Secure Session on Mobile: An Exploration on Combining Biometric, TrustZone, and User Behavior

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

***Abstract*—With the rise of Internet connected mobile devices, applications have migrated from PCs to mobile computing platforms. An important aspect, payment processing, faces new security challenges from these developments. Inasmuch, these advancements demand efforts from researchers and industry to meet increasing security needs. Threats can ensue from data loss, theft from lost, stolen, or decommissioned devices, information-stealing malware, and password peeping. We propose a secure framework for sensitive session driven applications which combines biometric-based continuous and implicit tracking of user identities, and TrustZone. This framework is accomplished through monitoring fingerprint authentication logs as well as detecting events when the phone has left the user's hands, all while in TrustZone, a platform for secure computation and storage on mobile devices. This solution leverages multiple onboard sensors as well as the ARM architecture to accomplish these feats. We conducted two user-studies acquiring smartphone users' usage statistics to investigate security and usability needs of our identity-tracking 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 this framework as a secure session-based payment system.**

***Index Terms*—TrustZone, Secure Session, Biometric, Sensor Fusion, User Behavior**

I. Introduction

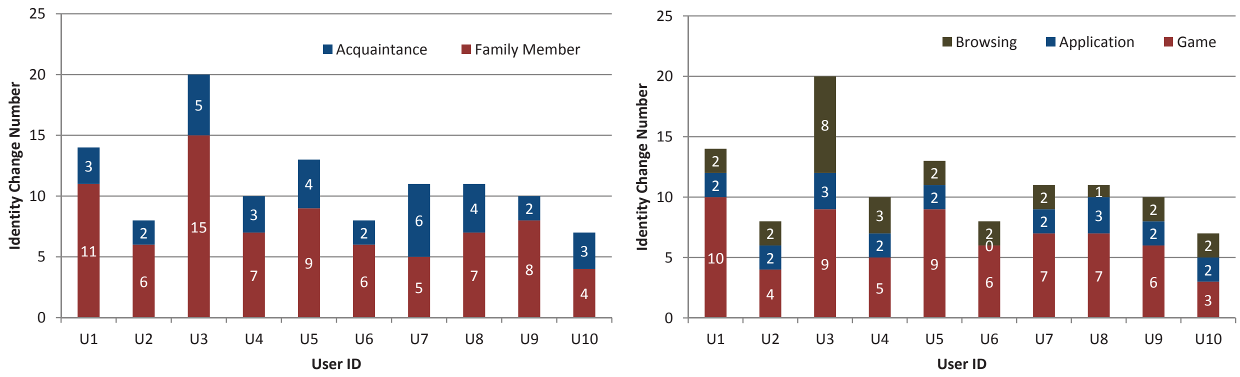
In a world increasingly dominated by technology, more and more sensitive informations such as transaction information for bank accounts, credit cards, trade secrets, and etc., is passed through mobile digital devices. While these new uses introduce more convenience and a richer experience, they also create new privacy and security issues. In response, these devices have now become targets for hackers. We can see these trends active in the market. From 2011 to 2012, mobile malware families ballooned by 58% [1], 32% of which were used to steal information [1]. In January 2012 alone, there were 32 million data breaches, of which 40% were caused by hackers [2]. In response to these shockingly large numbers, there is a need to process sensitive information in a way that is independent of a potentially infected operating system while monitoring physical events of the device to detect possible physical unauthorized use. Previous mobile user authentication technologies such as passwords only offer protections at the login-point. However, devices may be accessed by other impostors in phone theft scenarios rendering the password useless if already logged in [2], [3]. To investigate the frequency when smartphone devices will be utilized by guest users, we have implemented and installed an application on ten users' smartphones and continued to track the smartphone usage for one week. Data from this experiment reveals 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 or intentionally perform some operations the owner may not have intended. While in hacking scenarios, the password may be useless or ineffective in the case it is known.

First tackling the issue of hacking and known user-password scenarios, we introduce the use of a fingerprint authentication used in tandem with TrustZone. In any sensitive application a secure session must be established. A stage where payments may be made, as in the example of the bank application, the user must first request a session, the phone will then switch secure mode(TrustZone). A fingerprint, rather than a password, which can not be replicated or stolen nearly as easily will be used to authenticate the user. Once in TrustZone sensitive information will never leave a secure sandbox and if by chance a malicious user was to get the device it would be rendered useless in terms of accessing the stored payment information in TrustZone. TrustZone consists of a hardware enforced security environment providing code isolation together with secure software that provides both the fundamental security services and interfaces to other elements in the trusted chain, including smartcards, operating systems and general applications. TrustZone separates two parallel execution worlds: the non-secure normal execution environment, and a trusted, certifiable secure environment. The normal environment is used to complete daily tasks that do not utilize sensitive information such as listening to music, installing apps, and etc. All sensitive applications and services will run in the secure environment. This will effectively hamper many different techniques of hacking for this sensitive information.

While a good solution, this verification framework requests to authenticate identities each time when they access sensitive apps or information, sacrificing the user experience. To fix this issue we introduce continuous implicit biometric identity verification while in a session as a background process. This framework processes user semantics to deduct what state the phone is in (i.e., in right-hand, in left-hand, on table, or user switch). If the phone was to be set on a table or if the

MobiCASE 2014, November 06-07, Austin, United States

DOI 10.4108/icst.mobicase.2014.257767



(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

user was to switch, the session would be closed immediately protecting the user from possible unauthorized use. This is accomplished by the use of many on-device sensors such as the accelerometer, gyroscope, and etc. Once this data has been processed the phone state may then be determined. In order to prove the effectiveness of such technology we implemented 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 during the entire session. 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. Generated statistics of the ten-user one-week authentication data are displayed in 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.

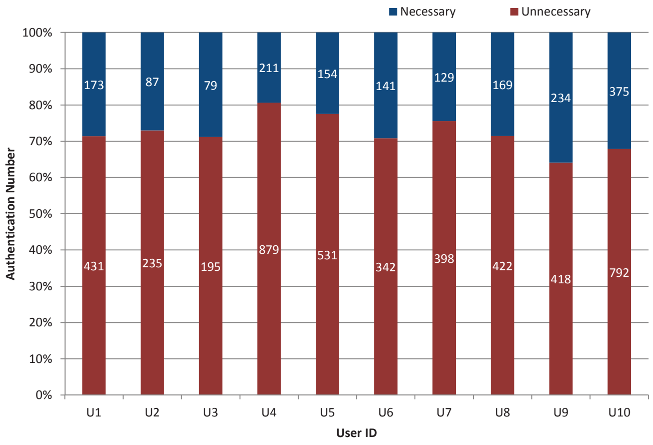


Fig. 2. Unnecessary authentication

Current methods of session based or sign-on security problems include single-stage verification and have not taken into account whether or not the mobile operating system or device has been compromised. If compromised, the hacker/user could steal anything from a password up to a certificate allowing them to wreak havoc upon the users wellbeing. Also current approaches do not adequately take into account whether the device has been physically stolen. Inasmuch, we have designed a TrustZone implemented approach that is becoming readily available to implement within the market on phones such as the IPhone 5S [4] and the upcoming Galaxy S5 [5]. Using our approach, the mobile device will enter the secure world. Now that the process has entered into the secure world as a service, it will be isolated from the normal operating system and any malicious services that may reside upon it. The phone will now continuously monitor sensors in the device to detect events where the user may set the phone down or hand it to another person as well as read the fingerprint of the user. Once initial verification via a submitted fingerprint has verified the user as the owner the session will be open to process or complete any requests made by the owner. In the example of a bank application tranactions would be requested within the session, a nonce would be returned from the bank server which would be signed in the secure mode using a certificate also stored in the secure mode. During the process the service remains in secure mode. If at any time a transfer or device placement event to occur the session would be closed immediately and the phone would leave secure mode. This all coupled will create a highly tamperproof solution to user verification transaction processing.

Our contributions are as follows:

- We design a framework for secure session-based applications, which leverages TrustZone and continuous biometric authentication schemes;
- The framework designed contributes to security heavily while not only maintaining, but improving device usability and convenience for the user;
- We implement identity verification schemes identifying its security properties and the results indicate our solution is practical.

II. Background

Of the main components, as shown in Fig. 3, TrustZone is the most crucial, as it controls the secure processing of sensitive data. TrustZone is an extension to the SOC (system-on-a-chip) ARM design covering everything from the processor to the memory to the peripherals. Using TrustZone design, the physical core is virtualized into two separate cores: one of which is referred to as the "secure world," and the other

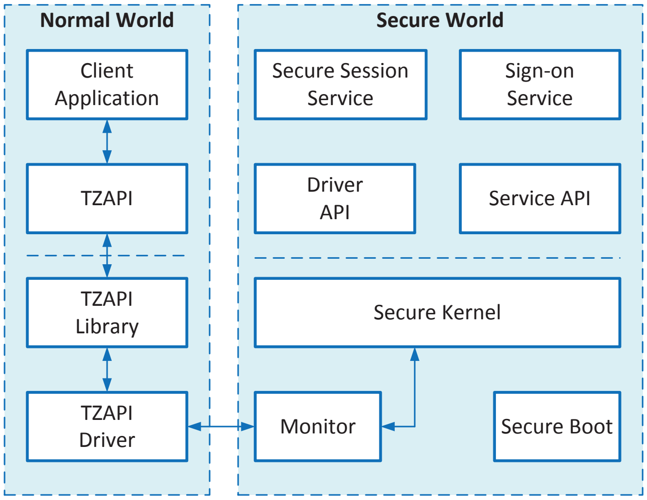


Fig. 3. The general software layout of TrustZone. Note that there is the secure world and the normal world. Our approach places a transaction service within the secure world, which will house the process detailed within Sec. IV

which is the "normal world." The secure world is used when processes will be interacting with or collecting sensitive data. This could range from using the keyboard, fingerprint readers, or extended to interactions with a bank app. In contrast, the normal world is used when there is no sensitive data being processed. This could include things such as playing games, taking pictures, and etc. The apps that run in each respective world can be chosen and defined as per what is deemed sensitive. These two modes are completely isolated from each other to stop data leaks. Secure services that are run can range from a complete operating system to trivial services. TrustZone achieves this functionality through the novel addition of monitor mode. The monitor is used as a faucet of communication between the two worlds. Inasmuch, the system is only as robust as the monitor, so it must be sensitive to what is transferred between the worlds. The monitor also handles the context switches between the two modes which naturally gives it the responsibility of saving the state and switching safely between the two modes. The way that a program will enter secure mode is by throwing an exception. The exceptions that may trap to the monitor causing a world switch are FIQ, IRQ, and external aborts. This effectively allows sensitive processes to be run in this mode such that they are secure from any malware that could be present within the normal world. This is the main motivation of the TrustZone architecture.

The other critical component of this system, as seen in Fig. 4, is a biometric fingerprint sensor. As the main focus of this paper is not fingerprint sensor technology, this will not be addressed in depth. The type of fingerprint reader is irrelevant. The sensor must first generate an image of the fingerprint. This can be through either optical, capacitance, or ultrasonic means [6]. Optical sensors use visible light to capture an image. This has major weaknesses in that if the finger is dirty it can be difficult to retrieve a good image. Capacitive readers use capacitance in order to generate an image using an array of sensors to detect the fingerprint. Last but not least, ultrasonic readers use high frequency sound waves to generate an image. Regardless of the way that the image is retrieved, it is first prepossessed where the picture is enhanced and adapted so that feature extraction can generate more reliable identity traits from the user. After feature extraction, a feature vector can be obtained containing discriminative properties (such as ridge ending, bifurcation, and short ridges). The last step is to compare this vector with existing templates where a matching score is generated. If this score is above a threshold, the fingerprint is accepted as valid. If not, then it is rejected [6], [7].

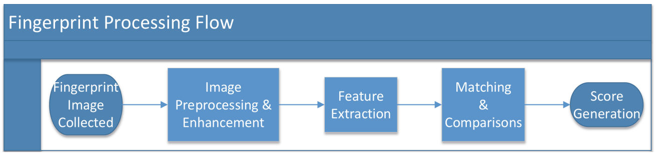

Fingerprint Processing Flow
Fingerprint Image Collected
Image Preprocessing & Enhancement
Feature Extraction
Matching & Comparisons
Score Generation

Fig. 4. The general flow of processing a fingerprint

III. Threat Model and Security Goal

The proposed system consists of three ends as shown in Fig. 5, the *Web Service*, the *Smart Phone* device and the *User*. The *User* tries to do transactions with the *Web Service* through the *Smart Phone*.

1) *Web Service* is fully trusted. It can always keep the secrets and follow the pre-defined protocols.
2) *User* is not trusted. A malicious user may try to cheat the *Smart Phone* and the *Web Service* by pretending the legal user and/or make illegal transactions;
3) *Smart Phone* is not fully trusted. We assume that the hardware of the *Smart Phone* is not compromised and supports the TrustZone security extensions. Due to the openness of some mobile platforms such as Android, it is not uncommon for users to have malicious applications installed. When running without the support of TrustZone, data stored in the smart phone and inputs from the user may be stolen or modified. When TrustZone is enabled, we assume all these information are properly protected.

Based on the above assumptions, the security goal of the proposed system is to protect the transactions between the *User* and the *Web Service*. However, side-channel attacks or physical attacks are not considered by the proposed system, since these attacks fall outside the defense capabilities of TrustZone technology.

IV. Secure Session based Mobile Sign-on Solution

In this section we provide detailed description of the proposed solution for secure session based mobile payment activity.

Solution overview: The online mobile payment scenario may be abstracted to a tripartite interaction protocol. The three parties included are the *User*, *Smart Phone*, and *Web Service*. The *User* wants to sign on the *Web Service* using a *Smart Phone*. In response, the *Web Service* issues a credential (e.g., public/private key pair) to the *User*, which is saved in the

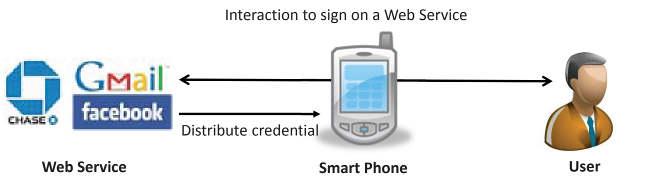


Fig. 5. A high level overview of secure online mobile sign on service. Notice that the phone acts as the hub between the user and the server, gathering information from each entity and distributing information back to the web server.

Smart Phone and protected with TrustZone. The *User* submits their biometric information (e.g., fingerprint) to log in to the *Smart Phone* system and the secure session. The mobile device utilizes motion sensors to detect user identity changes and determines if the secure session should be terminated or continued. When a new transaction takes place, the *Smart Phone* checks the secure session status and uses saved credentials to authenticate himself to the *Web Service* and complete the transaction.

Detailed description: The proposed system is constrained only by the architecture of the processor used in said device as well as the availability of a fingerprint biometric sensor. The processor used must be an ARMv6KZ or later application profile architecture. The reason for this is that ARM architectures prior to these do not have TrustZone implemented. Since TrustZone is hardware implemented there are no other options for older devices which does not provide hardware support for it. Once this requirement has been satisfied, the system may be very sensitive or lack some sensitivity as the process relies heavily upon the accuracy of the fingerprint sensor and the technology therein. It is assumed that when using this method, the quality of sensors will match the application i.e., an sensitive application will require and utilize high quality sensors. Please note that this process is session based, meaning, if the user requests three transactions, they will have to verify their fingerprint only one time to begin the session given the user using the device is constant.

As seen in Fig. 6, the software process is as follows: First the user will log on to some sensitive apps such as a bank app to pay bills. This process remains the same as is now. However, once the user requests a session, the device will throw an FIQ exception to enter the secure mode. This exception will be handled by the monitor, which verifies the app has permission to run services in secure mode. Upon validation, the monitor will complete the context switch into secure mode. Now that the service is running inside the secure mode it will not be affected by any malware infection that may be present inside the phone because once the context switch has been made, the process is running independently and completely isolated from everything within the normal mode. The application will now request the user to submit a fingerprint for verification. Because we are not leaving secure mode to collect or process the fingerprint, it too is safe. This, if stolen by hackers, could be detrimental as fingerprints are used in many settings to verify a user. This fingerprint will be processed as is explained in the background section. Once the fingerprint has been processed if the score generated is below a set threshold the verification will fail, service will end and the device will be context switched back into normal mode. However, if the fingerprint matching score is above the set threshold, the service will continue inside secure mode. The user is now free to request transactions to be made. The requests will be processed and for each request a nonce will be returned from the bank server. The nonce may contain transaction information such as time, the value of the transaction, and the recipient. Never leaving secure mode, the nonce will be signed using the certificate that too is stored in the secure mode and esnd the signature to the bank server to complete the transaction. Granted the user is the same and the session is continued the fingerprint will not be required for subsequent transactions. However if the session is ended by either the user or by the process due to detection of a user-switching event, a new session must be started (requiring a fingerprint). As seen in Fig. 7, the hardware process is as follows: Once the nonce has been received and the FIQ exception is thrown, it will be trapped by the monitor which, if a valid request, will carry out the context switch and pass the nonce via a register write. The APB (AXI to Advanced Peripheral Bus Bridge) will then request the I/O Controller enable the fingerprint sensor. Once complete, the processor will generate the score. Assuming the score is above the threshold the processor will generate the signature. The signature will be passed via a direct register write, and sent to bank. Then the bank can verify and complete the transaction.

V. Secure Session Service Framework

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

A. *Overview of Secure Session Service Framework*

Fig. 8 depicts the high-level architecture overview of Secure Session Service Framework. There are four main components within this framework: A Touch Fingerprint Sensing Module, which employs the fingerprint sensor deployed on new generation smartphones, *i.e.,* iPhone 5s and Galaxy S5, to identify the current user's identity (by fingerprint verification); a Fine-grained Activity Recognition Module that employs touch and motion sensors on the smartphone devices to detect a set of pre-defined smartphone physical motion activities; an Identity Reasoning Engine that analyzes the identity of current smartphone user based on the input from the previous two modules; and an Unauthorized Access Accountability Protection to protect the smartphone system which logs the unsuccessful secure session opening attempts. In normal usage scenarios, when the user requests to sign-on a web service, a password is required for each separate sign-on requests. However, using the further explained process, once a sign-on service is requested, the Touch Fingerprint Sensing Module detects and logs the user's identity and starts a new secure session. Once the identity is authorized and the new secure session begins, the Fine-grained Activity Recognition Module continues to track the touch screen usage data and the motion sensor data to

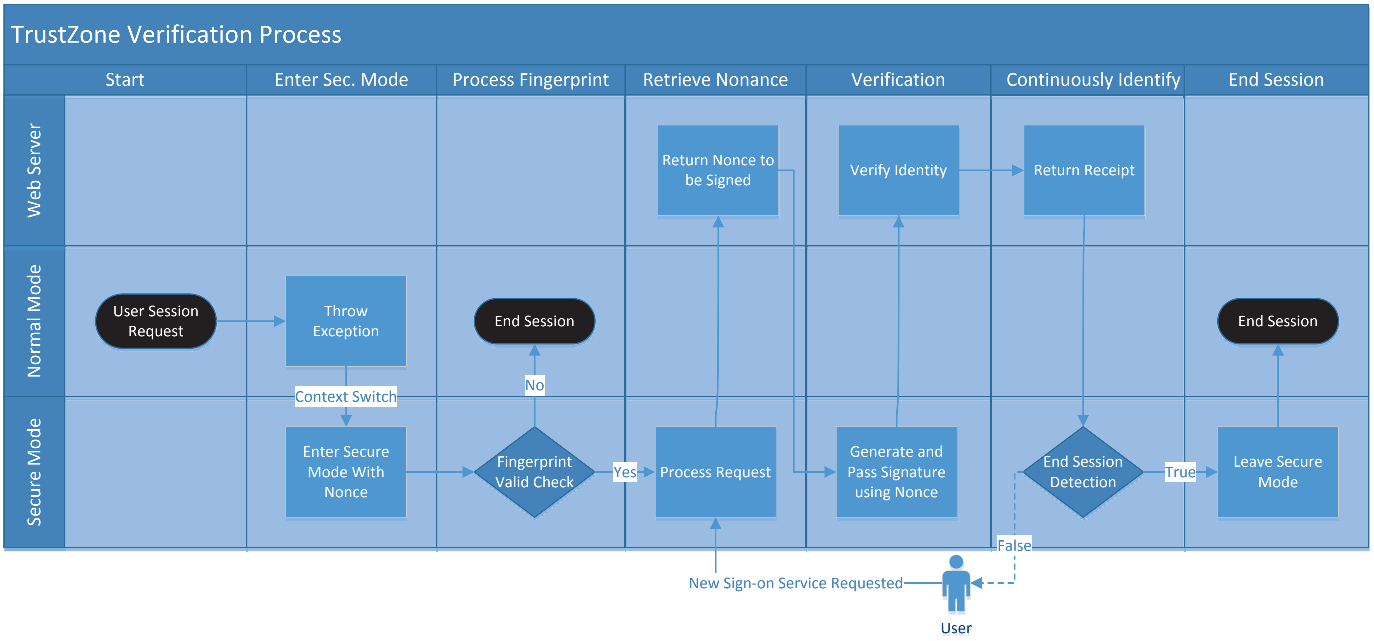


Fig. 6. A depiction of the authentication process of a sign-on request. Black ovals represent beginning or ending points, diamonds represent decisions, and squares are general processes. The process begins as a request for a session from the user made in normal mode. The system first enters the secure mode via an FIQ exception. The program will verify the user by fingerprint. If incorrect, the process will leave secure mode. If correct, the user is able to sign on the web service. Once a sign on service is requested a nonce is retrieved from the web server. Once received, the process will sign the nonce using the certificate stored in the secure mode and send this back to the web server to complete the sign-on process. While still in session the continuous identity verification will continue until another sign-on service is requested and restart the process at the ”Retrieve Nonce” step. If while waiting the device detects a user-switch event, the session will end automatically or if the user requests the session to end.

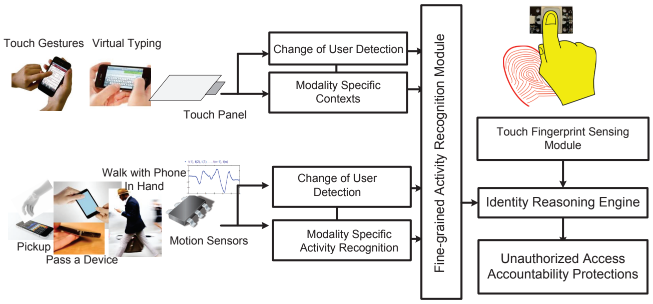

Touch Gestures
Virtual Typing
Touch Panel
Change of User Detection
Modality Specific Contexts
Walk with Phone In Hand
Pickup
Pass a Device
Motion Sensors
Change of User Detection
Modality Specific Activity Recognition
Fine-grained Activity Recognition Module
Touch Fingerprint Sensing Module
Identity Reasoning Engine
Unauthorized Access Accountability Protections

Fig. 8. Overview of Secure Session Framework

monitor subtle gestures, such as device-leaving-hand events. The subtle gestures are then sent to the Identity Reasoning Engine for detecting smartphone user identity changes. The secure session will be terminated when the Identity Reasoning Engine reports that the current user is not the owner of the device. Anytime the current user of the smartphone device wants to perform an online payment activity, the smartphone will check if the device is in a secure session, if so, the sign-on request can be successfully processed. Otherwise, the device will prompt a fingerprint authentication process and require user to verify his/her identity. And if the unauthorized user tries to sign-on a web service, the information will be logged for further processing.

The Touch Fingerprint Sensing Module is a mature technology on smartphone devices. The highlight and key point of Secure Session Service Framework are the Fine-grained Activity Recognition Module, the Identity Reasoning Engine and the Unauthorized Access Accountability Protections. We will discuss the details of these two modules respectively in the following.

B. *Fine-grained Activity Recognition Module*

To detect a device-leaving-hand events and in effect detecting an identity change event, we first define a set of subtle gestures and their corresponding context user statuses as listed in Fig. 9. Since we are solving identity-changing problems in the post-login stage, we only consider the smartphone physical motion status in unlocked state. Essentially, there are four statuses when the smartphone device is in an unlocked state, which are respectively: On Table, Using by the Left Hand (of a user), Using by the Right Hand (of a user), and Using 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 Pick-up 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

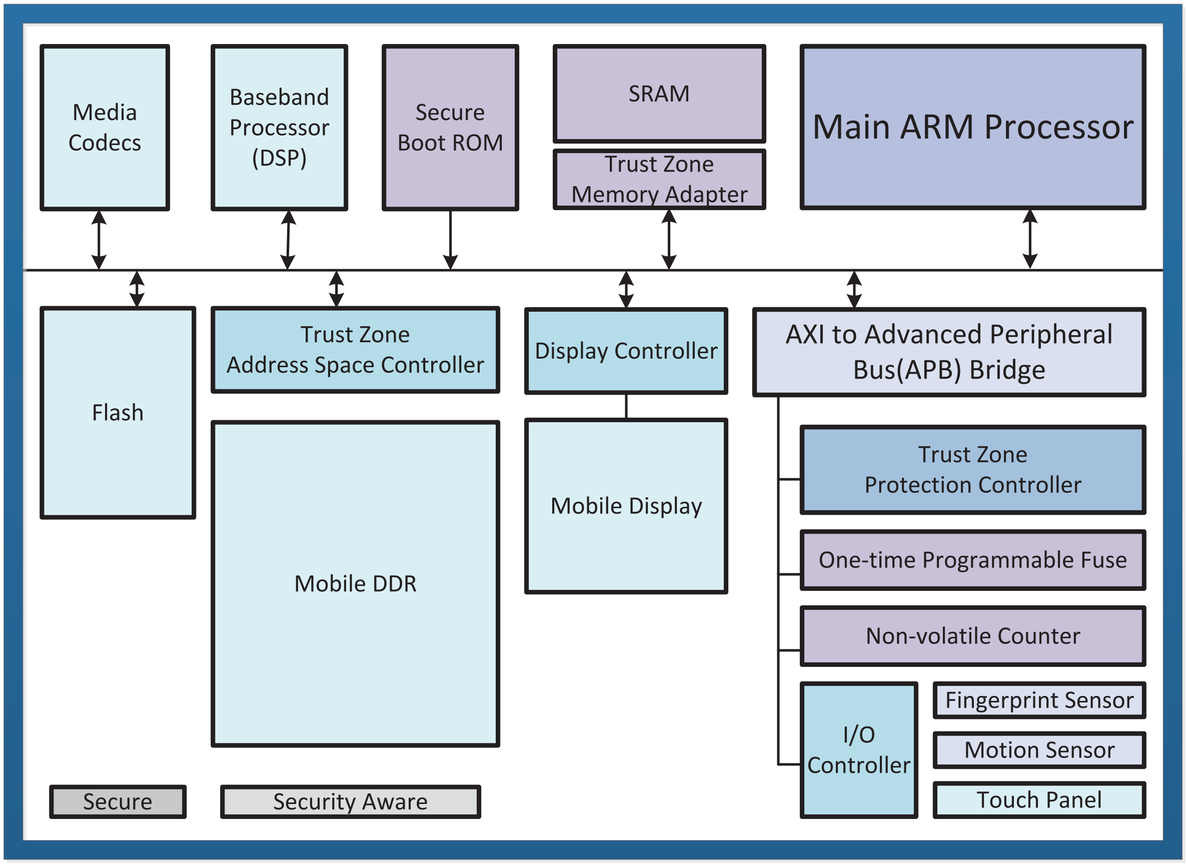


Fig. 7. The general hardware layout of TrustZone for our design. Note that within the IO Controller we have a fingerprint sensor and touch panel which will be protected by TrustZone.

the status or subtle gestures of the device. The Secure Session Service Framework 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 most activity recognition works, Secure Session Service Framework 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 Transfers between one user's hands. To solve the subtle gesture recognition in these complicated scenarios, we can leverage more accurate predictions (*i.e.,* On Table, Using by Right Hand, or context user status, such as walking) combined with the transition map(Fig. 9) to analyze those hard to detect subtle gestures. In the mean time, the Secure Session Service Framework 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).

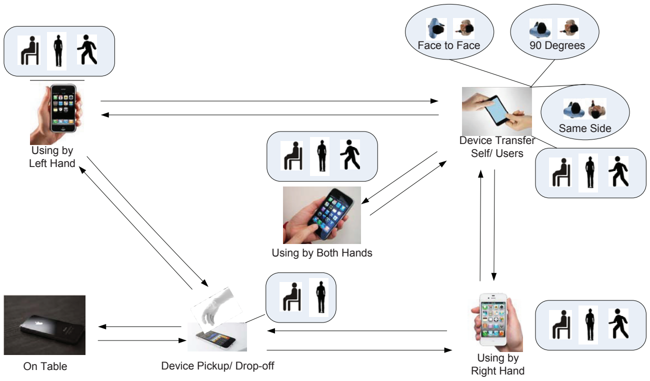


Fig. 9. Subtle gestures during user's normal smartphone device usage in the post-login stage. There are four physical motion status of the smartphone device and four subtle gestures between these status. User physical motion status and relative position between different users are also considered.

C. Identity Reasoning Engine

As long as we acquire the physical motion status and subtle gestures of the smartphone device listed in Fig. 9, we can combine them with the Touch Fingerprint Sensing Module to determine the current status of the user. The identity reasoning process is shown in Fig. 10. Because at the beginning of each session the Service can employ the Touch Fingerprint Sensing Module to log the identity of the current user we may safely

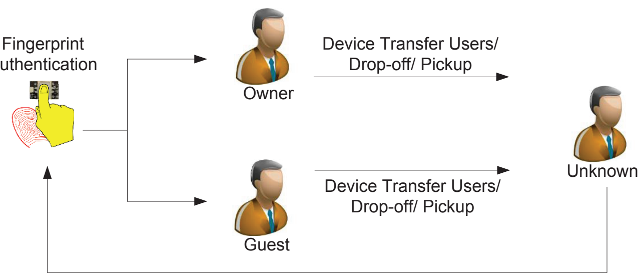

Fingerprint Authentication
Owner
Guest
Device Transfer Users/ Drop-off/ Pickup
Device Transfer Users/ Drop-off/ Pickup
Unknown

Fig. 10. The process of Identity Reasoning Engine

assume it is the verified user. Supposing that the current user is indeed the owner of the smartphone device, we then allow the session to continue until the device has been detected leaving the user's hands. It may be either transferred to other users or placed on a table in the Fine-grained Activity Recognition Module. The identity of the current user is considered as the smartphone owner within this session. If a device-leaving-hand event is detected *(*i.e.,) Device Transfer Users, Drop-off, Pickup), the identity will be logged as unknown and wait for another session start point to acquire an identity output from the Touch Fingerprint Sensing Module. Meanwhile, a guest user's identity can also be recognized by the fingerprint authentication, and will be marked as unknown when a device-leaving-hand event is detected. Since a guest user can never pass the fingerprint based identity verification, s/he can never start a secure session and can only access to those nonsensitive functions.

D. Unauthorized Access Accountability Protections

In the case that an unauthorized user attempts to enter a secure session (we are able to detect this when the fingerprint verification fails) we must take steps to prevent further access and deter unauthorized users. Because of the methods currently employed in fingerprint verification forbid fingerprint recording due to privacy concern, we are not able to record the fingerprint directly. However to combat the problem, in this process, when the fingerprint verification fails (which is obviously detectable) the gps coordinates are recorded with a time stamp and a picture taken using the front camera to capture the user. This information is emailed to specified users. This way immediate action may be taken to mitigate further and subsequent unauthorized access attempts as well as deal with current issues. This can greatly reduce the fraud that occurs in relation to the device as well as recover lost assests in the case they are able to access the session.

VI. Experiment

We implement the continuous biometric verification framework (which will be now refereed to as Secure Session Service Framework) as a background service that implicitly collects motion and touch screen data, and logs the user's identity when an attempt is made to sign-on a web service. The Secure Session Service Framework is installed on 13 smartphone users phones. The experiments consist of three sessions and the process is shown in Fig. 11.

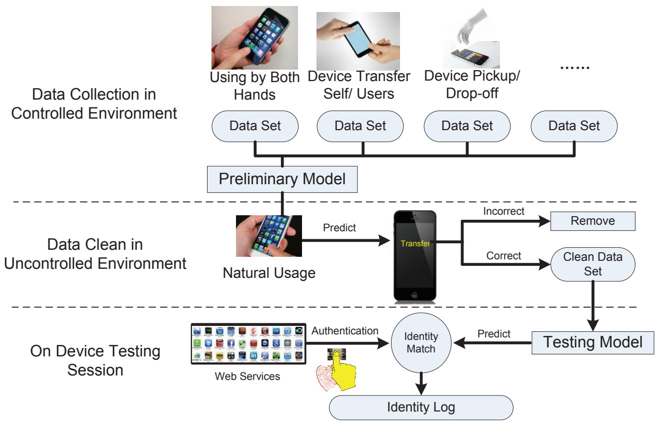


Fig. 11. Process of the experiments

A. Data Collection and Data Clean

In the data collection session, Secure Session Service Framework collected a set of phone usage data from users. Users followed the instructions provided by Secure Session Service Framework to perform a set of gestures and operations. This set includes 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 Secure Session Service Framework, 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. For example, when the Secure Session Service Framework records Device Transfers between different users' hands, it does not just collect the data of this subtle gesture, it starts recording when user unlocks the device. In addition, it records the subtle gesture sequence of Use by Right Hand, Device Transfer between different users' hands, Use by Right Hand, Use by Both Hands, and etc. So the data would be more close to user's natural usage.

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 Secure Session Service Framework 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 them to perform. If we aim to perform and detect device-leaving-hand events in uncontrolled environments, a more accurate set of training data is required. However, since all the subtle gestures listed such as Device Transfers or Device Pickups/Drop-offs happen in a very short time frames and any extra label actions would interfere with normal gestures, we decided to first train a model based on the data collected in the previous session. This model was then used to predict the current status or subtle gesture of the smartphone device and display it. 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 Secure Session Service Framework acquires clean data in an uncontrolled environment, it trains a new model based upon this new data set. After we install the new model on the device, Secure Session Service Framework still authenticates user's identity whenever a sign-on service request is being processed using fingerprint authentication and logs ground truth of user identity. Meanwhile, the testing model also outputs a prediction result of current user's identity. The Secure Session Service Framework matches the two user identity results and log them for further evaluation.

VII. Evaluation

In this section, we will first evaluate the performance of the Secure Session Service Framework, and then discuss its usage in TrustZone.

A. Performance Evaluation of the Secure Session Service Framework

We evaluated the performance of the proposed system in both security and usability aspects: i) How many times had unauthorized web service sign-on requests been reduced in comparison to a mobile system without post login access control and how many unauthorized web service sign-on requests accesses had been granted; ii) How many instances of unnecessary authentication had been reduced for the phone owners in contrast to a strict post login access control mechanism and how many instances of unnecessary authentication had been requested.

Fig. 12(a) depicts the identity match log results of the on-device testing session. The red bar represents the number of unauthorized accesses blocked by Secure Session Service Framework, while the blue bar marks the number of unauthorized access Secure Session Service Framework failed to detect. It is clear that in comparison to the mobile system without post login access control, Secure Session Service Framework greatly reduces unauthorized accesses (above 95% of unauthorized access request were denied) and only few times a guest user was allowed to perform a sensitive operation.

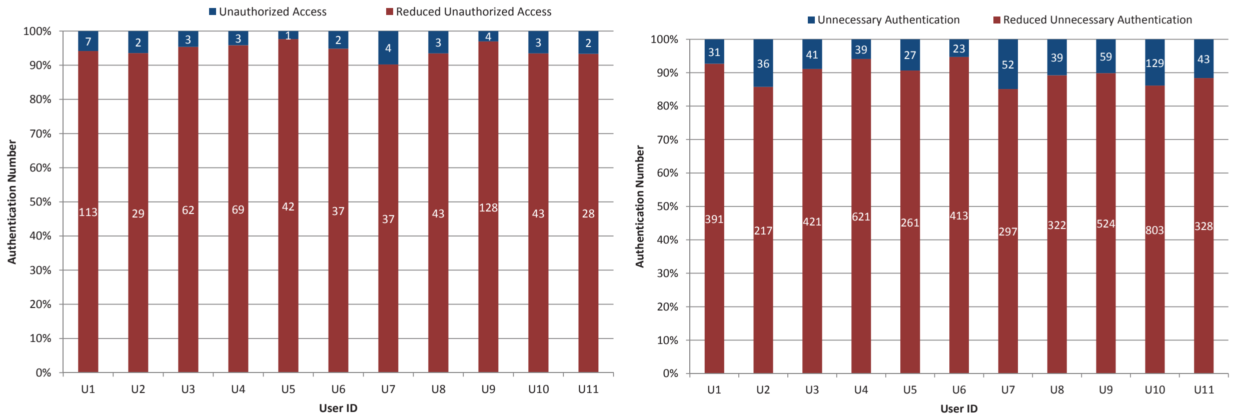
Fig. 12(b) depicts the usability enhancement results based upon the logged results. In comparison to the mobile system with strict post login access control that requires authentication every time when a user attempts to sign-on a web service, Secure Session Service Framework alleviates the user's burden of constant authentication when he/she attempted to perform such operations (themselves). Above 85% of authentications have been reduced by Secure Session Service Framework. Although Secure Session Service Framework may introduce a few unnecessary authentication events by falsely detecting a device-leaving-hand event, it still promotes the usability.

B. Discussion

Of the advantages of this approach, the most substantial one is the fact that this process uses a three-stage verification that may not be tampered with by any infection located in the normal mode of the mobile device. This is achieved by context switching into the secure mode. Once the nonce has been received from the web server the context switch is completed. Regardless whether infections see the nonce, it does no good without the certificate. So it is ok to possibly expose this to infections. Since after this time all infections have been isolated to the normal mode context (not in the secure mode). Thus, we may safely trust the secure context now being used. The first stage of verification, which is the fingerprint recognition, is also sensitive so it must be completed in TrustZone as well. Instead of sending this data directly to the bank server and risking fingerprint data stolen by hackers, we, as is the norm, use signature generation to verify the user to the server, which completes the second stage of verification scheme. This certificate is pertinent information and inasmuch must be stored in TrustZone as well. Then, granted the user does not end the session, continuous implicit biometric verification is utilized as well to further protect subsequent transactions within the session (which remains in secure mode as well). This continuous verification will monitor the users actions. In any case that the phone has possibly left the direct possession of the user the session is closed automatically. Whether the release of the device was intended or not. Throughout this whole process, no information is leaked to normal mode.

Besides the inherent protection offered by TrustZone, implicit continuous biometric verification, and fingerprint scanning technologies, there are a few other strong advantages within this approach. Currently, if a password is used, even with TrustZone, the hacker would be able to compromise the mobile device. While they would not be able to retrieve the certificate because of TrustZone, the certificate could be used to verify themselves effectively rendering the system ineffective. However, our approach does not employ these means. Since we are using fingerprint technology, the hacker would either have to have the person with them whose account they are trying to compromise present (who would naturally not allow this), or have collected a good fingerprint sample and replicate this fingerprint in a manner that is able to trick the fingerprint sensor. Although only few hackers would have this ability. Also it is important to note that since our method uses continuous implicit biometric verification rather than time as a session ending variable, our process can correctly handle phone theft while intra-session cases that would normally result in unauthorized use. Our method would end the session automatically once detecting the phone transfer and in effect render the phone incapable to carry out any subsequent web service sign-on request with a fingerprint.

Granted that some way an unauthorized user attempted to access/start a secure session in the phone. The current session, if one is ongoing would be ended and a new fingerprint verification would be required to start another. The Unauthorized Access Accountability Protections would record all data related to the denied access and how many attempts have been made. This after being emailed to the correct user would allow the device to be either recovered, or wiped as is per the norm. However since we record not only the face but also the location and time they may be used to apprehend the user



(a) Security Evaluation

(b) Usability Evaluation

Fig. 12. Performance evaluation of Secure Session Service Framework

and recover the device. This scheme also allows for the use in situations where the device is unknowingly taken and put back. Normally in this case the owner would not be aware of these attempts of that the phone had been stolen or tampered with. With our method this event would be recorded as well.

Currently, mobile devices only have the ability to accomplish fingerprint reading. As technology advances, the hardware in phones would increase in sophistication as well. Inasmuch if finger vein verification was integrated into hardware instead of fingerprint, the process would be much more robust. That said, fingerprints, as previously stated, may be dirty and make verification difficult. Finger vein scanning technology is not affected by the surface of the skin as veins are below the surface [8]. The process is also much more secure, in that as veins are hidden inside the body, there is little risk of forgery or theft [8]. While not infallible, this approach can offer increased security for settings that require it.

VIII. Related Work

The research of secure session based mobile payments draws from multiple areas, such as TrustZone, implicit and continuous identity authentication and Activity Recognition. We will discuss them respectively in the following.

TrustZone. Korean researchers have built a TrustZone-based platform for Android to prevent malware infections [8]. Also based upon TrustZone, Luo *et. al.* designed a dual operating system, one satisfying users application requirements and another acting as a secure OS providing specific security services [9]. Martin Pirker and Daniel Slamanig proposed a platform framework on TrustZone that may be used for arbitrary applications requiring a privacy-preserving online remote prepaid payment system suitable for micro as well as macro payments [10]. However, neither system provides a concrete design of user authentications on TrustZone. Researchers from ETH Zrich suggested utilizing TrustZone with a password to solve secure enrollment problem [11], but password authentication would be useless if the phone is lost and the password has been stolen.

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 [9], [10], GPS [11], touchscreen [12]–[17] *cite out paper*, microphone [18], and fingerprint sensor [19]–[22]. However, unlike previous works, we do not directly leverage the sensor readings and perform user authentication based upon this. In our method the fingerprint is retrieved to identify the user's identity and 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 [23]–[25]. In [26], Lu *et. al.,* proposed a continuous sensing engine for activity recognition on mobile platforms, which can detect five common physical activities: stationary, walking, cycling, running, and in a vehicle (i.e., car, bus). Yang *et. al.,* [27] also completed research on activity recognition by exploiting the accelerometer data. Different from the aforementioned works, the goal of this paper is not to detect a long term and stable motion but short term subtle gestures that take place in very short time frames to monitor users' identity changes by detecting device-leaving-hand events.

IX. Acknowledgement

This material is based upon work supported by the U.S. Department of Homeland Security under Award Number N66001-13-C-3002. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the opinions or policies of the U.S. Government. Mention of trade names or commercial products does not constitute their endorsement by the U.S. Government.

X. Conclusions and Future Work

In this paper we have proposed a novel method for multi-stage verification of identities in sensitive payments or otherwise sensitive sessions. Using this method, we are able to isolate the sensitive data and processing functions from the regular (normal mode) operating system and effectively isolate these processes from any malware or malicious software present in the normal operating system. Then to combat external factors we introduce a method to combines the results from the initial fingerprint for opening the session with data from both the motion sensors and touchscreen to continuously and implicitly verify the user identity inter-session. Because of the monitor, we can safely and reliably trust the secure mode with sensitive data and the processes of user verification. This three-stage verification method: The first stage being the fingerprint, the second which is continuous user verification, and the third, given the correct context, certificates used for signing which is more secure than single-stage verifications that are found in the majority of current session-based implementations. Moreover, if a phone is stolen while not in session, the hacker is no longer able to retrieve certificates and other sensitive data or easily replicate the fingerprint as if a password and no TrustZone was used. However, if inter-session the mobile device is still secure because a transfer will automatically close the session. This session can only be reopened using a fingerprint. This method is further strengthened by the use of a Unauthorized Access Accountability Protections method that will record all instances of unauthorized attempts at accessing a secure session, recording the picture of the imposters as well as the location and time, and emailing this to the appropriate person. This framework allows for protections for pre, during, and post events that may lead to unauthorized accesses. However, there is much room to grow in terms of how secure the process is as a whole with emerging technology in mobile devices as well as advancing technology such as finger vein scanning and the addition of more sensors for more accurate readings. While not impenetrable, this approach strengthens security in todays information-stealing age.

References

[1] Symantec, "Internet security threat report 2013," pp. 1–58, April 2013.

[2] 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*.

[3] 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*.

[4] Apple, "IPhone 5S Tech Specs," 2013. [Online]. Available: https://www.apple.com/iphone-5s/specs/

[5] Samsung, "Galaxy S5 Specs," 2014. [Online]. Available: http://www.samsung.com/global/microsite/galaxys5/specs.html

[6] V. Panchal, "A review on finger print recognition systems," *International Journal of Emerging Technologies in Computational and Applied Sciences*, pp. 1–6, June 2013.

[7] R. J, K. B. Raja, and V. K. R, "Fingerprint Recognition Using Minutia Score Matching," *arXiv.org*, Jan. 2010.

[8] J. Hashimoto, *Finger Vein Authentication Technology and Its Future*. IEEE, Nov. -1.

[9] 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.

[10] T. Feng, X. Zhao, and W. Shi, "Investigating mobile device picking-up motion as a novel biometric modality," in *Biometrics: Theory, Applications and Systems (BTAS), 2013 IEEE Sixth International Conference on*, Sept 2013, pp. 1–6.

[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*, 2011, pp. 141–148.

[14] T. Feng, X. Zhao, B. Carbunar, and W. Shi, "Continuous mobile authentication using virtual key typing biometrics." in *TrustCom/ISPA/IUCC*. IEEE, 2013, pp. 1547–1552. [Online]. Available: http://dblp.uni-trier.de/db/conf/trustcom/trustcom2013.html#FengZCS13

[15] T. Feng, Z. Liu, K.-A. Kwon, W. Shi, B. Carbunar, Y. Jiang, and N. Nguyen, "Continuous mobile authentication using touchscreen gestures," in *Homeland Security (HST), 2012 IEEE Conference on Technologies for*, Nov 2012, pp. 451–456.

[16] T. Feng, J. Yang, Z. Yan, E. M. Tapia, and W. Shi, "Tips: context-aware implicit user identification using touch screen in uncontrolled environments," in *Proceedings of the 15th Workshop on Mobile Computing Systems and Applications*. ACM, 2014, p. 9.

[17] X. Zhao, T. Feng, W. Shi, and I. Kakadiaris, "Mobile user authentication using statistical touch dynamics images," *Information Forensics and Security, IEEE Transactions on*, vol. 9, no. 11, pp. 1780–1789, Nov 2014.

[18] 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.

[19] T. Feng, Z. Liu, B. Carbunar, D. Boumber, and W. Shi, "Continuous remote mobile identity management using biometric integrated touch-display," in *Microarchitecture Workshops (MICROW), 2012 45th Annual IEEE/ACM International Symposium on*, Dec 2012, pp. 55–62.

[20] T. Feng, V. Prakash, and W. Shi, "Touch panel with integrated fingerprint sensors based user identity management," in *Technologies for Homeland Security (HST), 2013 IEEE International Conference on*, Nov 2013, pp. 154–160.

[21] P. Koundinya, S. Theril, T. Feng, V. Prakash, J. Bao, and W. Shi, "Multi resolution touch panel with built-in fingerprint sensing support," in *Design, Automation and Test in Europe Conference and Exhibition (DATE), 2014*, March 2014, pp. 1–6.

[22] P. Koundinya, X. Zhao, T. Feng, S. Theril, and W. Shi, "P-169: Touch-fingerprint display for supporting identity sensing," *SID Symposium Digest of Technical Papers*, vol. 45, no. 1, pp. 1610–1613, 2014. [Online]. Available: http://dx.doi.org/10.1002/j.2168-0159.2014.tb00430.x

[23] 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*, ser. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2009, vol. 5518, pp. 796–799. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-02481-8_120

[24] 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*, ser. IJCAI'05. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2005, pp. 766–772. [Online]. Available: http://dl.acm.org/citation.cfm?id=1642293.1642416

[25] L. Bao and S. Intille, "Activity recognition from user-annotated acceleration data," in *Pervasive Computing*, ser. Lecture Notes in Computer Science, A. Ferscha and F. Mattern, Eds. Springer Berlin Heidelberg, 2004, vol. 3001, pp. 1–17. [Online]. Available: http://dx.doi.org/10.1007/978-3-540-24646-6_1

[26] 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.

[27] 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*, ser. IMCE '09. New York, NY, USA: ACM, 2009, pp. 1–10. [Online]. Available: http://doi.acm.org/10.1145/1631040.1631042