Secure Pick Up: Implicit Authentication When You Start Using the Smartphone

Wei-Han Lee
Princeton University
weihanl@princeton.edu

Xiaochen Liu
University of Southern California
liu851@usc.edu

Yilin Shen
Samsung Research America
yilin.shen@samsung.com

Hongxia Jin
Samsung Research America
hongxia.jin@samsung.com

Ruby B. Lee
Princeton University
rblee@princeton.edu

ABSTRACT

We propose Secure Pick Up (SPU), a convenient, lightweight, in-device, non-intrusive and automatic-learning system for smartphone user authentication. Operating in the background, our system implicitly observes users' phone pick-up movements, the way they bend their arms when they pick up a smartphone to interact with the device, to authenticate the users.

Our SPU outperforms the state-of-the-art implicit authentication mechanisms in three main aspects: 1) SPU automatically learns the user's behavioral pattern without requiring a large amount of training data (especially those of other users) as previous methods did, making it more deployable. Towards this end, we propose a weighted multi-dimensional Dynamic Time Warping (DTW) algorithm to effectively quantify similarities between users' pick-up movements; 2) SPU does not rely on a remote server for providing further computational power, making SPU efficient and usable even without network access; and 3) our system can adaptively update a user's authentication model to accommodate user's behavioral drift over time with negligible overhead.

Through extensive experiments on real world datasets, we demonstrate that SPU can achieve authentication accuracy up to 96.3% with a very low latency of 2.4 milliseconds. It reduces the number of times a user has to do explicit authentication by 32.9%, while effectively defending against various attacks.

KEYWORDS

Authentication; Security; Privacy; Machine Learning; Smartphone; Dynamic Time Warping; Mobile System

1 INTRODUCTION

Mobile devices such as smartphones and tablets are rapidly becoming our means for entering the Internet and online social networks. They also store sensitive and personal information, such as email addresses or bank account information of users. The hardware of today's mobile devices is quite capable with multi-core gigahertz processors, and gigabytes of memory and solid-state storage. Their relatively low cost, ease of use and 'always on' connectivity provide a suitable platform for many day-to-day tasks involving financial transactions and sensitive data, making mobile devices attractive attack targets (e.g., see attacks against the Apple iOS and Google Android platforms in [24]).

Passwords are currently one of the most common forms for user authentication in mobile devices. However, they suffer from several weaknesses. Passwords are vulnerable to guessing attacks [2, 14, 22, 39, 40] or password reuse [7]. The usability issue is also a serious factor, since users do not like to have to enter, and reenter, passwords [32, 35]. A recent study in [5] shows that 64% of users do not use passwords or PINs as an authentication mechanism on their smartphones.

Recently, more and more smartphones are equipped with fingerprint scanners, making authentication through fingerprints quite popular. However, such mechanisms also suffer from several weaknesses. It is possible to trick the scanner by using a gelatin print mold over a real finger. In addition, the response time for the fingerprint scanner to unlock the smartphone is often more than one second [27], degrading the usability of fingerprint-based authentication.

Other biometric-based authentication mechanisms (e.g., via face and keystroke dynamics) are also unreliable and vulnerable to forgery attacks [36, 37]. For instance, an attacker can obtain a photo of the targeted user (e.g., via Facebook) and present it in front of the camera to spoof face recognition on smartphones. Furthermore, these authentication mechanisms require frequent user participation, hindering their deployment in real world scenarios. Hence, it is important to design secure and convenient authentication methods for smartphone users, the topic of this paper.

Behavior-based authentication mechanisms are recently proposed to implement convenient and implicit authentication which does not require frequent user participation and can reduce the user's efforts (e.g., the number of times) needed to unlock their smartphones. Behavior-based authentication is increasingly gaining popularity since mobile devices are often equipped with sensors such as accelerometer, gyroscope, magnetometer, camera, microphone, GPS and so on. Implicit authentication relies on a distinguishable behavioral pattern of the user, which is accomplished by building the users' profiles [4, 6, 9, 10, 15–20, 26, 29, 32, 38, 43]. If a newly-detected user behavior is consistent with the behavior profile stored in the smartphone, the device will have high confidence that no explicit authentication action is required. Otherwise, if the newly-detected behavior deviates significantly from the stored behavior profile, alternative explicit authentication mechanisms should be triggered, such as requiring the user to enter a password, PIN or checking his/her fingerprint.

Existing behavior-based authentication systems exploit machine learning techniques to achieve good security performance [4, 10, 18, 19, 38, 43]. However, these systems have several limitations for real world user authentication: 1) they need a large amount of training data (including other users' data) to learn an authentication classifier, which may violate users' privacy and thus hinder users' motivation to utilize these systems; 2) their training process is usually computationally complicated, which requires additional computational services, e.g., cloud computing, thus requiring users to trust the remote server and always have network connection; 3) their system updating process for capturing the user's behavioral drift over time is also quite complex.

Other behavior-based authentication mechanisms exploit specific contexts of users' behavior, e.g., how do users walk [26], and how do users answer a phone call [6], for authentication. However, their corresponding experiments require users to follow restricted patterns for authentication, e.g., walk straight ahead at the same speed [26] or answer a call when the phone is on a table in front of a user [6]. These constraints are unrealistic for extracting effective behavior patterns of users, making these systems impractical for real world authentication.

To address these issues, we propose a lightweight, in-device, non-intrusive and automatic-learning authentication system, called Secure Pick Up (SPU), which can be broadly deployed in real world mobile devices. Our system aims to utilize a simple and general behavioral pattern of smartphone users, the way people bend their arms when they pick up a phone to interact with the device, to implicitly authenticate the users. For a smartphone that installs our SPU application, the device starts extracting a user's pick-up pattern from his/her arm movements when picking up a phone, and then the system determines whether the current user is legitimate or not. If the user's current behavior conforms to the established behavior profile stored in the smartphone, the user passes the authentication and can have access to the smartphone. If the user's current behavior deviates from the established behavior profile, the device would present explicit authentication challenges, e.g., input of a password, PIN or fingerprint. If these backup explicit authentication mechanisms pass, the user is allowed access to the smartphone and the user's profile stored in the smartphone is updated consequently; otherwise, the user is denied access. This paper aims to answer the question of whether we could build and deploy such a model in a practical, convenient and secure manner on today's mobile devices. Our key contributions include:

- We design a behavior-based implicit authentication system, SPU, by exploiting users' behavioral patterns recorded by smartphone sensors when they bend their arms to pick up a phone. SPU can automatically learn a user's behavioral pattern in an accurate, efficient and stealthy manner. Furthermore, SPU does not require a large amount of training data of other users as previous work did, making our system easier to deploy in real world applications.
- Our system (including the profile updating process) can be implemented efficiently and entirely on personal smartphones. It does not require any additional computational services, e.g., cloud computing. To the best of our knowledge, it is the first using only a device's resources for implicit authentication, making SPU efficient and usable even without network access. For instance, our system can adaptively update the user's authentication model over time with rather low overhead, consuming negligible power of 2%.
- We propose an effective Dynamic Time Warping (DTW) algorithm to quantify similarities between users' pick-up patterns. More specifically, we modify the traditional DTW algorithm and propose a weighted multi-dimensional DTW technique to accommodate the multiple dimensions of sensor data in our setting, and to further improve authentication performance. Extensive experimental results verify the effectiveness of our method which can achieve high accuracy up to 96.3% in 2.4 milliseconds. Furthermore, we demonstrate that SPU can reduce a user's efforts by 32.9% to unlock his/her smartphone providing a more user-friendly experience and encouraging more users to protect access to their devices.
- Finally, our system is robust to various types of attackers, including the serious ones that observe victims' behaviors many times. For instance, our SPU can achieve 0% false acceptance rate (FAR) and 18% false rejection rate (FRR) for authenticating smartphone users under the worst case mimicry attacks (educated attacks).

2 SYSTEM DESIGN

The main objective of our SPU system is to increase the convenience for smartphone users by reducing their efforts (e.g., the number of times) to unlock the smartphone while guaranteeing their security through preventing unauthorized access to the smartphone. We now describe the threat model, design goals, key ideas and system architecture for SPU.

2.1 Threat Model

Compared to personal computers, smartphones are more easily lost or stolen, giving attackers more opportunity to obtain the sensitive data stored in the smartphones. We assume that the attackers have physical access to the smartphone and can even monitor and mimic the user's pick-up behavior. Therefore, they can launch mimicry attacks, to impersonate the legitimate user's behavior. Specifically, we consider three different levels of attacks as follows.

- Random Attack (RA): With no prior knowledge of the user's pick-up behavior, a RA attacker randomly picks up the smartphone and wishes to pass the authentication system. This is equivalent to a brute force attack against text-based password schemes.
- Context-Aware Attack (CAA): In a context-aware attack, an adversary knows the place where the user picks up his/her smartphone, but has not observed how the user does it.
- Educated Attack (EA): In an educated attack, an adversary has observed how and where the user picks up his/her smartphone.

In our SPU system, we consider a single-user model, which is in line with current smartphone usage scenarios. For multi-user models, our system can be generalized in a straightforward manner to incorporate multiple profiles (e.g., family members, guests) for progressive authentication as discussed in [21, 25]. Furthermore, we assume the availability of low-cost sensors in mobile devices for detecting a user's presence and behavior. Indeed, the sensors used in our implementation are the accelerometer and gyroscope, which are widely available in today's mobile devices. As more sensors become pervasive, they can easily be folded into our system.

2.2 Design Goals

Our system is designed to increase the convenience of smartphone users while guaranteeing their security, through implicitly authenticating the users in an unobtrusive manner. Furthermore, the whole authentication process should be implemented stealthily and efficiently. Overall, our design goals for the SPU system are:

- Accurate: the authentication system should not incorrectly authenticate a user.
- Rapid Enrollment and Updating: creating new user accounts or updating pick-up profiles for existing users should be quick.
- Rapid Authentication: the response time for the authentication system must be short, for the system to be usable in reality.
- Implicit: the authentication system should neither interrupt user-smartphone interactions nor need explicit user participation during the authentication process.
- Unobtrusive: the authentication system should be completely unobtrusive and should not invade the user's privacy; the user should be comfortable when using our system.
- Light-weight: the authentication system should not require intensive computations.
- Device only: the authentication system should work efficiently and entirely on mobile devices only even without network access. It should not depend on auxiliary training data of other users or additional computational capabilities, e.g., cloud computing.

2.3 Key Ideas

Our SPU system is designed to achieve all the design goals in Section 2.2. To increase the convenience for users and detect unauthorized access to the smartphone as soon as possible, it is required that we authenticate the users when they start using the smartphone. Therefore, we consider using the users' arm movements when they pick up their smartphones as a distinguishable behavior to authenticate the users. Our key idea stems from the observation that users' behavioral patterns are different from person to person when they start using their smartphones, from the time they pick up the phone to the time they press the *home* button or *power* button. More specifically, we extract the 'pick-up signal' from the user's arm movements measured by sensors (accelerometer and gyroscope) embedded in the smartphone.

To extract users' pick-up movements, we first define a particular user action and call it a 'trigger-action'. Here, we utilize the 'wake up' signal of a smartphone such as pressing the *home* button or *power* button in the sleep mode, as the trigger action [1]. Whenever a trigger-action is performed, we extract the pick-up signal from the measurements of the accelerometer and gyroscope (described below). That is to say, our system authenticates the user only when the smartphone is triggered to wake up from the sleep mode. Note that there is no necessity to authenticate the user when the smartphone is locked.

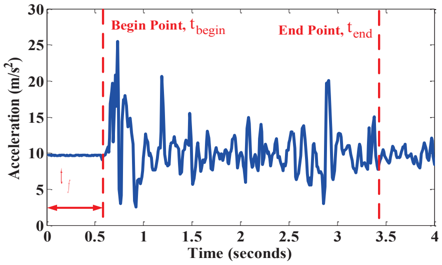


Figure 1: A real world instance of a user's pick-up movement. When a *wake up* signal is detected (*home* button or *power* button is pressed in the sleep mode) corresponding to the end point t_{end}, we backtrack the sensor measurements to find the begin point t_{begin} after detecting a flat signal lasting a period of t_f.

Figure 1 shows a real world instance for the extracted signal stream that describes a user's pick-up movements from measurements collected by the *accelerometer*. When our system detects the *home* button signal or *power* button signal during the *sleep* mode, we record the time as the end of the pick-up signal t_{end}, and back-track the accelerometer measurements to construct the pick-up signal. If we detect a flat signal lasting for a time period of t_f, we consider the end time of the flat signal as the beginning of the pick-up signal t_{begin} as shown in Figure 1.

In order to backtrack the pick-up signal, we need to record the entire time-series measurements of the accelerometer and gyroscope, while the smartphone is in the *sleep* mode. In Section 6, we will show that this sensor measurement process is efficient, only costing an additional 2% in power consumption of the smartphone.

Note that we only consider authenticating pick-up movements from a stable state in our SPU system. We will show in Section 5.2 that this type of pick-up movement (from a stable state) constitutes the most important pick-up characteristic of users.

After extracting the pick-up signal, we propose a weighted multi-dimensional Dynamic Time Warping algorithm to effectively quantify similarities between users' pick up movements for authentication (detailed process will be discussed in Section 4.2.2). More specifically, we modify the traditional DTW algorithm to accommodate the multi-dimensional sensor data in our setting, to further improve authentication performance.

We will show the distinguishable properties of users' pick-up patterns in Section 5. We will show that the pick-up signals are still distinguishable even under impersonation attacks in Section 5.3. Furthermore, our SPU system can significantly reduce users' efforts to unlock their smartphones as will be discussed in Section 5.4.

Unlike previous work, our SPU does not require a large amount of training data for learning a complex authentication classifier,

[1] In our experiments, we used the *home* button or *power* button as the 'trigger-action'. Our method can be easily integrated with new trigger-actions, e.g., the automatic wake-up feature in the iphone 7.

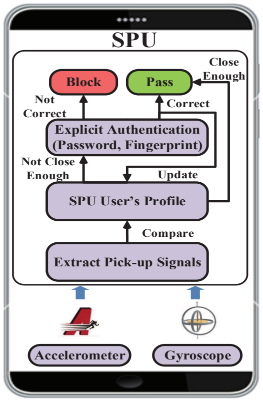


Figure 2: The flowchart of our SPU system.

and any additional computational capability of cloud servers, therefore more users would be motivated to use our system. In addition, our system can be easily combined with the state-of-the-art re-authentication systems [15, 16, 19, 38] to further improve the security of the smartphone.

2.4 System Architecture

Our system is designed for today's smartphones which are equipped with rich sensing capabilities. It could also be generally applied to tablets and other types of wearable devices such as smartwatches. Figure 2 shows the flowchart of our SPU system. System operation is in four phases:

Enrollment: When a user first enrolls in our SPU system, he/she is asked to pick up his/her smartphone in the same way as in his/her normal life. Our system then establishes the user's pick-up profile by extracting the pick-up signal and storing it in the smartphone.

Extracting pick-up signals: Our system keeps monitoring and recording the measurements of the accelerometer and gyroscope when the smartphone is in the sleep mode until it is picked up. We extract the pick-up signals from these sensor measurements in the enrollment phase and afterwards (detailed process discussed in Section 2.3).

Authentication: After extracting the pick-up signal, we compare the new incoming measurements (signal) with the user's pick-up profile stored in the smartphone by utilizing our proposed weighted multi-dimensional DTW technique (will be discussed in Section 4.2.2).

Post-Authentication: If the pick-up signal is authenticated as coming from the legitimate user, this testing passes and the current user can access the information and resources in the smartphone. Otherwise, the smartphone would request an explicit authentication, e.g., password, PIN or fingerprint, from the current user. We emphasize, however, that the desired response to such situations is a matter of policy. Furthermore, the stored user's profile will be updated to accommodate the user's behavioral drift if the correct explicit authentication is provided. Otherwise, no access to the smartphone is allowed.

3 DATA COLLECTION

3.1 Sensor Selection

There are various built-in sensors in today's smartphones, from which we aim to choose a small set of sensors that can accurately represent a user's pick-up behavioral pattern. In this paper, we consider the following two sensors that are commonly embeded in current smartphones: the accelerometer and the gyroscope [11].

These two sensors represent different levels of information about the user's behavior, and are often called a 6-axis motion detector. The accelerometer records larger motion patterns of users such as how they move their arms or walk [26], while the gyroscope records fine-grained motions of users such as how they hold the smartphone [42]. Furthermore, these sensors do not require the user's permission when requested by mobile applications [12], making them useful for background monitoring as in our implicit authentication systems.

3.2 Dataset Collection

We utilize the open-source Android system as our implementation platform. We develop an Android application to implement SPU on Andriod smartphones. Note that our methods are not limited to this platform and can be easily applied to other platforms such as the Apple iOS platform on an iPhone.

In our experiments, each data sample is a time-series measurement collected by the accelerometer and gyroscope, which captures the user's behavioral pattern when picking up the smartphone. In our user study, we consider three experimental scenarios and describe the detailed settings for each experiment as follows. All the participants were shown the app that is installed in their phones. All of the participants volunteered to participate in our experiments. There is no security breach on users' data in smartphones since we collect data and do the authentication attempts offline.

The first experiment was conducted under a lab setting, aiming to provide fundamental intuition for our SPU system. We collected sensor data from 24 users whose detailed demographics are described in Section 5.1. We asked each user to pick up the smartphone in 6 different places while sitting or standing [2]. For each scenario, we collected 10 samples of the pick-up movement for each user, under the 12 situations (6 places × 2 user states). Therefore, we collected $2,880$ (i.e., $24 \times 12 \times 10$) pick-up samples in total. We will describe the detailed analysis for the first experiment in Section 5.1.

The second experiment was conducted under a more realistic setting which is designed to verify the effectiveness of our SPU system in real world applications. The same 24 users were invited to install our application on their own smartphones and use them freely in their normal lives for a week. From the collected data, we extracted $3,115$ pick-up movement samples for these users. We will analyze the overall authentication performance of our system in real world scenarios in Section 5.2.

[2] 2 places are at a user's right hand side, another 2 places are in front of the user, and another 2 places are at a user's left hand side. In each of these three directions, one place is close while the other place is far.

Our third experiment was designed to analyze the security performance of SPU in defending against multiple attacks (e.g., impersonation attacks) as discussed in Section 2.1. In this experiment, we randomly select 6 out of the 24 users as victims and randomly select 12 out of the other 18 users (different from the victims) as adversaries. The experimental setting is the same as the first experiment. The only difference is that the adversaries are trying to mimic the victims under different levels of prior knowledge. Specifically, these adversaries perform the three attacks in Section 2.1 respectively, and the detailed attack processes are described as follows:

- Random Attack (RA): The random attacker tries to use the victim's smartphone without knowing any information about the victim. In total, we collected $12 \times 6 \times 10 = 720$ samples[3] of the pick-up signals under the random attack.
- Context-Aware Attack (CAA): We provided a context-aware attacker who is informed of the place where the victim picked up the smartphone. Note that these attackers have not observed how the victim picked up the smartphone. We also collected 720 pick-up samples under the context-aware attack.
- Educated Attack (EA): The victim user's behavior was recorded by a VCR and is clearly visible to the attacker. The attacker was asked to watch the video and mimic the victim's behavior to the best of his/her ability. In total, we also collected 720 pick-up samples under the educated attack.

We will discuss the security analysis for the third experiment in Section 5.3.

4 SPU AUTHENTICATION ALGORITHMS

We now describe the design of our authentication algorithm which aims to achieve the design goals in Section 2.2.

Previous implicit authentication algorithms exploit machine learning techniques to achieve good authentication performance [4, 10, 18, 19, 26, 38, 43]. However, we identify characteristics that the smartphone implicit authentication exhibits that are not well aligned with the requirements of machine-learning techniques. These include: 1) lack of training data especially those of other users; 2) fundamental limitations in computation capabilities for the training process and the updating process.

To overcome these challenges, we aim to design an implicit, lightweight and in-device authentication algorithm by matching the new incoming pick-up signal with the pick-up profile stored in the smartphone, instead of the complicated machine learning techniques of previous methods. Furthermore, the time duration of a pick-up movement varies across time and across users, and typically is within the range of 0.5 to 4 seconds. Therefore, our matching process should also automatically cope with time deformations and different speeds associated with time-dependent sensor data.

Towards these goals, we consider using the dynamic time warping technique [23] to carefully measure the distance between two time-series sensor data which may vary in time or speed. In DTW, the sequences are warped in a nonlinear fashion to match each other. It has been successfully applied to compare different speech patterns in automatic speech recognition and other applications in the data mining community. Furthermore, we propose an effective weighted multi-dimensional DTW to accommodate our setting where the collected sensor data are of multiple dimensions, thus taking the different distinguishing power of each sensor dimension into consideration.

4.1 Data Pre-processing

Our system keeps monitoring and collecting the measurements of the accelerometer and gyroscope in the background, while the smartphone is in *sleep* mode. When the *wake up* signal (e.g., *home* button or *power* button is pressed in the sleep mode) is detected, our SPU records the time as the ending of the pick-up signal and back-tracks the collected data to find the beginning of the pick-up signal, as described earlier in Section 2.3.

4.2 DTW-based Authentication Algorithm

4.2.1 One-Dimensional DTW. DTW is a well-known technique to find the optimal alignment between two given (time-dependent) sequences $X := (x_1, x_2, \ldots, x_N)$ of length $N \in \mathbb{N}$ and $Y := (y_1, y_2, \ldots, y_M)$ of length $M \in \mathbb{N}$ under certain restrictions. While there is a surfeit of possible distance measures for time-series data, empirical evidence has shown that DTW is exceptionally difficult to beat. Ding et al. in [8] tested the most cited distance measures on 47 different datasets, and no method consistently outperforms DTW. Therefore, in our system, we utilize DTW to measure the distance between users' pick-up signals.

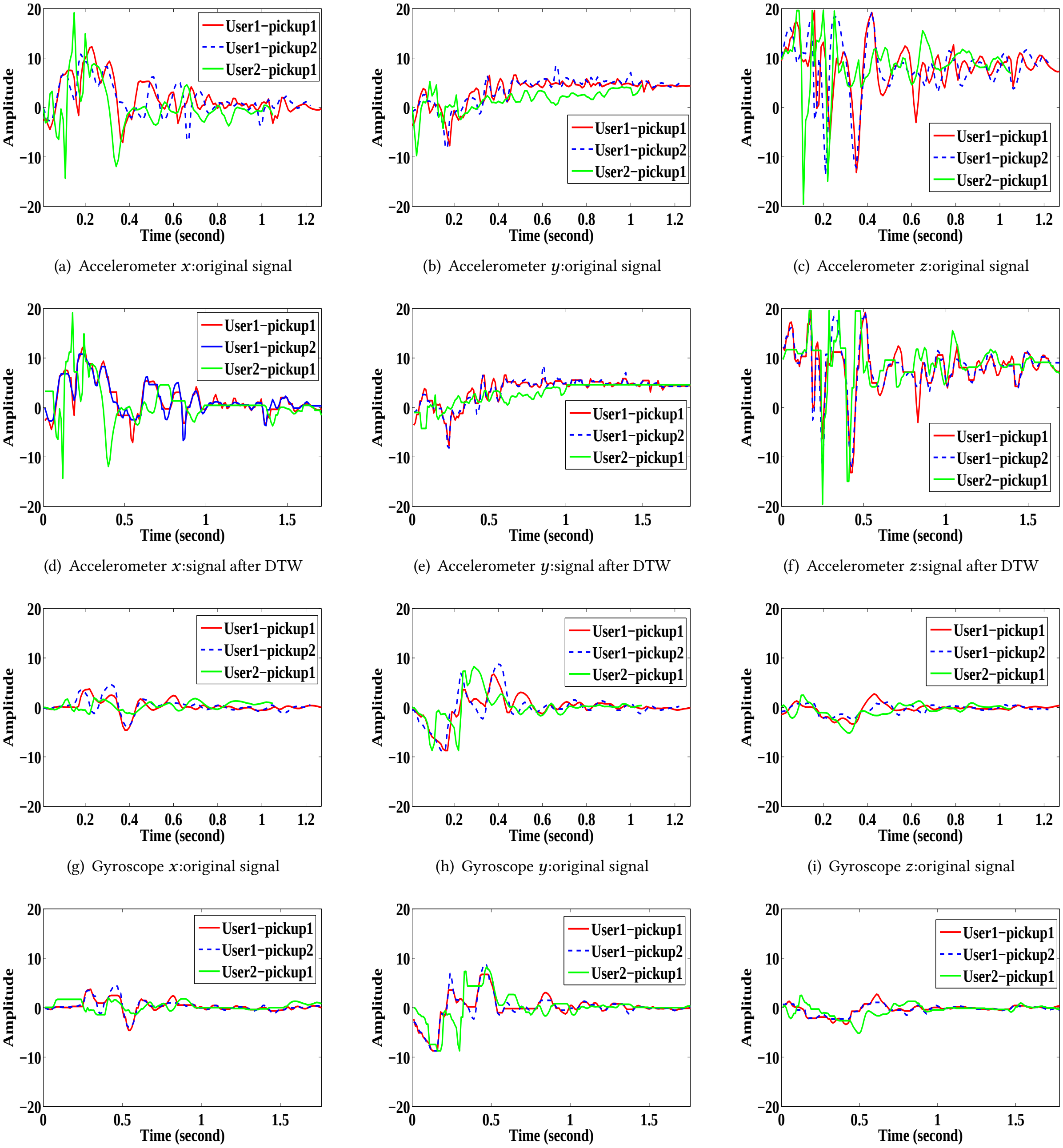
DTW calculates the distance of two sequences using dynamic programming [1]. It constructs an N-by-M matrix, where the (i, j)-th element is the minimum distance (called local distance) between the two sequences that end at points x_i and y_j respectively. An (N, M)-warping path $p = (p_1, p_2, \cdots, p_L)$ is a contiguous set of matrix elements which defines an alignment between two sequences X and Y by aligning the element x_{n_l} of X to the element y_{m_l} of Y. The boundary condition enforces that the first elements of X and Y as well as the last elements of X and Y are aligned to each other. The total distance $d_p(X, Y)$ of a warping path p between X and Y with respect to the local distance measure d is defined as $d_p(X, Y) = \sum_{l=1}^{L} d(x_{n_l}, y_{m_l})$. Therefore, the DTW for one dimensional time-series data can be computed as

$$DTW_1(X, Y) = \min d_p(X, Y) \tag{1}$$

4.2.2 Multi-dimensional DTW. Different from the popular one-dimensional signal (such as speech signal), each pick-up signal in our setting is multi-dimensional (6 dimensions in total including 3 dimensions for accelerometer and 3 dimensions for gyroscope), which is a practical challenge for applying the DTW algorithm to our system. In order to address this challenge, we develop a weighted multi-dimensional DTW by carefully analyzing the distinguishing powers of different sensor dimensions.

Baseline Approach: We first consider an existing approach to process multi-dimensional signals [33] with DTW as the baseline approach. Consider two k-dimensional time-series signals $X := [X_1, X_2, \ldots, X_k]$ and $Y := [Y_1, Y_2, \ldots, Y_k]$, where X_i and Y_i are one

[3] In our experiments, we considered 12 attackers, 6 victims and 10 repeated iterations for each user's pick-up movement.



(a) Accelerometer x:original signal

(b) Accelerometer y:original signal

(c) Accelerometer z:original signal

(d) Accelerometer x:signal after DTW

(e) Accelerometer y:signal after DTW

(f) Accelerometer z:signal after DTW

(g) Gyroscope x:original signal

(h) Gyroscope y:original signal

(i) Gyroscope z:original signal

(j) Gyroscope x:signal after DTW

(k) Gyroscope y:signal after DTW

(l) Gyroscope z:signal after DTW

Figure 3: The visualization of pick-up signals extracted from the accelerometer and gyroscope on three different dimensions. We randomly select two pick-up signals from the same user (red solid and blue dashed dark lines) and a pick-up signal from another user (green light lines). We observe that the distance between two pick-up signals corresponding to the same user is smaller than that from a different user, which lays the foundation for our implicit authentication algorithm. We also observe that different dimensions of sensors may have different powers to distinguish users. For instance, the accelerometer is better than gyroscope in matching the same user's pick-up signals and differentiating different user's patterns, which demonstrates the necessity of our proposed weighted multi-dimensional DTW algorithm.

dimensional time-series signals for each i. Assuming that each dimensional signal is independent of each other, the DTW algorithm under the multiple dimensions setting can be computed as the average over each dimension where

$$DTW_k(X, Y) = \frac{1}{k} \sum_{i=1}^{k} DTW_1(X_i, Y_i) \qquad (2)$$

Weighted Multi-dimensional DTW: However, the above baseline approach considers each dimensional signal as contributing equally to the final matching performance, which is an unrealistic assumption. In real world scenarios as in our settings, different dimensions corresponding to different sensors may have varying degrees of influence on the matching performance, since they reflect different levels of a user's behavioral characteristics. Therefore, we propose our weighted multi-dimensional DTW for discriminating the distinguishing powers of different sensor dimensions as:

$$DTW_k(X, Y) = \sum_{i=1}^{k} w_i DTW_1(X_i, Y_i) \qquad (3)$$

where w_i is the weight for the i-th dimensional signal.

Figure 3 further demonstrates the various distinguishing power for each sensor dimension. We randomly select two pick-up signals corresponding to the same user and one pick-up signal corresponding to another user and compute the distance between these signals after implementing the one-dimensional DTW according to Eq. 1. From Figure 3, we observe that the distance between two pick-up signals corresponding to the same user is much smaller than that from a different user, which lays the basic foundation for our implicit authentication algorithm. We also observe that the accelerometer is more powerful than the gyroscope in matching the same user's pick-up signals and differentiating different users' pick-up signals, which demonstrates the empirical necessity of our proposed weighted DTW algorithm. The reason is that a user's pick-up movement is dominated by the translation which is relevant to the accelerometer, while the rotation relevant to the gyroscope is less significant.

We further analyze the weights for each dimension of accelerometer and gyroscope by varying their weights from 0.1 to 0.9 on the axis of x, y, z with summation equal to one. We observe that when each dimension corresponding to the same sensor is equally weighted, the overall authentication performance is the best (with highest authentication accuracy). In addition, we also vary the weights from 0.1 to 0.9 on the accelerometer and the gyroscope with summation equal to one. We observe that the best performance (highest authentication accuracy) is achieved when the ratio between the weight of the accelerometer and that of the gyroscope is 0.6 to 0.4. Our observations further demonstrate that the accelerometer is more informative than the gyroscope in improving the authentication performance.

In summary, our SPU system realizes implicit, lightweight and in-device authentication for smartphone users, which consists of sensor data collection, pick-up signal extraction and weighted multi-dimensional DTW processing. If the distance (computed by our multi-dimensional DTW) between two time-series signals is close enough (less than a threshold θ), the user passes the authentication and can have access to the smartphone. The detailed process for selecting a proper distance threshold θ will be described in Section 5.2.1.

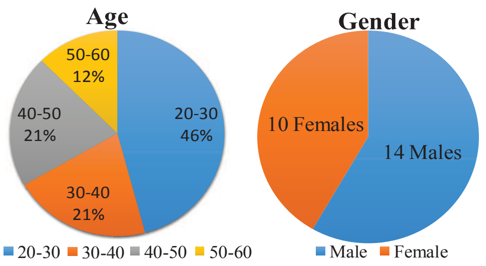


Figure 4: The demographics of users in our experiments.

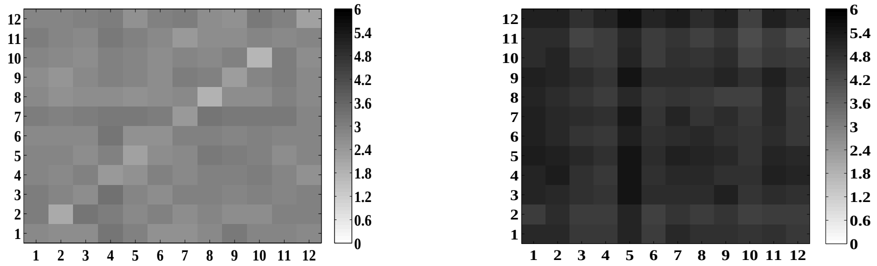
4.3 System Updating

The updating process in previous authentication mechanisms usually involves retraining the authentication classifiers, which is computationally complicated and typically requires additional computing power such as the use of cloud computing. In comparison, we develop an efficient and lightweight updating process to accommodate the user's pick-up behavioral drift over time.

Our system would automatically update the user's profile in the device whenever the user fails the implicit authentication but successfully passes the subsequent explicit authentication. Our updating process is implemented by averaging the currently stored pick-up profile and the newly-detected pick-up signal. The key challenge for this updating process is that the previous profile and the newly-detected instance may not be of the same length. To solve this problem, we utilize our multi-dimensional DTW algorithm to first scale the two signals to the same length and then average them to obtain the updated user's profile for future authentication. We will show the effectiveness of our system updating process in Section 5.2.

5 EXPERIMENTS

To verify the effectiveness of our SPU system, we carefully analyze our collected data (as discussed in Section 3.2) and evaluate the authentication performance of SPU under different experimental scenarios and different system parameters. More specifically, the objectives for our experimental analysis are: 1) to provide empirical confirmation of our system that people's arm movements while they pick up the smartphone can be utilized as a distinguishable behavioral pattern for authentication, as will be discussed in Section 5.1; 2) to investigate the overall authentication performance of SPU under real world usage scenarios, as will be discussed in Section 5.2; 3) to understand the influence of different system parameters on our system, as will be discussed in Section 5.2.1; 4) to verify the effectiveness of our system updating process (recall Section 4.3), as will be discussed in Section 5.2.2; 5) to demonstrate the robustness of our system in defending against various impersonation attacks, as will be discussed in Section 5.3; 6) to verify the necessity of combining the accelerometer and gyroscope in our system, as will be discussed in Section 5.4.



(a)

(b)

Figure 5: The heat map of applying weighted multi-dimensional DTW to our dataset. The average DTW distances between different pick-up signals in 12 contexts is collected for all 24 users from (a) the same user, and (b) different users. We can see that the DTW distances from the same user are much lower than that from different users, thus verifying the fundamental intuitions of our proposed algorithm.

5.1 Fundamental Intuition for Our System

Our first experiment was conducted under a lab setting (as described in Section 3.2), aiming to demonstrate the fundamental intuition and empirical confirmation for our SPU system. In this experiment, we asked each of the 24 users to pick up his/her smartphone in 6 different places while sitting or standing and repeat each movement for 10 iterations. Figure 4 shows the demographics of the 24 users in our experiments. The average age of the participants is 34.3 years old while the median is 31 years old. There are 14 males and 10 females.

After extracting the pick-up signals according to Section 2.3, we measure the distance between any two pick-up instances by exploiting the weighted multi-dimensional DTW technique as described in Section 4.2.2. In our algorithm, the weights for the accelerometer signal and the gyroscope signal are selected as 0.6 and 0.4 respectively, and each of the 3 dimensions of the same sensor is weighted equally (recall analysis in Section 4.2.2).

Figure 5(a) shows the average DTW distances of any two instances of pick-up signals corresponding to the same user. Both the x-axis and y-axis represent the 12 different pick-up scenarios (6 different places and 2 user states, i.e., sitting or standing). Lighter squares represent smaller DTW distances. In Figure 5(a), we observe the smallest DTW distances along the diagonal squares since they represent the distances between two pick-up signals corresponding to the same place and user state. By comparing the diagonal squares and the non-diagonal squares in Figure 5(a), we know the DTW distances across different pick-up scenarios do not vary drastically, demonstrating the robustness of our system under different context scenarios.

Figure 5(b) shows the average DTW distances of any two instances of pick-up signals corresponding to different users. From Figure 5, we observe that the DTW distances between pick-up signals corresponding to the same user are much lower than that between different users, which lays the fundamental intuition for our system that utilizes users' pick-up movements as distinguishable behavioral patterns for authentication.

5.2 Realistic Usage Scenario

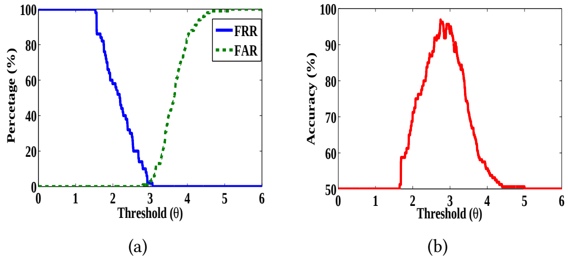
Our second experiment was conducted under a more realistic setting, where the same 24 users (shown in Figure 4) were invited to install our SPU application on their own smartphones and use them freely in their normal lives for a week (7 days)[4].

From the collected data, we extracted $3,115$ pick-up signals according to Section 2.3. That is to say, we can detect 18.54 (i.e., 3115/7/24) pick-up samples for each user per day (with standard deviation 10.54). We also recorded the number of times users unlock their smartphones, which is $8,736$ in a week. Therefore, the average number of times each user unlocks his/her smartphone is 52 (i.e., 8736/7/24) per day (with standard deviation 27.31).

Note that our system does not detect all the movements when the users try to unlock their smartphones, since we only extract pick-up signals starting from a stable state. In our experiment, we can detect 35.6% (i.e., 18.54/52, which correspond to the pick-up signals starting from a stable state) of users' pick-up movements when they try to unlock their smartphones. Therefore, we can save more than one third of the time that users need to unlock their smartphones explicitly. Furthermore, we also compute the DTW distance between other types of pick-up signals (e.g., picking up the smartphone from a bag or from a pocket) to investigate whether there are other pick-up patterns of users that can be utilized for authentication. Our observations show that the distance between other types of pick-up signals (not from a stable state) corresponding to the same user is very large, demonstrating that other types of pick-up signals can not be utilized as distinguishable patterns for user authentication. Therefore, the pick-up movements starting from a stable state which are extracted by our SPU system, constitute the most important pick-up characteristics of users. Our following experimental analysis are implemented on these detected pick-up movement samples.

5.2.1 Determining the Distance Threshold. A significant challenge in implementing our system is how to select a proper value for the distance threshold θ between the newly-detected pick-up

[4] We also let them use our application for another week for evaluating our system updating mechanism as discussed in Section 5.2.2.



(a)

(b)

Figure 6: (a) FAR, FRR and (b) accuracy, varying with different distance threshold θ. We observe that when $\theta = 3.1$, the FRR is 0% and FAR is less than 10%. When $\theta = 2.8$ the FRR is 7.6% and FAR is 0%, resulting in an authentication accuracy higher than 96.3%. Therefore, θ can tradeoff the usability of our system (lower FRR) and users' security (lower FAR)

signal and the stored pick-up profile of the user, which is an important system parameter to balance the trade-off between the usability of our system and the security of smartphone users. A smaller θ provides higher security, while a larger θ would result in better usability.

Here, we utilize false acceptance rate (FAR) and false rejection rate (FRR) as metrics to quantify the authentication performance of our system. FAR is the fraction of other users' data that is misclassified as the legitimate user's. FRR is the fraction of the legitimate user's data that is misclassified as other users' data. For security protection, a large FAR is more harmful to the smartphone users than a large FRR. However, a large FRR would degrade the convenience of using our system. Therefore, we aim to investigate the influence of the distance threshold θ in balancing FAR and FRR, in order to choose a proper θ for our system.

Figure 6(a) shows the FAR and FRR with varying values of the distance threshold θ. We observe that FAR is less than 10% and FRR is 0% when $\theta = 3.1$. The FAR drops to 0% and FRR increases to 7.6% when $\theta = 2.8$. Therefore, θ is a trade-off between the usability of our system (lower FRR) and the security of smartphone users (lower FAR). In Figure 6(b), we observe that the authentication accuracy is higher than 96.3% when θ is around 2.8. Combining Figure 6(a) and Figure 6(b), we choose $\theta = 2.8$ in our experiments from now on and in our published system, aiming at minimizing FAR and maximizing the security of the smartphone users.

5.2.2 Incorporating the System Updating Process. In order to verify the effectiveness of our system updating process as described in Section 4.3, we let the same 24 users use their smartphones freely for another week. More specifically, we randomly divided the users into two groups. The 12 users in the first group installed our SPU application which incorporates the updating process, while the other group installed another version of SPU without the updating process. After careful analysis, we observed that the users in the first group needed to explicitly unlock their smartphones (at the same time, their pick-up profiles would be updated in the SPU system) 17 times per day on average. For the other group without system updating, the users needed to explicitly unlock their smartphones 35 times per day on average. We can see that incorporating the system updating process can further reduce 52% of times for users to unlock the smartphones. These observations show the effectiveness of our system updating process and the advantage of our system in increasing smartphone users' convenience.

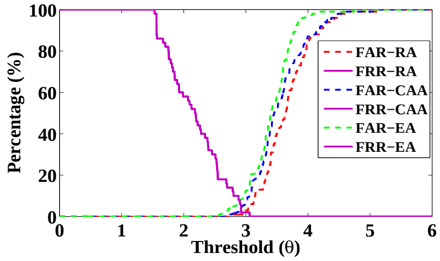


Figure 7: The FAR and FRR of SPU under various impersonation attacks.

5.3 Security Analysis

In our third experimental setting as described in Section 3.2, we aim to evaluate how robust our SPU system is in defending against various types of impersonation attackers (random attack, context-aware attack and educated attack).

For each of the three attacks, we computed FAR and FRR curves under different distance thresholds θ as shown in Figure 7, based on which we have the following observations: 1) SPU can effectively defend against random attacks. Here, 'random' attack indicates a brute force attack where the attacker picks up the smartphone randomly without knowing any information about the victim. 2) When the distance threshold $\theta = 2.5$, the FAR becomes 0% for all the three attacks and the corresponding FRR is 18%. Note that the FRR curve for the three attacks are the same since it evaluates the ratio that the victim is rejected by our system, which is irrelevant to the attacker's capability. 3) Furthermore, the user can defend against different levels of attacks by adjusting the distance threshold θ. These results suggest that our SPU system is more robust against random (brute force) attacks than other types of impersonation attacks (context-aware attacks and educated attacks) since these advanced attackers usually have access to partial information about the user's pick-up movements (recall Section 2.1 and Section 3.2).

In summary, SPU can defend against most realistic attacks robustly and effectively. Even with a strong attacker (i.e., an insider attacker), our system performs gracefully.

5.4 Further Experiments

We further demonstrate the necessity and advantages of combining the common sensors, acclerometer and gyroscope, in our SPU system. In Table 1, we observe that using the combination of accelerometer and gyroscope can achieve better performance than using each sensor individually, with the authentication accuracy up to 96.3%. Furthermore, our SPU can reduce the number of explicit authentications a user must do by 32.9% (i.e., 35.6% × (1 − 7.6%)) on average, where 35.6% is the ratio of detected pick-up signals (recall Section 5.2) and 7.6% is the FRR by using the combination of accelerometer and gyroscope.

Table 1: The authentication accuracy by using accelerometer and gyroscope with distance threshold $\theta = 2.8$.

	Accuracy	FAR	FRR
Accelerometer	90.9 %	6.4%	11.8%
Gyroscope	85.2 %	13.7%	15.2%
Acc+Gyr	96.3 %	0%	7.6%

Table 2: The authentication accuracy by using three motion sensors with distance threshold $\theta = 2.8$.

	Accuracy	FAR	FRR
Magnetometer	36.7%	54.4%	62.4%
Acc+Mag	67.2%	37.2%	48.7%
Gyr+Mag	54.8%	41.9%	57.1%
All three sensors	72.5%	27.6%	34.4%

Next, we went a step further to investigate whether our SPU system could benefit from more sensors than just the accelerometer and gyroscope. More specifically, we analyze the authentication performance of SPU when incorporating the measurements of a magnetometer and its combinations with the accelerometer and gyroscope. We consider the magnetometer since we can construct the popular 9-axis motion detector of the smartphone by combining the 3-axis measurements of magnetometer with the 3-axis measurements of each of accelerometer and gyroscope. An interesting observation shown in Table 2 is that incorporating the magnetometer into our SPU system does not improve the overall authentication accuracy - in fact, it degrades the authentication accuracy! Using more sensors is not always better! The reason is that the magnetic field is rather sensitive to the direction of the smartphone, which makes it vary significantly when the same user picks up the smartphone in different directions - thus degrading the overall authentication performance.

These observations substantiate our choice of using only the accelerometer and gyroscope in our system.

6 OVERHEAD ANALYSIS

We now evaluate the system overhead of SPU on personal smartphones to demonstrate the applicability of our system in real world scenarios. In our source code, the DTW algorithm is implemented in the C language by using the Native Development Kit (NDK) in Android 5.1. We test our system on a Google Nexus5 with 2.3GHz, Krait 400 processor, 16GB internal storage and 2GB RAM, using Android 5.1.

6.1 Power Consumption

There are four different testing scenarios: 1) Phone is locked and SPU is off; 2) Phone is locked and SPU keeps running; 3) Phone is under use and SPU is off; 4) Phone is under use and SPU is running.

For cases 1) and 2), the test time is 12 hours each. We charge the smartphone battery to 100% and check the battery level after 12 hours. The average difference of battery charged level from 100% is reported in Table 3. For cases 3) and 4), *the phone under use* means that the user keeps unlocking and locking the phone. During the unlocked time, the user keeps typing notes. The period of unlocking and locking is two minutes and the test time in total is 60 minutes.

Table 3: The power consumption under four different scenarios.

Scenario	Power Consumption
1) Phone locked, SPU off	1.1%
2) Phone locked, SPU on	2.9%
3) Phone unlocked periodically, SPU off	1.5%
4) Phone unlocked periodically, SPU on	3.5%

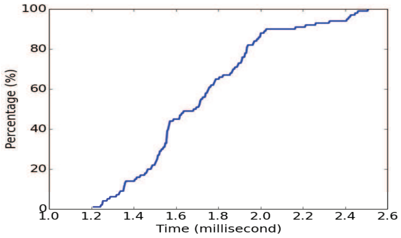


Figure 8: Cumulative distribution function of decision-making time in SPU. We can find that more than 90% of decision-making processes can be completed within 2 milliseconds and all the processes can be finished within 2.4 milliseconds.

Table 3 shows the result of our power consumption test on battery usage. We find that in cases 1) and 2), the SPU-on mode consumes 1.8% more battery power than the SPU-off mode each hour. We believe the extra cost in battery consumption caused by SPU will not affect user experience in daily use. For cases 3) and 4), SPU consumes 2% more battery power performing 30 SPU implicit authentications in one hour, which is also an acceptable cost for daily usage.

6.2 Response Time

Figure 8 shows the cumulative distribution function of decision-making time in SPU authentication. We find that more than 90% of the decision-making computations can be completed within 2 milliseconds and all can be finished within 2.4 milliseconds. This result shows that the latency caused by the SPU system for authentication is low enough to be user-friendly and reasonable for normal usage.

7 RELATED WORK

User authentication is one of the most important issues in smartphone security. Password-based authentication approaches are based on possession of secret information, such as passwords or PINs. Biometric-based approaches make use of distinct personal features, such as fingerprint or iris patterns. Behavior-based authentication identifies a user based on his/her behavioral pattern that is observed by the smartphone. Compared with the password-based and the biometric-based authentication, the behavior-based authentication is more convenient for smartphone users with good resilience to forgery attacks.

7.1 Password-based Authentication

The objective of most password-based authentication mechanisms, e.g., PIN or passwords, is to secure the phone from unwanted access. However, these methods require frequent participation of the user. This often leads to interruptions to the smartphone user, e.g. continuously prompting him/her with some challenges. As a result, many smartphone users tend to completely remove such authentication methods [35]. Our SPU system can overcome these weaknesses, which increases the convenience for smartphone users while guaranteeing their security, as shown in Section 5.

7.2 Biometric-based Authentication

Biometric-based authentications study static physical features of humans. Currently, there are many different physiological biometrics for authentication, such as face patterns, fingerprints [13], and iris patterns [28]. Biometric-based authentication systems involve an enrollment phase and an authentication phase. A user is enrolled by providing his/her biological data such as fingerprint or iris pattern. The system extracts these patterns from the provided data and stores the extracted patterns for future reference. During the authentication phase, the system compares the observed biological data against the stored data to authenticate a user.

However, biometric-based authentications also require frequent user participation, and hence is also an explicit authentication mechanism. For example, fingerprint authentication always requires the user to put his/her finger on the fingerprint scanner. On average, the response time is longer than 1 second [27], which is also much longer than the 2.4 milliseconds of our SPU system. Hence, unlike our implicit SPU authentication, these biometric-based approaches requiring user compliance are not as convenient as our SPU system.

7.3 Behavior-based Authentication

Another thread of authentication research measures the behavioral patterns of the user, where a user is identified based on his/her behavioral patterns, such as hand-writing pattern [10, 38], gait [26] and GPS location patterns [4].

With the increasing development of mobile sensing technology, collecting measurements through sensors built within the smartphone and other devices is now becoming not only possible, but quite easy through, for example, Android sensor APIs. Mobile sensing applications, such as the CMU MobiSens[41], run as services in the background and can constantly collect sensors' data from smartphones. Sensors can be either hard sensors (e.g., accelerometers) that are physically-sensing devices, or soft sensors that record information of a phone's running status (e.g., screen on/off). Therefore, sensor-based implicit authentication mechanisms have become very popular and applicable for behavior-based authentication.

In [4], an *n*-gram geo-based model is proposed for modeling a user's mobility pattern. They use the GPS sensor to detect abnormal activities (e.g., a phone being stolen) by analyzing a user's location history, and their algorithm can achieve 86.6% accuracy. However, the access to GPS require users' permissions, and cannot be done implicitly.

Nickel et al. [26] exploited a user's walking pattern to authenticate a smartphone user by using the *k*-NN algorithm. Conti et al. [6] utilized the user's movement of answering a phone call to authenticate a smartphone user. Shrestha et al.[34] utilized a tapping pattern to authenticate a user when the user does an NFC transaction. However, their experiments had strict restrictions on the users' behavior where the users have to walk or answer a phone call following a specific script (e.g., walk straight ahead at the same speed [26] or answer the phone which is on a table in front of a user [6]). These restrictions are impractical for a real use.

Users' behavior on a touch screen is one of the most popular research directions in behavior-based authentication [3, 10, 19, 31, 38]. Trojahn et al. [38] developed a mixture of a keystroke-based and a handwriting-based method to realize authentication by using the screen sensor. Their approach has achieved 11% FAR and 16% FRR. Frank et al. [10] studied the correlation between 22 analytic features from touchscreen traces and classified these features using *k*-NN and SVM. Li et al. [19] proposed another behavior-based authentication method where they exploited five basic movements (sliding up, down, right, left and tapping) and their related combinations, as the user's behavioral pattern features, to perform authentication. However, touch screen based authentications may suffer from a simple robotic attack [30].

SenSec [43] constantly collects data from the accelerometer, gyroscope and magnetometer, to construct gesture models while the user is using the device. SenSec has shown that it can achieve 75% accuracy in identifying owners and 71.3% accuracy in detecting the adversaries. Lee et al. [18] monitored the users' general behavioral patterns and utilized SVM techniques for user authentication. Their results show that the authentication accuracy can be higher than 90% by using a combination of sensors. However, these methods require a large amount of privacy sensitive training data from other users, and significant external computation power for learning the behavior models, unlike our in-device SPU authentication method.

In fact, almost all the existing behavior-based authentication mechanisms [4, 10, 18, 19, 26, 38, 43] heavily rely on a powerful remote server to share the tasks and take a relatively long time to complete the authentication process. In comparison, our SPU is a lightweight, in-device, non-intrusive and automatic-learning authentication system, which would increase the convenience for smartphone users while enhancing their security.

8 DISCUSSION AND FUTURE WORK

Our SPU system increases the convenience for smartphone users while enhancing their security. We will make SPU open source software, suitable for extensions with future research and experiments.

Future research can include more context-detection techniques to detect fine-grained pick-up patterns for users and embed it with SPU to further increase the convenience and security for smartphone users.

Users' pick-up patterns may vary when they are using other types of devices, e.g., tablets or smartwatches. It would be an interesting future direction to extend SPU to these mobile devices.

Furthermore, the combination of multiple devices may possibly provide better authentication performance for the SPU system.

9 CONCLUSION

We proposed a novel system, Secure Pick Up (SPU), to implicitly authenticate smartphone users in a lightweight, in-device, non-intrusive and automatic-learning manner. Unlike previous work, SPU does not require a large amount of training data (especially those of other users) or any additional computational power from a remote server, which makes it more deployable and desirable for many users.

Our key insight is that the user's phone pick-up pattern is distinguishable from others, using smartphone sensor measurements. We propose a weighted multi-dimensional dynamic time warping algorithm to effectively measure the distance between pick-up signals in order to determine the legitimate user versus others.

Extensive experimental analysis shows that our system achieves authentication accuracy up to 96.3% with negligible system overhead (2% power consumption). Furthermore, our evaluation shows that SPU can reduce by 32.9% the number of explicit authentications a user must do, and can defend against various impersonation attacks effectively. Overall, SPU offers a novel feature in the design of today's smartphone authentication and provides users with more options in balancing the security and convenience of their devices.

REFERENCES

[1] Dimitri P Bertsekas. 1995. *Dynamic programming and optimal control.* Athena Scientific Belmont, MA.

[2] Joseph Bonneau. 2012. The science of guessing: analyzing an anonymized corpus of 70 million passwords. In *IEEE Symposium on Security and Privacy.*

[3] Daniel Buschek, Alexander De Luca, and Florian Alt. 2016. Evaluating the Influence of Targets and Hand Postures on Touch-based Behavioural Biometrics. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems.* ACM, 1349–1361.

[4] Senaka Buthpitiya, Ying Zhang, Anind K Dey, and Martin Griss. 2011. n-gram Geo-trace modeling. In *Pervasive Computing.*

[5] ConsumerReports. 2013. Keep your phone safe: How to protect yourself from wireless threats. *Consumer Reports, Tech.* (2013).

[6] Mauro Conti, Irina Zachia-Zlatea, and Bruno Crispo. 2011. Mind how you answer me!: transparently authenticating the user of a smartphone when answering or placing a call. In *ACM symposium on Information, computer and communications security.*

[7] Anupam Das, Joseph Bonneau, Matthew Caesar, Nikita Borisov, and XiaoFeng Wang. 2014. The tangled web of password reuse. In *Network and Distributed System Security Symposium.*

[8] Hui Ding, Goce Trajcevski, Peter Scheuermann, Xiaoyue Wang, and Eamonn Keogh. 2008. Querying and mining of time series data: experimental comparison of representations and distance measures. *VLDB Endowment* (2008).

[9] Nathan Eagle and Alex Sandy Pentland. 2009. Eigenbehaviors: Identifying structure in routine. *Behavioral Ecology and Sociobiology* (2009).

[10] Michael Frank, Ralf Biedert, En-Di Ma, Ivan Martinovic, and Dong Song. 2013. Touchalytics: On the applicability of touchscreen input as a behavioral biometric for continuous authentication. *Information Forensics and Security* (2013).

[11] Google. N/A. Android sensor manager. http://developer.android.com/guide/topics/sensors/sensors_overview.html. (N/A).

[12] Google. N/A. Android system permission. http://developer.android.com/guide/topics/security/permissions.html. (N/A).

[13] Lin Hong and Anil Jain. 1998. Integrating faces and fingerprints for personal identification. *Pattern Analysis and Machine Intelligence, IEEE Transactions on* (1998).

[14] Patrick Gage Kelley, Saranga Komanduri, Michelle L Mazurek, Richard Shay, Timothy Vidas, Lujo Bauer, Nicolas Christin, Lorrie Faith Cranor, and Julio Lopez. 2012. Guess again (and again and again): Measuring password strength by simulating password-cracking algorithms. In *IEEE Symposium on Security and Privacy.*

[15] Wei-Han Lee and Ruby Lee. 2016. Implicit Sensor-based Authentication of Smartphone Users with Smartwatch. In *International workshop on hardware and architectural support for security and privacy.*

[16] Wei-Han Lee and Ruby Lee. 2017. Implicit Smartphone User Authentication with Sensors and Contextual Machine Learning. In *International Conference on Dependable Systems and Networks.*

[17] Wei-Han Lee and Ruby B Lee. 2015. Implicit Authentication for Smartphone Security. In *Information Systems Security and Privacy.* Springer.

[18] Wei-Han Lee and Ruby B Lee. 2015. Multi-sensor authentication to improve smartphone security. In *International Conference on Information Systems Security and Privacy.*

[19] Lingjun Li, Xinxin Zhao, and Guoliang Xue. 2013. Unobservable Re-authentication for Smartphones. In *Network and Distributed System Security Symposium.*

[20] Chien-Cheng Lin, Deron Liang, Chin-Chun Chang, and Ching-Han Yang. 2012. A new non-intrusive authentication method based on the orientation sensor for smartphone users. In *Software Security and Reliability, IEEE International Conference on.*

[21] Yunxin Liu, Ahmad Rahmati, Yuanhe Huang, Hyukjae Jang, Lin Zhong, Yongguang Zhang, and Shensheng Zhang. 2009. xShare: supporting impromptu sharing of mobile phones. In *International conference on Mobile systems, applications, and services.*

[22] Jiaxin Ma, Weining Yang, Min Luo, and Ninghui Li. 2014. A study of probabilistic password models. In *IEEE Symposium on Security and Privacy.*

[23] Meinard Müller. 2007. Dynamic time warping. *Information retrieval for music and motion* (2007).

[24] Carey Nachenberg. 2011. A window into mobile device security: Examining the security approaches employed in Apple's iOS and Google's Android. *Symantec Security Response* (2011).

[25] Xudong Ni, Zhimin Yang, Xiaole Bai, Adam C Champion, and Dong Xuan. 2009. DiffUser: Differentiated user access control on smartphones. In *Mobile Adhoc and Sensor Systems, International Conference on.*

[26] Claudia Nickel, Tobias Wirtl, and Christoph Busch. 2012. Authentication of smartphone users based on the way they walk using k-NN algorithm. In *Intelligent Information Hiding and Multimedia Signal Processing.*

[27] PhoneArena. N/A. Fingerprint scanners comparison: Galaxy S6 vs iPhone 6 vs Note 4 vs Huawei Mate7 vs Meizu MX4 Pro. http://www.phonearena.com/news/Fingerprint-scanners-comparison-Galaxy-S6-vs\-iPhone-6-v (N/A).

[28] Miao Qi, Yinghua Lu, Jinsong Li, Xiaolu Li, and Jun Kong. 2008. User-specific iris authentication based on feature selection. In *Computer Science and Software Engineering, International Conference on.*

[29] Oriana Riva, Chuan Qin, Karin Strauss, and Dimitrios Lymberopoulos. 2012. Progressive Authentication: Deciding When to Authenticate on Mobile Phones.. In *USENIX Security Symposium.*

[30] Abdul Serwadda and Vir V Phoha. 2013. When kids' toys breach mobile phone security. In *Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security.* ACM.

[31] Michael Sherman, Gradeigh Clark, Yulong Yang, Shridatt Sugrim, Arttu Modig, Janne Lindqvist, Antti Oulasvirta, and Teemu Roos. 2014. User-generated free-form gestures for authentication: Security and memorability. In *Proceedings of the 12th annual international conference on Mobile systems, applications, and services.* ACM.

[32] Elaine Shi, Yuan Niu, Markus Jakobsson, and Richard Chow. 2011. Implicit authentication through learning user behavior. In *Information Security.* Springer.

[33] Mohammad Shokoohi-Yekta, Yanping Chen, Bilson Campana, Bing Hu, Jesin Zakaria, and Eamonn Keogh. 2015. Discovery of meaningful rules in time series. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining.*

[34] Babins Shrestha, Manar Mohamed, Sandeep Tamrakar, and Nitesh Saxena. 2016. Theft-resilient mobile wallets: transparently authenticating NFC users with tapping gesture biometrics. In *Proceedings of the 32nd Annual Conference on Computer Security Applications.* ACM.

[35] Confident Technologies. N/A. Survey Shows Smartphone Users Choose Convenience Over Security. http://confidenttechnologies.com/news_events/survey-shows-smartphone-users-choose\-conven (N/A).

[36] Chee Meng Tey, Payas Gupta, and Debin Gao. 2013. I can be you: Questioning the use of keystroke dynamics as biometrics. Network and Distributed System Security Symposium.

[37] Shari Trewin, Cal Swart, Larry Koved, Jacquelyn Martino, Kapil Singh, and Shay Ben-David. 2012. Biometric authentication on a mobile device: a study of user effort, error and task disruption. In *Annual Computer Security Applications Conference.*

[38] Matthias Trojahn and Frank Ortmeier. 2013. Toward mobile authentication with keystroke dynamics on mobile phones and tablets. In *Advanced Information Networking and Applications Workshops, International Conference on.*

[39] Sebastian Uellenbeck, Markus Dürmuth, Christopher Wolf, and Thorsten Holz. 2013. Quantifying the security of graphical passwords: The case of android unlock patterns. In *ACM conference on Computer and communications security*.

[40] Matt Weir, Sudhir Aggarwal, Breno De Medeiros, and Bill Glodek. 2009. Password cracking using probabilistic context-free grammars. In *IEEE Symposium on Security and Privacy*.

[41] Pang Wu, Jiang Zhu, and Joy Ying Zhang. 2013. Mobisens: A versatile mobile sensing platform for real-world applications. *Mobile Networks and Applications* (2013).

[42] Zhi Xu, Kun Bai, and Sencun Zhu. 2012. Taplogger: Inferring user inputs on smartphone touchscreens using on-board motion sensors. In *ACM conference on Security and Privacy in Wireless and Mobile Networks*.

[43] Jiang Zhu, Pang Wu, Xiao Wang, and Joy Zhang. 2013. Sensec: Mobile security through passive sensing. In *Computing, Networking and Communications, International Conference on*.