 3 shows the high-level architecture overview of IdentityTracker. There are three main components within the IdentityTracker framework: A Touch Fingerprint Sensing Module,

978-1-4799-1737-2/15/$31.00 ©2015 IEEE

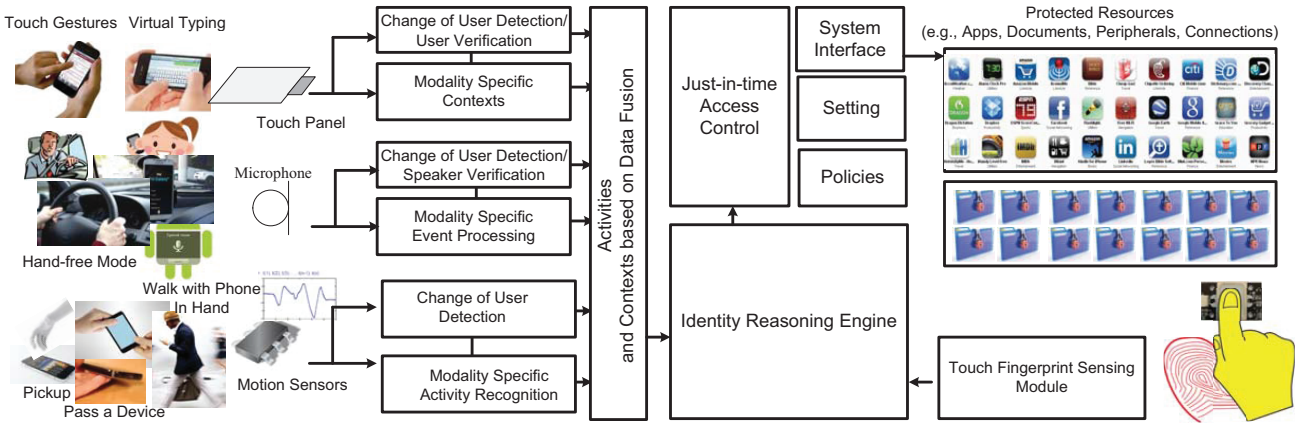


Fig. 3. Design of IdentityTracker

which employs the fingerprint sensor deployed on new generation smartphones, *i.e.,* iPhone 6 and Galaxy S5, to identify the current user's identity (by fingerprint verification); An app-level Just-in-time Access Control that manages the access permissions based on configurations of the policy and the current user identity; And most importantly, a Fine-grained Activity Recognition Module that employs touch, voice and motion sensors on the smartphone devices to detect device-leaving-hand events and monitor user identity changes, and an Identity Confirmation Module that confirms user's identity using touch and speech based user verification solutions. In normal usage scenarios, when the user unlocks the device with his/her fingerprint, the Touch Fingerprint Sensing Module detects and logs the user's identity. Once the identity is recognized, the Fine-grained Activity Recognition Module continues to track the touch screen usage data and the motion sensor data to monitor subtle gestures, such as device-leaving-hand events to detect user identity changes. While the Identity Confirmation Module will keep tracking and confirms user's identity using extracted input features. At anytime the current user of the smartphone device may want to access a mobile application. The app-level Just-in-time Access Control will check the current identity of the smartphone user as well as the policy of the mobile application. If the application is not locked, no matter what identity (of the current user), he/she can access it as there is no access control. Otherwise, the Just-in-time Access Control will react based upon the identity of the current user: block the application if the current user is a guest or give access permission to the owner.

B. Fine-grained Activity Recognition Module

The Touch Fingerprint Sensing Module and the Just-in-time Access Control are mature technologies on smartphone devices. The highlight and key point of IdentityTracker is the Fine-grained Activity Recognition Module and the Identity Confirmation Module. To detect the device-leaving-hand events, we first define a set of subtle gestures and their corresponding context user statuses as listed in Fig. 4. Since

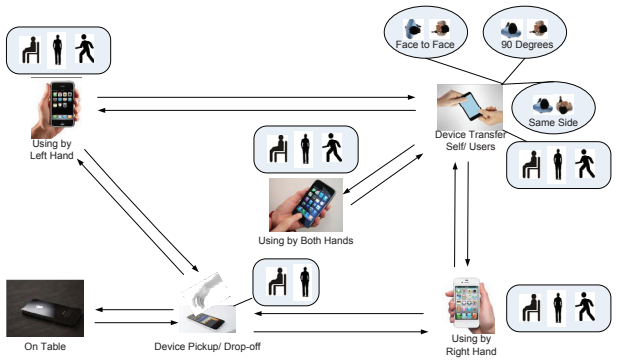


Fig. 4. Subtle gestures for leaving-hand-events detection

we are solving identity-changing problems in the post-login stage, we only considered the device while in an unlocked state. Essentially, there are four statuses when the smartphone device is in unlock state, which are respectively: On Table, Use by Left Hand (of a user), Use by Right Hand (of a user), and Use by Both Hands (of a user). There are four subtle gestures between these four status that trigger device-leaving-hand events, which respectively are Device Pickup from table, Device Drop-off to table, Device Transfer between the same user's hands, and Device Transfer between different users' hands. Concurrently, we need take the user's status into consideration since the motion sensor reading may be affected by different user statuses. During normal usage, there are three main user statuses: sitting, standing, and walking. While during Device-Transfer events between different users, users may have different relative positions(*i.e.,* Face to Face, 90 Degrees or in the Same Side).

To analyze the aforementioned concepts, touch and motion data are processed separately and then combined to predict the status or subtle gestures of the device. IdentityTracker extracts touch trace information, including touch point location, angle and length, contact size, and speed information to analyze which hand the user is using the device with. Similarly to

978-1-4799-1737-2/15/$31.00 ©2015 IEEE

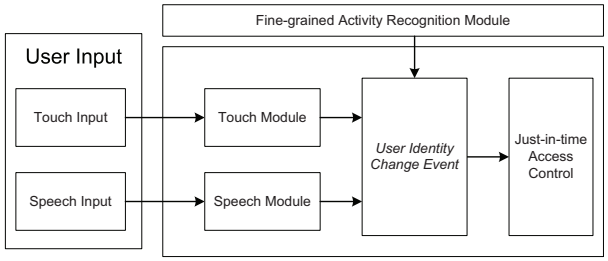


Fig. 5. Design of IdentityValidator

most activity recognition works, IdentityTracker also employs motion sensors, such as the accelerometer and gyroscope, to detect photo motion activities. The collected motion sensor data is pre-processed in frequency domain and value domain with a sliding window size of 16 sensor readings. By employing SVM on the extracted features, Holding the Device, On Table, or Device Transfer may be detected with ease in most cases. However, there are some complicated scenarios, such as walking and Device Transfer between different a user's own hands. To solve the subtle gesture recognition in these complicated scenarios, we can leverage those more accurate predictions (*i.e.,* On Table, Using by Right Hand, or context user status, such as walking) combined with the transition map(Fig. 4) to analyze those hard to detect subtle gestures. In the mean time, IdentityTracker can also utilize touch data to filter out some misclassified Device Transfer events (Since there is a touch event on the touchscreen, it is not possible the device is transferring, or the user cannot transfer the device from right hand to right hand).

C. Identity Confirmation Module

Fig. 5 shows the high-level architecture overview for Identity Confirmation Module. Identity Confirmation Module is mainly composed of two components: a Touch Verification Function and a Speech Verification Function. Touch Verification Function and Speech Verification Function respectively analyze all user's inputs on the smartphone device, including the touch inputs and speech inputs, and confirms the result from Fine-grained Activity Recognition Module. If the change of user identity change event detected by the Fine-grained Activity Recognition Module is confirmed, the Identity Confirmation Module will communicate with the Just-in-time Access Control. The Just-in-time Access Control then react to the current user based on the current user's identity.

1) Touch Verification Function: Touch Verification Function (Fig. 6) collects the touch gestures input data and running application context information from the *Multi-touch Driver* and *Running Application Context Listener* respectively. The collected raw data are then transferred to the *Multi-touch Gesture Engine* for data pre-processing and feature extraction. Then the pre-processed data are combined with the running application context information to generate a *Multi-touch Data Library* consisting of gesture templates. The *Touch Gesture Based User Authentication Module* compares incoming touch gestures with the templates in the *Multi-touch Data Library* to

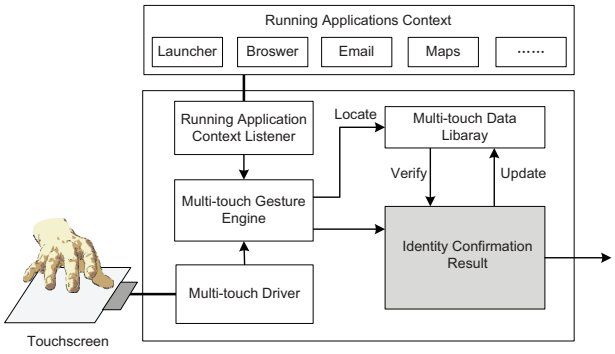


Fig. 6. Touch Module

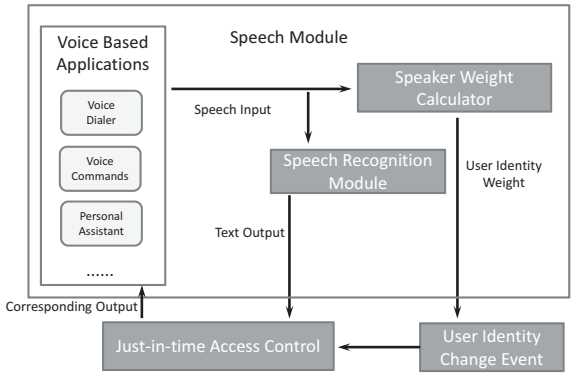


Fig. 7. Design of Speech Module

confirm the current user's identity and send the result to the Just-in-time Access Control.

2) Speech Verification Function: The way user interacts with smartphones using speech can be categorized into two classes, long conversations for telephone calls or recording, and short commands for speech based commands and messages. Besides the normal long conversation based speaker recognition, a highlight of the Speech Verification Function in IdentityTracker is that it engages in integrating speaker sensing and identity management with speech recognition. A diagram of the approach is presented in Fig. 7. The solution extends the Android speech recognition API with speaker recognition and identity management support. The new components include, an application interface that detects context running application and responds to the applications , an identity manager module that controls and enforces policies on whose speech an application should respond to and how to respond, and a speaker recognition module.

IV. Experiment

We implement IdentityTracker as a background service that implicitly collects motion and touch screen data, and logs

978-1-4799-1737-2/15/$31.00 ©2015 IEEE

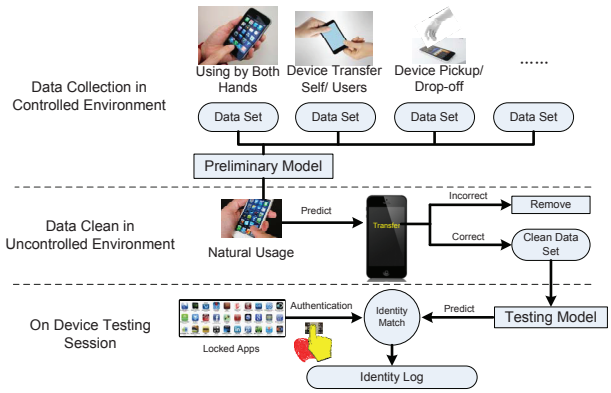


Fig. 8. Process of the experiments

the user's identity when an attempt is made to access to a locked application. The IdentityTracker app is installed on 13 smartphone users phones. The experiments consist of three sessions and the process is shown in Fig. 8.

A. Data Collection and Data Clean

In the data collection session, IdentityTracker collects a set of phone usage data from users. Users follows the instructions provided by IdentityTracker to perform a set of gestures and operations. Those gestures and operations are predefined, including phone operation on left hand, phone transfer from left hand to right hand, phone operation on right hand,phone operation on both hands, and phone transfer to another user. Although the gestures are predefined by IdentityTracker, users have freedom to perform the gestures in their own ways. The collected data are used to train a preliminary model. The model will be used for the next steps.

After the preliminary model is trained, we use it to classify user gestures and display the results to users. Users are required to provide feedback to the system, e.g., the correctness of the classification results. Users' feedback will be used to improve the primary model and generate a new model for the testing session. Although IdentityTracker attempts to ask users to perform their natural usage during the last data collection session, they may still be affected by the tasks we asked they to perform. If we aim to perform device-leaving-hand events, detection in uncontrolled environment, it needs to first receive a more accurate set of training data. However, since all the subtle gestures listed such as Device Transfers or Device Pickups/Drop-offs happen in a very short time frame and any extra label actions would interfere with the normal gestures. In response, we decided to first train a model based on the data collected in the previous session, and employ it to predict the current status or subtle gesture of the smartphone device and display it on the smartphone device. If the prediction is correct, the data will be recorded and labeled, otherwise the user can click on the display panel and the data will be labeled as misclassified and will not be used for final model training.

B. On Device Testing Session

As long as IdentityTracker acquires the clean data in an uncontrolled environment, it will train a new model based upon this new data set. After we install the new model on the device, IdentityTracker will still authenticate user's identity whenever a locked app is being accessed using fingerprint authentication and logs ground truth user identity. In the mean time, the testing model will also output a prediction result of current user's identity. The IdentityTracker will match the two user identity results and log them for further evaluation.

V. Evaluation

We evaluate the performance of our system in both security and usability aspects: i) How many times has unauthorized app access been reduced in comparison to a mobile system without app-level access control and how many unauthorized app accesses has been granted; ii) How many instances of unnecessary authentication have been reduced for the phone owners in contrast to a strict app-level access control mechanism and how many instances of unnecessary authentication have been requested.

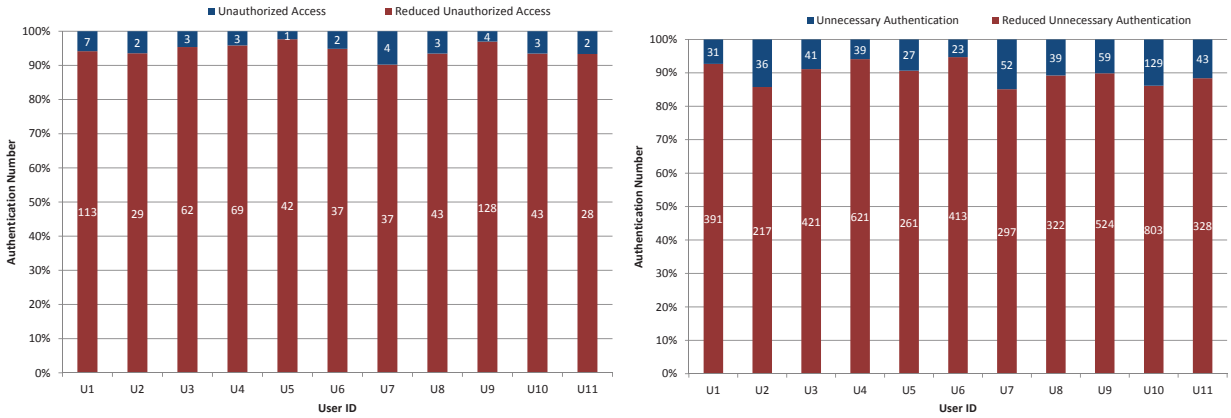
Fig. 9(a) shows the identity match log results of the on-device testing session. The red bar represents the number of unauthorized accesses blocked by IdentityTracker, while the blue bar marks the number of unauthorized access IdentityTracker failed to detect. It is clear that in comparison to the mobile system without app-level access control, IdentityTracker greatly reduces unauthorized accesses (above 90% of unauthorized access request were denied) and only very few times was a guest user allowed access to a locked application.

Fig. 9(b) shows the usability enhancement results based upon the logged results. In comparison to the mobile system with strict app-level access control that requires authentication every time a user atempts to access to a locked application, IdentityTracker alleviates the user's burden of constant authentication when he/she atempts to access a locked application (themselves). Above 85% of unnecessary authentications may be reduced by IdentityTracker. Although IdentityTracker may introduce a few unnecessary authentication events by falsely detecting a device-leaving-hand event, it still promotes the usability.

VI. Conclusions and Future Work

Throughout we have introduced and further explained IdentityTracker: a framework that not only tracks the user's identity through the fingerprint sensors, it combines these results with data from both the motion sensors and interaction inputs, including touch screen usage and speech inputs, while concurrently managing app-level access on smartphones in post-login stages. The idea focuses around motion and touch data to detect device-leaving-hand events based upon a predefined set of phone status, user status, and subtle gestures. To evaluate the performance of said system, we conducted two sessions of data collection to collect and clean the training data in both a controlled and uncontrolled environment. We then tested the trained model during the user's natural usage. Results have shown that our approach improves user security by not granting

978-1-4799-1737-2/15/$31.00 ©2015 IEEE



(a) Security Evaluation (b) Usability Evaluation

Fig. 9. Performance evaluation of IdentityTracker

app-level access to unauthorized guest users while at the same time promoting usability by greatly reducing the amount of unnecessary authentications for the smartphone's owner.

REFERENCES

[1] Making calls fifth most popular use for smartphones, says report. *http://www.whathifi.com/news*.

[2] A. Aviv, K. Gibson, E. Mossop, M. Blaze, and J. Smith. Smudge attacks on smartphone touch screens. In *4th USENIX conference on Offensive technologies*, 2010.

[3] L. Bao and S. Intille. Activity recognition from user-annotated acceleration data. In A. Ferscha and F. Mattern, editors, *Pervasive Computing*, volume 3001 of *Lecture Notes in Computer Science*, pages 1–17. Springer Berlin Heidelberg, 2004.

[4] T. Brezmes, J.-L. Gorricho, and J. Cotrina. Activity recognition from accelerometer data on a mobile phone. In *Distributed Computing, Artificial Intelligence, Bioinformatics, Soft Computing, and Ambient Assisted Living*, volume 5518 of *Lecture Notes in Computer Science*, pages 796–799. Springer Berlin Heidelberg, 2009.

[5] A. K. Karlson, A. B. Brush, and S. Schechter. Can i borrow your phone?: understanding concerns when sharing mobile phones. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*.

[6] J. Lester, T. Choudhury, N. Kern, G. Borriello, and B. Hannaford. A hybrid discriminative/generative approach for modeling human activities. In *Proceedings of the 19th international joint conference on Artificial intelligence*, IJCAI'05, pages 766–772, San Francisco, CA, USA, 2005. Morgan Kaufmann Publishers Inc.

[7] Y. Liu, A. Rahmati, Y. Huang, H. Jang, L. Zhong, Y. Zhang, and S. Zhang. xshare: supporting impromptu sharing of mobile phones. In *Proceedings of the 7th international conference on Mobile systems, applications, and services*.

[8] H. Lu, A. Bernheim Brush, B. Priyantha, A. Karlson, and J. Liu. Speakersense: Energy efficient unobtrusive speaker identification on mobile phones. In *IEEE Pervasive Computing*. 2011.

[9] H. Lu, J. Yang, Z. Liu, N. D. Lane, T. Choudhury, and A. T. Campbell. The jigsaw continuous sensing engine for mobile phone applications. In *Proceedings of the 8th ACM Conference on Embedded Networked Sensor Systems*, SenSys '10, pages 71–84, New York, NY, USA, 2010.

[10] J. Mantyjarvi, M. Lindholm, E. Vildjiounaite, S. marja Makela, and H. Ailisto. Identifying users of portable devices from gait pattern with accelerometers. In *IEEE International Conference on Acoustics, Speech, and Signal Processing*, 2005.

[11] P. Marcus, M. Kessel, and C. Linnhoff-Popien. Securing mobile device-based machine interactions with user location histories. In *Security and Privacy in Mobile Information and Communication Systems*. Springer Berlin Heidelberg, 2012.

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

[13] W. Shi, J. Yang, Y. Jiang, F. Yang, and Y. Xiong. Senguard: Passive user identification on smartphones using multiple sensors. In *IEEE 7th International Conference on Wireless and Mobile Computing, Networking and Communications*, pages 141–148, 2011.

[14] J. Yang. Toward physical activity diary: motion recognition using simple acceleration features with mobile phones. In *Proceedings of the 1st international workshop on Interactive multimedia for consumer electronics*, IMCE '09, pages 1–10, New York, NY, USA, 2009. ACM.

978-1-4799-1737-2/15/$31.00 ©2015 IEEE