SenGuard: Passive User Identification on Smartphones Using Multiple Sensors

Weidong Shi [#1], Jun Yang [*2], Yifei Jiang [†3], Feng Yang [‡4], Yingen Xiong [*5]

[#] *Computer Science Department, University of Houston, 4800 Calhoun Rd. Houston, TX 77004*
[1] larryshi@ymail.com

[*] *Nokia Research Center, 955 Page Mill Road, Palo Alto, CA 94304-1003*
[2] junyang2005@gmail.com, [5] yingen.xiong@nokia.com

[†] *Department of Computer Science, Colorado University at Boulder, Boulder, CO 80309-0430 USA*
[3] jiangyifei@gmail.com

[‡] *School of Computer and Communication Sciences, Ecole polytechnique federale de Lausanne, Lausanne, Switzerland*
[4] feng.yang@epfl.ch

***Abstract*—User identification and access control have become a high demand feature on mobile devices because those devices are wildly used by employees in corporations and government agencies for business and store increasing amount of sensitive data. This paper describes SenGuard, a user identification framework that enables continuous and implicit user identification service for smartphone. Different from traditional active user authentication and access control, SenGuard leverages availability of multiple sensors on today's smartphones and passively use sensor inputs as sources of user authentication. It extracts sensor modality dependent user identification features from captured sensor data and performs user identification at background. SenGuard invokes active user authentication when there is a mounting evidence that the phone user has changed. In addition, SenGuard uses a novel virtualization based system architecture as a safeguard to prevent subversion of the background user identification mechanism by moving it into a privileged virtual domain. An initial prototype of SenGuard was created using four sensor modalities including, voice, location, multitouch, and locomotion. Preliminary empirical studies with a set of users indicate that those four modalities are suited as data sources for implicit mobile user identification.**

I. Introduction

In recent years, mobile handheld devices have rapidly evolved from simple communication devices to mobile personal computers. Those personal computing devices such as smartphones, MID (Multimedia Internet Devices), tablets offer a vast array of useful applications to their owners. Increasingly, a lot of corporations and government agencies hand out mobile computing systems to their employees. As a consequence, important personal private data and business information such as personal private information, trade secrets, confidential information, credentials are stored on those mobile handheld systems, which results in increased demands for user identification and access control. However, it remains a challenge how to protect private personal and business data stored on a mobile computing system and enforce proper access control with a solution that can achieve proper balances among user friendly, cost, and security. Traditional one-shot active authentication and access control approaches that explicitly ask a mobile user to authenticate/identify himself/herself suffer from multitude of drawbacks when implanted on a consumer mobile handheld system.

First, prompting a mobile user frequently for entering a password is extremely user-unfriendly in design. Imagine that a smart phone user is requested to authenticate himself when the phone switches from a power saving or hibernation mode to a full power mode, or each time the user receives a phone call or instant message. A survey conducted in [1] showed that password restriction on smartphones more annoying to users than other deficiencies such as lack of coverages, small screen sizes, or poor voice quality. This suggests that implicit and passive user authentication that requires little or no active user involvement is a desirable feature by smartphone users.

Second, the design of consumer mobile devices is extremely sensitive to cost, size, and power efficiency. Integrating a dedicated biometric device such as a fingerprint scanner to a consumer mobile device is less attractive because of increased cost, additional space requirement, and extra power footprint.

Third, most biometrics and active one-shot user authentication approaches identify a user during login and then allow the mobile system to be used from that point forward under the assumption that the actual physical user is always the same as the login user. However, this assumption may not be valid in many scenarios such as when a different user has the same physical access to the system after initial login or biometric verification, or the system is stolen or lost. According to Turk and Altinok [2], the underlying assumptions of one-shot user authentication approaches such as, a user can always be identified at a single point in time, and the user remains constant for the duration of a login session, may not be true in many practical contexts and environments. An implicit and continuous user identification scheme that can monitor the actual user of a mobile device and establish user identity continuously without frequent and heavy user involvement does offer many desirable properties over active one-shot user authentication approaches.

In this paper we designed and evaluated an implicit mobile user identification system. The system can implicitly and continuously verify a mobile user. The system leverages the facts that a mobile computing device is personal and there are multiple low cost user identification capable sensors already integrated with today's mobile systems for other purposes and functionalities. An individual sensor may yield poor identification accuracy. However, combining the output of multiple sensors together can dramatically improve user identification accuracy. As a consequence, the mobile

978-1-4577-2014-7/11/$26.00

computing system becomes more secure without sacrificing user friendliness. Only when the system finds that the actual physical user has changed, it invokes the traditional active user authentication process that explicitly asks the user to authenticate himself.

The system is well suited for user identification on mobile handheld devices for the following reasons. A mobile device is a personal computing device, which reduces the complexity of implicit user identification. Second, today's mobile devices are equipped with multiple functional sensors. Those sensors can offer rich sources of information for ambient user identification.

The main contributions of our work include (i) design of a space-time multi-modality classifier that is optimized for, power efficiency, accuracy, user-friendliness, and sensor data availability; (ii) empirical study and evaluation of the applicability of multiple functional mobile sensors for implicit and continuous user identification, including multi-touch sensing input, handheld device accelerometer inputs, cellular modem location input, and voice input; and (iii) development of a system solution for protecting and securing an implicit and continuous user identification engine so the engine cannot be easily terminated by users.

Here is the structure of this paper. Section 2 gives some related work. The details of architecture and technology are described in Section 3. Preliminary experiment results and analysis are explained in Section 4. Some open issues are discussed in Section 5, and the final conclusions of the paper are presented in Section 6.

II. Related Works

The prior arts can be categorized into two groups including implicit user identification and multi-modality pattern classification, especially, multi-modality biometrics.

For desktops and personal computers, researchers in the past have explored the feasibility of applying keystroke dynamics and typing patterns for user identification. Keystrokes can be continually sampled by intercepting output from a keyboard. In a study by Clarke et al. [3] on users perceptions of authentication on mobile devices, the results showed that a system that can implicitly and continuously perform user authentication in the background without disrupting the normal user-mobile device interaction is a desired solution by mobile device users. Ailisto et al. [4] used accelerometers in television remote controls to identify individuals. Cuntoor et al. [5] and Gafurov et al. [6] experimented user identification using gait analysis and recognition. Koreman and Morris et al. [7] proposed a continuous multi-modal based approach for user identification. In [1], Jakobsson et al proposed an implicit user authentication framework and studied using recorded phone call history and location for continuous user authentication.

There has been a body of literature on combining multiple biometric inputs to produce aggregated user identification results. In [8], Indovina et al. identified that biometric integration can occur on the feature level, or the score level. In feature level integration all of the initial features from measurements are grouped together into a single feature vector for classification. Although the most information is available at this point, feature-level integration suffers from the so-called curse of dimensionality. Additionally, the features of some measurements may not always be available.

Muncaster and Turk [9] attempted to achieve a continuous, score-level multi-modal system by classifying based on a weighted sum of scores from each modality. The weighting factor was chosen in such a way that would capture the reliability of the modality, and was decreased monotonically with the time since the last measurement. Recently, Sim et al. [10] developed a continuous multi-modal biometrics system using a hidden Markov model (HMM). In this work, the users identity was the sole hidden variable that would emit biometric measurements. The authors were successful in integrating results from a fingerprint biometric classifier with a face classifier and developed a model that intuitively separated the uncertainty that arose from the dynamic model from the uncertainty that arose due to the sensor model.

III. SenGuard Architecture

A. Hypervisor Based SenGuard Framework

There are many challenges to design and implement an implicit and continuous user identification system. For instance, a user who has access to the system at current moment can terminate or abort any background running user identification service. Moreover, power efficiency is very critical to mobile handheld devices. An implicit and continuous user identification system should not have significant impact on battery life.

SenGuard addresses those challenges through several novel designs. The system protects the implicit user identification engine by running it in a protected or isolated domain from the domain that a normal user interacts with. This is achieved by using mobile virtualization. MeeGo [11] is a complete mobile Linux system. It supports both ARM and Intel x86 based mobile platforms. It is relatively straightforward to port MeeGo as a hypervisor guest by leveraging Linux based open source virtualization efforts such as Xen [12] or KVM. Meanwhile, running a smartphone OS as a hypervisor guest has many additional advantages such as improved security, increased flexibility in power management, enhanced manageability, support for multiple usage contexts (e.g., sharing a smartphone for both business and personal usage), etc.

In prototype of SenGuard, handset MeeGo is executed as an unprivileged Xen guest. Beneath it is a mobile Xen hypervisor that interfaces with the hardware. There is a privileged domain (domain 0) that handles all I/O operations and executes SenGuard service. After a smartphone is booted, domain 0 will run in headless mode. The MeeGo guest domain runs in full display mode. A smartphone user interacts with the MeeGo guest by default for normal smartphone operations. Domain 0 is hidden from the user. It requires privilege escalation to access domain 0.

Since domain 0 is a privileged I/O domain, most device drivers including those for managing sensors are situated in domain 0, see Figure 1. The MeeGo guest domain accesses smartphone I/O devices such as modem, sensors, and microphone through virtualized I/O devices. Virtualized I/O devices in the MeeGo guest receive forwarded I/O data from domain 0. In typical setting, software driver of a

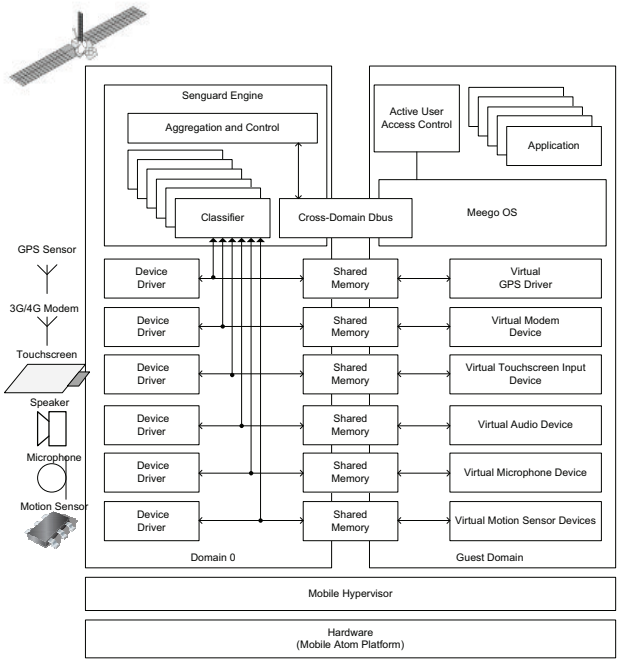


Figure 1. Mobile Hypervisor Based SenGuard Architecture

virtualized I/O device in the MeeGo guest domain communicates and exchange data with domain 0 using cross-domain shared memory. SenGuard in domain 0 snoops sensor inputs received by the devices drivers in domain 0. The captured sensor data such as multi-touch input event traces, voice input, motion sensor inputs are routed by SenGuard to a set of modality dependent classifiers that can process the sensor inputs and return a quantitative judgement whether the sensor inputs match with recorded sensor input patterns of the default smartphone owner.

SenGuard applies sensors adaptively and opportunistically. At first only passive and low power sensors are active (preferably snooping sensor inputs from normal user-mobile device interactions). If the performance of the classifier is good enough, the system just uses these sensors. If the results of the classifier are not satisfiable, the system activates other sensors one by one from low power consumption sensors to high power consumption sensors until the output of the aggregated classifier yields acceptable results.

To aggregate results from multiple sensors, SenGuard uses a sliding window based classifier aggregator. The aggregator is optimized for power consumption, sensor availability, and accuracy. Since not every sensor's information can be available or usable at certain time, the aggregation has to be dynamic in space domain. The aggregator must be able to handle a flexible size of feature set from multi-modality sensors. When a subset of sensors are available, the aggregated second-level classifier can still work and deliver robust performance. This property is essential to passive user identification using multiple sensors.

When evidences from multiple sensor inputs indicate the actual user is most likely different from the default smartphone owner, SenGuard will invoke active user authentication, which often requires the user to enter a credential for continued access to smartphone functions.

For exposing context information of the MeeGo domain to the privileged domain 0, we implemented a cross-domain Dbus that allows domain 0 to access MeeGo system and session Dbus messages. Domain 0 and SenGuard use these messages as sources of MeeGo context. In addition, domain 0 and SenGuard can interact with the MeeGo system and Window manager using Dbus messages.

B. Sensors

Accelerometer, microphone, cell ID location and touch screen are well known sensors on mobile devices. Accelerometer can measure user's motion states and step rate when user is moving. Microphone can detect voice activities when user is talking near the phone. Cell ID location can detect user's significant places such as home and office. Touch screen can monitor user's touching activities on his phone screen.

1) Motion: SenGuard opportunistically captures mobile user's motion sensing inputs and uses the inputs for user identification. One of the main motion input channels is accelerometer which can be found almost on every today's smartphone. Before accelerometer data are admitted for implicit user identification, the system applies activity recognition first to detect the context under which the motion data is captured. To achieve this goal, the system uses the JigSaw engine [13], a continuous sensing engine for activity recognition on mobile platform. JigSaw pipeline can robustly detect five common physical activities, stationary, walking, cycling, running, and in a vehicle (i.e., car, bus). After an activity is detected, the system will further extract physical features that can be used for user classification.

For activity recognition, raw accelerometer data is first captured and then broken into frames. If necessary, a one-off calibration process is applied before preprocessing occurs. During preprocessing, the raw readings are converted into gravitational units and normalized. During preprocessing there are a number of internal stages, beginning with normalization which converts the raw inputs into gravitational units (i.e., g) using device-specific parameters trained from calibration. Normalized accelerometer data is processed by admission control, where extraneous movements of the phone are efficiently filtered out. Examples of extraneous movements are transitional movements such as taking the phone out of a pocket or standing up. Admitted frames are further translated into an orientation independent global coordinate system making any subsequent processing insensitive to the phone orientation. The final transformed output is fed to the feature extraction stage. The extracted feature vector is then provided to the activity classification which can recognize five common physical activities. After the system determines what kind of activity a mobile user is doing, it further invokes an activity specific user classifier that extracts features for classifiing users using the transformed accelerometer data.

2) Voice: Microphone input processing is based on the Jigsaw microphone pipeline [13]. The stream of audio data from the microphone is divided into frames by the preprocessing stage. During preprocessing, the internal steps of ad-

mission control and duty cycling are applied to dynamically regulate the resources used by the pipeline. Following this step, the feature extraction stage extracts a combination of features for detecting human voice. The voice engine uses a voice classification stage to determine whether a frame contains common and easily identified sound classes using a very efficient and accurate decision tree classifier. If the frame contains human voice then the voice data is forwarded for a voice based binary user classifier.

The reason that SenGuard uses voice data as a modality for user identification is that the microphone is a voice communication device. It is easier to intercept voice data when a user makes a call.

To do the voice recognition, three steps are needed. The first one is feature extraction, the second one is training, and the last one is classification. During the feature extraction step, SenGuard uses 12 features which are the characteristic information of a user's voice data. The 12 features are the same as in [14], i.e., the standard deviation of energy entropy, zero crossing rate, spectral rolloff, spectral centroid, the mean of energy entropy, signal energy, zero crossing rate, spectral rolloff, spectral centroid, and spectral flux, and the standard deviation by mean ratio of signal energy and spectral flux. To calculate these features, SenGuard first divides the voice data into W frames, calculated the energy entropy, signal energy, zero crossing rate, spectral rolloff, spectral centroid, and spectral flux of each frame using the corresponding method in [14], and then computed all the statistics. For the second step, SenGuard collects voice data from a common database of multiple users and build a training data set. Each element of the data set is a vector of the features extracted from the voice data. SenGuard learns the parameters of voice binary classifier using the training data set. For the last step, when SenGuard captures the voice data of the smartphone's current user, it can extract the features from the data and use voice binary classifier to determine whether the user is the smartphone owner.

3) Location History: A smartphone user's mobility trace is another important feature for user identification. On one hand, mobility trace of individual user has consistent patterns not only because a user always takes familiar routes when he/she travels, but also because different traces of the same user are regular from one place to another, such as commute between home and work, shopping traces from home to a favorite shopping center. On the other hand, different users often have different mobility traces because they have distinct location preferences, e.g. live or work at different places, and route preferences.

One biggest challenge for designing trace based user identification algorithm is accurately identifying patterns of a smartphone owner while distinguishing the owner's patterns from others. SenGuard's location-based user identification was motivated from three observations:

- Compared with other sensory data such as GPS and Wi-Fi scans, cell ids are widely available on most handheld devices at no extra cost.
- Different users have much less common cell id sequence than common single cell id because cell id sequence is decided by not only where the user is at any moment but also the mobility trace of getting to those places.
- Individual user has many same or similar cell id sequences in his/her daily mobility traces.

Therefore SenGuard exploits cell id sequence as features for identifying smartphone owner.

SenGuards's cell id based user identification algorithm involves two steps: 1) learning cell id sequence patterns, and 2) user identification.

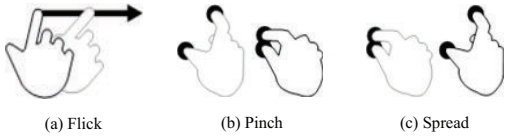
Step (1) has two main tasks: collecting cell id trace and learning the sequence patterns. Cell information represents user's location at a coarse level. And compared with other beacon types location information (Wi-Fi APs, etc.) and direct positioning information (GPS, A-GPS, etc.), it is more energy efficient and can be obtained at almost any given time and location.

Given a trace of cell ids, SenGuard learns the cell id sequence using a sliding window w with length of n. The length n, represents the number of cell ids included in the window. The cell id sequence q in each sliding window is saved. A set of unique cell id sequences $(q_1...q_m)$ collected in the learning phase are used as patterns of a smartphone user.

In Step (2), SenGuard's location based algorithm continuously collects cell id sequence q, then measures the Levenshtein distance between q and each sequence in pattern $q_1...q_m$. SenGuard sets a threshold t for the distance measurement, if $d(q, q_i) > t$, the algorithm indicates that the current user is not the smartphone owner.

4) Multi-Touch: Multi-touch inputs embed characteristics that are user specific and can be used for detecting smartphone owner. Smartphone touch inputs can be classified into several categories, touch gesture (e.g., flick, spread, pinch, drag, tap), virtual typing (e.g, typing using a touch based keyboard, entering a phone number), touch based drawing (drawing shapes using fingers). For each category, it is plausible to extract user specific features from multi-touch traces collected from a smarphone user. Comparing with other similar input modalities, such as mouse motion and keyboard dynamics that have been studied intensively in the past [3], [15], applying multi-touch for user identification is relatively new.

Though mouse motion, keyboard dynamics, and multi-touch inputs are all data produced by motion of human hand



(a) Flick (b) Pinch (c) Spread

Figure 2. Example Multi-touch Gestures

Figure 3. Sample Touch Features

and fingers, they have their own distinctive characteristics. For example, multi-finger touch gestures are unique data. Previous studies applicable to mouse motion and physical keyboard typing can not be applied directly to this type of new data. New studies are required to discover features applicable to multi-touch inputs for user identification.

Using spread touch gesture as an example, Figure 3 shows three subfigures plotted based on a real trace of spread touch from a user. In Figure 3(a), size of each plotted circle increases with touch pressure. In Figure 3(b), size of each plotted circle increases with time. While, in Figure 3(c), size of each plotted circle is based on the ratio of two touch axes (major and minor). The data was collected from a stantum resistive multi-touch device [16]. Figure 3(a) illustrates some of the features that SenGuard extracts from multi-touch gesture. For touch trace of each finger, SenGuard computes its least square linear gradient (shown as two red lines) as a feature. In addition, SenGuard computes the angle between the two gradient values (shown as red angle) as another feature. The distances between the two fingers at the beginning and end (shown as green and blue lines) are also used as features. Furthermore, SenGuard divides each gesture into three segments. The first segment corresponds to the beginning of a touch motion. The second segment, also the longest segment, marks the main touch motion. The third segment corresponds to the end of a touch motion. It is common that a smartphone user may apply different levels of touch pressure at different stages of a touch gesture. Figure 3(a) shows three touch pressure samples collected for the three touch trace segments (shown as yellow circles).

Linux supports multi-touch input device using evdev and recently added multi-touch input protocol. An vendor can implement a multi-touch device driver. The device driver can send out multi-touch events. SenGuard snoops the multi-touch events sent from the device driver and sends them to a multi-touch analysis and user identification engine situated in the privileged domain 0. The guest Meego domain contains a virtual multi-touch device driver. The device driver receives multi-touch events forwarded from domain 0 and injects them into the X server running in the guest Meego domain. The guest domain uses Qt Multi-touch Framework for responding to multi-touch gestures.

C. Classification Aggregation

Each modality dependant classifier can be an one-to-many binary classifier that only takes care of the activity difference between a smartphone owner and others. Each classifier can achieve certain level of identification accuracy. An aggregator can combine results from multiple classifiers and yield an aggregated measurement, see Figure 4 for detailed description. The aggregation can be done by designing a second-level meta classifier using the outputs of each first-level classifier. This kind of aggregation can boost the overall accuracy, better than simply using only one output of each individual classifier. By combing results from multiple sensing modalities over a time window, SenGuard can deliver results with higher recall of unauthorized phone usages and prevent false alarm event of authorized phone usages.

IV. Experimental Data Collection

Nokia N900 phones are used to collect accelerometer and cell ID location data. Touch screen data is collected from a separate multi-touch prototyping device. We don't collect microphone voice data and evaluate its performance as speaker identification is a well-studied area.

A. Accelometer

The N900 phone is equipped with a built-in accelerometer that is a tri-axial MEMS motion sensor (LIS302DL) made by STMicro. It has dynamically user selectable full scales of $\pm 2g/\pm 8g$ (where g is the gravitational acceleration, $g = 9.81 m/s^2$) and it is capable of measuring accelerations with an output data rate of 100 Hz or 400 Hz. The digital output has 8-bit representation with each bit equaling to $18mg$. The configuration of sensor device on N900 phones is set to $\pm 2g$. The sampling frequency of N900 accelerometer sensor is 100 Hz.

Accelerometer based user identification need data collection for training and generating a binary classifier. User walking data is recorded to monitor everyday human movements. During data collection, each participant carried N900 in different body positions. Before the participant performs each activity, he was required to input his name and phone's body position as labels for training and testing purpose. Each activity should last at least 20 minutes. The data set should be collected from people varying in age, gender, height and weight to include all kinds of varieties. Based on this principle, we collected a data set from eight people, five males and three females. The data set is processed according to a sliding window of 512 data points (about 5 seconds) with 50% overlapping. After signal projection into vertical and horizontal components, the following features are calculated from each component as in [13]: mean, standard deviation, mean crossing rate, spectral peaks, spectral sub-band energies, spectral sub-band energy ratios.

B. Location History

Many smartphones are capable of detecting the id of the current cell that the smartphone is connected to. the cell information includes four parts, namely, mobile country code (mcc), mobile network code (mnc), local area code (lac), and cell id (cid). For simplicity and also privacy concerns, SenGuard hashes each cell information into an internal id and refers to the hash code as the "cell id". In SenGuard, the cell id is collected using callback mechanism. SenGuard triggers logging function only when cell id information changes.

C. Touchscreen

For multi-touch evaluation, we used a multi-touch prototyping system developed in house. The system consists a 4.3inch Stantum resistive multi-touch screen, a 640x480 mobile LCD display, and a circuit board connecting with the Stantum touchscreen and LCD. The circuit board contains both a touch controller and a LCD controller. The circuit board can be connected to a computer using a USB cable. The computer can control what is displayed over the LCD. The computer can also receive touch inputs from the touch controller. Stantum resistive multi-touch supports unlimited

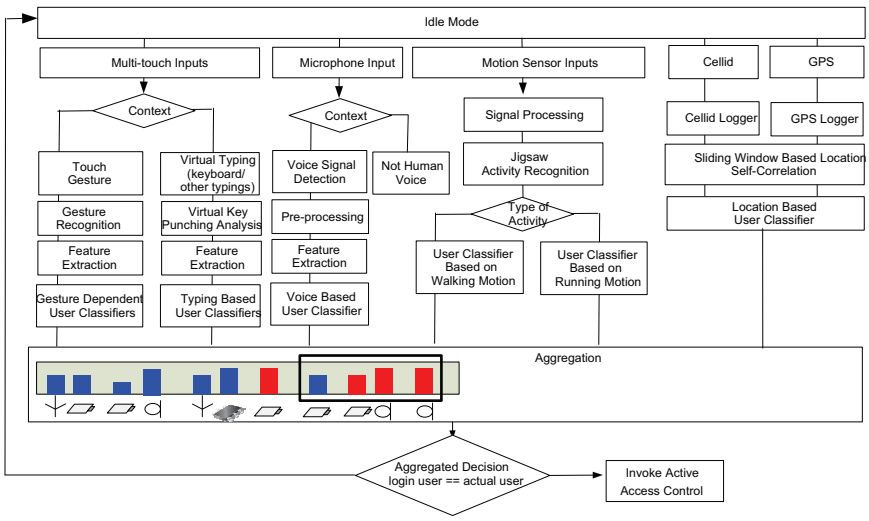


Figure 4. Multi-modality User Identification Classifier

touch contacts. It can sample touch inputs at a scanning rate of 180Hz. It can detect 256 level of finger-pressure. The overall response latency is below 5ms.

As a preliminary study, we collected touch inputs from seven users (three females and four males). Each user was asked to perform a set of tasks using multi-touch. Those tasks include, controlling a smartphone UI using flick touch gesture (left-to-right flick, right-to-left flick), mobile web browsing using pinch and spread touch gestures, entering sentences using virtual touch keyboard, entering phone numbers through touch, performing arithmetic calculations using touch inputs, dragging icons, and drawing simple shapes using finger touch. Each task was repeated multiple times by the same user.

V. Initial Evaluation Results

A. Classifier Results

Table I
CONFUSION MATRICES OF BINARY CLASSIFIERS

actual \classified	User A	Others
User A	0.971	0.029
Others	0.036	0.964

actual \classified	User B	Others
User B	0.955	0.045
Others	0.018	0.982

actual \classified	User C	Others
User C	0.968	0.032
Others	0.036	0.964

actual \classified	User D	Others
User D	0.958	0.042
Others	0.036	0.964

1) Accelometer: Confusion matrices of four Naive Bayes binary classifiers of user A, B, C and D are listed in Table I respectively. Every binary classifier of one user shows high recall rate of unauthorized usages and low false-alarm rate of authorized usages. It can be seen different user's classifier has slight different performance. The results show accelerometer signature of walking can represent user's biometric information with high confidence.

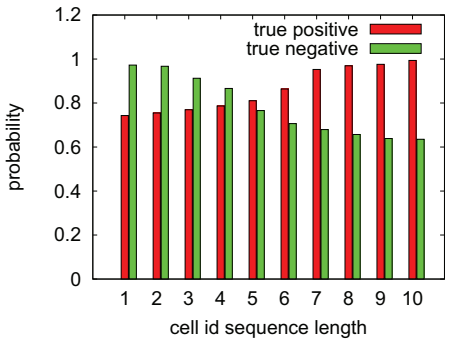


Figure 5. Detection Accuracy using different cell id sequence length.

2) Location History: We evaluated the location-based identification approach by using two metrics: true positive rate and true negative rate. True positive rate represents the accuracy of detecting access from an unauthorized user. If true positive rate is low, it means that most detected unauthorized accesses by SenGuard are actually legitimate accesses from the owner. It indicates that the system prompts the smartpone owner for entering a password too frequently. True negative rate represents the accuracy of detected authorized access, e.g. the device is carried by its owner. If true negative rate is low, it means that SenGuard misses finding unauthorized accesses.

Figure 5 displays true positive and true negative of SenGuard's location matching algorithm using different cell id sequence length. It shows that when length of cell id sequence increases, true positive rate increases as well but true negative rate decreases. This means when SenGuard increase the length of cell id sequence, its location matching algorithm has better performance on finding out accesses from unauthorized users (high recall), but has worse performance on recognizing device owner from them (low

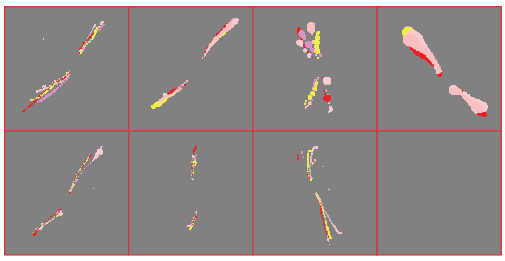


Figure 6. Sample Spread Profiles of Seven Users

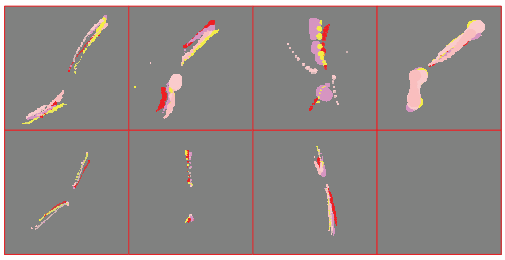


Figure 7. Sample Pinch Profiles of Seven Users

precision). In order to balance the recall and precision, by default SenGuard sets cell id sequence length to 6.

3) Touchscreen: In SenGuard, touch inputs are first classified into different categories, such as virtual typing, multi-touch gesture (e.g., flick, spread, pinch). Then, SenGuard extracts gesture specific features from the inputs. After that, the features are sent to a detection engine that can match the features against a user's profile. SenGuard uses different sets of features for different touch gestures.

Figure 6 shows sample spread touch traces of seven tested users. Each subfigure contains plotted traces of one user. In each subfigure, traces from different test trials are plotted using different colors. For each trace, size of trace dots increases with level of touch pressure.

As indicated by Figure 6, each user has his/her own distinctive spread touch style. No two of the seven users share the exact same spread touch style. For the same user, there is high degree of consistency that the same user will exhibit the same spread touch style. Though collected from different trials, some of the spread touch traces of the same user overlap with one another almost perfectly.

Using the set of features described earlier for spread gesture, we can tell all the seven users apart with high level of accuracy. SenGuard extracts the features from spread gesture by converting each pre-processed trace into a vector of features. Then the feature vector is fed into a classifier trained for each user.

Figure 7 shows sample pinch touch traces of seven tested users. Similar to the plotted spread traces, each subfigure contains plotted traces of one user. In each subfigure, traces from different test trials are plotted using different colors. For each trace, size of trace dots increases with level of touch pressure. Similar to the case of spread gesture, each user exhibits distinctive touch style of pinch gesture. For the same user, the same touch style can be observed across different trials. Again, using the set of features described earlier for spread gesture, we can tell all the seven users apart accurately.

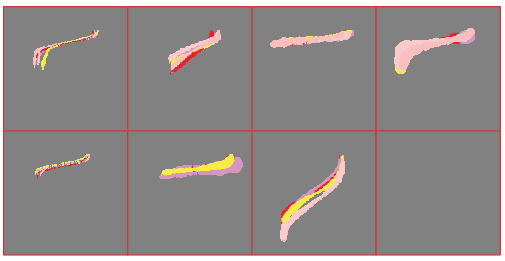


Figure 8. Sample Flick Profiles of Seven Users

Figure 8 shows sample flick touch traces of seven tested users. Different from pinch and spread, flick is a single figure gesture. Each subfigure contains plotted traces of one user. In each subfigure, X-axis denotes screen location translation and Y-axis denotes time. Each subfigure shows traces from different test trials using different colors. For each trace, size of trace dots increases with level of touch pressure. By observing the traces, one can find that for each trace, there was a finger acceleration stage, a steady movement stage (middle section of each flick trace), and a de-acceleration stage. SenGuard extracts steady touch pressure, minor/major ratio, steady finger moving speed, and acceleration/de-acceleration speed as features. The set of features provides good results for differentiating the users. SenGuard extracts those features from flick gesture by converting each pre-processed trace into a vector of features and then sends the feature vector to a classifier trained for each user.

4) Voice: Speaker identification is a well studied area. Over 95% accuracy has been reported in the literature before [17]. So we believe roughly the same performance (maybe a little degradation due to the voice quality from phone's microphone) can be achieved by voice based user identification after human voice is detected.

B. Discussion

We have evaluated the capabilities of four individual modalities to perform user identification. Each modality happens at certain time with its specific duty cycles. For example, accelerometer based user identification is triggered when user is walking, while voice based user identification is in place when human voice is detected around the phone. On the other hand, location based user identification need longer time than accelerometer or voice based user identification, as it is according to user's location pattern change. So we need a sliding time window to aggregate all available outputs from each modality classifier running in the background and design a majority based meta classifier to make the final call. When only part of biometric classifiers are functional, the aggregated second-level classifier can still work and deliver robust performance. This property is essential to passive user identification using multi-modality sensors.

Results from preliminary studies using captured mobile sensor inputs validates the promises of SenGuard as a continuous and implicit user identification solution. Though from a relatively small user set, classification results indicate that it is feasible to implement a non-intrusive and continuous

user authentication system based on sensor data intercepted from normal user-smartphone interaction.

It should be pointed out that SenGuard is not designed to differentiate every person in the world. It is meant to be a critical component of a complete mobile user authentication solution with a purpose tp attain better useable security through balancing security and user friendliness. When SenGuard is insufficient, active user authentication will kick in as the last line of defense.

VI. Future Work

SenGuard represents a first step by incorporating multi-modality user identification into a mobile system. By no means SenGuard has solved all the challenges of supporting implicit user identification on mobile platforms. However, we do believe SenGuard is among the first to demonstrate the viability of the concept with empirical results. SenGuard is an ongoing project that requires more in-depth future evaluations and further customization. Our prototyping system and preliminary results reveal many interesting aspects of SenGuard that can be launch points for further research. For example, one challenge that any mobile implicit user identification system including SenGuard is facing is that, how to ensure continuous service when a mobile system is in power saving or powered off state. At this moment, we are experimenting with a prototype that offloads SenGuard from domain 0 to a standalone low power MCU (micro-controller unit). The MCU provides both an I2C interface that connects to several sensors (e.g., accelerometer, GPS) and high speed serial interface that connects with micro-phone and touchscreen controller. The MCU is separately power managed. Depending on battery level, the MCU can provide continuous SenGuard service even when the main smartphone SoC is powered off.

VII. Conclusion

This paper presents SenGuard, a mobile user identification management solution that can provide continuous and implicit user authentication service on a mobile system. SenGuard is implemented over an open source mobile operating system. It uses a novel virtualization based system architecture that drastically improves protection of its user identification mechanism by moving it into a privileged virtual domain. In addition, virtualization allows implicit and continuous capture of interactive inputs from multiple sensor modalities. An initial prototype of SenGuard was created using four sensor modalities, voice, location, multitouch, and locomotion. Preliminary empirical studies with a small set of human users indicate that those four modalities are suited as data sources for implicit mobile user identification. The preliminary results also suggest a number of interesting avenues for further research.

References

[1] M. Jakobsson, E. Shi, and R. Chow, "Implicit authentication for mobile devices," in *4th USENIX Workshop on Hot Topics in Security (HotSec '09)*, Montreal, Canada, August 2009.

[2] A. Altinok and M. Turk, "Temporal integration for continuous multimodal biometrics," in *Multimodal User Authentication '03*, Santa Barbara, CA, 2003, pp. 11–12.

[3] N. L. Clarke and S. M. Furnell, "Authenticating mobile phone users using keystroke analysis," *Int. J. Inf. Secur.*, vol. 6, pp. 1–14, December 2006.

[4] 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.

[5] K. R. Cuntoor, A. Kale, A. N. Rajagopalan, N. Cuntoor, and V. Krger, "Gait-based recognition of humans using continuous hmms," in *Fifth IEEE International Conference on Automatic Face and Gesture Recognition*, 2002, pp. 321–326.

[6] D. Gafurov, K. Helkala, and T. Soendrol, "Biometric gait authentication using accelerometer sensor," *Int. J. Inf. Secur.*, vol. 1, pp. 51–59, 2006.

[7] J. Koreman, A. C. Morris, D. Wu, S. Jassim, H. Sellahewa, J. Ehlers, G. Chollet, G. Aversano, H. Bredin, S. Garcia-salicetti, L. Allano, B. L. Van, and B. Dorizzi, "Multi-modal biometric authentication on the securephone pda," in *Proc. MMUA Workshop on Mult-Modal User Authentication*, Toulouse, France, May 2006.

[8] M. Indovina, U. Uludag, R. Snelick, A. Mink, and A. Jain, "Multimodal biometric authentication methods: A cots approach," in *Proc. MMUA 2003, Workshop on Multimodal User Authentication*, 2003, pp. 99–106.

[9] J. Muncaster and M. Turk, "Continuous multimodal authentication using dynamic bayesian networks," in *Proc. MMUA Workshop on Mult-Modal User Authentication*, Toulouse, France, May 2006.

[10] T. Sim, S. Zhang, R. Janakiraman, and S. Kumar, "Continuous verification using multimodal biometrics," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, pp. 687–700, April 2007.

[11] MeeGo, http://meego.com/downloads/releases/1.1/meego-v1.1-handset.

[12] P. Barham, B. Dragovic, K. Fraser, S. Hand, T. Harris, A. Ho, R. Neugebauer, I. Pratt, and A. Warfield, "Xen and the art of virtualization," in *SOSP '03: Proceedings of the nineteenth ACM symposium on Operating systems principles*. New York, NY, USA: ACM, 2003, pp. 164–177.

[13] 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*, ser. SenSys '10, New York, NY, USA, 2010, pp. 71–84.

[14] T. Giannakopoulos, D. Kosmopoulos, A. Aristidou, and S. Theodoridis, "Violence content classification using audio features," *Advances in Artificial Intelligence, Lecture Notes in Computer Science*, vol. 3955/2006, pp. 502–507, 2006.

[15] C. S. Ikehara and M. E. Crosby, "User identification based on the analysis of the forces applied by a user to a computer mouse," in *Proceedings of the 36th Annual Hawaii International Conference on System Sciences (HICSS'03) - Track 5 - Volume 5*. Washington, DC, USA: IEEE Computer Society, 2003, pp. 130.1–. [Online]. Available: http://portal.acm.org/citation.cfm?id=820752.821585

[16] Stantum, http://www.stantum.com/en/.

[17] J. Joseph P. Campbell, "Speaker recognition: A tutorial," in *Proceedingd OF the IEEE*, september 1997, pp. 1437–1462.