Theft-Resilient Mobile Wallets: Transparently Authenticating NFC Users with Tapping Gesture Biometrics

Babins Shrestha
University of Alabama at Birmingham
babins@uab.edu

Manar Mohamed
University of Alabama at Birmingham
manar@uab.edu

Sandeep Tamrakar
Aalto University
sandeep.tamrakar@aalto.fi

Nitesh Saxena
University of Alabama at Birmingham
saxena@cis.uab.edu

ABSTRACT

The deployment of NFC technology on mobile phones is gaining momentum, enabling many important applications such as NFC payments, access control for building or public transit ticketing. However, (NFC) phones are prone to loss or theft, which allows the attacker with physical access to the phone to fully compromise the functionality provided by the NFC applications. Authenticating a user of an NFC phone using PINs or passwords provides only a weak level of security, and undermines the efficiency and convenience that NFC applications are supposed to provide.

In this paper, we devise a novel gesture-centric NFC biometric authentication mechanism that is fully transparent to the user. Simply "tapping" the phone with the NFC reader – a *natural* gesture *already* performed by the user prior to making the NFC transaction – would unlock the NFC functionality. An unauthorized user cannot unlock the NFC functionality because tapping serves as a "hard-to-mimic" biometric gesture unique to each user. We show how the NFC tapping biometrics can be extracted in a highly robust manner using multiple – motion, position and ambient – phone's sensors and machine learning classifiers. The use of multiple sensors not only improves the authentication accuracy but also makes active attacks harder since multiple sensor events need to be mimicked simultaneously. Our work significantly enhances the security of NFC transactions without adding any extra burden on the users.

1. INTRODUCTION

Mobile devices, especially smartphones, are rapidly becoming ubiquitous. These devices open up immense opportunities for everyday users offering valuable resources and services. A prime example of one such service, now getting widely deployed on smartphones, is Near Field Communications (NFC). NFC allows the phone to communicate with any other NFC device (an external contactless reader or another NFC phone) when they are in close proximity, typically upon tapping to one another. This facilitates many important applications in day-to-day life including payments (using the phone essentially as a digital wallet), access control for buildings [6, 13, 43] and vehicles [8, 44], and public transit ticketing [15, 18], to name a few. The NFC technology, especially mobile payments, is already popular in many countries (e.g., China and Japan) [25] and has been gaining momentum in many other countries (e.g., the US). Introduction of Apple Pay [3], Android Pay [2] and Samsung Pay [37] have further boosted the growth of NFC payments.

Given the rise of NFC deployments, a natural concern pertains to the security of NFC phones and NFC applications. One obvious and serious threat is that of loss or theft of NFC phones – an unauthorized entity in physical possession of an NFC phone can fully compromise the NFC functionality leading to severe consequences (e.g., making hefty purchases on behalf of the user or entering the user's office premises). Given many current mobile users do not lock their phones (e.g., with a PIN or pattern) [14], the abuse of NFC services becomes a real threat. A report by Boyles et al. [7] estimates that nearly one third of cell phone owners have experienced a lost or stolen phone, and 12% have had another person access the contents of their phone in a way that made them feel their privacy was invaded. Consumer Reports reported that 3.1 million smartphones were stolen in 2013, nearly double the year before [9]. While Lookout reported that 1 in 10 smartphone owners are victims of phone theft from a survey conducted in 2014 [24].

To address this problem, many NFC apps (e.g., Google Wallet) authenticate the user prior to making an NFC transaction with a PIN or password. This approach, however, has two major problems. First, given PINs or passwords are only short and weak secrets (especially in the context of mobile phones with small form factors), they can be easily guessed or brute-forced [1, 31, 48]. Second, typing in the PIN or password for *each NFC transaction* can be tedious and potentially annoying for the user, thereby significantly undermining the usability of NFC technology as it was inherently designed for easy and fast transactions [30, 45].

Given the rather poor security and usability offered by

ACSAC '16, December 05-09, 2016, Los Angeles, CA, USA

DOI: http://dx.doi.org/10.1145/2991079.2991097

PINs/passwords for the purpose of NFC user authentication, we set out to investigate a fully transparent and hard to compromise authentication mechanism. In short, we propose a *transparent behavioral biometrics* [21, 47] mechanism drawn from the gesture involving the tapping of a phone with a transaction terminal (e.g., an external NFC reader at point of service) while completing an NFC transaction. The tapping gesture is performed by the user prior to making a purchase using an NFC phone anyway and therefore no additional burden is required from the user in our approach. Our approach is not a general user authentication mechanism, for example, to unlock the phone, but is a specific user authentication mechanism to authorize the NFC transactions.

Our Contributions: The main contributions of this paper are summarized below:

1. **Transparent Authentication of NFC Users**: We show how the user's phone tapping gesture naturally exhibited during an NFC transaction can be uniquely detected in a robust manner using *multiple* – motion, position and ambient – sensors and *machine learning classifiers*. Such a biometric authentication would be very hard, if not impossible, for an attacker to compromise as the attacker needs to mimic the victim user's subtle hand movement and phone orientations measured by *multiple* different sensors. Thus, we believe that our work significantly enhances the security of NFC phones without adding any extra burden on the users.

2. **Gesture Detection Design and Implementation**: We design and implement the NFC tapping gesture detection biometrics as part of the proposed authentication mechanism. The NFC tapping gesture involves holding the phone in hand, and tapping and holding onto a NFC transaction terminal until the user is notified about the transaction completed/denied message. We extract multiple features from the phone's different sensors when a user taps her phone to NFC transaction terminal and implement machine learning approach to identify if the sensor data corresponds to the owner of the device (or not).

3. **Experiments and Evaluation**: To demonstrate the effectiveness of our approach, we have collected data from multiple users in near real-life scenarios emulating typical NFC transaction settings, and report on the overall accuracy of our authentication mechanism. Our results show that NFC tapping biometrics can be extracted with a high overall accuracy (92% on an average), while it does not seem possible for even a trained active attacker to succeed in mimicking the tapping gesture of a victim user.

Paper Outline: The rest of this paper is organized as follows. In Section 2, we outline our system and threat model, and our design goals. In Section 3, we elaborate on our approach including the underlying system architecture. In Section 4, we describe the design of our app system. Next, in Section 5, we report on our data collection procedures. In Section 6, we present the design and evaluation results for our NFC tap biometrics system. Finally, in Section 8, we discuss other properties of our system, including resistance to active attacks and NFC unauthorized reading.

2. BACKGROUND

2.1 System and Threat Model

We assume that a user owns an NFC-enabled phone that she uses to make NFC transactions with transaction terminals for payments or public transit ticketing. As our payment device in this paper, we focus on an NFC-enabled phone with an NFC transaction application. The transaction terminal accepts the payment when the owner taps his/her phone to the reader. We assume that the phone is only used by its owner and not shared with others.

In our threat model, we assume that the phone is in the possession of an attacker. The attacker might have stolen the device or could be performing a lunch-time attack. Lunch-time attack [32] is an attack scenario where the owner might have left the device in an office for a short time during which the attacker can access the device and perform malicious activities. In such a situation, the attacker would have access to the device for a limited time. However, the attacker has complete control over the device physically.

We assume that the phone's OS kernel is healthy and the attacker is unable to alter the kernel control flow. Strengthening the kernel is an orthogonal problem [33, 39]. We also assume that attacker cannot manipulate device's onboard sensor hardware. In other words, the attacker only has physical access to the device but does not have internal control of the device.

The attacker attempts to make transactions by tapping the stolen phone at an NFC transaction terminal. The NFC terminal at the merchant side is not compromised. However, the terminal is not aware of the fraudulent transaction. The merchant will process the transaction in a normal fashion as if the actual user is making the transaction. The goal of our system is to prevent such an attacker from utilizing the NFC transaction functionality when the attacker taps the stolen phone to make transactions. We want to achieve this transparently without involving additional effort from the NFC user other than tapping.

2.2 Design Goals

We consider following design goals for our authentication approach to be useful in practice.

- *Lightweight:* The authentication mechanism should be lightweight in terms of the various resources available on the device, such as memory, computation and battery power.

- *Efficient:* The approach should not incur high delay affecting the overall usability of the system. The entire authentication process should be completed within few seconds.

- *Robust:* The approach should have low error rates. The owner of the device must be able to authenticate to the phone with a high probability, while the impostor should be denied access with a high probability.

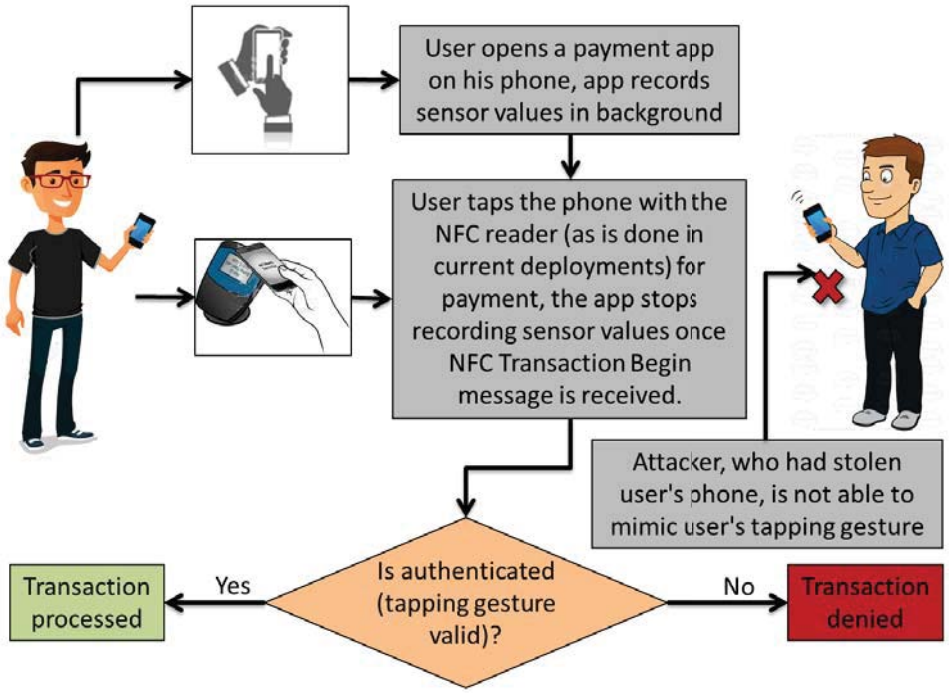

Figure 1: Overview of our system. The user gets authenticated just based on the uniqueness of his tapping gesture, a form of behavioral biometrics. The process is completely transparent to the user – no additional work is needed beyond what is currently done in NFC systems.

- *Consistency & Transparency:* The approach should not require the user to perform additional action while making a transaction: when the user makes an NFC transaction, she takes out her NFC-enabled phone, opens an app for payment, taps the phone to the transaction terminal, holds the phone until the transaction completes and removes her phone from the transaction terminal. The approach should not alter this model, thereby making the approach consistent and transparent to the users. The user should not be required to perform additional actions such as explicit gestures [23, 42] or passwords or PIN entry. Since these tasks add burden to the users which may degrade usability and, therefore, reduce chances of adoption.

3. OUR APPROACH: TAP BIOMETRICS

3.1 Background and Overview

Different user authentication approaches have been used based on "something you know", "something you have" and "something you are". In this paper, we set forth to authenticate users while they use the NFC applications based on "something you are". This approach has advantage over the first two approaches since people forget things (e.g., passwords) and lose things (e.g., access tokens). In other words, our approach implements biometric authentication to authenticate users. The biometric characteristics of an individual are believed to be easily measured accurately but hard to impersonate by others. Such biometric characteristics can again be classified into two different categories, physiological biometrics [20, 21] and behavioral biometrics [47]. In physiological biometrics, the authentication system uses physiological features of the user such as her facial structure, fingerprint or retina pattern, whereas in behavioral biometrics, the authentication system uses behavior of the user such as her keyboard typing pattern, or walking pattern. Physiological biometric authentication requires the user to perform some explicit actions such as using camera for face recognition or scanning finger/retina/iris while behavioral biometrics are usually transparent to the users and recognizes the user implicitly. In this paper, we use behavioral biometric characteristics to authenticate the user while she performs a tapping gesture before completing an NFC transaction.

When a user makes an NFC transaction using her NFC-enabled device (let's say an NFC phone), she taps her phone to the transaction terminal and holds it for a while. When the transaction completes or gets interrupted, she removes her phone away from the terminal. These steps are illustrated in Figure 1. Tapping a phone to an NFC transaction terminal involves a particular motion of her phone which can be measured using different embedded sensors on the phone. The motion sensors and the position sensors can give us information about how the phone was moved. Also, there may be significant changes in the pressure as detected by the device when moved. This can also be used to analyze how the device was moved. We observe in Section 6 that the features extracted from pressure sensor are indeed one of the discriminating features for the machine-learning classifiers.

In this paper, we show that the tapping gesture performed by a user before making NFC transactions is unique to the user and can be detected in a robust manner using machine learning classifiers and multiple sensors available on

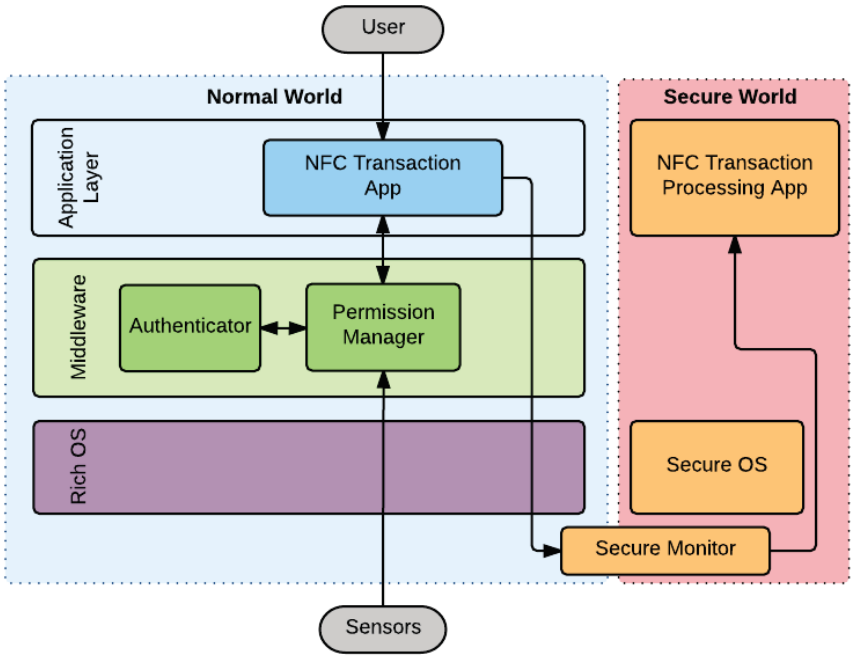

Figure 2: Our System Architecture: Control Flow

the phone. In the following section, we will demonstrate that our approach meets all of our design goals introduced in the prior section.

In our model, we add another layer of a security check on top of the default authentication system of Android and that of an NFC transaction app. Android can authenticate a user via different options such as passwords, PINs, face recognition, or fingerprint scanner. However, many users do not prefer to lock their phones. Also, using PINs or fingerprint scans for each transaction can be burdensome. These mechanism require explicit action and is not transparent to the users. We provide a way to authenticate the user before making the transaction in a way that does not require any explicit user action – just tapping the phone to the terminal (as is done currently) is sufficient. Our approach is invisible to users and requires no additional actions from the users. Our approach accurately identifies legitimate users and prevents unauthorized NFC transactions. It can also work seamlessly with other authentication methods, such as PINs or fingerprint scans when used, to achieve strong two-factor security.

3.2 System Architecture

Figure 2 depicts the control flow for our approach. Our system analyzes the collected sensor values and compares with a pre-registered template of the user's tapping gesture. Our system grants permissions to complete NFC transactions if and only if the sensor values match with the user's tapping gestures. Our system includes four modules: (1) *NFC Transaction App* which provides the user interface and handles NFC communication, (2) *Transaction Processing Module* which processes the NFC transaction messages, (3) *Authenticator Module* which is a trained classifier that uniquely identifies the user's tapping gesture, and (4) *Permission Manager* that reads the sensor values, communicates with the Authenticator Module and grants the NFC Transaction App with the permission to interact with the Transaction Processing Module.

We assume that the Transaction Processing Module executes as a *Trusted application* inside trusted execution environment (TEE), e.g. ARM TrustZone [4]. ARM TrustZone divides a device platform into two execution environments, namely, *normal world* and *secure world*. The normal world is used to host rich Operating Systems (OS), like Android OS, and user applications while it allows processing of security sensitive codes in isolation within the secure world. The two worlds communicate with each other via secure monitor.

In our approach, the trusted application is responsible for processing transaction specific messages, handling necessary cryptographic operations and maintaining secrets like keys required for NFC transactions. On the other hand, NFC Transaction App running on the normal world handles user interactions and NFC communication. To authenticate a user based on her tapping gesture, our system begins collecting information from different sensors as soon as the user opens NFC Transaction App. Our system also records the time when the phone receives the first NFC message from the NFC transaction terminal. At this point, the user must have tapped her phone to the NFC transaction terminal and she is holding her phone towards the terminal to complete the NFC transaction.

Whenever the NFC Transaction App starts, it informs the Permission Manager to indicate that it has started. Permission Manager immediately starts collecting the sensor val-

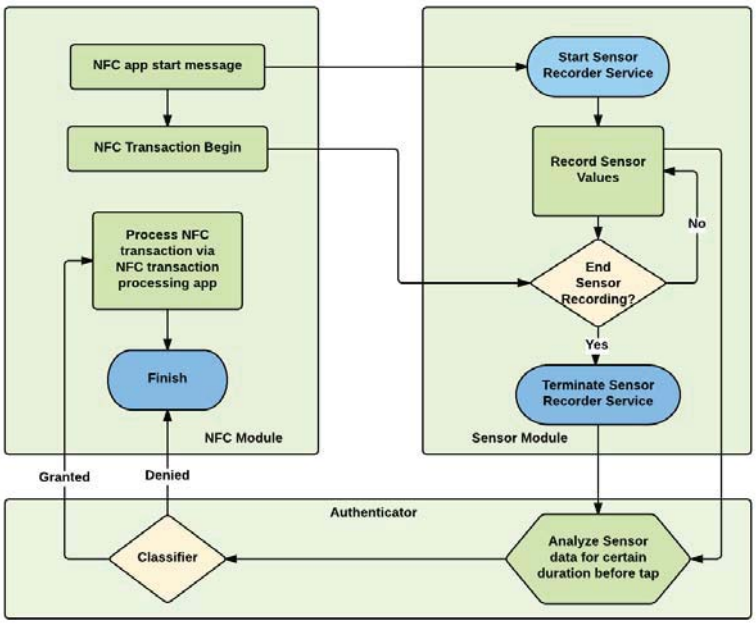

Figure 3: Sensor data collection flowchart.

ues. When the NFC Transaction App requires to process transaction messages, it requests the Permission Manager by sending NFC event begin time. The Permission Manager sends the set of appropriate sensor values to the Authenticator. Once the Authenticator confirms the tapping gesture as belonging to the user, the Permission Manager permits the NFC Transaction App to interact with the transaction module to complete the NFC transaction.

4. APPLICATION DESIGN

To develop and evaluate our authentication mechanism based on tap gesture biometrics, we first needed to collect the tap gesture data from different users. After the data collection, different features were to be generated to robustly identify individual user data from other user data. We chose to implement our system in the Android OS. For the data collection, we created two modules: (1) *NFC Transaction Module* for a user to perform the tap gesture on a NFC transaction terminal which simulates NFC transactions, and (2) *Sensor Module* to record sensor values when the user performs the tap so that underlying data can be analyzed and later used to identify the user.

4.1 NFC Transaction Module

Android provides NFC Host Card Emulation APIs that allows the NFC-enabled phone to acts as a contactless card and allows NFC applications to communicate with external contactless readers. We designed our NFC module to simulate a real-world NFC transaction application. For this, we chose to implement an NFC based public transit ticketing system. We designed and implemented both NFC ticketing application on the phone and the ticket reader application that controls the NFC transaction terminal. Both applications use a shared 128-bit AES key to authenticate each other during an NFC transaction. Specifically, we used three-pass mutual authentication protocol of MIFARE DESFire EV1 [1] as Kasper et al. [22] elaborated. Ticketing applications based on Mifare DESFire are widely used by public transit authorities around the world. NFC ticketing is only one aspect of an NFC transaction, nevertheless, it can be used as an analogy to understand user's NFC tapping gesture during any NFC transaction (e.g., for payments or building entry).

4.2 Sensor Module

Android platform provides several sensors that allow developers to monitor the motion of the device, the position of the device or the environment in which the device is. To be specific, the Android platform provides three broad categories of sensors, namely, motion sensors which measure acceleration forces and rotational forces along three axes, position sensors which measure physical position and orientation of the device, and environmental sensors which measure various environment parameters such as humidity, light illumination, ambient temperature, pressure and so on.

We created an Android service such that whenever the service is called by another activity or service, the service starts recording selected sensor values. The sensors we considered in our app are listed in Table 1. The sensors values are logged along with the timestamps so that they can be used for statistical analysis later on. When the calling app sends the stop service command to the service, the service stops recording the sensor data. The flow chart of this data collection process is shown in Figure 3.

[1]MIFARE DESFire EV1: http://www.nxp.com/products/identification_and_security/smart_card_ics/mifare_smart_card_ics/mifare_desfire/series/MIFARE_DESFIRE_EV1_4K.html

Figure 4: A user tapping an NFC reader at waist-flat position. In waist-flat scenario, the NFC reader is kept at the height of 0.75-1 m from the ground and horizontally on the table.

5. DATA COLLECTION

To develop and evaluate our approach, we first needed to collect data from multiple users. We also wanted to capture various types of gestures that users make while tapping their NFC-enabled phone to the terminals installed for different types of NFC applications. There is no standard instruction on how NFC transaction terminals should be placed, e.g. they can be placed horizontally, vertically or at certain angle from the surface where they are placed at the NFC transaction terminals. We designed our data collection engine to capture four different scenarios based on how the NFC transaction terminals may be installed: (1) *Waist-Flat*: horizontally at the height of 0.75-1 meter above the ground, (2) *Waist-Angular*: at 45 degree angle with horizontal surface at the height of 0.75-1 meter above the ground, (3) *Chest-Angular*: at 45 degree angle with the vertical surface at the height of 1-1.5 meter above the ground, and (4) *Chest-Vertical*: vertically at the height of 1-1.5 meter above the ground. A user tapping an NFC reader in the waist-flat scenario is shown in Figure 4. We implemented an app as discussed in the Section 4 and collected data using Google Nexus 5 as our phone model. We used NFC reader ACR 122U as the transaction terminal.

As the user opens the app to make NFC transactions, our system runs in the background as a service as mentioned in the Section 4.2. We continuously recorded the sensor values for the experiment and detailed analysis, however, in the real-life implementation, the sensors can be turned off as soon as the transaction success message is received or shortly thereafter.

For data collection, we invited volunteers to our lab via word of mouth. These volunteers were university students from different countries situated in the US and Finland. There were a total of **20 volunteers** (17 male and 3 female, between the age of 25-35) who participated in our study. We only observed four left handed users, while rest of them were right handed, and none of them swapped their phone from one hand to other during the experiment. The experiment was performed in lab settings. We provided a smartphone to the volunteers and asked them to tap it to the reader. Each user opened the app, tapped to the reader to initiate an NFC transaction and held it there until he/she was notified about the transaction complete message as displayed on the phone. Then the user brought the phone away from the reader. We asked each user to pause for few seconds before he/she tapped again for another transaction.

In one session, we asked the user to tap and perform the transaction five times for each of the four different reader positions mentioned above, i.e., after the user tapped the reader five times, we changed the position of the reader to a different setting. Hence, in a session, we collected 20 tap gesture samples from each user (five each for four different reader positions). We conducted six sessions collecting 120 tap gesture samples for each user (30 samples of data for each of the four positions of the reader). These six sessions were conducted in time spans ranging from either one day to six days depending upon the availability of the volunteers. However, each session had sufficient gap to break the user's rhythm of tapping and add variation to the user's hand motion. Our University's Institutional Review Board approved the study.

6. TAP BIOMETRICS DETECTION: DESIGN AND EVALUATION

6.1 Set-Up and Design

In order to evaluate the feasibility of the proposed tap gesture biometrics as an authentication scheme, we utilized the machine learning approach based on the underlying readings of the motion sensors, the position sensors and the ambient pressure sensors (the different sensor employed are listed in Table 1).

Classifier: We utilized the Random Forest classifier in our analysis. Random Forest is an ensemble approach based on the generation of many classification trees, where each tree is constructed using a separate bootstrap sample of the data. In order to classify a new input, the new input is run down on all the trees and the result is determined based on majority voting. Random trees have been shown to be a strong competitor to Support Vector Machine (SVM), and its performance frequently outperforms SVM [29]. Random Forest is efficient, can estimate the importance of the features, and is robust against noise [29].

Features: For each of the position and the motion sensor instances, we calculated the square root of the sum of squares for that instance's axes components (X, Y, Z), such that it captures the significance of all the three axes. Then, we calculated the mean and the standard deviation of all the instances in the sample that corresponds to a single tap. This gave us twenty features, which we used for training and testing of the Random Forest classifier.

The twenty features were used as input to train the classifier to differentiate a user from other users. We evaluated two training models for the classification task: (1) *scenario-specific model*, and (2) *general model*. The scenario-specific model requires each user to train a classifier on all reader (transaction terminal) positions (described in Section 5) before using the app. This model assumes that the classifier

Table 1: Sensors employed for authenticating users based on tap gesture.

Sensor	Type	Function
Accelerometer	Motion	Acceleration force including gravity
Gravity		Force of gravity on the device
Gyroscope		Rate of rotation of the device
Linear Acceleration		Acceleration force excluding gravity
Rotation Vector		Rotation vector of the device (uses geomagnetic field and gyroscope)
Game Rotation Vector	Position	Rotation vector of the device (does not use geomagnetic field)
Geomagnetic Rotation Vector		Rotation vector of the devices (uses magnetometer)
Magnetic Field		Earth's magnetic field
Orientation		Position of a device relative to the earth's frame
Pressure	Environment	Ambient air pressure

knows or is informed about the position of the reader (i.e., the scenario for the transaction). The generalized model, in contrast, uses all the data from all different scenarios of the user and builds a global classifier per user regardless of the reader position. Moreover, we have tested multiple gesture duration by utilizing the sensor data of one, two and three seconds before the transaction begins. Our goal was to determine the optimal duration of the tapping gesture which can uniquely identify each user.

In all of the classification tasks, the positive class corresponds to the tap gesture of the legitimate user and the negative class corresponds to impersonator (other user). Therefore, true positive (TP) represents the number of times the legitimate user is granted access, true negative (TN) represents the number of times the impersonator is rejected, false positive (FP) represents the number of times the impersonator is granted access and false negative (FN) represents the number of times the correct user is rejected.

As performance measures for our classifiers, we used Precision, Recall and F-measure (F1 score), as shown in Equations (1) to (3). Precision measures the security of the proposed system, i.e., the accuracy of the system in rejecting impersonators. Recall measures the usability of the proposed system as low recall leads to high rejection rate of the legitimate users. F-measure considers both the usability and the security of the system. To make our system both usable and secure, ideally, we would like to have F-measure as close as 1.

$$precision = \frac{TP}{TP + FP} \tag{1}$$

$$recall = \frac{TP}{TP + FN} \tag{2}$$

$$F\text{-}measure = 2 * \frac{precision * recall}{precision + recall} \tag{3}$$

6.2 Classification Results

General Model: As mentioned in Section 5, we collected data from 20 users. Each user performed a total of 120 taps. We divided the collected data into 20 sets based on the users' identities (ids). In order to build a classifier to authenticate a user based on her tapping biometrics, we defined two classes. The first class contains the Tap data from a specific user, and the other class contains randomly selected Tap data from other users. We analyzed three different duration of the tapping gesture, by considering one, two and three seconds before the transaction begins.

After running a 10-fold cross validation, we obtain results for different duration and different scenarios. The results show that one second of sensor data is enough for authenticating the user, shown with high F-Measure, recall and precision. Increasing the gesture duration did not improve the accuracy; it would rather decrease the accuracy as it may incorporate random user movement before the actual tapping gesture starts. We summarize the results for different scenarios with one second duration of the tap gesture in Table 2. The results for longer durations of the tap gesture (two seconds and three seconds) are shown in Tables 4 and 5 in Appendix A. These results suggest that increasing the tap duration does not seem to increase the accuracy and therefore one second duration seems optimal. Hence, the rest of the experiments reported in this paper are conducted with the one second duration of the tap gesture.

In our experiment, 12 out of the 20 users performed all the tapping in one day, and, for this sub-group of users, the average and standard deviation (for tapping duration of 1 second before) were 0.97 (0.03) for these users. The data collection from the rest of the users spanned between 4 and 22 days, and, for these users, the average and standard deviation of the F-Measure dropped to 0.88 (0.03). In practice, the classification models can be re-trained as the user makes new successful transactions such that the accuracy does not drop as the time gap between the testing and training data increases.

Scenario-Specific Model: In our scenario-specific model, we divided the collected data into 80 sets based on the user's ids and the scenario's (reader positions) id. In order to build a classifier to authenticate the user based on the tapping in a given specific scenario, we define two classes. The first class has the tap gesture data from a specific user in a specific scenario, and the other class contains randomly selected data from other users corresponding to the same scenario.

The classification results are calculated after running a 10-fold cross validation and shown in Table 2. The classification accuracy for the scenario-specific model is less than its correspondent in the general model. This may be due to the reduced number of instances in each of the files (30 versus

Table 3: The results for the active attack. The performance of the classifier built using 120 taps for generalized classification model as well as using 30 taps for different scenario specific classification model for the particular victim is shown. The last column shows the attack success rate FPR (False Positive Rate) for the corresponding classifier. FPR represents the rate at which the attacker was falsely classified as the victim. The attacker was not successful at all in mimicking the victim's tap gesture.

	Victim			Attacker
	F-measure	Recall	Precision	FPR
Generalized	0.98	0.98	0.99	0
Chest-Angular	0.98	1	0.97	0
Waist-Flat	0.97	0.97	0.97	0
Chest-Vertical	0.97	1	0.94	0
Waist-Angular	0.98	1	0.97	0

Table 2: The results for Generalized and Scenario-specific models. Each column shows average (Avg) and standard deviation (S.D.) for F-Measure, Recall and Precision for tapping duration of one second. Precision captures the security of the system while recall captures the usability of the system. F-measure accounts for both precision and recall.

	F-Measure	Recall	Precision
	Avg (S.D.)	Avg (S.D.)	Avg (S.D.)
Generalized	0.93 (0.05)	0.97 (0.03)	0.91 (0.08)
Chest-Angular	0.89 (0.06)	0.92 (0.06)	0.87 (0.07)
Waist-Flat	0.91 (0.06)	0.95 (0.05)	0.88 (0.07)
Chest-Vertical	0.92 (0.07)	0.94 (0.05)	0.89 (0.09)
Waist-Angular	0.91 (0.06)	0.95 (0.04)	0.88 (0.08)

120 in the general model). However, both models seem to perform about equally well in detecting the tap biometrics.

6.3 Summary of Results

The results obtained from both the classification models show that the tap gesture can be detected in a robust manner and thus will serve as an effective method for authenticating the users of NFC devices. This is reflected in high precision, recall and F-measure for both models. The general model can be used in applications where the user can train the model with tapping gestures in different scenarios (reader positions). The scenario-specific model can be used in practice when the phone can acquire the knowledge about the reader position. This knowledge can be acquired either by asking the user about the reader position, although this will require some user involvement in the authentication process, or the terminal can send its position to the phone.

6.4 Power Analysis

Since our app records sensor values, we set forth to analyze if our system is lightweight. To measure the battery power consumption, we used *PowerTutor* [50]. *PowerTutor* is an app readily available on on Google PlayStore[2] which estimates the power/energy consumed by different apps installed on the phone. The app provides the power/energy consumed by apps based on various parameters such as screen brightness, CPU usage, Wi-Fi polling and so on. We compared the energy consumed by our app with *NFCtools*[3], one of the most popular apps for NFC in Google PlayStore. We logged the energy consumption for both apps accounting for CPU usage only.

We ran *PowerTutor* app to monitor the power consumption of all the apps on the phone. Then, we performed 20 taps with our app against the NFC reader, and then we performed 20 taps with *NFCtools* against an NFC tag. We observed that our app consumes 0.2 J of energy per tap compared to 0.13 J of energy per tap by *NFCtools*. This shows that our system is lightweight as it only uses an additional 0.07 J of energy for the sensor recordings.

7. ACTIVE ADVERSARIAL ATTACKS

From the analysis presented in Section 6, we can see that our approach is robust and can authenticate users with a high accuracy. That is, the approach can be effectively used to differentiate one user from the other. However, it is possible that the attacker may deliberately attempt to mimic the tapping gesture exhibited by a victim user. In this section, we assess our tap biometrics system against such an active adversary.

If the attacker tries to authenticate himself as the victim user, he has to move his hand in such a way that his hand motion as sensed by different sensors correlates significantly with the tapping gesture exhibited by the legitimate user. Even when the attacker observes how the user taps, it may still be difficult for the attacker to reproduce the tapping gesture as our gesture is sensed by multiple sensors and *all* of the sensor values should match with the user's template. Mimicking multiple sensor events simultaneously would be harder for the attacker and so our approach should provide better resistance to active attacks compared to systems that use single or fewer sensors. While robotic attacks have been reported against other authentication systems (such as the one developed in [38], such attacks will not apply to our system since authentication is to be performed by a real human user in the presence of retail personnel and using a robot to make a purchase at the terminal would clearly raise a suspicion.

We proceeded to evaluate the robustness of our system against human-based observation and active adversarial attacks. For our evaluation, we designed an active attack that aimed at maximizing the attacker capabilities in defeating

[2]https://play.google.com/store/apps/details?id=edu.umich.PowerTutor

[3]https://play.google.com/store/apps/details?id=com.wakdev.wdnfc

our system. If our system could defeat this attacker, it could also defeat other weaker attackers. To this end, we asked one of our users to serve the role of the victim while a researcher served the role of an expert attacker. The victim and the attacker had similar body structures, i.e., their height and weight were similar, which would have facilitated the attacker to better mimic the victim's tapping gesture. We asked the victim user to perform his tapping for 30 times in each of the four reader orientation scenarios while the attacker recorded a clear video of him tapping. After the total 120 taps were collected from the victim, we built the classifier for this user following the procedure described in Section 6. The attacker then closely watched the previously recorded video and practiced to re-create the victim's tapping gesture against a dummy reader several times while receiving a feedback from his friend (a colluding attacker). This simulated the attacker's training phase performed at home (i.e., not at the retail store in the presence of the authentication terminal/reader). Finally, during the actual attack phase, the attacker performed 20 taps and each of these taps was tested against the victim's classifier built earlier.

The success rate of this active attacker against our authentication systems is shown in Table 3. From these results, we can claim that even when an attacker practices and mimics the hand motion of the victim, he cannot succeed. Also, we can claim that we have a strong attacker as the attacker is fully trained watching the victim's tap video recording and getting feedback from a colluding attacker. Moreover, the victim we chose matched the body structure of the attacker which may further facilitate the attack. Since our system can defeat such strong attacker, it can, therefore, defeat attacks in other scenarios where the victim's structure is different from the attacker's and/or where the attacker cannot fully observe the victim and train.

8. DISCUSSION

8.1 Defeating Unauthorized NFC Reading

In our approach, the NFC transaction will not be processed until the user is authenticated. Hence, our approach can serve as a defense mechanism to unauthorized reading of NFC information as well as relay attacks [34, 35] where an unauthorized reader in close physical proximity of the phone tries to leech the NFC information and perform fraudulent transaction.

8.2 Dealing with Authentication Errors

Our system is robust and has very low error rates as shown in the Section 6. However, our machine learning approach learns from the data that user has provided during the training phase. If the user trains the classifier with tap gestures from one hand and later taps using the other hand, the system may fail to authenticate the legitimate user. Such cases can be avoided by either training the classifier with both hands or user switching back to correct hand while tapping. Further study would be needed to analyze the handedness of the users and their tapping behavior.

In situations where an authentication error (false negative) occurs, i.e., when the legitimate user is denied the transaction, we may need a fallback approach. In the traditional payment approach, for example, when the access is denied once due to an error, the user has to swipe the card again. We can follow the same model by requesting the user to tap the phone again to the NFC transaction terminal. If this process fails again, the device may not belong to the user and the transaction can be blocked. As a fallback strategy, the user may be authenticated using PINs or fingerprints.

8.3 Power Efficiency

One of the design goals of our system is to be light-weight as high power consumption may reduce the usability of the system. Since the authentication procedure in our approach lasts for no more than few seconds, our approach is quite power efficient and light-weight as shown in Section 6.4. The sensors are activated as soon as the NFC transaction app is turned on and are deactivated as soon as the corresponding NFC message is received. Only those sensor data which falls within the considered time window (up to 3 seconds in our case) will be used by the classifier to make the authentication decision. The detection approach itself is very lightweight and requires negligible amount of power.

8.4 User Transparency

Our approach is triggered as soon as the app, which needs to process the NFC transaction, is turned on. The sensor data is recorded from the start of that app to the point when either the user has been authenticated or denied. This entire process of authentication is transparent to the legitimate user. This property satisfies one of our design goals of being transparent and having a consistent usage model to existing NFC usage.

9. RELATED WORK

There exists prior work that aims at improving security based on sensors and sensor data. In this section, we review some of this prior work relevant to our paper that utilizes on-board sensors to improve the security of authentication and authorization.

Shrestha et al. [41] proposed authorizing an app utilizing hand movement of the user for calling, snapping, and NFC tapping. In their work, the system tries to identify a corresponding gesture as calling, snapping or tapping when an app requests for a permission to access these sensitive resources. They use machine learning classifiers utilizing different sensors to identify if the hand motion matches with the corresponding gesture. In contrast to this work, we use hand motion as a biometric measure to authenticate the users while they use the motion to distinguish the call, snap or tap gesture with other gestures. Their work aims at preventing against malware which is trying to access the sensitive services without user awareness, but it cannot protect in case of theft or device misuse by other users, which is what our work is geared towards.

The work by Conti et al. [10] is well-aligned with ours. They authenticate users by analyzing the hand movements while making/answering phone calls. They investigated if such motion can be used as biometric authentication measure. They have used Dynamic Time Warping (DTW) algorithm to analyze and detect the gesture of making/answering phone calls in contrast to our approach where we use machine learning classifiers to identify if the tapping gesture is performed by the owner. They have only considered accelerometer and orientation sensors while our approach considers ten different sensors including not only motion and position sensors but also environment sensor (ambient pressure) for better accuracy. Although our approach may be

used to authenticate the users while making/answering the phone calls as well, we focus on tap gesture biometrics, especially when user wants to make transaction with an NFC-enabled device.

Hong et al. [19] propose Waving Authentication (WA), a biometric authentication based on waving hand along with the phone. WA utilizes accelerometer sensor to extract 8 features, train SVM classifiers to build a model and authenticate users with this model. However, this approach is not transparent to the user, unlike our method.

Gascon et al. [17] have analyzed typing motion behaviour of the user to continuously authenticate a user on smartphones. It records the touch input along with the timestamps when the keys are pressed or released. They use different sensors such as accelerometer, gyroscope, and orientation sensors and extract 2376 dimensional vector representing the typing motion behaviour of the user. They use linear Support Vector Machine (SVM) classifier to identify if the typing motion belongs to user or not. Other work [11, 16, 36, 40] share the similar philosophy to authenticate users based on the touch gesture. They either use only the touch sensor or use the touch sensor in conjunction with different inertial sensors.

Some of the other work that focus on behavioral biometric measure to authenticate users, include, but not limited to the voice pattern recognition [27, 49], the walking pattern [12, 26], and the tapping pattern [5, 28, 46]. To authenticate with the voice pattern or the tapping pattern on touch screen requires users to perform extra action while authenticating themselves. Although these are biometric measures to authenticate users, these are not transparent when a user is trying to make an NFC transaction with the phone and, hence, adds extra burden to users.

10. CONCLUSION AND FUTURE WORK

In this paper, we presented an approach to authenticate a user transparently before making an NFC transaction. The approach captures the user's hand movement and identifies the user based on the sensor data recorded by the device. The gesture is very unique to the user and is difficult for the attacker to mimic. We presented the design and implementation of the proposed authentication approach. Our results suggest that our approach could be very effective in authenticating users and preventing misuse of NFC services in case of theft or loss of NFC phone, without necessitating any additional user burden.

Our future effort will be focused on exploring new features from the available sensors as well as utilizing new sensors as they are introduced by the phone manufacturers and the operating systems. We also plan to further evaluate our approach with different smartphone models and a wider range of users. Moreover, to further improve the authentication accuracy, we plan to use a smartwatch (when available) along with the smartphone for detecting the tapping gesture from the user. Using multiple devices may provide a broader range of features (e.g., wrist movements and hand movements) which may increase the accuracy of the system and further reduce the chances for the attacker to mimic the gesture.

Acknowledgments

We would like to thank ACSAC 2016 anonymous reviewers for their useful feedback. We would also like to thank all of our participants at UAB and Aalto university for participating in our biometrics study. This work has been funded by NSF CNS-1526524 grant.

References

[1] A. Adams and M. A. Sasse. Users are not the enemy. *Commun. ACM*, 42(12), 1999.

[2] Android. Android âĂŞ android pay. Available online at https://www.android.com/pay/.

[3] Apple. Apple pay - apple. Available online at http://www.apple.com/apple-pay/.

[4] ARM. ARM Security Technology Building a Secure System using TrustZone Technology. Technical report, April 2009.

[5] S. Azenkot, K. Rector, R. Ladner, and J. Wobbrock. Passchords: secure multi-touch authentication for blind people. In *Proceedings of the 14th international ACM SIGACCESS conference on Computers and accessibility*, pages 159–166. ACM, 2012.

[6] R. Boden. Yale introduces residential deadbolt with nfc unlocking. Available online at http://www.nfcworld.com/2015/01/06/333372/yale-introduces-residential-deadbolt-nfc-unlocking/.

[7] J. L. Boyles, A. Smith, and M. Madden. Privacy and data management on mobile devices. Available online at http://www.pewinternet.org/2012/09/05/privacy-and-data-management-on-mobile-devices/.

[8] C. Busold, A. Taha, C. Wachsmann, A. Dmitrienko, H. Seudié, M. Sobhani, and A.-R. Sadeghi. Smart keys for cyber-cars: secure smartphone-based nfc-enabled car immobilizer. In *Proceedings of the third ACM conference on Data and application security and privacy*, pages 233–242. ACM, 2013.

[9] ConsumerReports.org. 3.1 million smart phones were stolen in 2013, nearly double the year before: Consumer reports. Available online at http://pressroom.consumerreports.org/pressroom/2014/04/my-entry-1.html.

[10] M. Conti, I. Zachia-Zlatea, and B. Crispo. 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*, 2011.

[11] A. De Luca, A. Hang, F. Brudy, C. Lindner, and H. Hussmann. Touch me once and i know it's you!: Implicit authentication based on touch screen patterns. In *SIGCHI Conference on Human Factors in Computing Systems*, CHI, 2012.

[12] M. O. Derawi, C. Nickel, P. Bours, and C. Busch. Unobtrusive user-authentication on mobile phones using biometric gait recognition. In *Intelligent Information Hiding and Multimedia Signal Processing (IIH-MSP),*

2010 Sixth International Conference on, pages 306–311. IEEE, 2010.

[13] K. Dyer. Czech firm releases universal nfc id system. Available online at http://www.nfcworld.com/2013/06/19/324720/czech-firm-releases-universal-nfc-id-system/.

[14] S. Egelman, S. Jain, R. S. Portnoff, K. Liao, S. Consolvo, and D. Wagner. Are you ready to lock? CCS '14, 2014.

[15] L. Finžgar and M. Trebar. Use of nfc and qr code identification in an electronic ticket system for public transport. In *Software, Telecommunications and Computer Networks (SoftCOM), 2011 19th International Conference on*, pages 1–6. IEEE, 2011.

[16] M. Frank, R. Biedert, E. Ma, I. Martinovic, and D. Song. Touchalytics: On the applicability of touchscreen input as a behavioral biometric for continuous authentication. *IEEE Transactions on Information Forensics and Security*, Jan 2013.

[17] H. Gascon, S. Uellenbeck, C. Wolf, and K. Rieck. Continuous authentication on mobile devices by analysis of typing motion behavior. In *Sicherheit*, 2014.

[18] J. Gautam, Y. Kumar, and A. Gupta. Existing scenario of near field communication in transport sector. In *Signal Processing and Integrated Networks (SPIN), 2014 International Conference on*, pages 327–332. IEEE, 2014.

[19] F. Hong, M. Wei, S. You, Y. Feng, and Z. Guo. Waving authentication: Your smartphone authenticate you on motion gesture. In *Proceedings of the 33rd Annual ACM Conference Extended Abstracts on Human Factors in Computing Systems*, pages 263–266. ACM, 2015.

[20] S. Huopio. Biometric identification. *Eye*, 3(1), 1988.

[21] A. Jain, L. Hong, and S. Pankanti. Biometric identification. *Communications of the ACM*, 43(2):90–98, 2000.

[22] T. Kasper, I. von Maurich, D. Oswald, and C. Paar. Chameleon: A versatile emulator for contactless smartcards. In K.-H. Rhee and D. Nyang, editors, *Information Security and Cryptology - ICISC 2010*, volume 6829 of *Lecture Notes in Computer Science*, pages 189–206. Springer Berlin Heidelberg, 2011.

[23] H. Li, D. Ma, N. Saxena, B. Shrestha, and Y. Zhu. Tapwave-rub: Lightweight malware prevention for smartphones using intuitive human gestures. In *Conference on Security and Privacy in Wireless and Mobile Networks (WiSec)*, pages 25–30, 2013.

[24] Lookout.com. Phone theft in america. Available online at https://www.lookout.com/resources/reports/phone-theft-in-america.

[25] P. Macgougan. Asia pacific to lead growth in global mobile payments. Available online at http://go-mashmobile.com/mcommerce-2/news-mcommerce-2/asia-pacific-to-lead-growth-in-global-mobile-payments-2828/.

[26] J. Mäntyjärvi, M. Lindholm, E. Vildjiounaite, S.-M. Mäkelä, and H. Ailisto. Identifying users of portable devices from gait pattern with accelerometers. In *Acoustics, Speech, and Signal Processing, 2005. Proceedings.(ICASSP'05). IEEE International Conference on*, volume 2, pages ii–973. IEEE, 2005.

[27] J. A. Markowitz. Voice biometrics. *Communications of the ACM*, 43(9):66–73, 2000.

[28] D. Marques, T. Guerreiro, L. Duarte, and L. Carriço. Under the table: tap authentication for smartphones. In *Proceedings of the 27th International BCS Human Computer Interaction Conference*, page 33. British Computer Society, 2013.

[29] R. Maxion, K. S. Killourhy, et al. Keystroke biometrics with number-pad input. In *Dependable Systems and Networks (DSN), 2010 IEEE/IFIP International Conference on*. IEEE, 2010.

[30] R. McMillan. Google declares war on the password. Available online at http://www.wired.com/2013/01/google-password/all/.

[31] R. Morris and K. Thompson. Password security: a case history. *Commun. ACM*, 22(11), 1979.

[32] S. Müller. On the security of an rsa based encryption scheme. In *Information Security and Privacy*, pages 135–148. Springer, 1999.

[33] N. L. Petroni, Jr. and M. Hicks. Automated detection of persistent kernel control-flow attacks. In *Conference on computer and communications security*, 2007.

[34] M. Roland, J. Langer, and J. Scharinger. Practical attack scenarios on secure element-enabled mobile devices. In *Near Field Communication (NFC), 2012 4th International Workshop on*, pages 19–24. IEEE, 2012.

[35] M. Roland, J. Langer, and J. Scharinger. Applying relay attacks to google wallet. In *Near Field Communication (NFC), 2013 5th International Workshop on*, pages 1–6. IEEE, 2013.

[36] N. Sae-Bae, K. Ahmed, K. Isbister, and N. Memon. Biometric-rich gestures: A novel approach to authentication on multi-touch devices. In *Conference on Human Factors in Computing Systems*, 2012.

[37] Samsung. Samsung pay - safe and simple mobile payments. Available online at http://www.samsung.com/us/samsung-pay/.

[38] A. Serwadda and V. V. Phoha. When kids' toys breach mobile phone security. In *Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security*, pages 599–610. ACM, 2013.

[39] A. Seshadri, M. Luk, N. Qu, and A. Perrig. Secvisor: a tiny hypervisor to provide lifetime kernel code integrity for commodity oses. In *Symposium on Operating systems principles*, SOSP, 2007.

[40] M. Shahzad, A. X. Liu, and A. Samuel. Secure unlocking of mobile touch screen devices by simple gestures: You can see it but you can not do it. In *Proceedings of the 19th annual international conference on Mobile computing & networking*, pages 39–50. ACM, 2013.

[41] B. Shrestha, M. Mohamed, A. Borg, N. Saxena, and S. Tamrakar. Curbing mobile malware based on user-transparent hand movements. In *Pervasive Computing and Communications (PerCom), 2014 IEEE International Conference on*, pages 221–229. IEEE, 2015.

[42] B. Shrestha, N. Saxena, and J. Harrison. Wave-to-access: Protecting sensitive mobile device services via a hand waving gesture. In *Cryptology and Network Security (CANS)*. 2013.

[43] C. Sorrel. Lockitron: Unlock your home with your cellphone. Available online at http://www.wired.com/2011/05/lockitron-unlock-your-home-with-your-cellphone/.

[44] R. Steffen, J. Preißinger, T. Schöllermann, A. Müller, and I. Schnabel. Near field communication (nfc) in an automotive environment. In *2nd International Workshop on Near Field Communication*, pages 15–20. IEEE, 2010.

[45] L. Tung. Google looks to ditch passwords for good with nfc-based replacement. Available online at http://www.zdnet.com/article/google-looks-to-ditch-passwords-for-good-with-nfc-based-replacement/.

[46] J. O. Wobbrock. Tapsongs: tapping rhythm-based passwords on a single binary sensor. In *Proceedings of the 22nd annual ACM symposium on User interface software and technology*, pages 93–96. ACM, 2009.

[47] R. V. Yampolskiy and V. Govindaraju. Behavioural biometrics: a survey and classification. *International Journal of Biometrics*, 1(1):81–113, 2008.

[48] J. Yan, A. Blackwell, R. Anderson, and A. Grant. Password memorability and security: Empirical results. *IEEE Security and Privacy*, 2(5), 2004.

[49] H. Yang, Y. Xu, H. Huang, R. Zhou, and Y. Yan. Voice biometrics using linear gaussian model. *Biometrics, IET*, 3(1):9–15, 2014.

[50] L. Zhang, B. Tiwana, Z. Qian, Z. Wang, R. P. Dick, Z. M. Mao, and L. Yang. Accurate online power estimation and automatic battery behavior based power model generation for smartphones. In *Proceedings of the eighth IEEE/ACM/IFIP international conference on Hardware/software codesign and system synthesis*, pages 105–114. ACM, 2010.

APPENDIX

A. RESULTS FOR OTHER DURATIONS

In this section, we show the performance of the classifiers for generalized as well as for different scenarios using 10-Fold cross validation for two and three seconds duration of tap gesture. The result for two seconds duration of tap gesture is summarized in Table 4 while that for three seconds duration of tap gesture is summarized in Table 5. The accuracy did not change significantly even when longer duration than one second of tap gesture was used to classify. Moreover, the performance of the classifier dropped in some occasions when longer duration was used.

Table 4: The results for Generalized and Scenario-specific models for tapping duration of two seconds. Each column shows average (Avg) and standard deviation (S.D.) for F-Measure, Recall and Precision. Precision and recall captures the security and the usability of the system, respectively. F-measure accounts for both precision and recall.

	F-Measure	Recall	Precision
	Avg (S.D.)	Avg (S.D.)	Avg (S.D.)
Generalized	0.93 (0.05)	0.95 (0.04)	0.9 (0.07)
Chest-Angular	0.89 (0.06)	0.93 (0.05)	0.86 (0.09)
Waist-Flat	0.92 (0.05)	0.95 (0.06)	0.90 (0.06)
Chest-Vertical	0.91 (0.04)	0.93 (0.04)	0.89 (0.05)
Waist-Angular	0.89 (0.07)	0.92 (0.06)	0.86 (0.08)

Table 5: The results for Generalized and Scenario-specific models for tapping duration of three seconds. Each column shows average (Avg) and standard deviation (S.D.) for F-Measure, Recall and Precision. Precision and recall captures the security and the usability of the system, respectively. F-measure accounts for both precision and recall.

	F-Measure	Recall	Precision
	Avg (S.D.)	Avg (S.D.)	Avg (S.D.)
Generalized	0.92 (0.06)	0.96 (0.03)	0.89 (0.08)
Chest-Angular	0.90 (0.05)	0.93 (0.06)	0.87 (0.05)
Waist-Flat	0.90 (0.06)	0.93 (0.05)	0.87 (0.08)
Chest-Vertical	0.90 (0.06)	0.93 (0.05)	0.88 (0.08)
Waist-Angular	0.89 (0.06)	0.91 (0.07)	0.87 (0.07)