Implicit Authentication for Smartphone Security

Wei-Han Lee(✉) and Ruby B. Lee

Princeton Architecture Lab for Multimedia and Security (PALMS),
Department of Electrical Engineering, Princeton University,
Princeton, NJ, USA
{weihanl,rblee}@princeton.edu
http://www.springer.com/lncs

Abstract. Common authentication methods based on passwords, or fingerprints in smartphones, depend on user participation. They do not protect against the threat of an attacker getting hold of the phone after the user has been authenticated. Using a victim's smartphone, the attacker can launch impersonation attacks, which threaten the data that can be accessed from the smartphone and also the security of other users in the network. In this paper, we propose an implicit authentication method using the sensors already built into smartphones. We utilize machine learning algorithms for smartphones to continuously and implicitly authenticate the current user. We compare two typical machine learning methods, SVM and KRR, for authenticating the user. We show that our method achieves high performance (more than 90 % authentication accuracy) and high efficiency. Our method needs less than 10 s to train the model and 20 s to detect an abnormal user. We also show that the combination of more sensors provides better accuracy. Furthermore, our method enables adjusting the security level by changing the sampling rate.

Keywords: Smartphone · Security · Authentication · Support Vector Machine (SVM) · Sensors · Accelerometer · Orientation sensor · Magnetometer · Android

1 Introduction

In recent years, the use of mobile devices like smartphones and tablets has increased dramatically. Smartphones are becoming an important means for accessing various online services, such as online social networks, email and cloud computing. Many applications and websites allow users to store their information, including passwords and other security-critical information. Users also save various contacts, photos, schedules, email, messages and other personal information in their smartphones. They do not want personal and sensitive information to be leaked to others without their permission. However, the smartphone is easily stolen, and the attacker can have access to the personal information stored in the smartphone. Furthermore, the attacker can steal the victim's identity and

O. Camp et al. (Eds.): ICISSP 2015, CCIS 576, pp. 160–176, 2015.
DOI: 10.1007/978-3-319-27668-7_10

launch impersonation attacks in networks, which could threaten the victim's sensitive information like his bank account and confidential data stored in the cloud, as well as the security of networks, especially online social networks. Therefore, providing reliable access control of the information stored on smartphones, or accessible through smartphones, is very important. But, first, it is essential to be able to authenticate the legitimate user of the smartphone, and distinguish him or her from other unauthorized users. It is also important to continue to authenticate a user, since his smartphone may be taken over by an attacker after the legitimate user has been authenticated.

Passwords are currently the most common form of authentication. However, they suffer from several weaknesses. Passwords are vulnerable to attacks because they are easily guessed. They suffer from social engineering attacks, like phishing, pretexting, etc. The usability issue is also a serious factor, since users do not like to have to enter, and reenter, passwords or pins. A study [1] shows that 64 % of users do not use passwords or pins as an authentication mechanism on their smartphones. Hence, this paper proposes a means of implicit and continuous authentication, beyond the initial authentication by password, pin or biometric (e.g., fingerprint).

Implicit authentication does not rely on the direct involvement of the user, but is closely related to his/her biometric behavior, habits or living environment. We propose a form of implicit authentication realized by building the user's profile based on measurements from various sensors already present in a typical smartphone. Specifically, sensor measurements within the smartphones can reflect users' behavior patterns and environment characteristics. The recent development and integration of sensor technologies in smartphones, and advances in modeling user behavior create new opportunities for better smartphone security.

In this paper, we propose a multi-sensor-based system to achieve continuous and implicit authentication for smartphone users. The system leverages data collected by three sensors: accelerometer, orientation sensor, and magnetometer, in a smartphone, and then trains a user's profile using the SVM machine learning technique. The system continuously authenticates the current user without interrupting user-smartphone interactions. The smartphone's security system is alerted once abnormal usage is detected by our implicit authentication mechanism, so that access to sensitive information can be shut down or restricted appropriately, and further checking and remediation actions can be taken. Our authentication mechanism can adaptively update a user's profile every day considering that the user's pattern may change slightly with time. Our experimental results on two different data sets show the effectiveness of our proposed idea. It only takes less than 10 s to train the model everyday and 20 s to detect abnormal usage of the smartphone, while achieving high accuracy (90 %, up to 95 %).

We arrived at our three-sensor solution by first testing the performance on a single-sensor-based system, considering each of the accelerometer, the orientation sensor and the magnetometer. We found that the authentication accuracy for measurements from the orientation sensor alone is worse than that of the accelerometer alone or the magnetometer alone. Then, we test a two-sensor-based system,

using pairwise combinations from these three sensors. This showed that the combination of multiple sensors can improve the accuracy of the resulting authentication. We then combined the measurements from all three sensors, and showed that while there was a slight performance improvement, this incremental improvement is much less than going from one to two sensors, and the authentication accuracy is already 90 %, reaching 95 %. We also show that our method allows the users to adjust their security levels by changing the sampling rate of the collected data. Furthermore, we compare our method with another popular machine learning method, kernel ridge regression (KRR), and show that our proposed method outperforms KRR.
The main contributions of our paper are summarized below:

- We propose a multi-sensor-based system to achieve continuous and implicit authentication, which is accurate, efficient and flexible.
- We compare our three-sensor-based method with single-sensor and twosensor-based methods on two real data sets. Our three-sensor-based method is shown to have the best performance.
- We also analyze the balance between the authentication accuracy and the training time. We give a reasonable trade-off with respect to the sampling rate and the data size, that is practical and meaningful in the real world environment of commodity smartphone users.
- We compare our SVM method with a method based on KRR, and show that our SVM method outperforms the KRR method.

Table 1. Sensors enabled in some popular smartphones.

Sensor	Nexus 5	iphone 5s	Galaxy S5
accelerometer	Yes	Yes	Yes
gyroscope	Yes	Yes	Yes
magnetic field	Yes	Yes	Yes
light	Yes	Yes	Yes
proximity	Yes	Yes	Yes
pressure	Yes	No	Yes
orientation	Yes	No	No
temperature	No	No	No
GPS	Yes	Yes	Yes
Network	Yes	Yes	Yes
MIC	Yes	Yes	Yes
camera	Yes	Yes	Yes

Table 2. Sensor measurements, common usage and whether applications need the user's permission to access measurements.

Sensor	Description	Common use	Permission
accelerometer	Measures the acceleration force on all three physical axes	Motion detection	No
orientation	Measures degrees of rotation on all three physical axes	Rotation detection	No
magnetometer	Measures the geomagnetic field for all three physical axes	compass	No
gyroscope	Measures a device's rate of rotation on all three physical axes	Rotation detection	No
light	Measures the ambient light level	Environment detection	No
proximity	Measures the proximity of an object relative to the view screen	Phone position during a call	No
pressure	Measures the ambient air pressure	Environment detection	No
temperature	Measures the ambient temperature	Environment detection	No
GPS	Positioning	Positioning	Yes
network	Provide user connection to internet	Connectivity, location, surfing patterns	Yes
microphone	Record voice	Speech recognition	Yes
camera	Record image	Face recognition	Yes

2 Background

2.1 Smartphone Inputs and Sensors

A unique feature of a smartphone is that it is equipped with a lot of sensors. Table 1 lists some common sensors in some of the most popular smartphones. Table 2 lists the sensors' functionality, description of the measurements made, what it can be used for in terms of user or smartphone authentication, and whether Android permissions are required to read the sensor's measurements.

Smartphone sensor information include measurements from an accelerometer, orientation sensor, magnetometer, gyroscope, ambient light, proximity sensor, barometric pressure and temperature. Other more privacy sensitive inputs include a user's location as measured by his GPS location, WLAN, cell tower ID and Bluetooth connections. Also privacy sensitive are audio and video inputs like the microphone and camera. These privacy sensitive inputs require Android permissions. The contacts, running apps, apps' network communication patterns, browsing history, screen on/off state, battery status and so on, can also help to characterize a user. Since we would like to perform implicit authentication, we prefer those sensors that do not require explicit Android permissions, and are commonly available on smartphones.

2.2 Related Work

Table 3 summarizes and compares our work with past work on sensor-based authentication.

With the increasing development of mobile sensing technology, collecting many measurements through sensors in smartphones is now becoming not only

Table 3. Comparison of our three-sensor SVM method with state-of-the-art research in implicit authentication (if the information is given in the paper cited, otherwise it is shown as n.a. (not available)). FP is false positive rate and FN is false negative rate. train means the time for training the model and test means the time for detecting the abnormal usage. The script column shows whether a user has to follow a script. If a script is required, we can not achieve implicit authentication without user participation.

	Devices	Sensors	Method	Accuracy	Detecting time	Script
Our method	Nexus 5 Android	orientation, magnetometer, accelerometer	SVM	90.23 %	train:6.07s test:20s	No
Kayacik et al. [2]	Android	light, orientation, magnetometer, accelerometer	temporal &spatial model	n.a.	train: n.a. test:≥122s	No
Zhu et al. [3]	Nexus S	orientation, magnetometer, accelerometer	n-gram language model	71.3 %	n.a.	Yes
Buthpitiya et al. [4]	n.a.	GPS	n-gram model on location	86.6 %	train:n.a. test:≥30 min	No
Trojahn et al. [5]	HTC Desire	screen	keystroke &hand-writing	FP:11 % FN:16 %	n.a.	Yes
Li et al. [6]	Motorola Droid	screen	sliding pattern	95.7 %	train: n.a. test:0.648s	Yes
Nickel et al. [7]	Motorola Milestone	accelerometer	K-NN	FP:4 % FN:22 %	train:1.5 min test:30s	Yes

possible, but quite easy through, for example, Android sensor APIs. Mobile sensing applications, such as the CMU MobiSens [8], run as a service in the background and can constantly collect sensors' information from smartphones. Sensors can be either hard sensors (e.g., accelerometers) that are physicallysensing devices or soft sensors that record information of a phone's running status (e.g., screen on/off).

Continuous authentication on smartphones is likely to become an interesting new research area, given the easily accessible data today in smartphones.

In [2], a lightweight, and temporally &spatially aware user behavior model is proposed for authentication based on both hard and soft sensors. They considered four different attacks (uninformed outsider, uninformed insider, informed outsider and informed insider) and showed that even the informed insider can be detected in 717 s. However, they did not quantitatively show the accuracy of their method. In comparison, our method not only clearly shows high accuracy performance but also requires much less detection time (e.g., we only need 20 s to detect an abnormal user while training the profiles for less than 10 s.)

SenSec [3] constantly collects data from the accelerometer, orientation sensor and magnetometer, to construct the gesture model while the user is using the device. SenSec is shown to achieve an accuracy of 75 % in identifying users and 71.3 % in detecting the non-owners. However, they ask users to follow a script, i.e., a specific series of actions, for authentication. In comparison, we do not need users to follow a specific script while still getting better authentication accuracy, higher than 90 %.

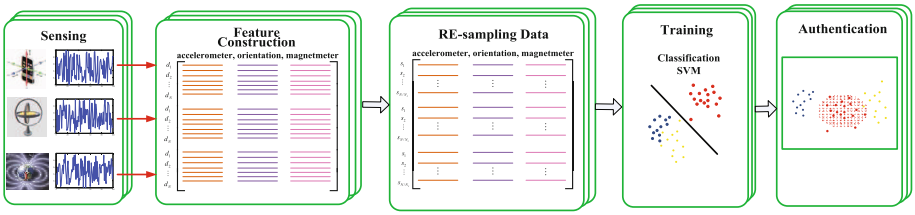


Fig. 1. In our method, we first construct a vector at each sample time by using sensors' data. For example, our three-sensor-based method uses 9 values from the accelerometer, magnetometer and orientation sensor in a smartphone. After that, we re-sample the data collected from the sensors. Then, we train the re-sampled data with the SVM technique to get a user's profile. Based on the user's profile, we can do the implicit authentication.

In [4], an n-gram geo-based model is proposed for modeling a user's mobility pattern. They use the GPS sensor to demonstrate that the system can detect abnormal activities (e.g., a phone being stolen) by analyzing a user's location history, and the accuracy they achieve is 86.6 %. However, they just utilize a single sensor for authentication, which largely limits their performance. By exploiting multiple sensors, our method achieves better accuracy.

Biometric-based systems have also been used to achieve continuous and unobservable authentication for smartphones [5–7]. However, they ask users to follow a script for authentication. In comparison, we do not need users to follow a specific script while still getting good authentication accuracy. [5] developed a mixture of a keystroke-based and a handwriting-based method to realize authentication through the screen sensor. Their approach has 11 % false acceptance rate and 16 % false rejection rate. [6] proposed another biometric method to do authentication for smartphones. They exploited five basic movements (sliding up, down, right, left and tapping) and the related combinations as the user's features, to perform authentication. An accelerometer-based biometric gait recognition to authenticate smartphones using k-NN algorithm was proposed in [7]. Their work is based on the assumption that different people have different walking patterns. Their process only takes 30 s. However, their approach asks the users to follow a script, where they just record the data when the user is walking. In comparison, we do not need the user to follow any script, which means that we can provide continuous protection without user interaction, while their approach can only guarantee security for walking users.

The fact that sensors reflect an individual's behavior and environment can be used for authentication as well as for new attacks. [9] proposed an attack to infer a user's input on a telephone key pad from measurements of the orientation sensor. They used the accelerometer to detect when the user is using a smartphone, and predicted the PIN through the use of orientation sensor measurements.

Sensors also reflect environmental information, which can be used to reveal some sensitive information. By using measurements from an accelerometer on a smartphone to record the vibrations from a nearby keyboard [10], the authors

could decode the context. In [11], the authors show that the gyroscope can record the vibration of acoustic signals, and such information can be used to derive the credit card number.

3 Key Ideas

Some past work only consider one sensor for authentication [4–7]. We will show that the authentication accuracy can be improved by taking other sensors into consideration. We propose a multi-sensor-based technology with a machine learning method for implicit authentication, which only takes a short time to detect the abnormal user, but also needs less than 10 s to retrain the user's profile. First, we collect the data from the selected sensors. Then, we use the SVM technique as the classification algorithm to differentiate the usage patterns of various users and authenticate the user of the smartphone.

Our methodology can be extended to other sensors in a straight-forward manner. Figure 1 shows our methodology, and the key ideas are presented below.

3.1 Sensor Selection

There are a lot of sensors built into smartphones nowadays as shown in Tables 1 and 2. With smartphones becoming more connected with our daily lives, a lot of personal information can be stored in the sensors. The goal is to choose a small set of sensors that can accurately represent a user's characteristics. In this paper, we experiment with three sensors that are commonly found in smartphones: accelerometers, orientation sensors and magnetometers. They also represent different information about the user's behavior and environment: the accelerometer can detect coarse-grained motion of a user like how he walks [7], the orientation sensor can detect fine-grained motion of a user like how he holds a smartphone [9], and the magnetometer measurements can perhaps be useful in representing his environment. Furthermore, these sensors do not need the user's permission to be used in Android applications (Table 2), which is useful for continuous monitoring for implicit authentication.

Also, our method using these three sensors does not need the user to perform a sequence of actions dictated by a script hence facilitating implicit authentication. Note that our method is not limited to these three sensors, but can be easily generalized to different selections of hard or soft sensors, or to incorporate more sensors.

3.2 Data Sets and Re-sampling

We use two data sets, a new one which we collected locally by ourselves which we call the PU data set, and another data set which we obtained from the authors of a published paper [2], which we call the GCU data set.

The PU data set is collected from 4 graduate students in Princeton University in 2014 based on the smartphone, Google Nexus 5 with Android 4.4. It contains

sensor data from the accelerometer, orientation sensor and magnetometer with a sampling rate of 5 Hz. The duration of the data collected is approximately 5 days for each user.

Our pseudo code for implicit data collection in Android smartphones is given in Listing 1. Our application contains two parts. The first part is an Activity, which is a user interface on the screen. The second part is a Service, which is running in the background to collect data. Each sensor measurement consists of three values, so we construct a vector from these nine values from three sensors. We use different sampling rates as a factor in our experiments, to construct data points.

We use the second data set, called the GCU dataset version 2 [2], for comparison. This is collected from 4 users consisting of staff and students of Glasgow Caledonian University. The data was collected in 2014 from Android devices and contains sensor data from wifi networks, cell towers, application use, light and sound levels, acceleration, rotation, magnetic field and device system statistics. The duration of the data collected is approximately 3 weeks. For better comparison with our PU data set, we only use the data collected from the accelerometer, orientation sensor and magnetometer.

Listing 1. Pseudo code for PU dataset collection using Android smartphones.

```
In Activity.java
protected onCreate(Bundle Instance){
 register a BroadcastReceiver;
 set ContentViews and Buttons on the screen;
}
private start_button = new Button.OnClickListener() {
 start Service.java to collect and record data;
}
private stop_button = new Button.OnClickListener() {
 stop Service.java;
}
In Service.java
private onStart(Intent intent, int startId) {
 get Sensor Service ss;
for (Sensor s : sensors) {
 ss register a sensorEventListener s;
}
private sensorEventListener = new SensorEventListener() {
 public onSensorChanged(SensorEvent event) {
  case Sensor.TYPE_ACCELEROMETER:{
    record data with time stamp in memory.
    send data to Activity.java and show on the screen.
  }
  case Sensor.TYPE_ORIENTATION:{
    record data with time stamp in memory.
    send data to Activity.java and show on the screen.
  }
  case Sensor.TYPE_MAGNETIC_FIELD:{
    record data with time stamp in memory.
    send data to Activity.java and show on the screen.
  }
 }
}
```

The sensor measurements originally obtained are too large to process directly. Hence, we use a re-sampling process to not only reduce the computational complexity but also reduce the effect of noise by averaging the data points. For

example, if we want to reduce the data set by 5 times, we average 5 contiguous data points into one data point. In Sect. 4, we will show that the time for training a user's profile can be significantly reduced by re-sampling.

3.3 Support Vector Machines

The classification method used by prior work typical did not give very accurate results. Hence, we propose the use of the SVM technique for better authentication accuracy.

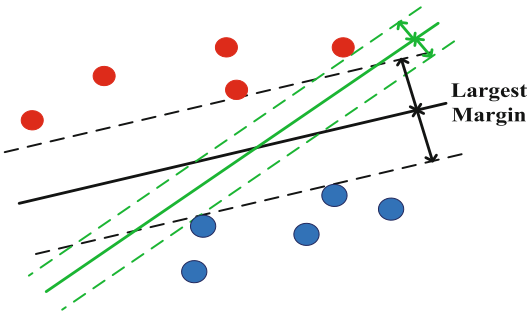


Fig. 2. Illustrating SVM. The purpose of SVM is to find the largest margin separating two groups of data. The black dotted lines represent the largest margin, whereas the green dotted lines do not give the largest margin (Color figure online).

Support Vector Machines (SVMs) are state-of-the-art large margin classifiers, which represent a class of supervised machine learning algorithms first introduced by [12]. SVMs have recently gained popularity for human activity recognition on smartphones [13]. In this section, we provide a brief review of the related theory of SVMs [12,14].

After obtaining the features from sensors, we use SVM as the classification algorithm in the system. The training data is represented as $\mathcal{D} = \{(\boldsymbol{x}_i, \boldsymbol{y}_i) \in \mathcal{X} \times \mathcal{Y} : i = 1, 2, \ldots, n\}$ for n data-label pairs. For binary classification, the data space is $\mathcal{X} = \mathbb{R}^d$ and the label set is $\mathcal{Y} = \{-1, +1\}$. The predictor $\boldsymbol{w}$ is $\mathcal{X} \to \mathcal{Y}$. The objective function is $J(\boldsymbol{w}, \mathcal{D})$. The SVM finds a hyperplane in the training inputs to separate two different data sets such that the margin is maximized. Figure 2 illustrates the concept of SVM classification. A margin is the distance from the hyperplane to a boundary data point. The boundary point is called a support vector and there may exist many support vectors. The most popular method of training such a linear classifier is by solving a regularized convex optimization problem:

$$\boldsymbol{w}^* = \text{argmin}_{\boldsymbol{w} \in \mathbb{R}^d} \frac{\lambda}{2} \|\boldsymbol{w}\|^2 + \frac{1}{n} \sum_{i=1}^{n} l\left(\boldsymbol{w}, \boldsymbol{x}_i, y_i\right) \tag{1}$$

where

$$l(\boldsymbol{w}, x, y) = \max\left(1 - y\boldsymbol{w}^T\boldsymbol{x}, 0\right) \tag{2}$$

The margin is $\frac{2}{||w||}$ in SVM. So, Eq. 1 minimizes the reciprocal of the margin (first part) and the misclassification loss (second part). The loss function in SVM is the Hinge loss (Eq. 2) [15].

Sometimes, we need to map the original data points to a higher dimensional space by using a kernel function so as to make training inputs easier to separate. In our classification, we label the smartphone owner's data as positive and all the other users' data as negative. Then, we exploit such a model to do authentication. Ideally, only the user who is the owner of the smartphone is authenticated, and any other user is not authenticated. In our experiments, we selected LIBSVM [16] to implement the SVM. The input of our experiment is n positive points from the legitimate user and n negative data points from randomly selected n other users. The output is the user's profile for the legitimate user.

3.4 Kernel Ridge Regression

For comparison, we utilize another popular classification method, kernel ridge regression (KRR) [17], to train the user's model. The KRR is a regularized least square method for classification and regression. It is similar to an SVM, except that a different objective is being optimized, which does not put emphasis on points close to the decision boundary. The solution depends on all the training examples instead of a subset of support vectors. The classifier is obtained by solving an optimization problem:

$$\boldsymbol{w}^* = \text{argmin}_{\boldsymbol{w}\in\mathbb{R}^d}\rho\|\boldsymbol{w}\|^2 + \frac{1}{n}\sum_{i=1}^{n}\left(\boldsymbol{w}^T\boldsymbol{x} - y\right)^2 \tag{3}$$

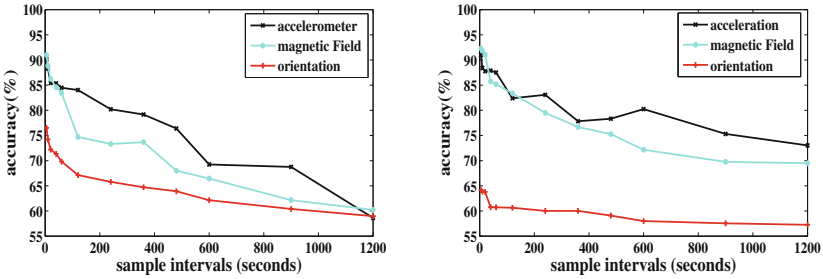
An advantage of kernel ridge regression is that the optimization solution has an analytic solution, which can be solved efficiently. The solution of KRR is as follows [18]:

$$\boldsymbol{w}^* = \left[\boldsymbol{S} + \rho\boldsymbol{I}\right]^{-1}\boldsymbol{X}\boldsymbol{y} \tag{4}$$

4 Experimental Results

Figure 1 shows the steps in our experiments. The following are some settings in our experiments:

- We use both the PU data set and the GCU data set.
- We use accelerometer, magnetometer and orientation sensors (can be extended to other sensors).
- We re-sample the data by averaging the original data, with the sampling rate changing from 1 s to 20 min.
- Each data is a 9-dimensional vector (three values for each sensor). We use SVM to train the data to obtain a user's profile.



(a) PU data set (b) GCU data set

Fig. 3. Authentication accuracy for single sensor system in (a) the PU data set, and (b) the GCU data set. Higher sampling rates give better accuracy for each sensor. The accelerometer and magnetometer have better performance than the orientation sensor. The reason is that both of them record a user's longer term characteristics, where the accelerometer somehow represents a user's walking style and the magnetometer records a user's general environment. However, the orientation sensor represents how the user holds a smartphone, which is more variable.

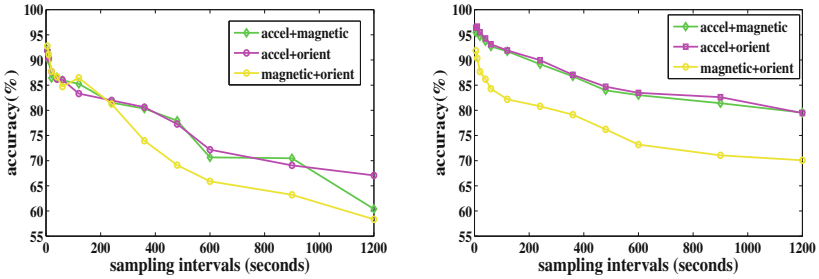
- We label one user's data as positive and the other users' data as negative, and randomly pick equivalent data from both positive and negative sets.
- We experiment with data from one sensor, a pair of two sensors, and all three sensors to train the user's profile. We show that multi-sensor-based authentication indeed improves the authentication accuracy.
- In our experiments, we use 10-fold cross validation, which means that the size of training data over the size of training data and testing data is 1/10.

4.1 Single-Sensor Authentication

From Fig. 3, we observe the single-sensor-based system in both the PU data set and the GCU data set. First, we find that the accuracy increases with faster sampling rate because we use more detailed information from each sensor. Second, an interesting finding is that the accelerometer and the magnetometer have much better accuracy performance than the orientation sensor, especially for the GCU data set. We think this is because they both represent a user's longer-term patterns of movement (as measured by the accelerometer) and his general environment (as measured by the magnetometer). The orientation sensor represents how the user holds a smartphone [9], which may be more variable. Therefore, the accelerometer and magnetometer have better authentication accuracy. The difference is more marked in the GCU data set, but the overall relative accuracy of the three sensors is the same in both data sets. The accuracy is below 90 % even for fast sampling rates like 10 s (see also Table 4).

4.2 Two-Sensor Authentication

Figure 4 shows that for all pairwise combinations, accuracy increases with faster sampling rate. The combination of data from two sensors indeed gives better



(a) PU data set (b) GCU data set

Fig. 4. Authentication accuracy with SVM for a combination of two sensors, for (a) the PU data set, and (b) the GCU data set. The higher sampling rate gives better accuracy for each sensor.

authentication accuracy than using a single sensor (see Table 4). The average improvement from one sensor to two sensors is 7.4 % in PU data set (14.6 % in GCU data set) when the sampling rate is 20 s. Another interesting finding is that using a combination of magnetometer and orientation sensors is worse than the other two pairs which include an accelerometer. In fact, the combination of magnetometer and orientation sensors is not necessarily better than using just the accelerometer (see also Table 4). Therefore, choosing good sensors is very important. Also, using higher sampling rate gives better accuracy.

4.3 Three-Sensor Authentication

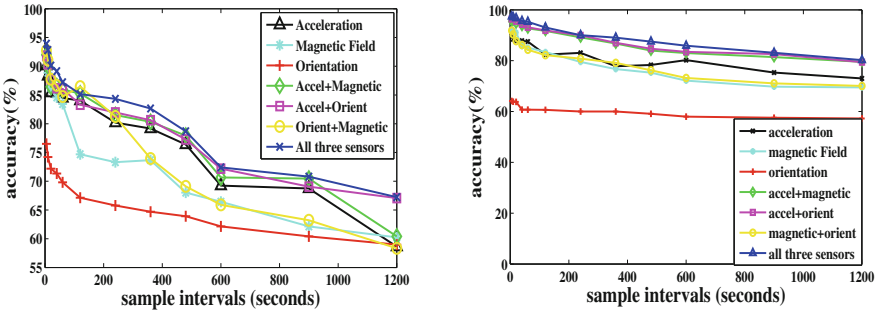
Now, we compare the three-sensor-based system with one and two sensor-based authentication experiments. From Fig. 5 and Table 4, we observe that the three-sensor results give the best authentication accuracy, as represented by the top line with triangles in both data sets, seen more clearly as the highest value in each column in Table 4. Again, we find that the accuracy increases with faster sampling rates because we use more detailed information from each sensor.

4.4 Training Time vs. Sampling Rate

In the rest of the evaluations below, we use the three-sensor-based system, since it has the best authentication accuracy.

From Fig. 5 and Table 4, when the sampling rate is higher than 4 min (samples every 240 s or less), the accuracy in the PU data set is better than 80 %, while that in the GCU data set is better than 90 %. The average improvement from two sensors to three sensors is 3.3 % in PU data set (4.4 % in GCU data set) when the sampling rate is 20 s. Furthermore, when the sampling rate is higher than 20 s, the accuracy in the PU data set is better than 90 %, while that in the GCU data set is better than 95 %.

Figure 6 and Table 5 shows that a higher sampling rate (smaller sampling interval) needs more time to train a user's profile. The time exponentially

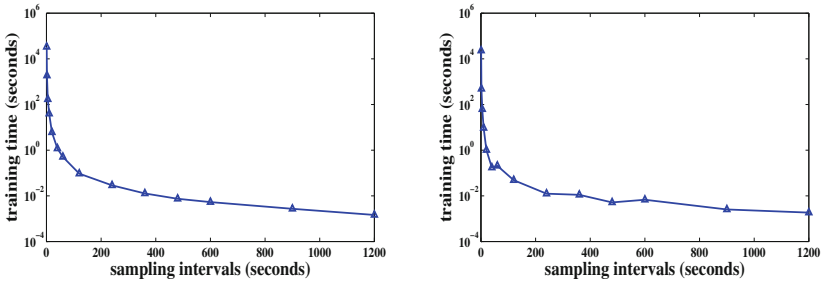


(a) PU data set (b) GCU data set

Fig. 5. Authentication accuracy for single sensors, two sensors and three sensors, for the PU data set and the GCU data set. The higher sampling rate has better accuracy for each combination of sensors. Two sensors give better accuracy than using a single sensor, and three sensors further improves the accuracy.

increases with the increase of the sampling rate. It is a trade-off between security and convenience. However, the good news is that when the sampling interval is about 20 s, it only needs less than 10 s in the PU data set (and roughly 1 s in the GCU data set) to train a user's profile, but the accuracy is higher than 90 % (and 95 % in the GCU data set), as seen from Table 4. It means that a user only needs to spend less than 10 s to train a new model to do the implicit authentication for the whole day in the PU data set and only 1 second for the GCU data set.

These findings validate the effectiveness of our method and its feasibility for real-world applications. Furthermore, our method can be customized for users. They can change their security level by changing the sampling rate of the sensors in their smartphones.



(a) PU data set (b) GCU data set

Fig. 6. (a),(b) Represent respectively the time for training a user's profile by using the SVM algorithm for three-sensors-based system in the PU data set and the GCU data set.

Table 4. The accuracy (%) vs. sampling rate in both PU data set and GCU data set for all combinations of 1, 2 or 3 sensors.

Sampling rate (s)	5	10	20	40	60	120	240	360	480	600	900	1200
acc(PU)	90.1	88.3	85.4	85.3	84.5	84.0	80.2	79.2	76.4	69.2	68.8	58.6
mag(PU)	91.0	88.9	86.2	84.6	83.4	74.7	73.3	73.7	68.0	66.4	62.2	60.2
ori(PU)	76.5	74.2	72.2	71.3	69.8	67.1	65.8	64.7	63.9	62.1	60.4	59.0
acc+mag(PU)	92.0	90.0	86.4	86.6	85.9	85.3	81.5	80.3	77.9	70.6	70.5	60.4
acc+ori(PU)	91.8	90.3	87.7	86.2	86.1	83.3	82.0	80.6	77.3	72.2	69.1	67.1
mag+ori(PU)	92.8	91.1	87.7	86.7	84.7	86.5	81.3	74.0	69.1	65.9	63.2	58.3
all(PU)	93.9	92.8	90.1	89.1	87.2	85.2	84.3	82.7	78.7	72.4	70.8	67.2
acc(GCU)	91.0	88.4	87.8	87.9	87.5	82.4	83.1	77.8	78.3	80.2	75.3	73.0
mag(GCU)	92.3	91.2	91.0	85.7	85.2	83.4	79.5	76.7	75.3	72.2	69.8	69.5
ori(GCU)	64.2	63.9	63.8	60.8	60.7	60.6	60.0	60.0	59.1	58.0	57.5	57.3
acc+mag(GCU)	95.5	95.8	94.7	93.7	92.7	91.8	89.2	86.7	84.0	83.1	81.4	79.6
acc+ori(GCU)	96.4	96.6	95.5	94.3	93.1	92.0	90.0	87.1	84.7	83.5	82.7	79.4
mag+ori(GCU)	91.8	90.3	87.7	86.2	84.3	82.2	80.8	79.1	76.2	73.2	71.1	70.1
all(GCU)	97.4	97.1	96.7	95.7	95.3	93.1	90.0	89.1	87.5	85.9	83.1	80.2

Table 5. Time for training a user's profile by using the SVM algorithm for three sensors, for (a) the PU data set and (b) the GCU data set, respectively. We can see that the smaller sample interval (higher sampling rate) needs more time to train a user's profile. Therefore, we need to find a trade-off sampling rate to balance performance and complexity.

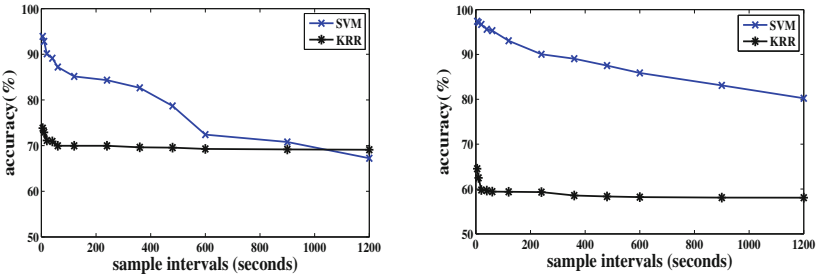
Sampling interval	1	2	5	10	20	40	60
training time (PU data set)	33502s	1855s	170.72s	39.85s	6.07s	1.19s	0.51s
training time (GCU Data Set)	23101s	485s	62.41s	9.43s	1.02s	0.21s	0.17s

4.5 Accuracy and Time vs. Data size

Figure 8 shows another trade-off between security and convenience. We choose a sampling interval of 10 min and a training data size ranging from 1 day to 5 days in the PU data set (and 1 day to 15 days in the GCU data set). The blue dashed line with triangles shows that the accuracy increases with the increase of training data size. The black solid line with circles shows that the training time increases with the increase of training data size.

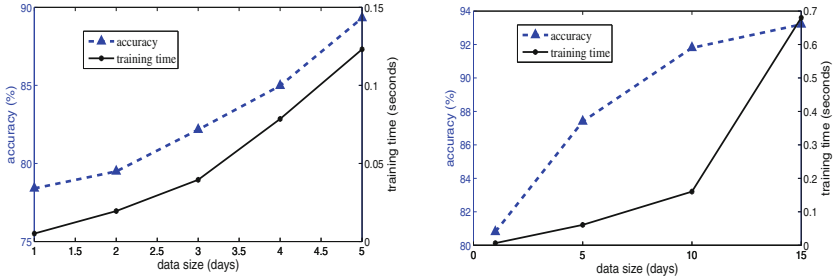
4.6 Comparison with KRR

In order to compare our SVM performance with other machine learning methods, we apply another popular machine learning method, kernel ridge regression (KRR) to train the user's model. Figure 8 compares the performance of SVM



(a) PU data set (b) GCU data set

Fig. 7. Comparison of authentication accuracy between SVM and KRR for a combination of three sensors, for (a) the PU data set, and (b) the GCU data set. Using SVM has much better performance than KRR.



(a) PU data set (b) GCU data set

Fig. 8. (a),(b) represent the authentication accuracy and the training time with different training data size for three sensors in the PU data set and the GCU data set. Blue dashed lines show that the larger data size has better accuracy because we use more information about the user. Black solid lines show that larger data size usually needs longer training time (Color figure online).

and KRR by using all three sensors. Figure 8 shows that using SVM gives much better authentication performance than using KRR.

5 Conclusions

In this paper, we utilize three sensors: the accelerometer, the orientation sensor and the magnetometer, which are all commonly built into smartphones today. We apply the SVM technique as the classification algorithm in the system, to distinguish the smartphone's owner versus other users, who may potentially be attackers or thieves. In our experiments, we compare the authentication results for different sampling rates and different data sizes, which shows a trade-off between accuracy performance and the computational complexity. Furthermore, we experiment with data from a single sensor and from a combination of two sensors, to compare their results with data from all three sensors. We find that

the authentication accuracy for the orientation sensor degrades more than that of the other two sensors. Therefore, the data collected from the orientation sensor is not as important as that from the accelerometer and magnetometer, which tend to measure more stable, longer-term characteristics of the user's coarse-grained movements and his general physical location, respectively (Fig. 7).

We also compared using KRR versus using our SVM method, and found that SVM gave much better authentication accuracy.

Utilizing sensors to do implicit user authentication is very interesting and promising. Our work also suggests some other interesting research directions. First, we can use more detailed sensors' information to further improve the authentication accuracy. Second, we can try to combine the time information with frequency information to potentially achieve a better user profile. Many other issues relating to the user's privacy remain. It is also interesting to launch an attack through the sensors' information. Since our research shows that indeed, sensors can represent a user's characteristic behavior and physical environment, sensors can be used for both new security defenses, e.g., implicit authentication, and for new attacks. By understanding these potential attacks, we may be able to design more secure sensor systems to further improve smartphone security.

Acknowledgements. This work was supported in part by the National Science Foundation under grant NSF CNS-1218817. Any opinions, findings, and conclusions or recommendations expressed in this work are those of the authors and do not necessarily reflect the views of NSF.

References

1. ConsumerReports, Keep your phone safe: How to protect yourself from wireless threats, Consumer Reports, Technical (2013)
2. Kayacık, H.G., Just, M., Baillie, L., Aspinall, D., Micallef, N.: Data driven authentication: on the effectiveness of user behaviour modelling with mobile device sensors. In: Mobile Security Technologies (2014)
3. Zhu, J., Wu, P., Wang, X., Zhang, J.: Sensec: mobile security through passive sensing. In: International Conference on Computing, Networking and Communications (2013)
4. Buthpitiya, S., Zhang, Y., Dey, A.K., Griss, M.: n-gram geo-trace modeling. In: Pervasive Computing (2011)
5. Trojahn, M., Ortmeier, F.: Toward mobile authentication with keystroke dynamics on mobile hones and tablets. In: 2013 27th International Conference on Advanced Information Networking and Applications Workshops (WAINA) (2013)
6. Li, L., Zhao, X., Xue, G.: Unobservable re-authentication for smartphones. In: Network and Distributed System Security Symposium (2013)
7. Nickel, C., Wirtl, T., Busch, C.: Authentication of smartphone users based on the way they walk using k-nn algorithm. In: 2012 Eighth International Conference on Intelligent Information Hiding and Multimedia Signal Processing (IIH-MSP) (2012)
8. Wu, P., Zhu, J., Zhang, J.Y.: Mobisens: a versatile mobile sensing platform for real-world applications. Mob. Netw. Appl. **18**(1), 60–80 (2013)

9. Xu, Z., Bai, K., Zhu, S.: Taplogger: inferring user inputs on smartphone touchscreens using on-board motion sensors. In: Proceedings of the Fifth ACM Conference on Security and Privacy in Wireless and Mobile Networks (2012)
10. Marquardt, P., Verma, A., Carter, H., Traynor, P.: (sp) iphone: decoding vibrations from nearby keyboards using mobile phone accelerometers. In: ACM Conference on Computer and Communications Security (2011)
11. Michalevsky, Y., Boneth, D., Nakibly, G.: Gyrophone: recognizing speech from gyroscope signals. In: USENIX Security (2014)
12. Vapnik, V.N., Vapnik, V.: Statistical Learning Theory, vol. 2. Wiley, New York (1998)
13. Anguita, D., Ghio, A., Oneto, L., Parra, X., Reyes-Ortiz, J.L.: Human activity recognition on smartphones using a multiclass hardware-friendly support vector machine. In: Bravo, J., Hervás, R., Rodríguez, M. (eds.) IWAAL 2012. LNCS, vol. 7657, pp. 216–223. Springer, Heidelberg (2012)
14. Cristianini, N., Shawe-Taylor, J.: An Introduction to Support Vector Machines and Other Kernel-Based Learning Methods. Cambridge University Press, Cambridge (2000)
15. Gentile, C., Warmuth, M.K.: Linear hinge loss and average margin. In: Conference and Workshop on Neural Information Processing Systems, vol. 11, pp. 225–231 (1998)
16. Chang, C.-C., Lin, C.-J.: LIBSVM: a library for support vector machines. ACM Trans. Intell. Syst. Technol. **2**, 27:1–27:27 (2011)
17. Hastie, T., Tibshirani, R., Friedman, J., Hastie, T., Friedman, J., Tibshirani, R.: The elements of statistical learning, vol. 2(1). Springer, New York (2009)
18. Hoerl, A.E., Kennard, R.W.: Ridge regression: biased estimation for nonorthogonal problems. Technometrics **12**(1), 55–67 (1970)