Inferring Smartphone Users' Handwritten Patterns by using Motion Sensors

Wei-Han Lee[1], Jorge Ortiz[2], Bongjun Ko[2] and Ruby Lee[1]
[1]*Princeton University, U.S.A.*
[2]*IBM Research, U.S.A.*

Keywords: Smartphone Sensors, Handwritten Pattern, Dynamic Timing Warping, Majority Voting.

Abstract: Mobile devices including smartphones and wearable devices are increasingly gaining popularity as platforms for collecting and sharing sensor data, such as the accelerometer, gyroscope, and rotation sensor. These sensors are used to improve the convenience of smartphone users, e.g., supporting the mobile UI motion-based commands. Although these motion sensors do not require users' permissions, they still bring potential risks of leaking users' private information reflected by the changes of sensor readings. In this paper, we investigate the feasibility of inferring a user's handwritten pattern on a smartphone touchscreen by using the embedded motion sensors. Specifically, our inference attack is composed of two key steps where we 1) first exploit the dynamic time warping (DTW) technique to differentiate any pair of time-series sensor recordings corresponding to different handwritten patterns; and 2) develop a novel sensor fusion mechanism to integrate information contained in multiple motion sensors by exploiting the majority voting strategy. Through extensive experiments using real-world data sets, we demonstrate the effectiveness of our proposed attack which can achieve 91.4% accuracy for inferring smartphone users' handwritten patterns.

1 INTRODUCTION

The ubiquity of mobile devices such as smartphones and wearable devices together with their ever-growing computing, networking, and sensing powers have been increasingly changing the landscape of our daily lives. These devices are often equipped with various embedded sensors including the Global Positioning System (GPS) sensor, camera, microphone, the environmental sensor (e.g., the ambient light sensor and the barometer), and the motion sensors (e.g., the accelerometer, gyroscope, rotation sensor). These sensors have been effectively utilized to improve the convenience of smartphone users. For instance, the GPS can be used for positioning and the motion sensors can be leveraged for mobile gaming.

Within these built-in sensors, some require users' permission to obtain access (such as the GPS, camera, microphone) because these sensors are explicitly utilized for collecting users' private information (such as location, image and speech). In comparison, motion sensors such as the accelerometer, gyroscope and rotation sensor do not require users' permissions, probably due to the assumption that data collected by these sensors is not sensitive. Motion sensors can provide recordings of acceleration, rotation and direction with high precision and accuracy, which can enable applications that provide convenient functions for the users. For example, a game can utilize the gravity sensor data of a smartphone to infer users' different gestures, such as tilt, shake, rotation, or swing (**?**). All the measurements of the accelerometer, gyroscope, and rotation sensor in smartphones running on the iOS system and the Android system can be accessed without requiring any user permission (Xu et al., 2012).

However, even motion sensors that do not require explicit permissions are still vulnerable to privacy attacks since their measurements are closely correlated with users' sensitive behavior patterns. With the increasing development of motion sensors in smartphones, the risks of leaking a user's sensitive information through an installed third-party application exploring motion sensors have raised more privacy and security concerns. For instance, Marquardt et al. (Marquardt et al., 2011) utilize the vibrations detected by smartphone accelerometer to infer the user' inputs to a nearby keyboard. Michalevsky et al. (Michalevsky et al., 2014) show that existing gyroscopes on smartphones are sufficiently sensitive to measure acoustic signals in the vicinity of

Lee, W-H., Ortiz, J., Ko, B. and Lee, R.
Inferring Smartphone Users' Handwritten Patterns by using Motion Sensors.
DOI: 10.5220/0006650301390148
In *Proceedings of the 4th International Conference on Information Systems Security and Privacy (ICISSP 2018)*, pages 139-148
ISBN: 978-989-758-282-0

the smartphone to identify the speaker' private information and even parse speech. These security and privacy breaches demonstrate that the motion sensors are possible side channels for attackers that aim to infer users' sensitive behavior patterns (Xu et al., 2012; Marquardt et al., 2011; Michalevsky et al., 2014).

In this work, we aim to infer users' handwritten patterns by exploiting the motion sensors embedded in the smartphones, which has not been investigated in the literature to the best of our knowledge. The fundamental intuition of our attack is that the users' handwritten behaviors and the changes of the smartphone motions are closely correlated with each other. During a writing event, the force from the users' finger on the touchscreen would cause changes of the motion sensor measurements, which would follow certain patterns corresponding to different contexts of handwritten behaviors. By utilizing the changes of motion sensor readings, the attacker can therefore infer users' handwritten patterns on the smartphone touchscreen.

More specifically, our proposed attack is composed of two key steps where we 1) first exploit the dynamic time warping (DTW) (Berndt and Clifford, 1994) technique to evaluate the similarity between any pair of time-series sensor recordings, in order to differentiate users' various handwritten patterns. This DTW-based similarity evaluation technique can be utilized for constructing template sensor signals corresponding to different handwritten patterns (under the training mode), as well as inferring the incoming handwritten event by matching the observed sensor measurement with the constructed template sensor signals (under the testing mode); and 2) develop a novel sensor fusion mechanism to integrate information contained in multiple motion sensors by exploring the majority voting strategy (Lam and Suen, 1997). In summary, our key contributions include:

- We investigate the unique patterns of users' handwritten events in terms of measurement changes of motion sensors embedded in the smartphones. Our observations raise the awareness of motion as a significant channel that may leak smartphone users' handwritten patterns.
- We exploit the dynamic time warping technique to measure the distance between any pair of time-series sensor recordings, in order to distinguish users' handwritten patterns on the smartphone touchscreen.
- We further propose a novel sensor fusion mechanism by leveraging the majority voting strategy to integrate information recorded by multiple sensors, in order to enhance the overall accuracy of inferring users' handwritten patterns.
- We present the design of our attack that utilizes observed motion sensor readings to stealthily record the user's inputs on the touchscreen. Extensive experiments on real-world data sets demonstrate the effectiveness of our attack which can infer the contexts of users' secret inputs with up to 91.4% accuracy.

2 BACKGROUND AND RELATED WORK

In this section, we will first describe the motion sensors embedded in the smartphones and then discuss existing attacks that aim to infer users' private information by exploiting measurements recorded by smartphone sensors.

2.1 Motion Sensors

Since motion sensors such as the accelerometer, gyroscope and rotation sensor are integrated into a smartphone, they bring the opportunity to assist navigation, location, etc., with knowledge about the motion of the smartphone users.

Accelerometer: the accelerometer measures the acceleration in m/s^2, which is the rate of change of velocity with time, of a smartphone along three axes: x-axis (lateral or left-right), y-axis (longitudinal or forward-backward), and z-axis (vertical or up-down) (Xu et al., 2012).

Linear Accelerometer: the linear accelerometer measures the acceleration effect of the smartphone movement, excluding the effect of the Earth's gravity on the device. It is typically derived from the accelerometer, where other sensors (e.g. the gyroscope) can help to remove the linear acceleration from the data. Linear acceleration units are shown in m/s^2 similar to the accelerometer.

Gyroscope: the gyroscope measures the rate of rotation in rad/s around a device's x, y, and z axis. The gyroscope is used to maintain and control the position, level or orientation of the smartphone based on the principle of angular momentum.

Rotation Sensor: the rotation (orientation) sensor measures the change of direction of the smartphone along three dimensions: x-axis (Azimuth), y-axis (Pitch), and z-axis (Roll) (Xu et al., 2012).

2.2 Inferences Derived from Smartphone Sensor Data

Mobile devices which are often equipped with sensors such as the accelerometer, gyroscope, rotation

sensor, camera, microphone, GPS and so on, are being used by mobile sensing systems to make sophisticated inferences about users. These inferences have enabled an entire ecosystem of context-aware applications such as traffic and environmental monitoring (Azizyan et al., 2009; Templeman et al., 2013; Mohan et al., 2008; Tung and Shin, 2015), behavior-based user authentication (Lee and Lee, 2017; Frank et al., 2013; Mare et al., 2014; Zhu et al., 2013; Lee et al., 2017; Riva et al., 2012; Conti et al., 2011), activity mode detection (Reddy et al., 2010; Bao and Intille, 2004; Luxton et al., 2011), and speech translation (Michalevsky et al., 2014; Lei et al., 2013).

While the smartphone sensory data has enabled context-aware applications, the same data can also be used by an adversary to make inferences about the private information of the users. Therefore, there exist fundamentally conflicting requirements between protecting privacy of users' sensitive information recorded by smartphone sensors and preserving utility of the same data for authorized context-aware applications. These private inferences include the identification of emotional state (Chang et al., 2011; Rachuri et al., 2010), speaker identity recognition (Nirjon et al., 2013; Liu et al., 2012), location tracking (Han et al., 2012; Brouwers and Woehrle, 2011; Nirjon et al., 2013; Kim et al., 2010), on-screen taps recognition (Miluzzo et al., 2012), onset of stress (Lu et al., 2012; Chang et al., 2011) and keystroke detection (Miluzzo et al., 2012; Xu et al., 2012; Liu et al., 2015; Marquardt et al., 2011; Owusu et al., 2012; Cai and Chen, 2011). Furthermore, many applications have access to data recorded by the motion sensors that do not require users' permissions, combinations of which can be maliciously used to predict more private information than what the applications advertise.

3 ATTACK OVERVIEW

3.1 Attack Goals

The objective of our attack is to infer the users' handwritten patterns by exploiting the motion sensors embedded in the smartphones. Since these motion sensors are usually considered as collecting insensitive information, our attack does not require any user permission to access the recordings of motion sensors such as the accelerometer, linear accelerometer, gyroscope and rotation sensor.

3.2 Attack Workflow

We explain the workflow of our attack which works under the training mode and the testing mode as follow:

In the training mode, when the user is interacting with the smartphone, we can record the handwritten characters entered on the touchscreen, and correlate these ground-truth information with the measurements of motion sensors collected during the handwritten events to generate the user's interaction patterns/templates.

In the testing mode, our attack keeps monitoring the measurements of motion sensors. When the user is performing sensitive inputs on the touchscreen, the acquired interaction patterns/templates in the training mode can be leveraged to infer the user's handwritten characters based on the measurements of motion sensors.

4 EXPERIMENTAL SETUP

Android Application Implementation: We develop an Android application to implement our privacy attack of inferring smartphone users' handwritten patterns. Specifically, we collect recordings from all the motion sensors including the *accelerometer*, *linear accelerometer*, *gyroscope*, and *rotation sensor* in a Google Nexus 5 (with 2.3GHz, Krait 400 processor, 16GB internal storage and 2GB RAM on Android 7.0 operating system), corresponding to the scenarios where the user entered 26 characters from A to Z on the touchscreen. The sampling frequency of our application is set to 50 Hz.

Sensor Data Collection: In our experiments, we collect sensor recordings of 10 users' handwritten events corresponding to the 26 characters, and we repeated this process for 10 times. Therefore, we collect $10 \times 26 \times 10 = 2600$ time-series sensor recordings in our data set. Furthermore, we use 10-fold cross validation in our experiments to generate the training and testing data, i.e., $9/10$ of our collected data is used as training data and the remaining $1/10$ is used as testing data. We repeated this process for 1000 iterations and reported the averaged results.

5 PROPOSED APPROACH

Our proposed inference attack is composed of two key techniques. First, we propose to exploit the dynamic time warping (DTW) algorithm (Berndt and Clifford, 1994) to quantify similarities between two

time-series sensor recordings, in order to distinguish users' handwritten patterns (detailed process will be discussed in Section 5.1). More specifically, under the training mode, we construct the *template sensor recordings* using DTW technique, by selecting the most representative sensor measurement corresponding to each handwritten pattern of the user. Under the testing mode, we evaluate the similarity between the incoming sensor signal and all the template sensor recordings by using DTW algorithm, from which we identify the closest template sensor recording and label the incoming sensor signal accordingly for each sensor dimension. Second, we develop a novel sensor fusion mechanism to generate the final inference result which integrates information contained in multiple motion sensors by leveraging the majority voting strategy (Lam and Suen, 1997) (as will be discussed in Section 5.2).

5.1 Evaluating Similarity of Sensor Recordings by using DTW

DTW is a well-known technique (Berndt and Clifford, 1994) to find the optimal alignment between two given (time-dependent) sequences $\boldsymbol{x} := (x_1, x_2, \ldots, x_N)$ of length $N \in \mathbb{N}$ and $\boldsymbol{y} := (y_1, y_2, \ldots, y_M)$ of length $M \in \mathbb{N}$ under certain restrictions. It has been successfully applied to compare different speech patterns in automatic speech recognition and other applications in the data mining community. 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 (Ding et al., 2008) tested the most cited distance measures on 47 different data sets, and no method consistently outperforms DTW. Therefore, we aim to exploit the DTW technique to carefully measure the distance between any pair of time-series sensor recordings which may vary in time or speed.

DTW calculates the distance of two sequences using dynamic programming (Bertsekas, 1995), where the sequences are warped in a nonlinear fashion to match each other. 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 $\boldsymbol{x}$ and $\boldsymbol{y}$ by aligning the element x_{n_l} of $\boldsymbol{x}$ to the element y_{m_l} of $\boldsymbol{y}$. The boundary condition enforces that the first elements of $\boldsymbol{x}$ and $\boldsymbol{y}$ as well as the last elements of $\boldsymbol{x}$ and $\boldsymbol{y}$ are aligned to each other. The total distance $d_p(\boldsymbol{x}, \boldsymbol{y})$ of a warping path p between $\boldsymbol{x}$ and $\boldsymbol{y}$ with respect to the local distance measure d is defined as

$$d_p(\boldsymbol{x}, \boldsymbol{y}) = \sum_{l=1}^{L} d(x_{n_l}, y_{m_l}) \tag{1}$$

Therefore, the DTW distance for two time-series data can be computed as

$$DTW(\boldsymbol{x}, \boldsymbol{y}) = \min d_p(\boldsymbol{x}, \boldsymbol{y}) \tag{2}$$

Constructing Template Sensor Recording: Under the training mode of our attack, we aim to construct the template sensor recording corresponding to each character and sensor dimension, i.e., $\boldsymbol{t}_{character,\ sensor,\ axis}$, where $character \in \boldsymbol{C} = \{A, \ldots Z\}$, $sensor \in \boldsymbol{S} = \{accelerometer,\ linear\ accelerometer,\ gyroscope\ ,\ rotation\ sensor\}$ and $axis \in \boldsymbol{A} = \{x,\ y,\ z\}$. Our objective is to identify the most representative sensor signal that is the closest to all the senor recordings of the same character. Specifically, for each $character \in \boldsymbol{C}$, we compute the DTW distance between any pair of sensor recordings in $\boldsymbol{R}_{character,\ sensor,\ axis} = \{\boldsymbol{r}_{character,\ sensor,\ axis}\}$ and select as template the one that achieves the smallest DTW distance, i.e.,

$$\begin{aligned} &\boldsymbol{t}_{character,\ sensor,\ axis} \\ &= \underset{\boldsymbol{r}_1 \in \boldsymbol{R}_{character,\ sensor,\ axis}}{\arg\min} \sum_{\boldsymbol{r}_2 \in \boldsymbol{R}_{character,\ sensor,\ axis}} DTW(\boldsymbol{r}_1, \boldsymbol{r}_2) \end{aligned} \tag{3}$$

Inferring Handwritten Pattern Corresponding to Each Sensor Dimension: Under the testing mode of our attack, we aim to infer the handwritten character corresponding to an incoming sensor recording $\boldsymbol{D} = \{\boldsymbol{d}_{sensor,\ axis}\}_{sensor \in \boldsymbol{S}, axis \in \boldsymbol{A}}$. Our inference attack consists of two steps: 1) identify the input character from data recorded by each sensor dimension; and 2) infer the input character by integrating the information contained in all the sensors. For the first step, we can calculate the DTW distance $Dist_{character,\ sensor,\ axis}$ between the input data $\boldsymbol{d}_{sensor,\ axis}$ and each template sensor recording $\boldsymbol{t}_{character,\ sensor,\ axis}$, from which we can identify the handwritten character corresponding to each sensor dimension as

$$\begin{aligned} &Infer(\boldsymbol{d}_{sensor,\ axis}) \\ &= \underset{character}{\arg\min}\, Dist_{character,\ sensor,\ axis} \\ &= \underset{character}{\arg\min}\, DTW(\boldsymbol{d}_{sensor,\ axis}, \boldsymbol{t}_{character,\ sensor,\ axis}) \end{aligned} \tag{4}$$

The second step of integrating the information contained in multiple sensors is described as follows.

5.2 Majority Voting based Sensor Fusion Mechanism

In order to achieve enhanced inference performance, we aim to exploit the majority voting strategy to integrate information recorded by all the sensors. Majority voting is previously applied in the classification domain (Lam and Suen, 1997), where it represents a mapping function that maps multiple classifiers' decisions into a single decision, i.e., it maps $\textbf{Class} \times \textbf{Class} \times \cdots \times \textbf{Class}$ to **Class**, where $\textbf{Class} = \{A, B, \ldots Z\}$ in our setting. The benefit of using majority voting is that it can reduce the noise and bias caused by a single classifier. Therefore, the decision made by majority voting is more robust and reliable. Specifically, assuming that we have m classifiers $Class_1, Class_2, \ldots, Class_m$, the *majority voting function* $MajorityVote(\cdot)$ can be described as

$$Class = MajorityVote(Class_1, Class_2, \cdots, Class_m) \tag{5}$$

where $MajorityVote(Class_1, Class_2, \cdots, Class_m)$ returns the most frequent result appearing within $Class_1, Class_2, \cdots, Class_m$.

With the knowledge of the inferred character corresponding to each sensor dimension $Infer(\boldsymbol{d}_{sensor,\ axis})$ in Eq. 4, we construct the set of inferred characters for all the sensors as

$$\boldsymbol{LABEL}(\boldsymbol{D}) = \{Infer(\boldsymbol{d}_{sensor,\ axis})\}_{sensor \in \boldsymbol{S},\ axis \in \boldsymbol{A}} \tag{6}$$

Then, we can infer the handwritten character corresponding to the incoming sensor recordings $\boldsymbol{D} = \{\boldsymbol{d}_{sensor,\ axis}\}_{sensor \in \boldsymbol{S},\ axis \in \boldsymbol{A}}$ as follows:

$$\begin{aligned} &Infer(\boldsymbol{D}) \\ =&MajorityVote(\boldsymbol{LABEL}(\boldsymbol{D})) \\ =&The\ Most\ Frequent\ Character\ in\ \boldsymbol{LABEL}(\boldsymbol{D}) \end{aligned} \tag{7}$$

Based on our analysis above, we therefore summarize our attack in Algorithm 1.

Algorithm 1: Our Attack of Inferring Users' Handwritten Patterns by Using Motion Sensors.

Input : The input sensor signal $\boldsymbol{D} = \{\boldsymbol{d}_{sensor,\ axis}\}_{sensor \in \boldsymbol{S},\ axis \in \boldsymbol{A}}$ and the template sensor recordings $\{\boldsymbol{t}_{character,\ sensor,\ axis}\}_{character \in \boldsymbol{C},\ sensor \in \boldsymbol{S},\ axis \in \boldsymbol{A}}$, where $\boldsymbol{C} = \{A, \ldots Z\}$, $\boldsymbol{S} = \{accelerometer, linear\ accelerometer, gyroscope, rotation\ sensor\}$ and $\boldsymbol{A} = \{x, y, z\}$;

Output: The detected character corresponding to the input sensor data $Infer(\boldsymbol{D})$;

Construct the set of inferred characters for all the sensors $\boldsymbol{LABEL}(\boldsymbol{D})$ as an empty set;
for *sensor in* $\boldsymbol{S}$ **do**
  for *axis in* $\boldsymbol{A}$ **do**
    for *character in* $\boldsymbol{C}$ **do**
      According to Eq. 2, calculate the DTW distance $Dist_{character,\ sensor,\ axis}$ between $\boldsymbol{d}_{sensor,\ axis}$ and $\boldsymbol{t}_{character,\ sensor,\ axis}$;
    end
    Label the identified character for each sensor dimension as
    $Infer(\boldsymbol{d}_{sensor,\ axis}) = \arg\min_{character} Dist_{character,\ sensor,\ axis}$;
    Update $\boldsymbol{LABEL}(\boldsymbol{D})$ as
    $\boldsymbol{LABEL}(\boldsymbol{D}) = [\boldsymbol{LABEL}(\boldsymbol{D}), Infer(\boldsymbol{d}_{sensor,\ axis})]$;
  end
end
Label the detected character corresponding to $\boldsymbol{D}$ as
$Infer(\boldsymbol{D}) = MajorityVote(\boldsymbol{LABEL}(\boldsymbol{D}))$
$= The\ Most\ Frequent\ Character\ in\ \boldsymbol{LABEL}(\boldsymbol{D})$;
return $Infer(\boldsymbol{D})$;

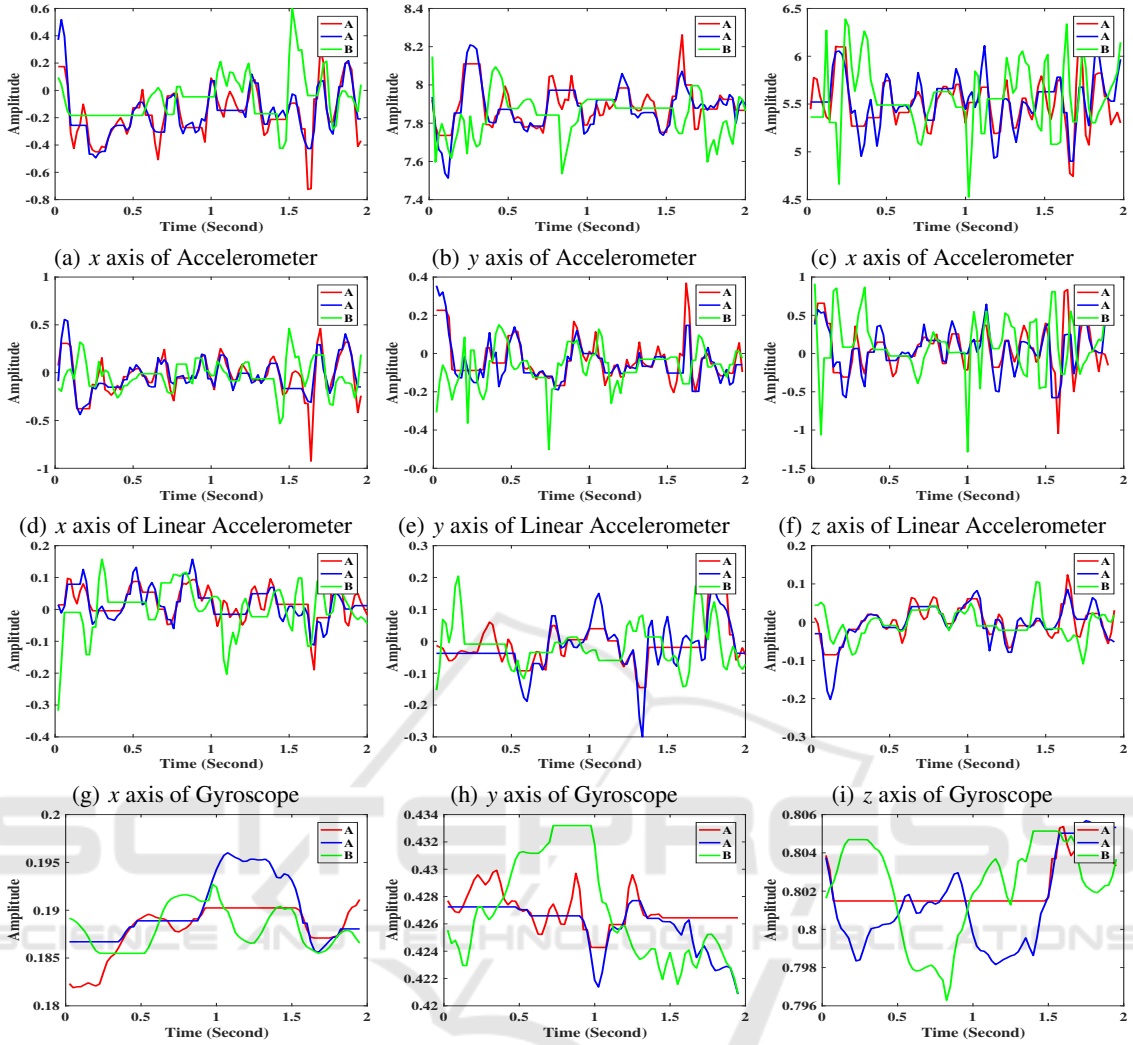
6 EVALUATION

6.1 Evaluation Metrics

In our experiments, we quantify the performance of our attack by using the metric of *accuracy*, which is computed as the ratio of correctly-inferred characters. More specifically, we can compute the *accuracy* of our attack by using measurement of each sensor dimension as

$$\begin{aligned} &Accuracy_{sensor,axis} \\ =&\frac{\sum_{\boldsymbol{d}_{sensor,\ axis}} \mathbb{I}(Infer(\boldsymbol{d}_{sensor,\ axis}) = Char(\boldsymbol{D}))}{\sum_{\boldsymbol{d}_{sensor,\ axis}} 1} \end{aligned} \tag{8}$$

where $\mathbb{I}(event)$ is the *indicator* function and $\mathbb{I}(event) = 1$ if *event* holds otherwise $\mathbb{I}(event) = 0$. $Char(\boldsymbol{D})$ represents the ground-truth character that is entered on the touchscreen corresponding to the input sensor signal $\boldsymbol{D}$. $Infer(\boldsymbol{d}_{sensor,\ axis})$ is the inferred



(a) x axis of Accelerometer (b) y axis of Accelerometer (c) x axis of Accelerometer

(d) x axis of Linear Accelerometer (e) y axis of Linear Accelerometer (f) z axis of Linear Accelerometer

(g) x axis of Gyroscope (h) y axis of Gyroscope (i) z axis of Gyroscope

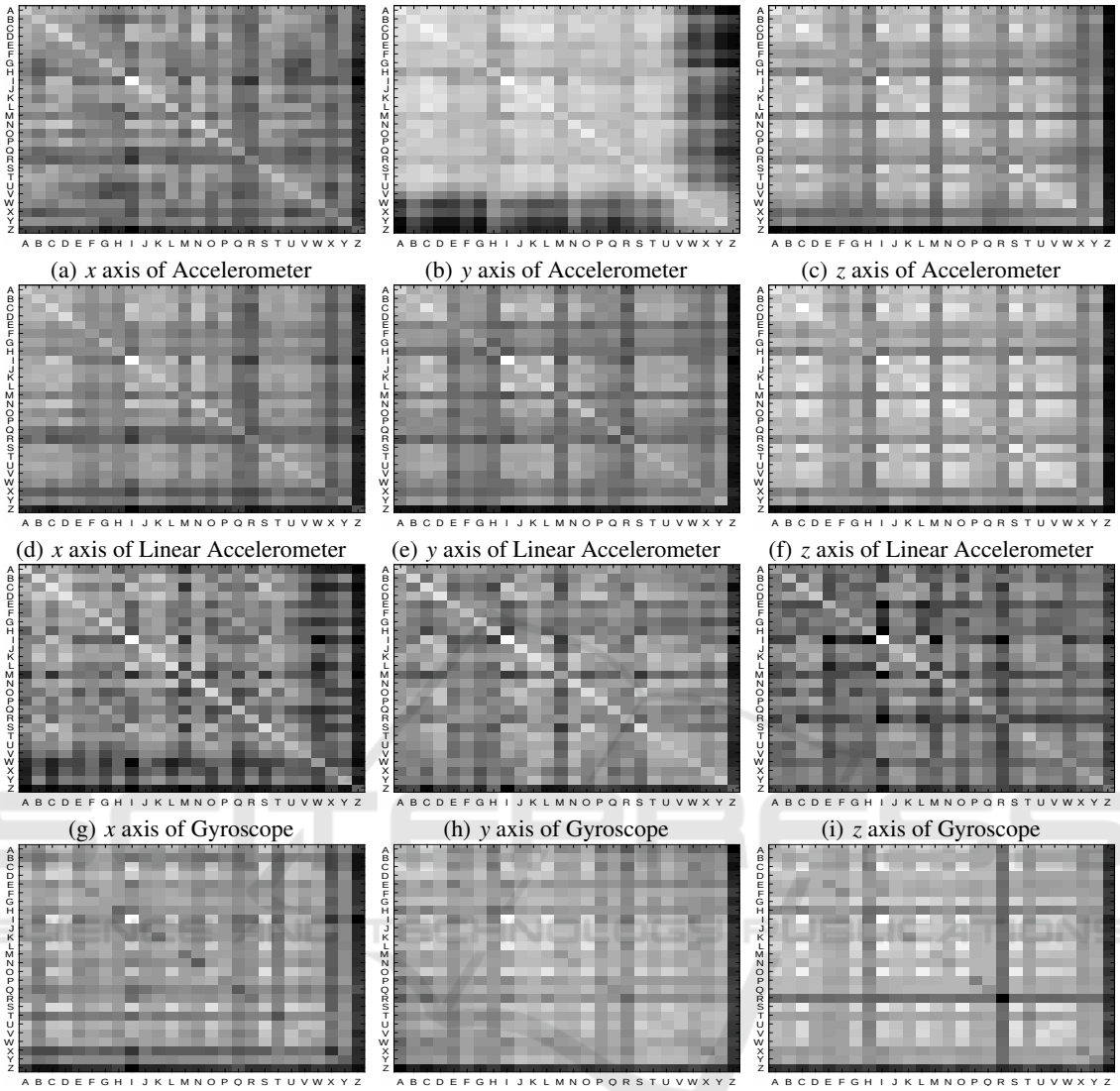
(j) x axis of Rotation Sensor (k) y axis of Rotation Sensor (l) z axis of Rotation Sensor

Figure 1: The visualization of handwritten signals extracted from the accelerometer, linear accelerometer, gyroscope and rotation sensor under the three different dimensions. We randomly select two handwritten signals from the same character A (red and blue lines) and a handwritten signal from another character B (green lines). We observe that the distance between two handwritten signals corresponding to the same character is smaller than that from a different character, which lays the foundation of our attack.

character by using data collected by each sensor dimension as shown in Eq. 4.

After applying the majority voting based sensor fusion mechanism in Section 5.2, the overall accuracy of our attack can be computed as

$$Accuracy = \frac{\sum_{\boldsymbol{d}_{sensor,\,axis},\ sensor\in\boldsymbol{S},\,axis\in\boldsymbol{A}} \mathbb{I}(Infer(\boldsymbol{d}_{sensor,\,axis}) = Char(\boldsymbol{D}))}{\sum_{\boldsymbol{d}_{sensor,\,axis},\ sensor\in\boldsymbol{S},\,axis\in\boldsymbol{A}} 1} \tag{9}$$



(a) x axis of Accelerometer (b) y axis of Accelerometer (c) z axis of Accelerometer

(d) x axis of Linear Accelerometer (e) y axis of Linear Accelerometer (f) z axis of Linear Accelerometer

(g) x axis of Gyroscope (h) y axis of Gyroscope (i) z axis of Gyroscope

(j) x axis of Rotation Sensor (k) y axis of Rotation Sensor (l) z axis of Rotation Sensor

Figure 2: The heatmap of DTW distance between sensor measurements corresponding to any pair of characters from A to Z by using different dimensions of motion sensors (the accelerometer, linear accelerometer, gyroscope and rotation sensor). A whiter color corresponds to a smaller DTW distance between two characters. We observe various distinguishing powers of motion sensors in differentiating users' handwritten patterns.

6.2 Effectiveness of Evaluating Sensor Similarity by using DTW Distance

In our attack, we utilize the DTW distance as described in Section 5.1 for evaluating the similarity between any pair of time-series sensor recordings. Figure 1 shows the distinguishing power of the motion sensors (the accelerometer, linear accelerometer, gyroscope and rotation sensor) to differentiate users' entered characters on the touchscreen. Specifically, we randomly select two time-series sensor recordings corresponding to the same character and one sensor recording corresponding to another character, and then compute the distance between these signals after implementing the DTW technique according to Eq. 2. From Figure 1, we observe that the distance between two sensor signals corresponding to the same character is much smaller than that from a different character, which lays the foundation for our attack. Figure 2 shows the heatmap of the DTW distance between sensor measurements corresponding to any pair of characters under each sensor dimension. From Figure 2, we observe that different sensors have various powers in matching the same character's handwritten gestures and distinguishing different characters' handwritten gestures, which demonstrates the empirical necessity

of our proposed majority voting based sensor fusion mechanism.

Table 1: Accuracy of Distinguishing Handwritten Patterns by Using Different Sensor Dimensions.

	X axis	*Y* axis	*Z* axis
Accelerometer	57.3%	54.6%	51.6%
Linear Accelerometer	49.9%	38.6%	48.4%
Gyroscope	75.1%	75.7%	51.6%
Rotation Sensor	35.0%	42.1%	30.6%

Table 1 shows the *accuracy* of inferring users' handwritten patterns by using different sensor dimensions according to Eq. 4. From Table 1, we know that the accuracy achieved by different sensor dimension varies from each other, and the gyroscope shows better distinguishing power than the other sensors. The reason is that a user's handwritten movement is dominated by the rate of rotation recorded by the gyroscope while the translation movement that is relevant to the accelerometer is less significant. We also observe that using the rotation sensor achieves much lower accuracy than the other sensors. The reason is that the absolute rotation values recorded by the rotation sensor is too sensitive to the handwritten movements, making it difficult for effective inference attack. This observation also provides a guide for us to explore the combination of the accelerometer, linear acclerometer and gyroscope in the practical deployment of our attack.

6.3 Effectiveness of Majority Voting based Sensor Fusion Mechanism

After applying the majority voting based sensor fusion mechanism as described in Section 5.2, our attack can achieve up to 91.4% accuracy through integrating information recorded by the accelerometer, linear accelerometer and gyroscope, which is much higher than using each sensor independently (recall Table 1). It is also interesting to know that the overall accuracy achieved by combining these three sensors and the rotation sensor is only 89.2%. This observation not only shows that utilizing more sensors does not necessarily result in better inference performance, but also demonstrates the effectiveness of only using the three sensors of the accelerometer, linear accelerometer and gyroscope in the practical attacks of inferring users' handwritten patterns.

7 DISCUSSION AND FUTURE DIRECTION

7.1 Handwritten Pattern Inferences using Motion Sensors is Practical

Our experimental results in Section 6 demonstrate the feasibility of inferring users' handwritten patterns by exploiting innocuous motion sensors. More specifically, by integrating the information recorded by the acclerometer, linear accelerometer and gyroscope, we can accurately infer users' handwritten patterns with up to 91.4% accuracy, whose performance is significantly better than using a single sensor or integrating these three sensors with the rotation sensor. This observation can serve as an effective guide for the design of practical attacks on users' handwritten patterns.

In this paper, we infer user's secret input independently (character by character). However, in reality, the input sequence may be correlated with each other for meaningful presentation (e.g., users' text message). Therefore, exploiting the correlation inherently existing between contiguous gestures to infer more secret information will be an interesting future direction.

7.2 Potential Countermeasurements

Our proposed inference attack demonstrates the fundamentally conflicting requirements between protecting privacy of users' sensitive information contained in smartphone sensors and preserving utility of the same data for authorized usage. Several sensor privacy protection mechanisms have been proposed in the literature (Beresford et al., 2011; Hornyack et al., 2011; Cornelius et al., 2008; Shebaro et al., 2014; Li and Cao, 2013) which, however, are often heuristic in nature and fail to provide rigorous privacy guarantees.

To overcome the limitations existing in previous sensor privacy protection mechanisms, potential countermeasurements for our inference attack include the differential privacy framework (Dwork, 2006) and its generalized variations (Kifer and Machanavajjhala, 2014; **?**), which can be leveraged to provide rigorous access control over smartphone sensors. Note that applying these privacy-preserving mechanisms often require the modification of smartphone operating systems which usually incur significant CPU/memory overhead and battery cost.

8 CONCLUSION

While the third-party applications relying on mobile sensors are becoming increasingly popular, the security and privacy issues related to these applications are not well understood yet. In this paper, we study the feasibility of inferring user's handwritten patterns on smartphone touchscreen by utilizing data collected by the motion sensors. In our method, we exploit the DTW technique to measure the similarity between any pair of time-series sensor data, aiming at distinguishing the user's different inputs on the touchscreen. For achieving enhanced inference accuracy, we propose a novel majority voting based sensor fusion mechanism through integrating information contained in multiple motion sensors. We present the design and implementation of our attack in an application that explores the measurements of motion sensors to stealthily infer the user's private inputs on the touchscreen. Extensive experiments using real-world data sets demonstrate the effectiveness of our attack which can achieve 91.4% accuracy for inferring users' handwritten patterns.

ACKNOWLEDGMENT

This research was sponsored by the U.S. Army Research Laboratory and the U.K. Ministry of Defence under Agreement Number W911NF-16-3-0001. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the U.S. Army Research Laboratory, the U.S. Government, the U.K. Ministry of Defence or the U.K. Government. The U.S. and U.K. Governments are authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation hereon.

REFERENCES

Azizyan, M., Constandache, I., and Roy Choudhury, R. (2009). Surroundsense: mobile phone localization via ambience fingerprinting. In *Proceedings of the 15th annual international conference on Mobile computing and networking*, pages 261–272. ACM.

Bao, L. and Intille, S. (2004). Activity recognition from user-annotated acceleration data. *Pervasive computing*, pages 1–17.

Beresford, A. R., Rice, A., Skehin, N., and Sohan, R. (2011). Mockdroid: trading privacy for application functionality on smartphones. In *Proceedings of the 12th workshop on mobile computing systems and applications*, pages 49–54. ACM.

Berndt, D. J. and Clifford, J. (1994). Using dynamic time warping to find patterns in time series. In *KDD workshop*, volume 10, pages 359–370. Seattle, WA.

Bertsekas, D. P. (1995). *Dynamic programming and optimal control*. Athena Scientific Belmont, MA.

Brouwers, N. and Woehrle, M. (2011). Detecting dwelling in urban environments using gps, wifi, and geolocation measurements. In *Workshop on Sensing Applications on Mobile Phones (PhoneSense)*, pages 1–5.

Cai, L. and Chen, H. (2011). Touchlogger: Inferring keystrokes on touch screen from smartphone motion. *HotSec*, 11:9–9.

Chang, K.-h., Fisher, D., and Canny, J. (2011). Ammon: A speech analysis library for analyzing affect, stress, and mental health on mobile phones. *Proceedings of PhoneSense*, 2011.

Conti, M., Zachia-Zlatea, I., and Crispo, B. (2011). Mind how you answer me!: transparently authenticating the user of a smartphone when answering or placing a call. In *Proceedings of the 6th ACM Symposium on Information, Computer and Communications Security*, pages 249–259. ACM.

Cornelius, C., Kapadia, A., Kotz, D., Peebles, D., Shin, M., and Triandopoulos, N. (2008). Anonysense: privacy-aware people-centric sensing. In *Proceedings of the 6th international conference on Mobile systems, applications, and services*, pages 211–224. ACM.

Ding, H., Trajcevski, G., Scheuermann, P., Wang, X., and Keogh, E. (2008). Querying and mining of time series data: experimental comparison of representations and distance measures. *Proceedings of the VLDB Endowment*, 1(2):1542–1552.

Dwork, C. (2006). Differential privacy. In *Automata, languages and programming*. Springer.

Frank, M., Biedert, R., Ma, E., Martinovic, I., and Song, D. (2013). Touchalytics: On the applicability of touchscreen input as a behavioral biometric for continuous authentication. *IEEE transactions on information forensics and security*, 8(1):136–148.

Han, J., Owusu, E., Nguyen, L. T., Perrig, A., and Zhang, J. (2012). Accomplice: Location inference using accelerometers on smartphones. In *Communication Systems and Networks (COMSNETS), 2012 Fourth International Conference on*, pages 1–9. IEEE.

Hornyack, P., Han, S., Jung, J., Schechter, S., and Wetherall, D. (2011). These aren't the droids you're looking for: Retrofitting android to protect data from imperious applications. In *Proceedings of the 18th ACM conference on Computer and communications security*, pages 639–652. ACM.

Kifer, D. and Machanavajjhala, A. (2014). Pufferfish: A framework for mathematical privacy definitions. *ACM Transactions on Database Systems (TODS)*, 39(1):3.

Kim, D. H., Kim, Y., Estrin, D., and Srivastava, M. B. (2010). Sensloc: sensing everyday places and paths using less energy. In *Proceedings of the 8th ACM Conference on Embedded Networked Sensor Systems*, pages 43–56. ACM.

Lam, L. and Suen, S. (1997). Application of majority voting to pattern recognition: an analysis of its behavior and performance. *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, 27(5):553–568.

Lee, W.-H. and Lee, R. B. (2017). Implicit smartphone user authentication with sensors and contextual machine learning. In *Dependable Systems and Networks (DSN), 2017 47th Annual IEEE/IFIP International Conference on*, pages 297–308. IEEE.

Lee, W.-H., Liu, X., Shen, Y., Jin, H., and Lee, R. B. (2017). Secure pick up: Implicit authentication when you start using the smartphone. In *ACM Symposium on Access Control Models and Technologies (SACMAT)*.

Lei, X., Senior, A. W., Gruenstein, A., and Sorensen, J. (2013). Accurate and compact large vocabulary speech recognition on mobile devices. In *Interspeech*, volume 1.

Li, Q. and Cao, G. (2013). Efficient privacy-preserving stream aggregation in mobile sensing with low aggregation error. In *International Symposium on Privacy Enhancing Technologies Symposium*, pages 60–81. Springer.

Liu, B., Jiang, Y., Sha, F., and Govindan, R. (2012). Cloud-enabled privacy-preserving collaborative learning for mobile sensing. In *Proceedings of the 10th ACM Conference on Embedded Network Sensor Systems*, pages 57–70. ACM.

Liu, X., Zhou, Z., Diao, W., Li, Z., and Zhang, K. (2015). When good becomes evil: Keystroke inference with smartwatch. In *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*, pages 1273–1285. ACM.

Lu, H., Frauendorfer, D., Rabbi, M., Mast, M. S., Chittaranjan, G. T., Campbell, A. T., Gatica-Perez, D., and Choudhury, T. (2012). Stresssense: Detecting stress in unconstrained acoustic environments using smartphones. In *Proceedings of the 2012 ACM Conference on Ubiquitous Computing*, pages 351–360. ACM.

Luxton, D. D., McCann, R. A., Bush, N. E., Mishkind, M. C., and Reger, G. M. (2011). mhealth for mental health: Integrating smartphone technology in behavioral healthcare. *Professional Psychology: Research and Practice*, 42(6):505.

Mare, S., Markham, A. M., Cornelius, C., Peterson, R., and Kotz, D. (2014). Zebra: Zero-effort bilateral recurring authentication. In *Security and Privacy (SP), 2014 IEEE Symposium on*, pages 705–720. IEEE.

Marquardt, P., Verma, A., Carter, H., and Traynor, P. (2011). (sp) iphone: Decoding vibrations from nearby keyboards using mobile phone accelerometers. In *Proceedings of the 18th ACM conference on Computer and communications security*, pages 551–562. ACM.

Michalevsky, Y., Boneh, D., and Nakibly, G. (2014). Gyrophone: Recognizing speech from gyroscope signals. In *USENIX Security Symposium*, pages 1053–1067.

Miluzzo, E., Varshavsky, A., Balakrishnan, S., and Choudhury, R. R. (2012). Tapprints: your finger taps have fingerprints. In *Proceedings of the 10th international conference on Mobile systems, applications, and services*, pages 323–336. ACm.

Mohan, P., Padmanabhan, V. N., and Ramjee, R. (2008). Nericell: rich monitoring of road and traffic conditions using mobile smartphones. In *Proceedings of the 6th ACM conference on Embedded network sensor systems*, pages 323–336. ACM.

Nirjon, S., Dickerson, R. F., Asare, P., Li, Q., Hong, D., Stankovic, J. A., Hu, P., Shen, G., and Jiang, X. (2013). Auditeur: A mobile-cloud service platform for acoustic event detection on smartphones. In *Proceeding of the 11th annual international conference on Mobile systems, applications, and services*, pages 403–416. ACM.

Owusu, E., Han, J., Das, S., Perrig, A., and Zhang, J. (2012). Accessory: password inference using accelerometers on smartphones. In *Proceedings of the Twelfth Workshop on Mobile Computing Systems & Applications*, page 9. ACM.

Rachuri, K. K., Musolesi, M., Mascolo, C., Rentfrow, P. J., Longworth, C., and Aucinas, A. (2010). Emotionsense: a mobile phones based adaptive platform for experimental social psychology research. In *Proceedings of the 12th ACM international conference on Ubiquitous computing*, pages 281–290. ACM.

Reddy, S., Mun, M., Burke, J., Estrin, D., Hansen, M., and Srivastava, M. (2010). Using mobile phones to determine transportation modes. *ACM Transactions on Sensor Networks (TOSN)*, 6(2):13.

Riva, O., Qin, C., Strauss, K., and Lymberopoulos, D. (2012). Progressive authentication: Deciding when to authenticate on mobile phones. In *USENIX Security Symposium*, pages 301–316.

Shebaro, B., Oluwatimi, O., Midi, D., and Bertino, E. (2014). Identidroid: Android can finally wear its anonymous suit. *Transactions on Data Privacy*, 7(1):27–50.

Templeman, R., Zahid Rahman, D. C., and Kapadia, A. (2013). Placeraider: Virtual theft in physical spaces with smartphones. In *in Network and Distributed System Security Symposium*. Citeseer.

Tung, Y.-C. and Shin, K. G. (2015). Echotag: accurate infrastructure-free indoor location tagging with smartphones. In *Proceedings of the 21st Annual International Conference on Mobile Computing and Networking*, pages 525–536. ACM.

Xu, Z., Bai, K., and Zhu, S. (2012). Taplogger: Inferring user inputs on smartphone touchscreens using onboard motion sensors. In *Proceedings of the fifth ACM conference on Security and Privacy in Wireless and Mobile Networks*, pages 113–124. ACM.

Zhu, J., Wu, P., Wang, X., and Zhang, J. (2013). Sensec: Mobile security through passive sensing. In *Computing, Networking and Communications (ICNC), 2013 International Conference on*, pages 1128–1133. IEEE.