Using In-Built Android Sensors to Map Physical Actions for NFC Security

Word Count: 4565

April 15, 2017

Nikhil Sharma

Abstract

NFC (Near-field Communication) is a set of communication protocols that allow two devices to send and receive data, like Bluetooth and WiFi-Direct functionalities. Initially launched in 2006, NFC allows two devices within 3-5 cm range to wirelessly pair and is used for transferring files such as pictures, videos, and other documents. According to HIS Technology, it is estimated by 2018 that there will be 844 million Android phones with the NFC chip in-built. Despite the wide-use of the protocols, there are emerging vulnerabilities within the technology that allow hackers to gain access into devices and interrupting the NFC transfer. Such vulnerabilities include NFC Data Corruption/Manipulation and Interception Attacks, both resulting in hackers hijacking the NFC transfer midway and inflicting harm to both devices. The current focus on strengthening NFC involves additional encryption to the communication protocol and the data transferred. A new focus is emerging throughout mobile phone security which involves the use of in-built sensors to secure the device and authenticate users attempting to gain access. Examples of this can be seen in fingerprint scanners, facial and voice recognition locks, and gait analysis. The goal of this project is to collect data from the in-built sensors within an Android device during various physical movements and establish the effectiveness of using the gathered data to authenticate a NFC transfer.

A Samsung Galaxy S7 was chosen to gather the sensor data from and to evaluate a potential NFC-sensor authentication mechanism. From the device, the accelerometer, gyroscope, light and proximity sensor were chosen to serve as the baseline for authentication. These sensors are all environment-dependent and display appropriate data variation relative to the short range of NFC transfers. After selecting the sensors, a program was developed in Android Studio that gathered sensor data from various physical movement such as flipping the phone upside down

rapidly, tilting the sensor forwards and backwards, and rotating the device at various angles. The sensor data was exported, analyzed, and fitted into a baseline so that a specific data subset could correlate to a specific movement. If a specific collection of data points could correlate to a specific action, then the receiving NFC phone could establish an action to be performed by the user on the sending NFC phone and only allow communication should the correct sensor data be read. The results show that data subsets could be isolated from the sensors during physical movements for all but 2 physical actions, flipping the phone face-up and flipping the phone face-down. This research project proved that in-built sensor can be used to authenticate NFC transfers using a variety of physical movements, however there would be a relatively small pool of physical actions to choose from and the range of motion would be limited.

Introduction

In a cybercriminal age where the focus has shifted to international corporations' and companies' firewalls being breached and exploited on a weekly basis, the protection provided to the average commercial smartphone device is under-satisfactory. In these smartphone devices, wireless file transferring infrastructures- such as Bluetooth, Wifi-direct, or NFC- are a major feature used by consumers to transfer files wirelessly from device to device easily. Unaware to the average consumer, devices like Wifi Pineapples, Stingray phone trackers, and other cellular network/wireless communication intercepting devices can be used to intercept the transfer of wireless information (Mulliner). These devices fool phones into routing the wireless transfer through the hacking device and can alter, delete, or attach malware to the existing wireless file transfer without the unsuspecting user's knowledge. Currently, there is no free protection against these hacking devices for the average non-military phone, and these devices are suspected in massive data dump attacks such as the News of the World hacking scandal (NowSecure). Use of

the spoofing devices by hackers in their local area is virtually undetectable, as users often associate the error in file transfer with “software glitches” or “the phone’s being stupid” (Snell). The best advice offered currently against these hackers for commercial smartphone users is constant vigilance over one’s smartphone and avoidance of unsecure networks, sound advice but impractical to achieve.

However, studies on the growth of cellular devices and their internal components show that there are in-built sensors found within 96% of current smartphones distributed globally (Ali). These sensors include accelerometers, gyroscopes, light, magnetic, and orientation-focused sensors. Current uses of these sensors include detecting when the phone has been rotated (orientation and gyroscope), the speed of a vehicle with the phone inside (accelerometer) and GPS location (magnetic) (Ali). My research topic focuses on the use of these easily accessed inbuilt sensors to improve smartphone security, specifically the NFC wireless protocol available on Android devices. The focus of the research is on these smartphone sensors and determining if they can be used to help improve the wireless transferring interfaces by utilizing the sensor data between the devices to authenticate the file transfer. Demonstratively speaking, for a file (music, picture, video, documents, etc) to be sent from one phone to another device, the user would be prompted to perform a randomly generated action with their phone in hand, such as quickly turning the phone upside down, before the file is transferred. This additional authentication step would safeguard against the hacking/spoofing devices mentioned earlier as any actions performed by the hacking devices would return in different sensor data from what is required, instantly drawing a red flag.

I believe this investigation into the uses of smartphone sensors to protect NFC transmission is needed because of the previously mentioned spread of cybercrimes and hackers

with the spoofing devices and malicious intents. Additionally, as the growth of technology continues in the upcoming years, wireless transmission of information and files will be key in efficiency and ease of access. Rather than carrying around USB cables and other obsolete wires, cell phones will incorporate more wireless transmission interfaces, as we have already seen the industry start investing in (Hao). Addressing and procuring solutions to the problem of unsecured transmission of wireless data between devices will be crucial before the implementation of new wireless interfaces to avoid major cybercrimes. Regardless of the sensors chosen for this project, the methodology used and sensor data gathered will contribute to future sensor-based solutions.

Introduction to NFC Technology & Vulnerabilities

NFC protocol is a method of wireless/contactless communication between two devices that allows the devices to share data without physically touching (Coskun). It is used widely across the world to conveniently transfer files between devices and is emerging as a payment option at commercial stores, prompting users to simply wave their phone over NFC scanners and pay for their items. The pairing of devices through NFC requires one of the devices to be the active device (also known as the reader or interrogator) which creates a radio frequency current that communicates with the other NFC compatible device (Mesropyan). Currently, the range on NFC connections varies between 5-10 cm but upgrades to the existing technology could extend this range to over 20 cm. Peer-to-peer communication through two active devices is established when a user wishes to transfer one file to another user's device. NFC is a relatively recent development to wireless communication as it was conceived in 2004 through the NFC Forum and was first implemented in 2006 within the Nokia 6131. While the relatively short-range of NFC protects it from long-range vulnerabilities, there are still flaws within the protocol that can be exploited by hackers to gain access to the device. NFC exploits can be broken down into three

broad categories: Eavesdropping, Data Corruption and Manipulation, and Interception Attacks. Eavesdropping attacks occur when an unauthorized device can listen into the NFC connection and extract the data being transferred without the users ever knowing. Data Corruption and Manipulation and Interception Attacks are based off the same principle and occur when a malicious user interferes with the data being sent mid-transfer. The hacker can render the data corrupt and useless to the receiving device, or they can inject their own malware so the receiving device gets infected and infects future devices it connects with. Throughout 2014 and 2015, there were over five thousand reported instances in which stores discovered malware within their NFC scanners which could have impacted hundreds of thousands of customers and infected their devices (Arcese).

Focusing on the Android Platform and Samsung Galaxy S7

NFC technology was predominately available in the Android platform and was relatively recently unveiled in the Apple iPhone, starting from the 2014 release of the iPhone 6. No generation of Apple tablets, the iPads, have the NFC technology within them. The Android platform has supported NFC since 2010, when the Samsung Nexus S was released. Due to Android's longer history with NFC, it was more appealing to incorporate the Android platform within the study of securing NFC connections. However, this was not the only factor that influenced my decision to exclude the Apple iOS operating system. My research also focuses on extracting sensor data from commercial phones and analyzing that data. It became abundantly clear that within the timeline (a period of a few months), it would be more efficient to use Android's open-source and well documented Android Studio platform to develop the data-extracting application rather than using Apple's proprietary application, Swift. Specifically, within the Android platform, the Samsung Galaxy S7 was chosen as the test phone from which

the sensor data would be extracted from and the NFC authentication developed for. The S7 model was chosen because it represented a good aggregate model within the Android framework as it is just over a year old per release date. The one year lifespan of the S7 means that there was time to address any NFC glitches and correct them (as opposed to any newer devices released) and that there will be a multitude of modern in-built sensors that can be used for data extraction.

In-Built Sensors Chosen to Target for Authentication

Within the Samsung Galaxy S7, there are numerous in-built sensors that are used intrinsically within the device for authentication. The fingerprint sensor can be used for user authentication to password-protect the device or any apps. The camera and sound sensors can be used for facial or voice recognition in many apps. Within my hypothesis, a physical motion would have to be conducted with the phone in hand so that the sensor data matches a baseline set for that specific motion. Once the sensor data is matched and authenticated, hypothetically, the NFC transmission would be allowed to continue. When deciding the in-built sensors to study, it is important to consider the relatively short range of NFC transmissions. Therefore, not all sensors are viable within this study and only those offering varying values even when a few centimeters apart are viable. For this reason, a sensor such as the temperature sensor would not be chosen because a difference of a few centimeters would not significantly vary the temperature reading. However, the accelerometer, gyroscope, light, and proximity sensor are all sensors within the Samsung Galaxy S7 that read environmental data liable to change over short distances. As a result, these four sensors were chosen for the study and the goal was to map each sensor to a specific movement conducted by a user with the phone in hand.

Literary Review of Similar Approaches and Methods

Sensors and Mobile Phones: Evolution and State-of-the-Art

This paper within the *Pakistan Journal of Science* examined a study at the Department of Computer Science at the University of Peshawar in which various professors extracted sensor data from a multitude of in-built sensors from commercial phones. The authors discuss the widespread availability of sensors within modern smartphone devices, approximating over 1 billion Android devices worldwide with an array of in-built sensors. The researches also test and collect data from various sensors found within both Android and Apple devices to measure the degree of accuracy within these sensors, providing the spread in error of the data collection from the sensors. This source proved that there are certain sensors such as temperature and pressure which remain relatively constant with short displacements, which would not be useful in NFC authentication due to the short range. This study is objective regarding sensors and the varying degrees of accuracy of them, with the calculations and methodology provided to the reader for verification purposes as well.

Authenticating a Mobile Device's Location Using Voice Signatures

Another study examined the findings of five researchers at Columbia University in their attempts to implement a sensor-based authentication system. In this experiment, researchers use data from a phone's GPS as well as the voice sensors (microphone) to analyze the speaking habits of users. In addition, the study displays the formulas and adjustments made to the incoming data to process it against the authoritative algorithm. While the findings don't directly tie into NFC sensor-based authentication in terms of sensors used, the authorization algorithm that is used as well as the corrections made to the incoming raw data provide a baseline for this

project's sensor data. The methodology used is provided within the study, and the calculations used for margin of error within the actual algorithm can be configured for this research project involving the Android phone's sensors.

To verify the authenticity of the study, a connection between the conducting professors and HP had to be explored to verify that there was no conflict of interest between the two. HP was not involved within this study, rather the laboratory that the study was conducted in was merely named after its sponsor (which happened to be HP). HP and these scientists had no prior connections, and HP has not commented on the results of this study to this time of writing.

Multi-sensor authentication to improve smartphone security

This source proved to be the closest in resemblance to this paper's research topic. This source, published by researchers from Princeton University, analyzes how modern sensors found within smartphones can be utilized for smartphone security. My research topic narrows the scope to Android phone sensors only, as well as providing the specific smartphone security feature, NFC. This source proved excellent as a broad indication of the security-related functionalities of modern sensors and applications of specific sensors-such as the accelerometer or gyroscope-towards security. Additionally, this source provides the data and graphs collected from the study to analyze each sensor's accuracy and potential to convey data values outside the acceptable margin of error. This provides an indication of magnitude for statistical Type I error, where the sensor data recorded values *not* within the acceptable margin, but the action performed with the sensor was correct.

The researchers conducted the study in accordance with the Department of Electrical Engineering at Princeton, at the Princeton Architecture Lab for Multimedia and Security. The

setting ensures that the data from the sensors within the study was accurate and there was no presence of lurking variables.

Vulnerability Analysis and Attacks on NFC-Enabled Mobile Phones

The research project focuses on vulnerabilities of NFC transmissions and the effectiveness of using in-built sensors to authenticate transmissions. This specific source is a white paper published by a researcher at the Fraunhofer Institute for Secure Information Technology. This paper details the risks of various attacks and exploits within the NFC protocol and how malicious users can bypass NFC security measures. Specifically, interception and data corruption attacks are elaborated upon within this paper, portraying in non-technical terms how the attacks are conducted and what level of intrusion the attacks result in. This precise information of the vulnerabilities of NFC protocol is crucial in understanding how the in-built sensors would have to interface with the existing NFC connections to address the weaknesses.

Method

Throughout the research process, a mixed approach with a focus on a scientific experimental approach was taken. Initially, the focus was on descriptively analyzing various sensors found within a variety of Android phones, using the modern Samsung Galaxy S7 as the base comparison. When choosing the sensors, a list of all sensors available within the S7 was compiled and categorized as motion-based, environment-based, or room/setting-based. Motion-based sensors had the highest variance and would change even with the slight movement of the phone. Environment-based sensors were dependent on the immediate environment of the phone's location, not as variant as the motion-based but better than the room/setting-based. The room/setting-based offer the least variance and tend to return the same data if the device is within the same room/location. These types of sensors are least helpful for the NFC authentication as

they would offer no variability to authenticate against. As previously stated, the sensors chosen were the accelerometer, gyroscope, light, and proximity sensors. The accelerometer and gyroscope both depend on the motion of the phone's axes which can correlate directly with a specific rotation/movement to be performed with the phone. The light and proximity sensors are both environmental sensors and the data will vary even a short distance apart. After having chosen the sensors, an Android app was developed that extracted the sensor data from the chosen sensors over a specific period correlating to a physical action with the device. Each physical action was performed thirty times under the same settings and variables to ensure data validity. After the sensor data is exported, statistical analysis was conducted for each physical action to correlate with a specific sensor data subset. Should a baseline for each physical action be established in relation to a sensor data subset, then the initial hypothesis will be proven correct and a mechanism can be proposed which would authenticate the NFC transfer with the in-built sensors.

The advantage of a mixed approach is that it allows for greater depth of analysis to be conducted during the descriptive approach and increased applications during the experimental approach. The findings and conclusions drawn from analyzing sensors found within Android phones will allow for efficient code and innovative methods of sensor authentication. The potential disadvantage of this research method is the risk of overextending and losing focus of the core topic while researching various sensors and studies. However, I believe this risk can be mitigated by simultaneously developing the authentication app along with the research. This will ensure my research is consistently focused on the end-goal of the experimental app development.

Ethical Considerations

This research does contain a demonstrative portion, which will require an Android phone for data validity. Due to the nature of programming required for the authentication app, it is possible that there could be damage done to the phones' internal software. To mitigate these concerns, phones of my own ownership and responsibility were used and were devoid of any personal or confidential information.

Physical Actions with Device

The physical actions to be performed with the device fell under three broad categories, each with its associated sensor(s). The three categories were: movement of the phone relative to the body, movement of the phone relative to the environment, and movement of the phone independent of other factors. Within movement of the phone relative to the body, there were actions such as placing the phone against one's chest (screen facing chest) and placing the phone face up on one's palm. The proximity sensor's data was utilized for this category due to nature of the actions. The next category was movement of the phone relative to the environment, targeting the in-built light sensor. Actions under this category included placing the phone in a dimly-lit setting (such as covering the phone with one's hand or place the phone face-down) and pointing the phone up to the sky. A possible limitation noted during testing was users may not always be in the proper setting to point the device up the sky, so the sky could be substituted for any light source. The final category of motion was movement of the device independent on external factors. This category targeted the gyroscope and accelerometer sensors, which measure rotational motion and acceleration respectively. These sensors do not rely upon environmental data, rather the motion of the actual device. Actions under this category included flipping the

phone from face-up to face-down, tilting the phone forwards and backwards (relative to the user), and moving the phone rapidly in a linear direction.

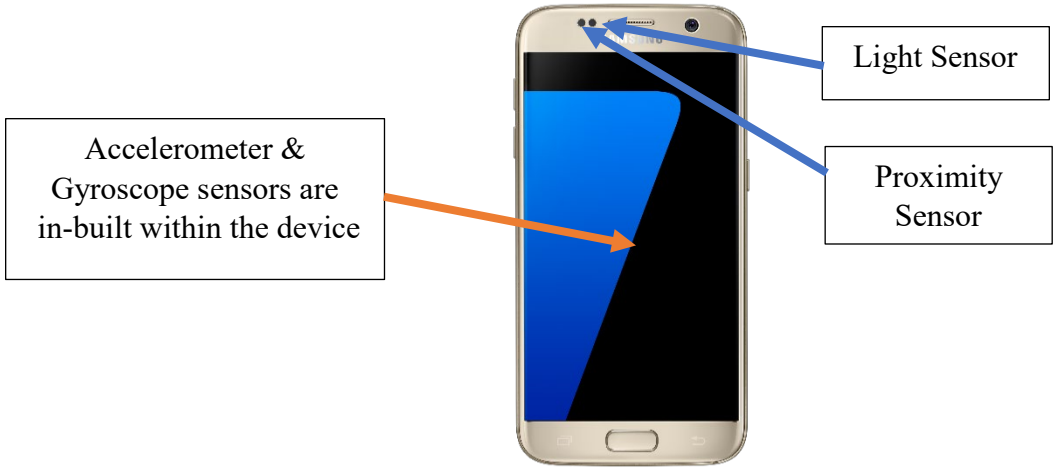
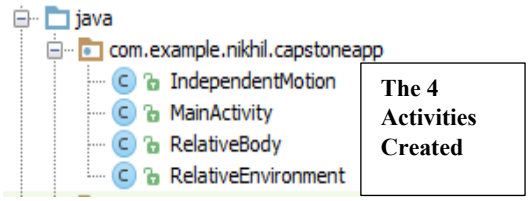


Figure A

Extraction of Sensor Data from Physical Activities

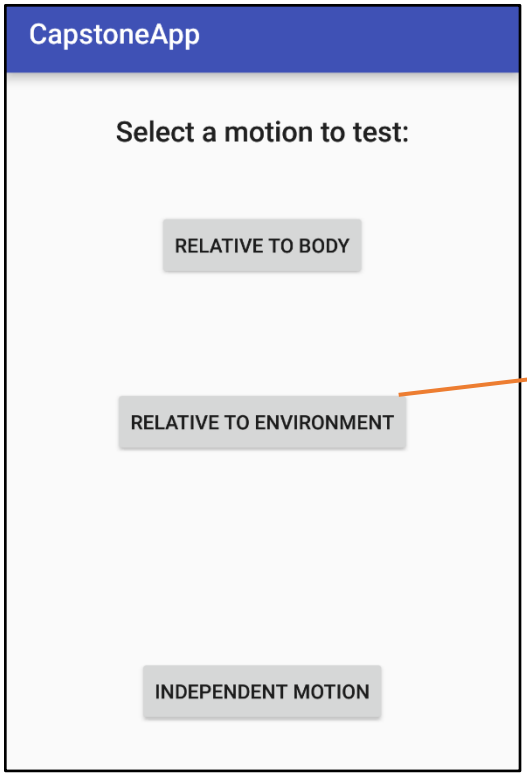
Sensor data extraction was done through Android Studio using open-source Java libraries for handling Android sensors. Open-source java libraries are pre-made code that software engineers at Google and Samsung made for developers to easily access device features, i.e. a phone's sensors. Android Studio is the official development environment for the Android platform and offers the most convenience and reliability in extracting data from Android devices.



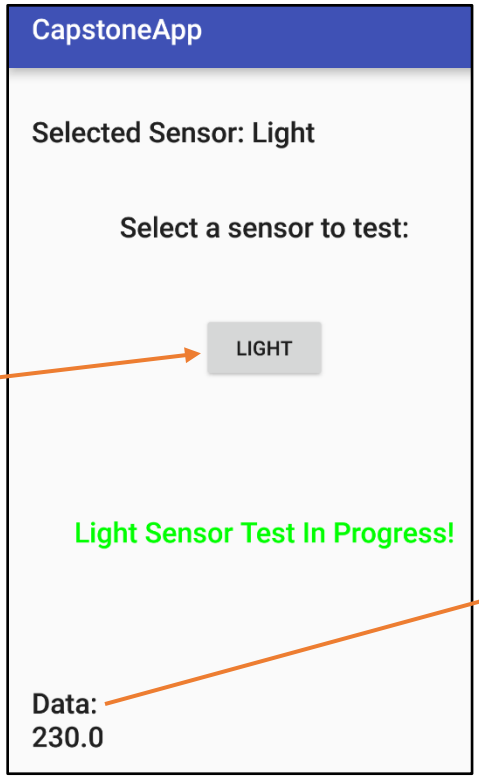
The 4 Activities Created

Within Android Studio, there are "activities" that allow for front-end User Interfaces, which is what the user sees when using the app, and there is back-end Java programming. The back-end Java programming is where the sensors are listed and the sensor data is extracted. For this project, there were a total of 4 activities, one activity for the home screen and three activities for each physical action type. The home screen activity, dubbed the "Main Activity" was the screen that contained buttons for each specific physical motion type. All other activities

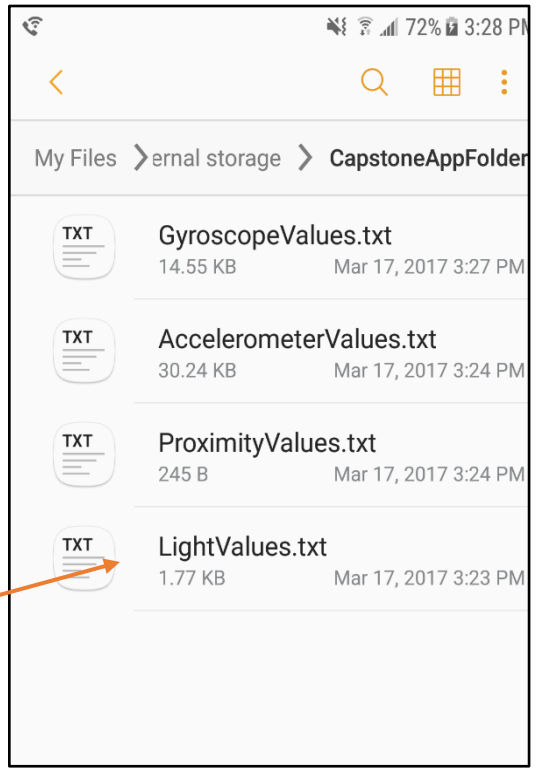
corresponded to a motion, the “IndependentMotion” activity handled the independent motion accelerometer and gyroscope sensors and the “RelativeBody” and “RelativeEnvironment” activities handled the proximity and light sensors respectively. The user interface was kept to the standard Android theme with only buttons and text labels implemented throughout the app. In the programming, each sensor had to be declared within the code and matched to the correct Android syntax, i.e. the proximity sensor’s specific syntax within Android Studio is “Sensor.TYPE_PROXIMITY”. Each sensor was mapped to a button and once the button corresponding to a sensor type was pressed, the SensorManager library would initiate a SensorManager variable, which controlled the specific sensor’s output. When declaring the sensor type, a delay of one second was specified so that the data readings would each be one second apart. This one second interval ensured that when the data readings were displayed in an Excel file, the data points would be sequential and evenly spaced out throughout the time interval of the physical motion. To obtain the data values, the SensorEventListener interface was implemented. This interface is another set of pre-made tools for Android Studio that allows the raw sensor data to be converted into integers and variables. In addition to being displayed on the screen of the phone, a text-file was created that would store the sensor values as well as a timestamp from when the value was taken. The timestamp allowed easy comparison of the values during the time range of the physical activity. Each physical action began with the phone placed upon a table face-up with the screen off. Each action was conducted 20 times. The path of extraction of sensor data from the phone to logging the values into a text-file can be seen in Figure B.



This is a screenshot of the Main Activity, which served as the home screen. Each button corresponds to a specific physical activity type, i.e. the "Relative to Environment" button is for the environment-dependent light sensor.



This is a screenshot of the "RelativeEnvironment" Activity. This was taken while data was being extracted. The data value shows up in the bottom left corner and gets logged to a text file.



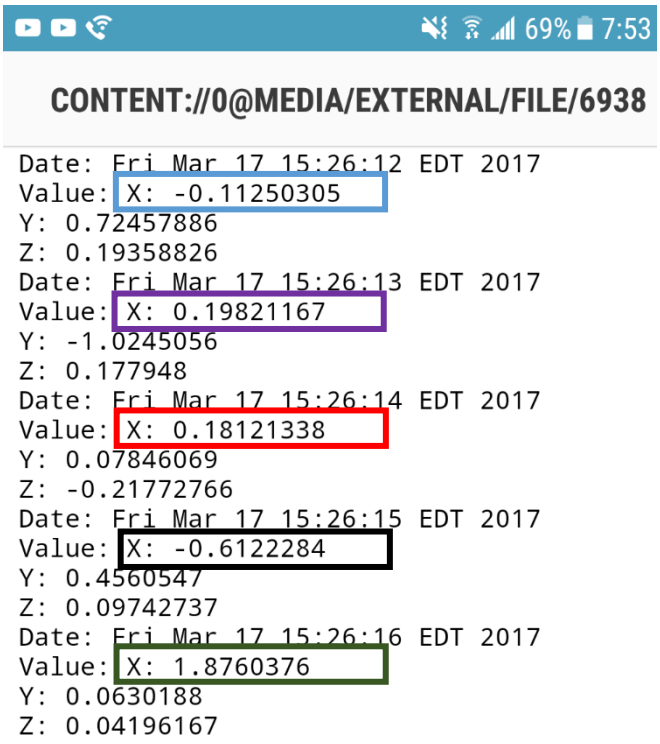
In addition to being displayed on the screen, the data values are also being logged in a text file within the phone. The data values from the text file were then converted into Excel and analyzed.

Figure B

Data Obtained and Results

The sensor data extracted from each physical motion was logged into a text file along with a timestamp. From this timestamp as well as timing the actual physical activity, a comparison could be drawn between moments throughout the physical activity and trends within the data. A chart summarizing the timestamps, trends, and physical activities can be found below as well a picture showing the conversion between raw sensor data and data analyzed within Excel.

Name of Physical Activity	**Type of Physical Activity**	**Average Duration of Physical Activity**	**Intended Sensor with Physical Activity**	**Trend/Baseline established with sensor data**
Placing phone against chest	Relative to Body	3 seconds	**Proximity**	Data value was 0
Placing phone in palm (face-up)	Relative to Body	2.5 seconds	**Proximity**	Data value was 8.000183
Placing phone face-down or in dimly lit setting	Relative to Environment	3.5 seconds	**Light**	Data value was <50
Placing phone facing sky/light-source	Relative to Environment	1.5 seconds	**Light**	Data value was >50
Flipping phone face-up	Independent to external factors	1 second	**Accelerometer**	**Inconclusive**
Flipping phone face-down	Independent to ext.	2.5 seconds	**Accelerometer**	**Inconclusive**
Tilting phone forwards (relative to user)	Independent to ext.	3.5 seconds	**Gyroscope**	Positive X-axis value
Titling phone backwards (rel. to user)	Independent to ext.	2 seconds	**Gyroscope**	Negative X-axis value
Moving phone rapidly in linear direction	Independent to ext.	1.5 seconds	**Accelerometer**	Positive X, Y, and Z-axes values



These two images show the process of converting the raw extracted sensor data into an Excel spreadsheet that can be analyzed to create data trends for each physical motion. From the raw sensor data's timestamp, a match could be made between the visible motion of the physical activity and the sensor data trend. The matching color boxes show the mapping from the raw sensor data into the worksheet. From comparing the timestamps with the visible motion, a trend could be established in relation to the sensor data. The following example shows the conversion between sensor data to sensor trend for one trial of the tilting phone forward (relative to user) physical action. From this instance, one can see that there is a positive X-axis sensor data trend when tilting the phone towards the user.

Action	**Sensor**	Sensor Data Time Total	
Tilting phone forwards (relative to user)	**Gyroscope**	4 seconds	
Time	**Visible Motion**	Sensor Data Value	**Sensor Data Trend**
0th second (starting)	Phone is lying face-up on stationary table	-0.11 X	Approximately 0 for each axis
1st second	User picks up phone and begins to tilt phone towards them slight	0.19 X	X value increases 0.2 °/s
2nd second	User stops tilting and holds phone stationary	0.18 X	All axes relatively constant
3rd second	User accidentally tilts phone backwards (heard wrong prompt)	-0.61 X	X value decreases by 0.7 °/s
4th second	User corrects themselves and tilts phone forward fully	1.87 X	X value increases by 2.4 °/s

Discussion of Data and Implementation of NFC-authentication

As the data above shows, a baseline and sensor data trend could be established for 7 out of the 9 physical motions. The two physical motions that could not be identified with sensor data were flipping the phone face-up and flipping the phone face-down. Originally, the accelerometer sensor was to be used solely to identify these two complementary motions. However, there could be no established data trend as the Y and Z axes would simultaneously fluctuate throughout the duration of the motion. This could be attributed to the quality of the accelerometer sensor located

within the Samsung Galaxy S7 or it could be a cause of numerous lurking variables include the one-year age of the phone or improper settings. Despite this, these two physical actions could be identified when the accelerometer and gyroscope sensors were both being used to measure sensor data. In the gyroscope sensor data, the Y-axis data would be negative when flipping the phone face-down and positive when flipping the phone face-up. The reason that the gyroscope data can't be used individually for these two physical motions is because the data values can be confused with tilting the phone forwards and backwards (relative to the user) as well. Only when comparing the fluctuating accelerometer's data with the gyroscope's Y-axis data can the physical motion of flipping the phone face-up and face-down be recognized. However, using two sensors to check a physical motion won't be practical in implementing within an NFC-authentication system. When authenticating the NFC file transfer, the sending device will have to simultaneously be extracting its own sensor data and sending that sensor data to the receiving device. From previous studies on modern Android phones, it is not possible to multi-extract sensor data while handling any other background connections (Lee, Wei-Han, and Ruby Lee).

However, there are still signs of progress from this research project. There are 7 physical motions that can be used as an authentication mechanism in the NFC-transfer. The mechanism would be implemented in a way such that before the NFC-communication is established between the two devices, the sending user will have to perform a randomly chosen physical action. Only when the sensor data matches the established sensor data trend will the NFC communication channel open between the two devices. Current limitations of such a mechanism include a small pool of physical actions to choose from a delay in NFC-transmission times with the inclusion of sensor data extraction. However, these are merely reflections on the data and could be proven wrong and the actual implementation of the NFC mechanism is out of this project's scope.

Conclusion and Further Developments

This study was initiated to identify in-built sensors within Android phones that could be used for NFC authentication through mapping physical actions against data trends and baseline values. The sensors identified were the accelerometer, gyroscope, light, and proximity sensors which all would be effective in the short-range variability of NFC transmissions. Physical actions were identified that would correlate to each sensor and could be used to authenticate the wireless transfers. The data showed that 7 out of the 9 physical actions proposed would be reliable authentication mechanisms, with the accelerometer being the least effective sensor and the proximity, light, and gyroscope sensors able to be mapped to the physical actions. While the under-100% result is enough to disprove my hypothesis that all the physical actions could be mapped to a single sensor data trend, the results of this project are promising for the future of sensor-data authentication, beyond just NFC transfers.

The encouraging results from the study show that it is possible to use physical actions as an authentication mechanism for applications that would require a relatively small pool of physical actions to choose from. The study also shows that in-built sensor extraction and comparison to baseline data can be done quickly within Samsung Galaxy S7 and newer models. While the NFC authentication mechanism was not implemented in this study due to its extended scope, the results of this study directly contribute to such an authentication mechanism.

Special Thanks To

I would like to thank Sriram Chellappan, Associate Professor at The Department of Computer Science and Engineering at University of South Florida. He provided mentorship

extracting data from the in-built sensors as well as providing me a lab within USF to conduct all the sensor extractions.

Works Cited

Ali, S., et al. "Sensors And Mobile Phones: Evolution And State-Of-The-Art." Pakistan Journal Of Science 66.4 (2014): 385-399. Academic Search Complete. Web. 16 Nov. 2016.

Arcese, G.; Campagna, G.; Flammini, S.; Martucci, O. Near Field Communication: Technology and Market Trends. *Technologies* 2014, *2*, 143-163.

Coskun, Vedat, Busra Ozdenizci, and Kerem Ok. "The Survey on Near Field Communication." Sensors (Basel, Switzerland). MDPI, 5 June 2015. Web. 15 Apr. 2017.

Brassil, Jack, Ravi Netravali, Pratyusa Manadhata, and Prasad Rao. Authenticating a Mobile Device's Location Using Voice Signatures. Piscataway, NJ: IEEE, 2012. 2012 IEEE 8th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob). IEEE. Web. 15 Nov. 2016.

Khalilzadeh, Jalayer, et al. "Security-Related Factors in Extended UTAUT Model for NFC Based Mobile Payment in the Restaurant Industry." Computers in Human Behavior, vol. 70, May 2017, pp. 460-474. EBSCOhost, doi:10.1016/j.chb.2017.01.001.

Hao, Xia, et al. "Using Smart Phone Sensors To Detect Transportation Modes." Sensors (14248220) 14.11 (2014): 20843-20865. Academic Search Complete. Web. 16 Nov. 2016.

"History of Near Field Communication." History of Near Field Communication - NearFieldCommunication.org. NearFieldCommunication.org, n.d. Web. 15 Mar. 2017.

Lee, Wei-Han, and Ruby Lee. "Multi-sensor Authentication to Improve Smartphone Security." Princeton University, 2015. Web. 2016.

Lee, Young-Seol and Sung-Bae Cho. "Layered Hidden Markov Models to Recognize Activity with Built-In Sensors on Android Smartphone." Pattern Analysis & Applications, vol. 19, no. 4, Nov. 2016, pp. 1181-1193. EBSCOhost, doi:10.1007/s10044-016-0549-8.

Liu, Ming. "A Study of Mobile Sensing Using Smartphones." International Journal of Distributed Sensor Networks. SAGE Publications Ltd, 01 Mar. 2013. Web. 15 Nov. 2016.

Mesropyan, Elena. "Sound-Based Payments as an Inclusive Technology for the Developing World." Lets Talk Payments. LTP Team, 28 Oct. 2016. Web. 15 Apr. 2017.

Mulliner, Collin. "Vulnerability Analysis and Attacks on NFC-Enabled Mobile Phones." 2009 International Conference on Availability, Reliability and Security (2009): n. pag. Mulliner. BETAVERSiON.NET. Web. 16 Nov. 2016.

NowSecure. "2016 NowSecure Mobile Security Report." NowSecure, 2016. Web. Winter 2016.

"Security Risks of Near Field Communication." NEAR FIELD COMMUNICATION. NearFieldCommunication.org, n.d. Web. 16 Nov. 2016.

"Sending Files to Another Device." Android Developers. Android Developer, n.d. Web. 16 Nov. 2016.

"Sensor Types." Android Open Source Project. Android, n.d. Web. 16 Nov. 2016.

Shoaib, Muhammad, et al. "Fusion Of Smart Phone Motion Sensors For Physical Activity Recognition." Sensors (14248220) 14.6 (2014): 10146-10176. Academic Search Complete. Web. 16 Nov. 2016.

Snell, Bruce. "Mobile Threat Report." Intel Security, 2016. Web. Winter 2016.