UNIVERSITY OF CALIFORNIA, MERCED

A Tap-Based Back-of-Device Mobile Users Authentication System

A Thesis submitted in partial satisfaction of the requirements
for the degree of Master of Science

in

Electrical Engineering and Computer Science

by

Satvik Kulshreshtha

Committee in charge:

Professor Mukesh Singhal, Chair
Professor Ahmed Sabbir Arif
Professor Wan Du

2019

The Thesis of Satvik Kulshreshtha is approved, and it is acceptable in quality and form for publication on microfilm and electronically:

Chair

University of California, Merced

2019

Table of Contents

Declaration

I hereby declare that no portion of the work referred to in this Project Thesis has been submitted in support of an application for another degree or qualification in this or any other university or institute of learning. If any act of plagiarism is found, I am fully responsible for every disciplinary action taken against me, depending upon the seriousness of the proven offence.

Acknowledgments

I would like to thank Professor Ahmed Sabbir Arif for his supervision and encouragement. I learned so much about the area of Human-Computer Interaction (HCI) working with him. Without his support, this work would not have been possible. I would also like to thank Professors Mukesh Singhal and Wan Du for their guidance. Thanks to all my colleagues at UC Merced: Jun Hyung Shin, Tapish Rathore, Sree Harsha and many others. I would like to acknowledge the Human-Computer Interaction Group members for their support. Special words of gratitude go to my friend Ojasvita, who has always been a major source of support when things became a bit discouraging. Finally, I would like to acknowledge the people who mean the world to me, my parents and my sister. I extend my respect to my parents, grandparents, and all the elders in the family. I cannot imagine a life without their love and blessings.

Abstract

A Tap-Based Back-of-Device Mobile Users Authentication System

by

Satvik Kulshreshtha

Master of Science

in

Electrical Engineering and Computer Science

University of California, Merced

We propose a novel tap-based mobile user authentication system that enables users to authenticate themselves by performing tap patterns on the back of the device. With this approach, the user first selects a pattern as her password, then performs it to authenticate herself. On each authentication attempt, the system compares the data from the built-in microphone and accelerometer with the password, then authenticates the user when they are similar. Since the proposed approach requires performing the tap patterns on the back of the device, it increases security by reducing the risks of shoulder surfing, smudge, and video attack. In a user study, the approach yielded 70% accuracy rate with just three samples, and was secure (17% successful attack rate) even in an ideal shoulder surfing threat model. Further, most participants found it easy-to-use, felt secure while using it, and wanted to keep using it on their devices.

Chapter 1

Introduction

The usable security community has focused much on investigating different user-centered attack on mobile devvices, particularly shoulder surfing [1–3], smudge attack [4, 5], and thermal attack [6]. Among these, shoulder surfing is arguably the most discussed threat, in which attackers attempt to obtain a user's personal identification numbers (PINs) and other confidential details by looking over her shoulder [1]. In recent years, different approaches have been applied to alleviate shoulder surfing attack, which include adding random cues [7–10], using micro gestures [11], enabling force input [12], forcing the attacker to observe multiple cues [2, 3], and masking user input [13, 14]. In addition to user authentication, many have explored approaches that aim at protecting mobile users from shoulder surfing text messages [15] and personal pictures [10]. Most of these works address a threat model where the attacker can clearly observe the process of password entry once. Other threat models include multiple observation attack [3, 16–18] and video attack [2, 19].

Many have also investigated smudge attack, in which the attacker studies the oily residues left on the touchscreen to discover a password [20]. Studies showed that this attack performs well for patterns since the residue left on the screen offers hints on where the pattern started and ended. Likewise, smudges offer hints on which digits were used to unlock a device [12]. The methods used to mitigate smudge attack include graphically transforming the visual cue on which the password is entered [4, 5], introducing a random element that leads to different smudges at every authentication attempt [5], and using multiple fingers to increase the complexity of the gesture patterns [21]. The smudge attack threat model considers clearly visible smudge traces and an optimal lighting condition to study the smudges on the screen [12]. Besides, the model assures that the attacker has access to the mobile device.

Thermal image attack, in contrast, exploits the properties of thermal imaging. More specifically, heat traces are transmitted from the user's fingers to the touchscreen during authentication. These traces fade away slowly [22], allowing thermal cameras to perceive which parts of the display have been touched even after the user had already entered the password. Similar to shoulder surfing, thermal attack leaks information about the order in which password and patterns are entered [20]. Unlike shoulder surfing, thermal attack can be performed after the user had left the device, which gives the attacker an advantage as she does not have to observe the user performing the authentication process. This makes this attack subtler. Mowery et al. [23] studied the effectiveness of thermal attack on ATMs with plastic keypads. They found out that thermal attack are possible even after the

user is authenticated. Some have explored the effectiveness of thermal attack on mobile and other touchscreen-based devices. In one such study, Andriotis et al. [6] were able to observe heat traces resulting from entering a gesture pattern for three seconds after authentication, which enabled them to retrieve parts of the pattern.

To minimize the risks of the above threats by increasing the security of user authentication, we introduce a new approach to authenticate mobile users that, inspired by the concept of "secret knocks", uses back-of-device tap patterns as passwords and can be used not only for user authentication but also in other applications, such as to secure peer-to-peer data transfer.

1.1 Objective & Overview

This work aimed at developing a fully automated system that can measure similarity scores between two tap patterns (namely the sample and an authentication attempt) and authenticate users when the scores are above predetermined thresholds. The primary objective of this work are as follows.

- Identify the best built-in sensors of a stock smartphone that can detect and classify back-of-device taps.
- Identify the best similarity score thresholds that can reliability compare two tap patterns using real-time sensor data in real-word scenarios.
- Design and develop a user authentication approach that can authenticate users by comparing two back-of-device tap patterns.
- Evaluate the proposed mobile user authentication approach in terms of usability and security in an empirical study.

The proposed user authentication system is a proof of concept implementation that consists of a Web application developed for an Android smartphone with HTML5, CSS and JavaScript. It uses Python scripts to compute data, and Flask framework to establish connections between the application and a server, as well as to gather data from the built-in sensors of an off-the-shelf smartphone. The goal was to enable mobile users to authenticate themselves taking the security of user-data into account. Although the proposed system focuses on back-of-device taps for user authentication, it can be integrated into other application, such as to the camera for taking pictures, for sharing data between devices, and to control a music player.

1.2 Organization

This thesis is organized as follows. Chapter 2 discusses the existing literature on mobile user authentication and the challenges that remain in the field. Chapter 3 describes the architecture of the proposed back-of-device tap-based user authentication approach. Chapter 4 presents the results of a pilot and two user studies that evaluated the reliability of built-in sensors in various settings, determine the best thresholds to compare tap patterns, and the usability and security of the proposed approach. Finally, Chapter 5 concludes this thesis with speculation on future extension of this work.

Chapter 2

Related Work

This chapter presents a comprehensive review of the existing user authentication approaches from the literature. Nowadays, mobile devices are developed to serve various functions, as well as to store sensitive information. The core objective of user authentication is to protect these information from unauthorized users. For this, researchers have proposed a variety of approaches. Over the past few years, there has been rich literature, investigating the use of secrete-knowledge based approaches for user authentication. The subsequent sections provide an overview of these approaches.

Nanavati et al. [24] classified mobile devices authentication in three fundamental approaches: i) secret-knowledge based, ii) token based, and iii) biometrics. The first uses a secret PIN or password to authenticate users. While it provides a standard level of protection along with cheap and quick authentication, this approach alone is inadequate to protect the data stored in mobile devices since users tend to use common PINs and passwords, and seldom share these with friends and family [25]. Besides, most users find this approach very inconvenient and do not have confidence in the protection it provides [26]. The second is a SIM (Subscriber Identification Module) based approach that requires the user to remove the device's SIM when its not being used. However, removing SIM is not only inconvenient but also not recommended by most service providers. The final approach is biometrics that relies on a unique feature of the user, that is, it identifies and verifies users based on human characteristics. There are two different types of biometrics: i) physiological, which includes fingerprint, facial recognition, and iris scanning, and ii) behavioral, which includes keystroke dynamics, mouse movements, and speech recognition.

In the earliest era of mobile phones, behavioral authentication focused on keystroke dynamics that authenticated users based on their typing actions on mobile keypads. Clarke et al. [27] developed a keystroke dynamics based authentication scheme that relied on the features of key hold-time and inter-key latency of a keypad. They reported an average equal error rate of 12.8% using a neural network classifier. Zahid et al. [28] designed a scheme with six keystroke features, including key hold-time, digraph time, and error rate to authenticate mobile phone users. They demonstrated that through combining a PIN based verification mechanism, their system could achieve an average error rate of 2% using a combined classifier of particle swarm optimizer, genetic algorithms, and a fuzzy classifier. A particle swarm optimizer is a simple algorithm effective for optimizing wide range of functions [29]. Zahid et al. used it with the recommended modifications [30, 31]. It starts by assigning particles initial velocities after creating the initial particles. Then the lowest function

value and best location value are determined after objective function at each particle location is calculated. The new velocity is chosen based on the individual particle's best location, its neighbor's best location, and its current velocity. Then the particle's location, its velocities, and the neighbor's velocities are updated interactively. This iteration continues until a stopping criterion is reached by the algorithm.

Touch dynamics has received much attention with the advent of touchscreen based phones. Generally, touch dynamics can provide more behavioral events like touch movement and multitouch actions than keystroke dynamics that only employs buttons as input method. However, there are some similarities between touch and keystroke dynamics as single touch events like touch press-up and press-down are similar to button-up and button-down events. For user authentication, keystroke and touch gestures can be combined for virtual keyboards or keypads. Zheng et al. [32] presented an authentication scheme that combined touch and tap features to validate a user when she enters her password. To build a normal profile for users, they further used a one-class algorithm based on nearest neighbor distance. They invited 80 participants for the evaluation and reported an average equal error rate of 3.65%. De Luca et al. [33] attempted to improve the performance of inputting password patterns against shoulder surfing attack by using touch gestures. Buriro et al. [34] intriduced Touchstroke, a bi-modal biometrics authentication scheme that took both the user's hand movements and touch typing into account. They mainly focused on a scenario where the user enters a text-independent 4-digit password.

Feng et al. [35] designed a touchscreen based authentication scheme that required users to wear a digital glove to verify themselves based on finger gestures. Using a random forest classifier, they reported a false acceptance rate of 4.66% and a false rejection rate of 0.13%. Meng et al. [36] developed a behavioral authentication scheme for smartphones that utilized 21 touch features. They reported an average error rate of 3% by using a neural network classifier. Frank et al. [37] proposed a touch behavioral authentication scheme that used a total of 30 touch features. In an evaluation, it yielded a median equal error rate of 4%. However, the authors cautioned that this approach could only be deployed as an optional, not standalone, authentication scheme. Sae-Bae et al. [38] focused on multi-touch behavior and proposed to authenticate users based on up to 22 multi-touch gestures that can be extracted from both hand and finger actions.

Some recent works have combined behavioral and other biometrics for added security. Smith-Creasey and Rajarajan [39] developed an authentication scheme by combining face and touch gestures. They reported an equal error rate of 3.77% using a stacked classifier approach. Shahzad et al. [40] proposed a scheme that utilized how users interact with touchscreens when performing a touch gesture and a signature. It accounted for touch velocity, device acceleration, and stroke time. Nguyen et al. [41] proposed a method that required users to draw a PIN on a touchscreen instead of typing. It used a PIN content analyzer and a drawing behavior analyzer to identify imposters. Arif et al. [12] proposed a different scheme that added pseudo pressure detection to the digit-lock method for an extra layer of security. With this approach, both the PIN and the force applied to enter the digits became the user's password. Another approach augmented up, down, left, and right gestures to the digit keys to enable the user to either tap or perform any one of the four gestures when selecting their PINs [11]. An evaluation showed that users felt secure when using this approach although it was error prone and slower than the conventional digit lock method. Meng et al. [42], on the other hand, combined a touch movement based authentication scheme with

the common pattern-lock method. A study suggested that this approach can increase security without degrading usability. Arif and Mazalek [43] proposed two new user authentication techniques. The first enabled user to select custom slide patterns as their passwords. It divided the screen into three different zones, where each zone acted as a distinct touch area. The second, a variation of the first method, also allowed users to pick one of the three available time-frames for each zone. Both methods performed relatively well in an evaluation, and users found the first method easy to use.

Some additional mechanisms can be used to enhance the performance of behavioral authentication. Meng et al. [44] identified that classifier performance can be unstable due to specific training data. They proposed an adaptive mechanism that can maintain the authentication accuracy. They then described a lightweight touch gesture based scheme, which yielded an average error rate of 2.46%. With the increasing capability of smartphones, behavioral authentication has also received some attention. A recent survey [45] on biometric authentication provides further details.

Chapter 3

Proposed System

We developed a tap-based user authentication approach that uses the concept of "secret knock", which was used during the prohibition era (1920-1933), on the back of the device to authenticate the user. It uses the built-in microphone and accelerometer of a stock smartphone to detect back-of-device taps. It was implemented taking the security aspect of user authentication into account.

We argue that the proposed approach is more secure compared to the existing authentication systems since the taps are performed on the back of the device, which reduces the risk of shoulder surfing. When under video surveillance, back-of-device taps also reduces the risk of getting the password caught or recorded on camera. Using this mechanism, users do not have to remember a password or a PIN, which is arguably more difficult than remembering a tap pattern. The proposed approach is also more secure than the pattern-lock authentication method since it does not leave any smudges on the touchscreen, which makes it relatively easy to guess a touch pattern.

The proposed approach can be used on various devices, including smartwatches, smartphones, and tablets. But as a proof of concept implementation, we used an off-the-shelf Android smartphone. To make the prototype user-friendly, we develop a Web application that uses HTML5, CSS and JavaScript [46] on front-end and Python on back-end to handle server side operations. It also uses Flask, a micro framework for Python for establishing connections between the application and the application server. Since the implementation is in development environment, it is tested over `localhost`. Mozilla Firefox browser is used as a client to run the Web application since other browsers do not allow the sensors to be monitored when an application is in the development environment. Rest APIs are used to connect application to the Data Access Objects since no real database is used in the implementation.

3.0.1 System Overview

Primarily, the proposed system involves performing patterns generated by taps on the back of the device that are recorded using the built-in microphone and accelerometer. A smartphone is considered for the development of this system as motion and orientation sensor are easily available on these devices as compared to older devices. The patterns generated were recorded in two forms: a wave audio file and acceleration (the rate of change in velocity) recorded by the microphone and the accelerometer, respectively. The data are then processed in two parts.

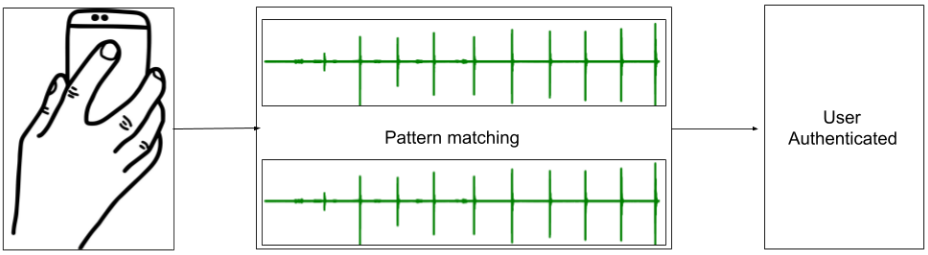

Figure 3.1: High-level system overview.

The first part includes Dynamic Time Warping (DTW) algorithm and Fast Fourier Transform (FFT), which are used for the wave audio file to match the audio pattern. DTW [47] processes similarity between two temporal sequences in time domain. The input signals (wave audio files) are read and the features are extracted using LibROSA, a package written in Python programming language for music and audio analysis. Mel-Frequency Cepstrum Coefficients (MFCCs) are a feature widely used in automatic speech and speaker recognition. After feature extraction, the minimum distance between two time-domain sequences is estimated, which helps in finding if the observed input is similar to the estimated input. Distance of zero implies a perfect match and gives a straight line. To analyze the frequency domain of the recorded data, Fast Fourier Transform (FFT) [47] is used. FFT reveals periodicity in input data and the relative strength of periodic component. FFT splits the input data into smaller components, hence makes finding similarity patterns in frequency domain more convenient. Mean Square Error (MSE) is calculated on FFT to find out the accuracy between two given input data, which are wave audio files. A MSE of zero implies estimated and observed values are matched with perfect accuracy.

An accelerometer sensor in a smartphone is an electro-mechanical device used to measure acceleration forces, the measurement of the rate of change in velocity that may be the force of gravity or vibrations. Smartphones and other mobile devices recognize their orientation through the use of an accelerator, which is a small device made up of axis-based motion sensing. Generally, accelerometers are made up of multiple axes. Most smartphones make use of three-axis models to determine the moment of impact. Therefore, second part of the implementation includes sharp peaks detection by calculating the relative maxima of the noisy data recorded by accelerometer.

3.0.2 System Architecture

The architecture of the proposed system is described as follows. It records microphone and accelerometer data generated by back-of-device taps, then applies two methods to authenticate users.

Figure 3.3 illustrates the method implemented for matching the wave audio file recorded using the built-in microphone of the device. After the data is recorded using the Web application on the front-end, the data is sent to the local server developed with Flask. Pattern matching method is implemented on this application server, which uses two algorithms to perform pattern matching. The two main modules of this method are the DTW and the FFT. The steps involved in DTW are: distance calculation between each pair of frames in the sample frequency. Aligning the frequency samples and finding the minimum distance between the samples. FFT includes blocking of the

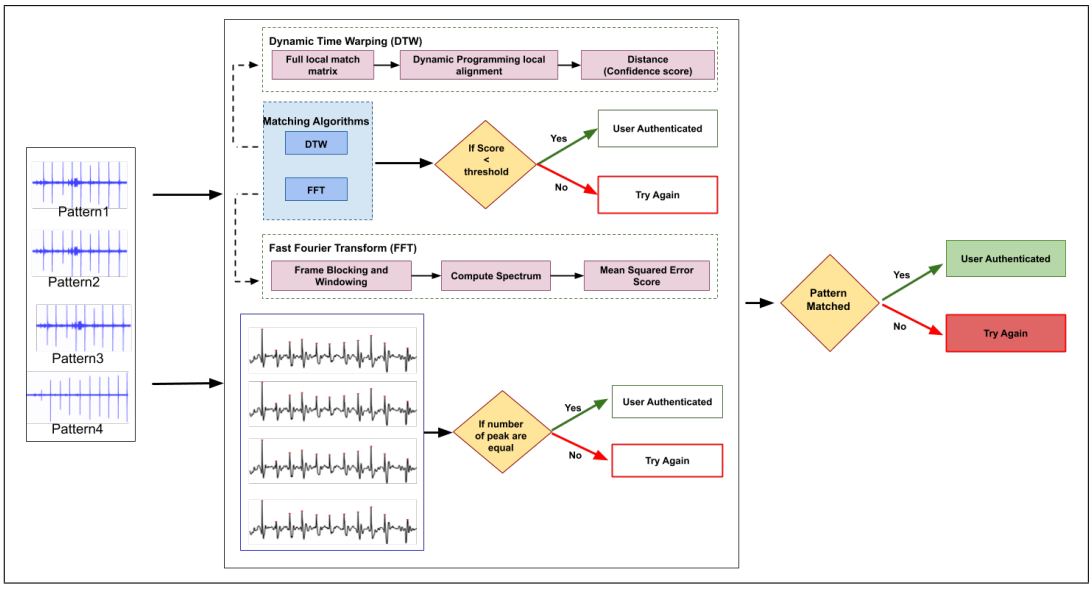

Figure 3.2: Architecture of the proposed system.

signal into N number of frames (this method is called frame blocking) followed by computing the frequency spectrum. Finally, the Mean Square Error is calculated.

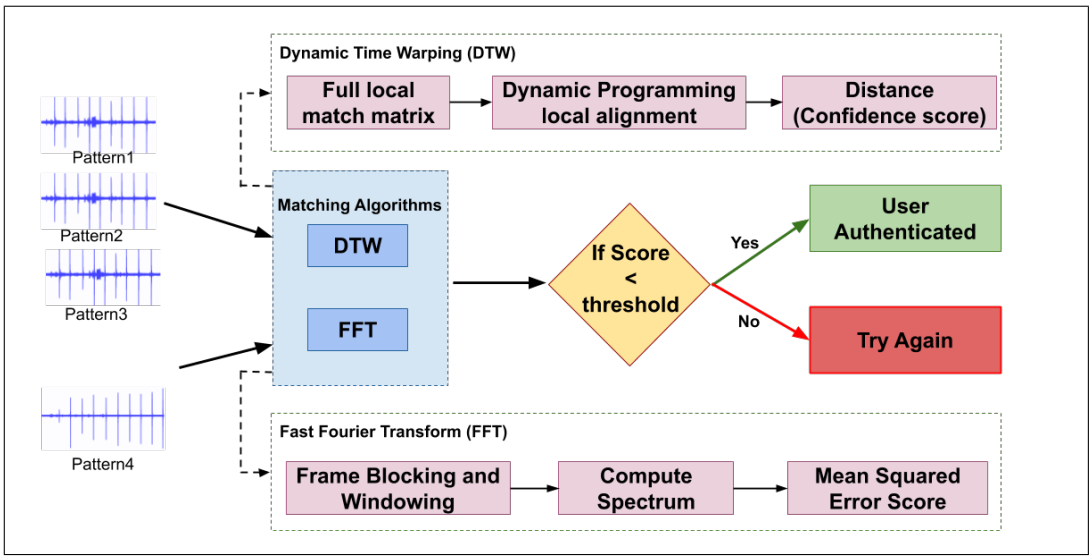

Figure 3.3: Processing of the audio data.

Figure 3.4 illustrates the method implemented for matching the peaks recorded using the accelerometer sensor. The algorithm for finding peak includes performing the continuous wavelets transform on the data followed by finding the maxima at each row in the matrix and filtering the

height and width. The parameters used to calculate the relative maxima of the given data are: the array in which the relative maxima has to be calculated, the axis of data, order that shows what points are needed for comparison, and the mode that determines how the edges of the vector are treated (sharp or soft). The tuple of integer array of maxima is returned at the end.

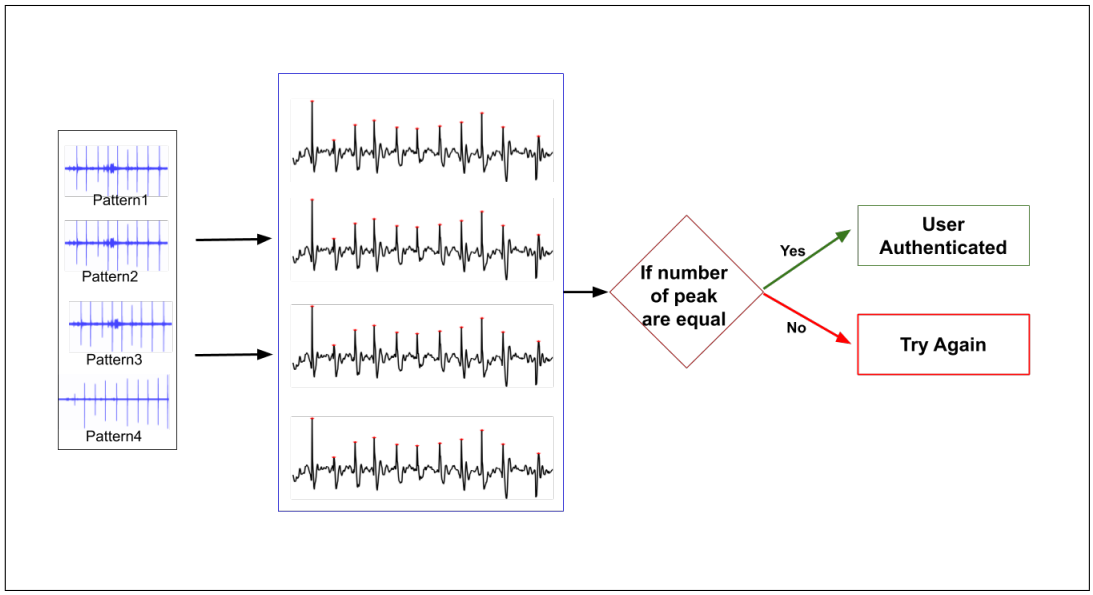

Figure 3.4: Processing of accelerometer data

The results from the two methods are combined together to match the patterns generated by the user by performing the back-of-device taps to determine whether the user is authenticated with the system or not. Both methods are applied on each input pattern generated by the user. Every pattern is compared with each other to find the similarity score in order to authenticate the user. The effectiveness of proposed framework is evaluated in terms of success rate and attack failure rate.

Chapter 4

Evaluation

We conducted a pilot and two user studies to evaluate the reliability of built-in sensors in various settings, determine the best thresholds to compare tap patterns, and the usability and security of the proposed approach.

4.1 Pilot Study

This pilot study was conducted to find out whether the system recognizes the taps on back-of-device and to find out which sensors are the most suitable for detecting these taps.

4.1.1 Apparatus

The device used for the study was Motorola G^5 Plus smartphone ($150.2 \times 74 \times 7.7$ mm), which weighs 155g, running Android OS version 7.0 (Nougat) at 1080×1920 pixels.

4.1.2 Design

We used a within-subjects design. The independent variables were sensor (Accelerometer, Gyroscope, and the Microphone) and setting. There were two settings in the study to record readings from microphone sensor and accelerometer/gyroscope sensor.For microphone sensor readings, four settings were considered: 1) indoor with no background noise, 2) indoor with background noise (noise such as, television sound and chatting), 3) outdoor, and 4) outdoor with heavy background noise (such as noise from a construction site). For Accelerometer and Gyroscope, two settings were considered: tap on back-of-device while 1) seated and 2) walking.

4.1.3 Participants

For this study, we recruited five male participants from the local university. Their age ranged from 22 to 29 years. They were not compensated for volunteering.

4.1.4 Procedure

In the study, participants were instructed to tap on back-of-device in the given conditions. The device was held in upright position during the study for every condition. The participants were asked to tap for ten times on the back of the device with a uniform interval of one second in all the conditions which makes a total of 300 patterns. The data were then sent to server for further computation.

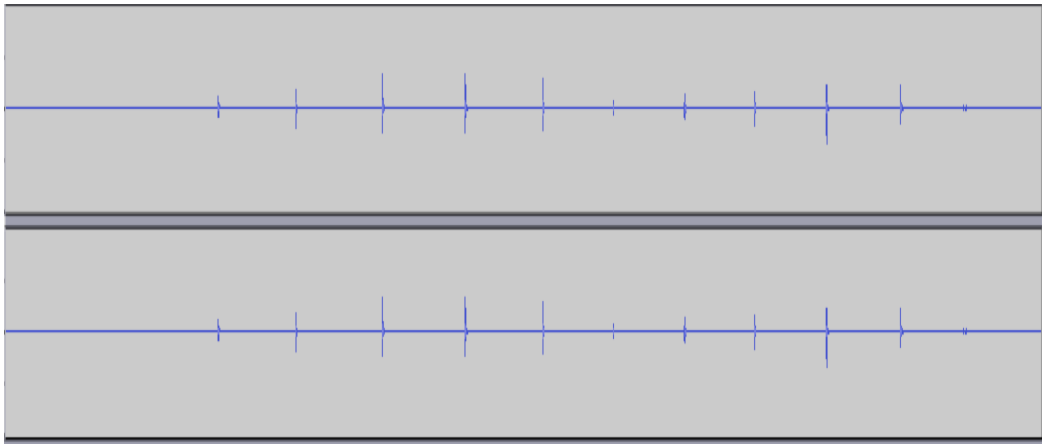

Figure 4.1: Data recorded by Microphone.

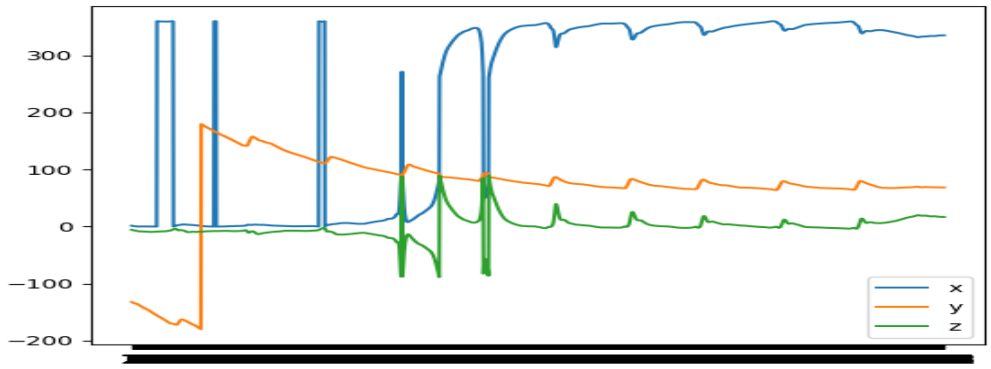

Figure 4.2: Data recorded by Gyroscope.

4.1.5 Observations

We made the following observations.

1. Data showed that the proposed system is able to detect the taps on the back-of-device.

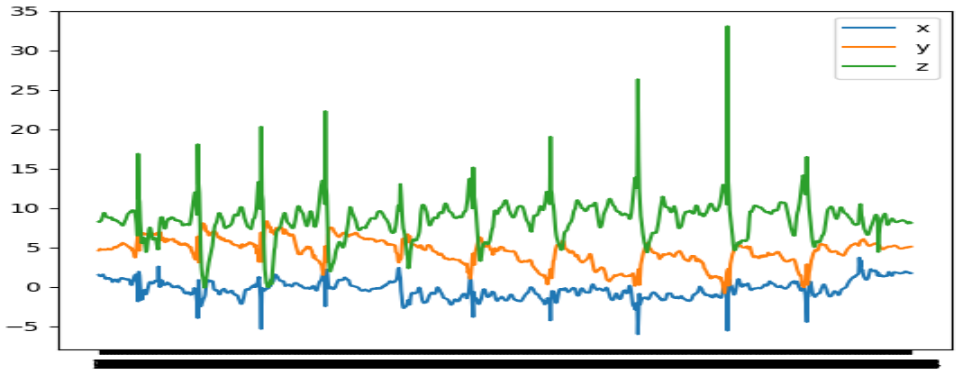

Figure 4.3: Data recorded by Accelerometer.

2. After comparing the data recorded by the three sensors, we conclude that the microphone and the accelerometer are the most reliable sensors to detect the taps on the back-of-device (see Figure 4.1, 4.2, and 4.3.

4.2 User Study 1

In order to develop the proposed real-time user authentication system, we need data in real scenarios to set appropriate threshold for all users so that we get high true positive rate and low false positive rate. The purpose of this study was to find out a threshold for the development of the system to authenticate the user with the device.

4.2.1 Apparatus

The device used for the study was Motorola G^5 Plus smartphone ($150.2 \times 74 \times 7.7$ mm), which weighs 155g, running Android OS version 7.0 (Nougat) at 1080×1920 pixels and has 60 to 120 frames per inch (fpi).

We used a Web application for this study that is developed using HTML5, JavaScript, and Python. The WebkitAudio API was used in this application. For frequency matching Fast Fourier Transform (FFT) and Dynamic Time Warping (DTW) algorithms were used. For establishing connection between client server, Flask , a microframework was used. The application launched with a plain full-screen view, which user touches in order to trigger the recording, and touches again to stop recording. Wave File Audio (.wav) file format used for collected knocks or pattern.

4.2.2 Design and Participants

We used a within-subjects design for this usability testing study. Fifteen participants volunteered to take part in this study. All the participants were recruited from local university community. Their average age was 26.2 years (SD = 3.05). Five of them were female and ten were male. All of them were frequent smartphone users, with on average 6.34 years (SD = 2.29) of experience with touch-screen devices. Fourteen of them were right-handed, one was ambidextrous. The ambidextrous user used the right hand to perform back-of-the-device tapping.

4.2.3 Procedure

First, the study procedure was explained to all participants. The study was conducted under regular environment considering there was a lot of background noise. The Web application developed for recording the tapping data was launched on the smartphone and was given to the participants. Participant were instructed to hold the device in any preferred orientation (e.g., portrait or landscape) and were allowed to use any hand for holding the device and tapping. Then were asked to tap a pattern or a rhythm of their choice on the back-of-device.

To start the study, participants touched the screen of device to trigger the sensor recording, and once they finish, they tapped the screen again to stop the sensor from recording data. They were allowed to re-select their pattern if they are not satisfied with their initial pattern choice. Manual record were kept of how the device was held by the participants to perform tapping during the study. They were asked to perform the experiment five times after recording the initial pattern (total six times) keeping the same pattern or rhythm they tapped at the first time.

There were mandatory breaks between every round of tapping to make sure participants did not feel fatigue and to observe if they remember the pattern correctly. Finally, all participants were

Figure 4.4: Two participants taking part in the study. Participant on the left is interacting to the application and performing back-of-device tapping. Participant performing back-of-device tapping, on the right.

asked to complete a short questionnaire.

4.2.4 Results

We calculated the standard deviation to detect outliers. Based on observed values, we excluded one participant because the value recorded were above ±3 standard deviation (SD). Therefore, total number of users became fifteen.

4.2.4.1 Maximum and Minimum Intensity of Peaks

The average maximum intensity of peaks was 6871.90 dB and the standard deviation recorded was 8131.16. The average minimum intensity and standard deviation of peaks was -7411.14dB and 9265.41, respectively. See Figure 4.5.

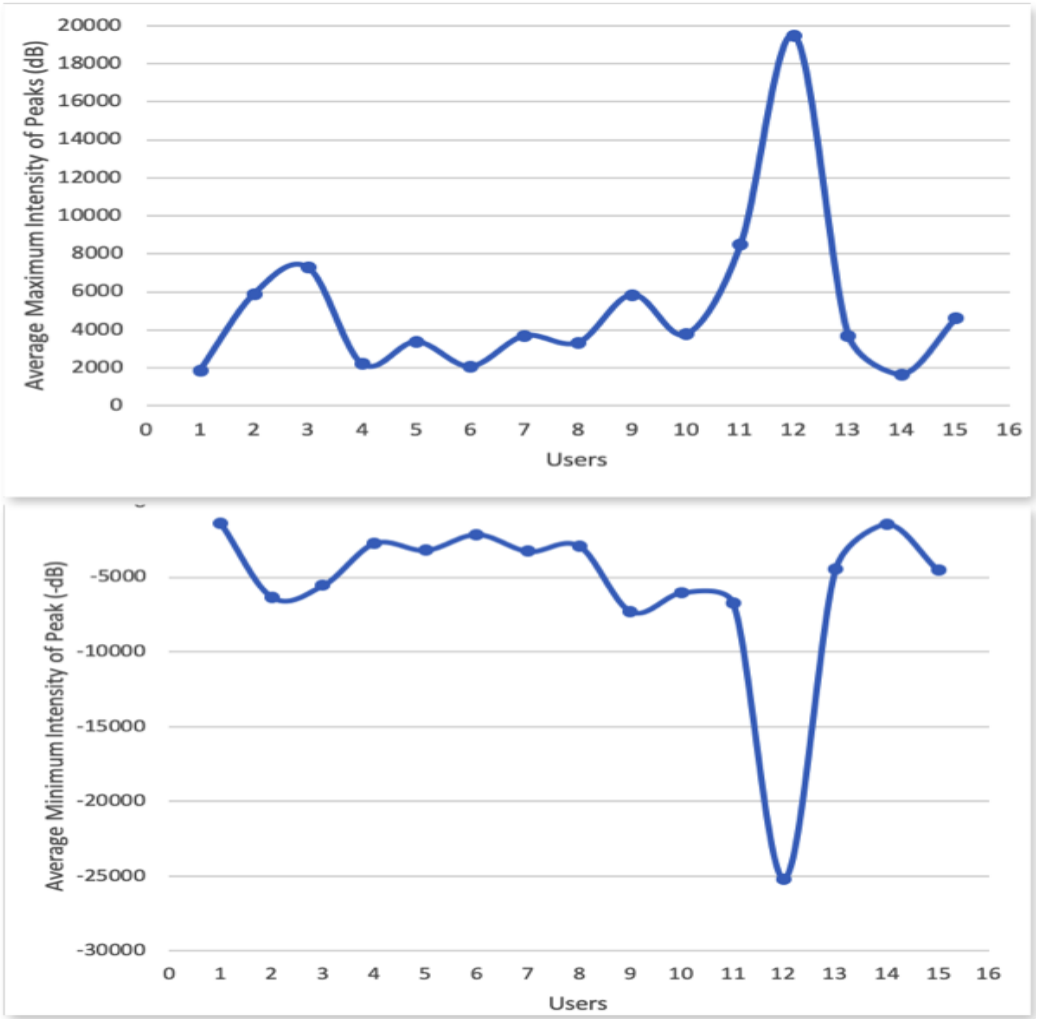

Figure 4.5: Average Maximum and Minimum intensity of peaks recorded per user.

4.2.4.2 Duration of Peaks

The average duration of peaks was 4.96 Seconds and the standard deviation recorded was 2.53. See Figure 4.11.

4.2.4.3 Number of Peaks

The average number of peaks recorded during the study was 6.93 and the standard deviation recorded was 4.81. See Figure 4.7.

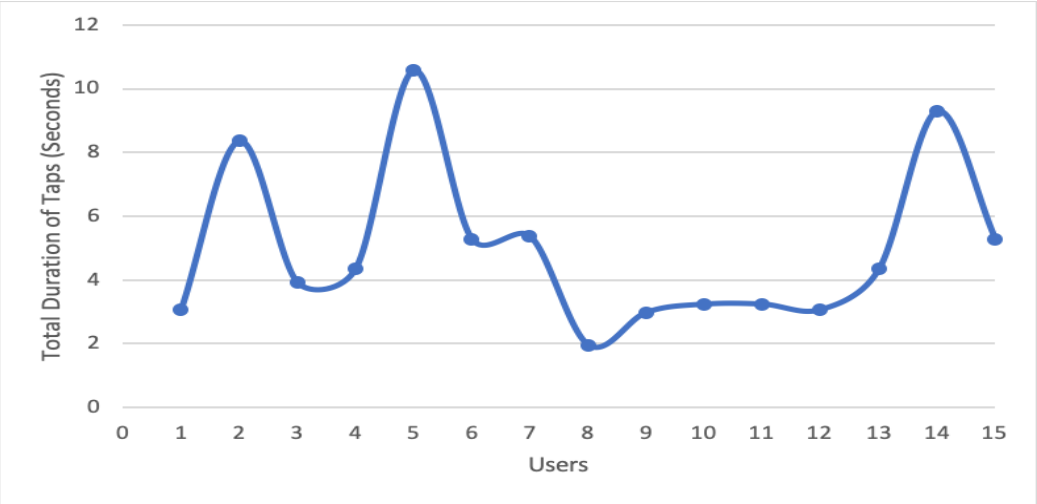

Figure 4.6: Total Duration of peaks per User.

4.2.4.4 DTW and FFT Scores

Mean and Standard deviation of the scores computed using DTW are 111.25 and 28.59 respectively. For FFT, the mean of the scores is 826.125 and its corresponding standard deviation is 1732.39.

4.2.4.5 Questionnaire

Upon completion of the study, all participants completed a short questionnaire. Four out of fifteen participants said that the pattern they tapped was music inspired. Five out of fifteen participants said that the pattern was their random choice that came to their mind and was easy to tap, remaining participants said they have a pattern that they are familiar with since childhood, hence chose that pattern.

All the participants agreed that the pattern they performed was easy to remember, caused no mental fatigue. Two out of fifteen participants felt that tapping on back-of-device caused physical fatigue. Thirteen out of fifteen participants said they will prefer using the back-of-device user authentication method in future, while two said they will prefer some other method of user authentication.

4.2.5 Discussion

In this study, we gathered data from fifteen participants who performed back-of-device tapping. After complete analysis of data collected, results shows that it is difficult to set a common threshold for every user as the intensity of taps may vary with the user. Results also revealed that user will prefer tap-based user authentication method in future. A couple of the participants also felt that this method of authentication is safer than existing method, such as PIN and Password, since it saves them from shoulder surfing.

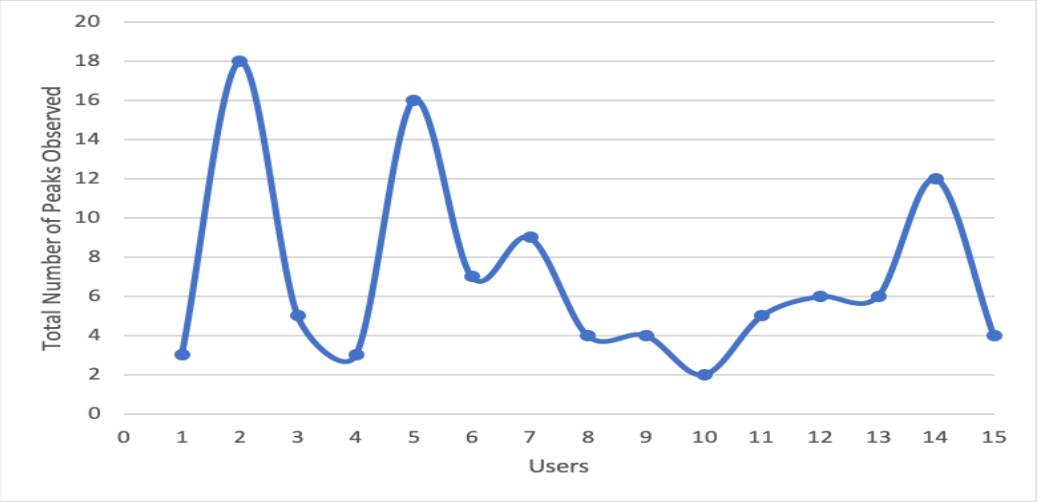

Figure 4.7: Average number of peaks recorded.

4.3 User Study 2

The purpose of this study was to get real-time data from the users and to test the user authentication system in real scenario and to evaluate the system such that we get high true positive rate and low false positive rate.

4.3.1 Apparatus

The device used for the study was Motorola G^5 Plus smartphone ($150.2 \times 74 \times 7.7$ mm), which weighs 155g, running Android OS version 7.0 (Nougat) at 1080×1920 pixels and has 60 to 120 frames per inch (fpi).

We used a Web application for this study that is developed using HTML5, JavaScript and Python. WebkitAudio API was used in this application for recording the microphone data. For frequency matching Fast Fourier Transform (FFT) and Dynamic Time Warping (DTW) algorithms were used. For establishing connection between client server, Flask, a micro framework was used. To detect and match the peaks formed in rate of change of velocity, sciPy-signal python module was used.

The application launched with a plain full-screen view, which users touched in order to trigger the recording of their patterns/rhythms, and touched again to stop recording. Wave File Audio (.wav) file format and an integer array of acceleration data (rate of change of velocity) were recorded and analyzed in order to match the pattern recorded by user.

4.3.2 Design and Participants

We used a within-subjects design for this study. Twelve participants volunteered to take part in this study. All the participants were recruited from local university community. Their average age was 26.16 years. Three of them were female and nine were male. All of them were frequent smartphone users, with on average 7.5 years of experience with touchscreen devices. Eleven of them were right-handed, one was ambidextrous. The ambidextrous user used the right hand to perform back-of-device tapping.

User	Attacker
12 participants ×	12 participants ×
3 tap rhythms ×	3 tap rhythms ×
5 attempts= 180 attempts	5 attacks= 180 attacks

4.3.3 Procedure

First, the study procedure was explained to all participants. The study was done under regular environment considering there was a background noise. The Web application developed for recording the tapping data was launched on the smartphone and was given to the participants.

The study was done in pairs where one participant was user and other one was attacker. First, the user was asked to record three patterns of his/her choice on the back-of-device and was asked to repeat it for five times making it a total of fifteen patterns from one user. Meanwhile, the attacker was asked to observe the pattern tapped by user over the shoulders. Secondly,the attacker was asked

Figure 4.8: Two participants performing the study. Participant on the left is interacting to the application and performing back-of-device tapping. Participant performing back-of-device tapping, on the right.

to replicate the pattern in ten attempts recorded by user in order to determine the success rate of the authentication system and to check whether the system resist shoulder surfing attack.

Then the participants switch their roles and repeated the study. They were given three attempts to record the training data and the final attempt was the test data that authenticate their pattern. To start the study, participants touched the screen of device to trigger the sensor recording, and once they finish, they tap the screen again to stop the sensor from recording data.

4.3.4 Results

All the participants held the device in portrait position with right hand and perform the tapping using index finger of the same hand.One participant used middle finger to perform some taps. It was observed that eleven out of twelve participants (N = 11) tapped on the center of the back of device while one participant performed at top right corner of the back of device. The average number of taps recorded during the study was 4.055 and the standard deviation recorded was 1.41 (Figure 4.9). The average maximum intensity of peaks was 10532.96 dB and the standard deviation recorded was 3913.64 (Figure 4.10). The average minimum intensity and standard deviation of peaks was -9749.26dB and 3666.73 respectively. The average duration of peaks was 2.70 Seconds and the standard deviation recorded was 0.62.(Figure 4.11).Eleven out of fifteen (SD = 1.75) successful attempts were recorded on an average. (Figure 4.12) and only 2.5 attacks out of fifteen on average were recorded successful attempts to crack the password.(Figure 4.13).

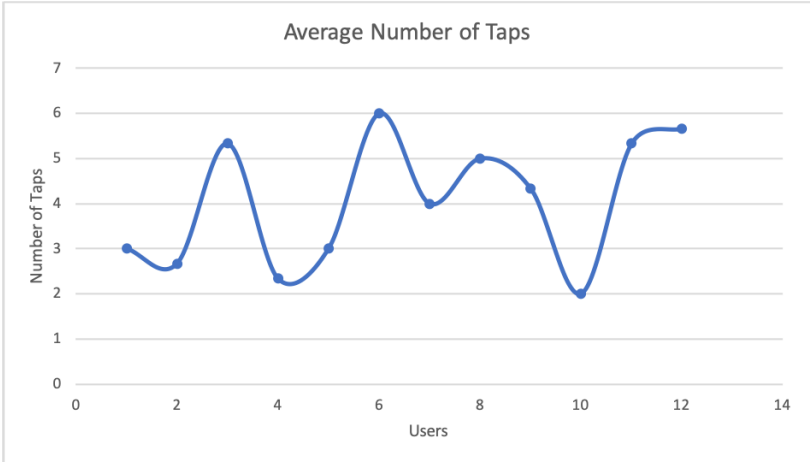

Figure 4.9: Average number of taps recorded per user.

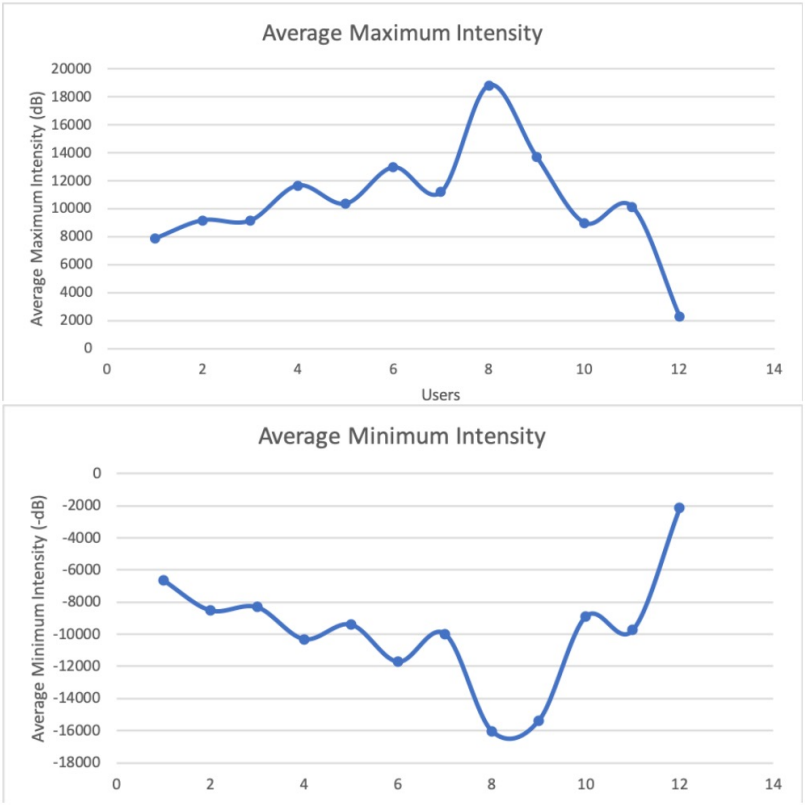

Figure 4.10: Average maximum and minimum intensity of peaks recorded per user.

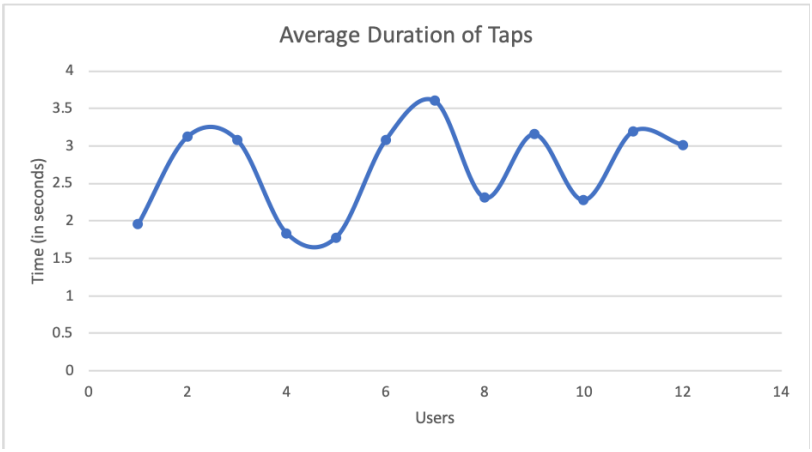

Figure 4.11: Total duration of peaks per user.

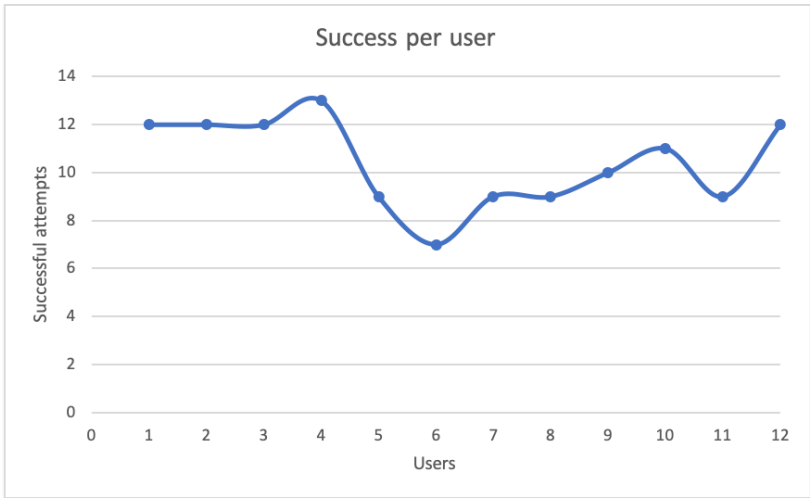

Figure 4.12: Successful attempts per user

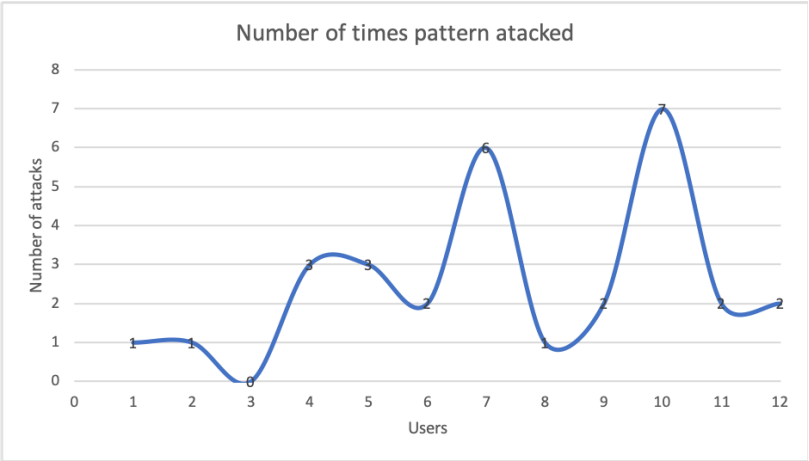

Figure 4.13: Number of times pattern hacked per user

4.3.5 Questionnaire

Upon completion of the study, all participants went through an informal interview session. Most participants(N = 7) selected the pattern as their favorite song or tunes. Rest of the participants (N = 5) selected the rhythm that are easy to remember. Results showed that mops of the participants (N = 11) found that the taps were easy to remember and remaining one participant found it difficult.
Most participants felt that performing the back-of-device rhythms did not cause any cognitive (N = 11) or physical (N = 10) stress, however a few participant felt slightly cognitive (n = 1) and physical (N = 2) stress for extensive use.
Eight out of fifteen participants wanted to use the back-of-device authentication method in future. Remaining three out of four participants liked the idea of back of the device tapping authentication but did not want to use the method in future due to technological issues such as high false negative rate. And one participant was undecided about it.
As a shoulder surfer, eleven out of twelve (N =11) participants responded that it was difficult to observe the tapping pattern. Many participants mentioned that they were able to crack some password because the setting provided to them to shoulder surf was ideal (quiet room and unobstructed view of the authentication process), but they also mentioned that it would be difficult to observe and crack the authentication pattern in real world.

User	Pattern hacked	Pattern not hacked
1	1	14
2	1	14
3	0	15
4	3	12
5	3	12
6	2	13
7	6	9
8	1	14
9	2	13
10	7	8
11	2	13
12	2	13

Table 4.1: Table showing successful attempts and failed attempts by attacker to hack the pattern created by the user.

4.3.6 Summary

In this study, we gathered data from twelve participants who performed back-of-device tapping. Eight out of fifteen participant felt that the proposed system is convenient and secure to use as it does not leave smudge on the screen and resist shoulder surfing. Results also revealed that user would prefer tap-based user authentication method in future. Study showed that when attacker tried to break the authentication pattern created by user, attacker failed on an average of 12.5 times out of fifteen attempts and succeeded to bypass the authentication on an average of 2.5 times out of fifteen attempts given (see Table 4.1 and 4.2).

User	Successful	Failed	TPR	FPR
1	12	3	0.80	0.20
2	12	3	0.80	0.2
3	12	3	0.80	0.2
4	13	2	0.86	0.13
5	9	6	0.60	0.40
6	7	8	0.46	0.53
7	9	6	0.60	0.40
8	9	6	0.60	0.40
9	10	1	0.66	0.33
10	11	4	0.73	0.26
11	9	6	0.60	0.40
12	12	3	0.86	0.20

Table 4.2: Table showing successful attempts and failed attempts by user to match the pattern.

The Effectiveness of the proposed system is evaluated in terms of True positive rate (TPR) and False positive rate (FPR), which is shown in the Receiver Operating Curve below (Figure 4.14). TPR and FPR are calculated using the following equations.

$$TPR = \frac{Authenticated\ User}{Authenticated\ User + Invalid\ User}$$

$$FPR = \frac{InvalidUser}{Invalid\ User + Authenticated\ User}$$

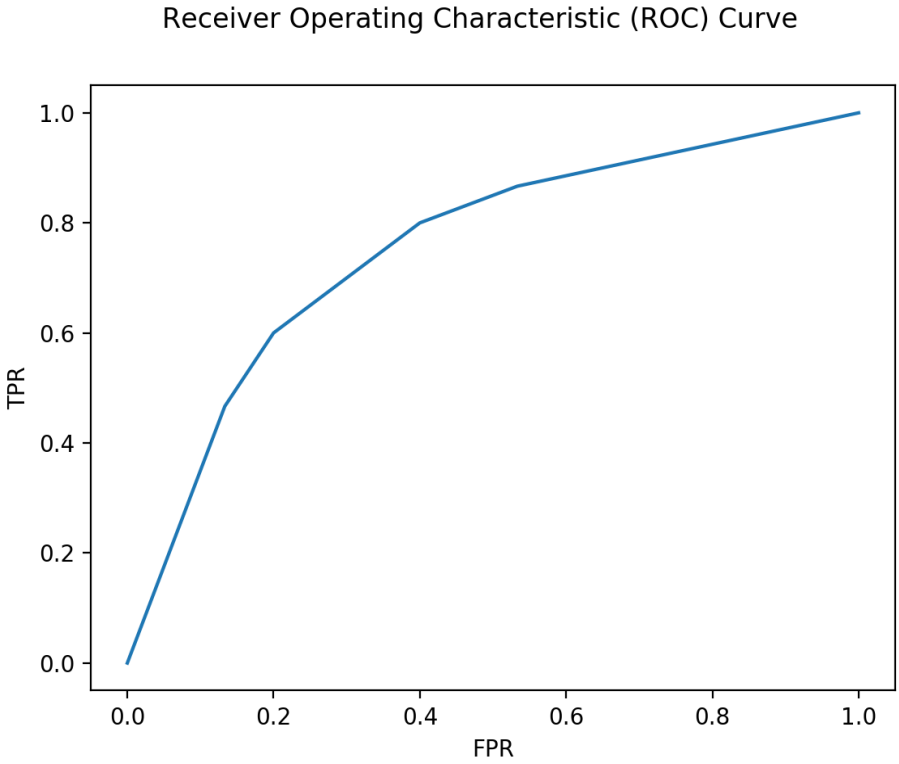

Figure 4.14: Receiver Operating Characteristic (ROC) curve.

4.3.7 Limitations

The proposed user authentication system is developed using a smartphone device since older models of the mobile devices might not have the sensors like accelerometer and gyroscope. So, we can say that the system only works with a smartphone that has the motion and the orientation sensors available in the device.

Since the user authentication system uses a pattern or a rhythm, it highly relies on the memory of the user. Pilot study showed that some users forgot their rhythm and changed the authentication pattern after first attempt. So, we can say that it follows the same memorability characteristics as similar techniques such as PIN, Password and lock pattern based authentication.

The proposed system might fail to detect the taps in loud noisy environment as the microphone will detect more noise than actual authentication pattern. Accelerometer might fail to recognize the pattern if the user waives his hand or the device while tapping on the back of the device as there will be a severe change in the rate of change of velocity and peaks will not form properly.

This user authentication technique requires one handed interaction with the device but many users used two hand to do the interaction. However, it is seen that user tends to use two hands with the other user authentication technique too.

After a study, it can be seen that the patterns that are either created using multiple fingers or the patterns that are too long have less chances of getting recognized by the tap based mobile authentication system.

Finally, there are different cases and covers available in market for the safety of devices that may alter the pattern that user actually want to create and the system might fail to recognize the pattern.

Chapter 5

Conclusion

In this work, we proposed a tap-based user authentication system using the concept of back-of-device taping. In order to get authenticated with the device, user taps a pattern or a rhythm on the back-of-device and the system records the user's pattern. The system uses in-built device sensors, such as the microphone and the accelerometer, to detect the back-of-device tapping. The proposed system is more secure than the existing methods of user authentication since the back-of-device tapping makes the system secure against attacks such as shoulder surfing and camera attacks.

The main contributions of this work are: (1) the proof-of-concept implementation of back-of-device tap detection using in-built sensor available in a mobile device (such as, a smartphone) and (2) user study that illustrates the performance of the proposed system in real-time.

We conducted a pilot study to test if the system is able to detect back-of-device tapping without any external hardware and using the in-built sensors available in the smartphones. Another user study was conducted with fifteen participants who volunteered to take part in this study. All the participants were recruited from local university community. The purpose of this study was to set an appropriate threshold in order to authenticate users with the device, which is used to further improve the system. The results of this study showed that it is difficult to set a common threshold for individual user since the tapping intensity varies with every user. So, instead of setting a common threshold we decided to take multiple samples of input pattern to determine different threshold for each user.

The results from the final study showed that the proposed system performed well in the given scenario. The results from the study also showed that when tried to break the authentication pattern created by user, attacker failed on an average of 12.5 times and succeeded to bypass the authentication on an average of 2.5 times out of fifteen attempts given, regardless of the fact that they were given a clear view of the password entry. Yet, more work is needed to make the system more secure. An informal interview conducted with the participants acting as attacker after the final study showed that the system performed well against shoulder surfing attack as it is difficult to recognize the pattern when tapping performed was on the back-of-the-device.

Chapter 6

Future Scope

There are things that needs to be investigated and explored. In the future scope of this approach, the implementation of the system can be done using different machine learning models, which may help increase the accuracy of this system as the model will train every time there is a new pattern and will try to produce accurate output.

Further research is needed to determine which algorithm works best for pattern matching to reduce the response time of the system. Some solution may suggest to develop native operating systems applications such as Android application or iOS applications rather than using a web based application. This proof of concept implementation can be integrated with other application to enhance the functionality of application, for example, It can be integrated with the camera to take picture just by tapping at the back-of-device. Another application could be sharing the data with another device over Bluetooth using back-of-device tapping, which will also make the data sharing more secure.

Finally, to achieve the better accuracy, more studies can be conducted in real-world scenarios with a larger sample to get more reliable data.

Bibliography

[1] M. Eiband, M. Khamis, E. von Zezschwitz, H. Hussmann, F. Alt, Understanding shoulder surfing in the wild: Stories from users and observers, in: Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems, CHI '17, ACM, New York, NY, USA, 2017, pp. 4254–4265. doi:10.1145/3025453.3025636.
URL http://doi.acm.org/10.1145/3025453.3025636

[2] A. De Luca, M. Harbach, E. von Zezschwitz, M.-E. Maurer, B. E. Slawik, H. Hussmann, M. Smith, Now you see me, now you don't: protecting smartphone authentication from shoulder surfers, in: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, ACM, 2014, pp. 2937–2946.

[3] M. Khamis, F. Alt, M. Hassib, E. von Zezschwitz, R. Hasholzner, A. Bulling, Gazetouchpass: Multimodal authentication using gaze and touch on mobile devices, in: Proceedings of the 2016 CHI Conference Extended Abstracts on Human Factors in Computing Systems, ACM, 2016, pp. 2156–2164.

[4] S. Schneegass, F. Steimle, A. Bulling, F. Alt, A. Schmidt, Smudgesafe: Geometric image transformations for smudge-resistant user authentication, in: Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing, UbiComp '14, ACM, New York, NY, USA, 2014, pp. 775–786. doi:10.1145/2632048.2636090.
URL http://doi.acm.org/10.1145/2632048.2636090

[5] E. von Zezschwitz, A. Koslow, A. De Luca, H. Hussmann, Making graphic-based authentication secure against smudge attacks, in: Proceedings of the 2013 International Conference on Intelligent User Interfaces, IUI '13, ACM, New York, NY, USA, 2013, pp. 277–286. doi:10.1145/2449396.2449432.
URL http://doi.acm.org/10.1145/2449396.2449432

[6] P. Andriotis, T. Tryfonas, G. Oikonomou, C. Yildiz, A pilot study on the security of pattern screen-lock methods and soft side channel attacks, in: Proceedings of the Sixth ACM Conference on Security and Privacy in Wireless and Mobile Networks, WiSec '13, ACM, New York, NY, USA, 2013, pp. 1–6. doi:10.1145/2462096.2462098.
URL http://doi.acm.org/10.1145/2462096.2462098

[7] A. Bianchi, I. Oakley, V. Kostakos, D.-S. Kwon, The phone lock: audio and haptic shoulder-surfing resistant pin entry methods for mobile devices, in: Tangible and Embedded Interaction, 2010.

[8] A. Bianchi, I. Oakley, D. S. Kwon, Spinlock: A single-cue haptic and audio pin input technique for authentication, in: E. W. Cooper, V. V. Kryssanov, H. Ogawa, S. Brewster (Eds.), Haptic and Audio Interaction Design, Springer Berlin Heidelberg, Berlin, Heidelberg, 2011, pp. 81–90.

[9] A. Bianchi, I. Oakley, D. S. Kwon, Counting clicks and beeps: Exploring numerosity based haptic and audio pin entry, Interacting with Computers 24 (5) (2012) 409 – 422. doi:https://doi.org/10.1016/j.intcom.2012.06.005.
URL http://www.sciencedirect.com/science/article/pii/S095354381200063X

[10] E. von Zezschwitz, S. Ebbinghaus, H. Hussmann, A. De Luca, You can't watch this!: Privacy-respectful photo browsing on smartphones, in: Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems, CHI '16, ACM, New York, NY, USA, 2016, pp. 4320–4324. doi:10.1145/2858036.2858120.
URL http://doi.acm.org/10.1145/2858036.2858120

[11] A. Arif, M. Pahud, K. Hinckley, W. Buxton, A tap and gesture hybrid method for authenticating smartphone users, in: Proceedings of the 15th International Conference on Human-computer Interaction with Mobile Devices and Services, MobileHCI '13, ACM, New York, NY, USA, 2013, pp. 486–491. doi:10.1145/2493190.2494435.
URL http://doi.acm.org/10.1145/2493190.2494435

[12] A. S. Arif, A. Mazalek, W. Stuerzlinger, The use of pseudo pressure in authenticating smartphone users, in: Proceedings of the 11th International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services, MOBIQUITOUS '14, ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), ICST, Brussels, Belgium, Belgium, 2014, pp. 151–160. doi:10.4108/icst.mobiquitous.2014.257919.
URL http://dx.doi.org/10.4108/icst.mobiquitous.2014.257919

[13] A. De Luca, E. von Zezschwitz, N. D. H. Nguyen, M.-E. Maurer, E. Rubegni, M. P. Scipioni, M. Langheinrich, Back-of-device authentication on smartphones, in: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI '13, ACM, New York, NY, USA, 2013, pp. 2389–2398. doi:10.1145/2470654.2481330.
URL http://doi.acm.org/10.1145/2470654.2481330

[14] J. Gugenheimer, A. D. Luca, H. Hess, S. Karg, D. Wolf, E. Rukzio, Colorsnakes: Using colored decoys to secure authentication in sensitive contexts, in: MobileHCI, 2015.

[15] M. Eiband, E. von Zezschwitz, D. Buschek, H. Hußmann, My scrawl hides it all: protecting text messages against shoulder surfing with handwritten fonts, in: Proceedings of the 2016 CHI Conference Extended Abstracts on Human Factors in Computing Systems, ACM, 2016, pp. 2041–2048.

[16] E. Hayashi, S. Das, S. Amini, J. Hong, I. Oakley, Casa: Context-aware scalable authentication, in: Proceedings of the Ninth Symposium on Usable Privacy and Security, SOUPS '13, ACM, New York, NY, USA, 2013, pp. 3:1–3:10. doi:10.1145/2501604.2501607.
URL http://doi.acm.org/10.1145/2501604.2501607

[17] I. Oakley, A. Bianchi, Keeping secrets from friends, Archives of Design Research 27 (3) (2014) 49–62.

[18] O. Wiese, V. Roth, See you next time: A model for modern shoulder surfers, in: Proceedings of the 18th International Conference on Human-Computer Interaction with Mobile Devices and Services, MobileHCI '16, ACM, New York, NY, USA, 2016, pp. 453–464. doi:10.1145/2935334.2935388.
URL http://doi.acm.org/10.1145/2935334.2935388

[19] E. Von Zezschwitz, A. De Luca, B. Brunkow, H. Hussmann, Swipin: Fast and secure pin-entry on smartphones, in: Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems, ACM, 2015, pp. 1403–1406.

[20] A. J. Aviv, K. Gibson, E. Mossop, M. Blaze, J. M. Smith, Smudge attacks on smartphone touch screens, in: Proceedings of the 4th USENIX Conference on Offensive Technologies, WOOT'10, USENIX Association, Berkeley, CA, USA, 2010, pp. 1–7.
URL http://dl.acm.org/citation.cfm?id=1925004.1925009

[21] I. Oakley, A. Bianchi, Multi-touch passwords for mobile device access, in: Proceedings of the 2012 ACM Conference on Ubiquitous Computing, UbiComp '12, ACM, New York, NY, USA, 2012, pp. 611–612. doi:10.1145/2370216.2370329.
URL http://doi.acm.org/10.1145/2370216.2370329

[22] E. Larson, G. Cohn, S. Gupta, X. Ren, B. Harrison, D. Fox, S. Patel, Heatwave: thermal imaging for surface user interaction, in: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, ACM, 2011, pp. 2565–2574.

[23] K. Mowery, S. Meiklejohn, S. Savage, Heat of the moment: Characterizing the efficacy of thermal camera-based attacks, in: Proceedings of the 5th USENIX Conference on Offensive Technologies, WOOT'11, USENIX Association, Berkeley, CA, USA, 2011, pp. 6–6.
URL http://dl.acm.org/citation.cfm?id=2028052.2028058

[24] S. Nanavati, M. Thieme, R. Nanavati, Biometrics: identity verification in a networked world, Vol. 20, John Wiley & Sons, 2002.

[25] U. Topkara, M. J. Atallah, M. Topkara, Passwords decay, words endure: Secure and re-usable multiple password mnemonics, in: Proceedings of the 2007 ACM Symposium on Applied Computing, SAC '07, ACM, New York, NY, USA, 2007, pp. 292–299. doi:10.1145/1244002.1244072.
URL http://doi.acm.org/10.1145/1244002.1244072

[26] N. L. Clarke, S. M. Furnell, Authentication of users on mobile telephones - a survey of attitudes and practices, Comput. Secur. 24 (7) (2005) 519–527. doi:10.1016/j.cose.2005.08.003. URL http://dx.doi.org/10.1016/j.cose.2005.08.003

[27] N. L. Clarke, S. M. Furnell, Authenticating mobile phone users using keystroke analysis, International Journal of Information Security 6 (1) (2007) 1–14. doi:10.1007/s10207-006-0006-6.

[28] S. Zahid, M. Shahzad, S. A. Khayam, M. Farooq, Keystroke-based user identification on smart phones, in: Proceedings of the 12th International Symposium on Recent Advances in Intrusion Detection, RAID '09, Springer-Verlag, Berlin, Heidelberg, 2009, pp. 224–243. doi:10.1007/978-3-642-04342-0-12.

[29] J. Kennedy, R. Eberhart, Particle swarm optimization, in: Proceedings of ICNN'95 - International Conference on Neural Networks, Vol. 4, 1995, pp. 1942–1948 vol.4. doi:10.1109/ICNN.1995.488968.

[30] E. Mezura-Montes, C. A. C. Coello, Constraint-handling in nature-inspired numerical optimization: past, present and future, Swarm and Evolutionary Computation 1 (4) (2011) 173–194.

[31] M. E. H. Pedersen, Good parameters for particle swarm optimization, Hvass Lab., Copenhagen, Denmark, Tech. Rep. HL1001.

[32] N. Zheng, K. Bai, H. Huang, H. Wang, You are how you touch: User verification on smartphones via tapping behaviors., in: ICNP, Vol. 14, 2014, pp. 221–232.

[33] A. De Luca, A. Hang, F. Brudy, C. Lindner, H. Hussmann, Touch me once and i know it's you!: implicit authentication based on touch screen patterns, in: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, ACM, 2012, pp. 987–996.

[34] A. Buriro, B. Crispo, F. Del Frari, K. Wrona, Touchstroke: smartphone user authentication based on touch-typing biometrics, in: International Conference on Image Analysis and Processing, Springer, 2015, pp. 27–34.

[35] T. Feng, Z. Liu, K.-A. Kwon, W. Shi, B. Carbunar, Y. Jiang, N. Nguyen, Continuous mobile authentication using touchscreen gestures, in: Homeland Security (HST), 2012 IEEE Conference on Technologies for, Citeseer, 2012, pp. 451–456.

[36] Y. Meng, D. S. Wong, R. Schlegel, et al., Touch gestures based biometric authentication scheme for touchscreen mobile phones, in: International Conference on Information Security and Cryptology, Springer, 2012, pp. 331–350.

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

[38] N. Sae-Bae, N. Memon, K. Isbister, K. Ahmed, Multitouch gesture-based authentication, IEEE transactions on information forensics and security 9 (4) (2014) 568–582.

[39] M. Smith-Creasey, M. Rajarajan, A continuous user authentication scheme for mobile devices, in: Privacy, Security and Trust (PST), 2016 14th Annual Conference on, IEEE, 2016, pp. 104–113.

[40] M. Shahzad, A. X. Liu, A. Samuel, Behavior based human authentication on touch screen devices using gestures and signatures, IEEE Transactions on Mobile Computing 16 (10) (2017) 2726–2741.

[41] T. V. Nguyen, N. Sae-Bae, N. Memon, Draw-a-pin: Authentication using finger-drawn pin on touch devices, Computers Security 66 (2017) 115 – 128. doi:https://doi.org/10.1016/j.cose.2017.01.008.
URL http://www.sciencedirect.com/science/article/pii/S0167404817300123

[42] W. Meng, W. Li, D. S. Wong, J. Zhou, Tmguard: a touch movement-based security mechanism for screen unlock patterns on smartphones, in: International Conference on Applied Cryptography and Network Security, Springer, 2016, pp. 629–647.

[43] A. S. Arif, A. Mazalek, Slide-to-unlock revisited: Two new user authentication techniques for touchscreen-based smartphones, in: Proceedings of the 11th International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services, MOBIQUITOUS '14, ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), ICST, Brussels, Belgium, Belgium, 2014, pp. 389–390. doi:10.4108/icst.mobiquitous.2014.257921.
URL http://dx.doi.org/10.4108/icst.mobiquitous.2014.257921

[44] Y. Meng, D. S. Wong, L.-F. Kwok, Design of touch dynamics based user authentication with an adaptive mechanism on mobile phones, in: Proceedings of the 29th Annual ACM Symposium on Applied Computing, SAC '14, ACM, New York, NY, USA, 2014, pp. 1680–1687. doi:10.1145/2554850.2554931.
URL http://doi.acm.org/10.1145/2554850.2554931

[45] W. Meng, D. S. Wong, S. Furnell, J. Zhou, Surveying the development of biometric user authentication on mobile phones, IEEE Communications Surveys Tutorials 17 (3) (2015) 1268–1293. doi:10.1109/COMST.2014.2386915.

[46] M. D. Ale(axew3), Audio recorder, https://github.com/Satvikkul/monowaw-AudioRecorder/.

[47] J. Huo, Dynamic time warping and fft: A data preprocessing method for electrical load forecasting, 2018.

Appendix A

Web application And Application Server

Recording sensor data and creating JSON

```
if (window.DeviceOrientationEvent && bool == true)
{
window.addEventListener("deviceorientation", processGyro);
}
else
{
alert("DeviceOrientation is not supported");
}

if(window.DeviceMotionEvent &&  bool == true)
{
window.addEventListener("devicemotion", motion);
}
else
{
alert("DeviceMotionEvent is not supported");
}
console.clear();

}
function processGyro(event)
{
if (bool == true) {
xG =document.getElementById("alpha").innerHTML=event.alpha;
yG =document.getElementById("beta").innerHTML=event.beta;
zG =document.getElementById("gamma").innerHTML =event.gamma;

arr_xG.push(xG);
arr_yG.push(yG);
```

```
arr_zG.push(zG);
}
}

function motion(event)
{
if (bool == true) {
xAcc = document.getElementById("X").innerHTML=event.
    accelerationIncludingGravity.x;
yAcc = document.getElementById("Y").innerHTML=event.
    accelerationIncludingGravity.y;
zAcc = document.getElementById("Z").innerHTML =event.
    accelerationIncludingGravity.z;
arr_xAcc.push(xAcc);
arr_yAcc.push(yAcc);
arr_zAcc.push(zAcc);
}
```

Getting accelerometer data

```
@app.route('/sensordata', methods=['POST'])
def add_message1():
content = request.json
xAcc = list(content['xAcc'].values())
xAcc1 = xAcc[1:]
yAcc = list(content['yAcc'].values())
yAcc1 = yAcc[1:]
zAcc = list(content['zAcc'].values())
zAcc1 = zAcc[1:]
lbl = content['lbl'][1:]

txtfile = "/path/to/file"
with open(txtfile, "w") as output:
writer = csv.writer(output, lineterminator='\n')
for val in zAcc1:
if val:
writer.writerow([val])

txtfile = "/path/to/file"
with open(txtfile, "w") as output:
writer = csv.writer(output, lineterminator='\n')
for val in lbl:
if val:
writer.writerow([val])
```

```
f = open('/path/to/file', 'r+')
data = f.read()
f.close()
for i in data:
if i:
acc_data = data.split()
x = np.array(acc_data)
x = np.asfarray(x, float)
noisy = array(x)
pkind = array(argrelmax(noisy, order=35))
number_of_peaks_acc = pkind.size
#print("Peaks: ",number_of_peaks_acc)
# readzAcc1()
#number_of_peaks_acc1 = readzAcc1()
return number_of_peaks_acc
return 'success'
```

Recording microphone data and sending it to server

```
A plugin for recording the output of Web Audio API nodes
    developed by Matt Diamond and Axew3 is used in this work. This
     code extends the work done by- https://github.com/mattdiamond
    /Recorderjs and https://github.com/axew3/monowaw-AudioRecorder
var XHR = new XMLHttpRequest();
XHR = new XMLHttpRequest();
XHR.onreadystatechange = function() {
if (XHR.readyState === 0 || XHR.readyState === 1) {
 //console.log(XHR.response;
} else if (XHR.readyState === 3) {
//console.log(XHR.response);
} else if (XHR.readyState === 4) {
 console.log(XHR.response + ' done');
 //var disp = XHR.responseText;
disp = XHR.responseText;
 // document.getElementById("output").innerHTML = disp;
check();

} else {
 //console.log(XHR.response);
}
}
XHR.open("GET", "http://localhost:8890/sensordata");
XHR.send();
```

```
function saveAudio() {
audioRecorder.exportWAV( doneEncoding );
}

function gotBuffers( buffers ) {
var canvas = document.getElementById( "wavedisplay" );
drawBuffer( canvas.width, canvas.height, canvas.getContext('2d'),
    buffers[0] );
audioRecorder.exportWAV( doneEncoding );
}

Recorder.setupDownload( blob, "knocks" + recordIndex + ".wav" );

function doneEncoding( blob ) {
if(recordIndex < 1)
{
Recorder.setupDownload( blob, "knocks" + recordIndex + ".wav" );
recordIndex++;
}
else
{
Recorder.setupDownload( blob, "knocks" + "1" + ".wav" );

}

}
function check(){
if(disp == "Match"){
YES();
}
else if (disp == "Mismatch") {
NO();
}
}

function YES() {
var x = document.getElementById("snackbar");
x.className = "show";
setTimeout(function(){ x.className = x.className.replace("show",
    ""); }, 3000);
}

function NO() {
```

```
var x = document.getElementById("snackbar1");
x.className = "show";
setTimeout(function(){ x.className = x.className.replace("show",
    ""); }, 3000);
// var out = "Start Taps";
// var xyz = document.getElementById("output").innerHTML = out;
}

function myFunction() {

//var x = document.getElementById("record").name;
document.getElementById("record").innerHTML = text + "<br>" +
    text0;

}

function myFunction1() {

//var x = document.getElementById("record").name;
document.getElementById("record").innerHTML = text1;

}
function toggleRecording( e ) {

if (e.classList.contains("knocks")) {

if (count = 1)
{
audioRecorder.stop();
myFunction1();
send();
//check();
e.classList.remove("knocks");
audioRecorder.getBuffers( gotBuffers );
count = 0;
}

} else {
// start recording

if (!audioRecorder)
return;
e.classList.add("knocks");
```

```
audioRecorder.clear();
myFunction();
//YES();
audioRecorder.record();

count = 1;

}

}
```

Returning result to server

```
@app.route('/sensordata', methods=['GET'])
def add_message():
time.sleep(0.5)
num_files = len(fnmatch.filter(os.listdir("/path/to/directory"),
    '*.wav'))
print(num_files)
if (num_files == 2):
rename1(), rename2(), rename3(), rename4()
if (num_files == 4):
MSE3 = getfft3()
minMSE, maxMSE = mseRange()
DTW3 = getdtw3()
minDtw, maxDtw = dtwRange()
number_of_peaks_acc1 = readzAcc1()
print("peak1: ",number_of_peaks_acc1)
number_of_peaks_acc = saveToDisk()
print("peak: ", number_of_peaks_acc)
# if ((minDtw <= DTW3 <= maxDtw) or (minMSE <= MSE3 <= maxMSE) or
    (number_of_peaks_acc==number_of_peaks_acc1)):
if ((number_of_peaks_acc==number_of_peaks_acc1)):
print("match")
return 'Match'
else:
print("mismatch")
return 'Mismatch'
else:
return "\n"
```

Calculating Fast Fourier Transform and Mean Square Error

```
fs_rate, signal = wavfile.read('/path/to/file')
peak_normalized_audio = pyln.normalize.peak(signal, -1.0)
```

```
#print ("Frequency sampling", fs_rate)
l_audio = len(signal.shape)
#print ("Channels", l_audio)
if l_audio == 2:
    signal = signal.sum(axis=1) / 2
N = signal.shape[0]
#print ("Complete Samplings N", N)
secs = N / float(fs_rate)
#print ("secs", secs)
Timestamp = 1.0/fs_rate
t = scipy.arange(0, secs, Timessample)
FFT = (scipy.fft(signal))
FFT_side = FFT[(range(N//2))] # one side FFT range
freqs = scipy.fftpack.fftfreq(signal.size, t[1]-t[0])
fft_freqs = np.array(freqs)
freqs_side = freqs[range(N//2)] # one side frequency range
fft_freqs_side = np.array(freqs_side)
fs_rate1, signal1 = wavfile.read('/path/to/file')
l_audio1 = len(signal1.shape)
if l_audio1 == 2:
    signal1 = signal1.sum(axis=1) / 2
N1 = signal1.shape[0]
secs1 = N1 / float(fs_rate1)
Timesample1 = 1.0/fs_rate1  # sampling interval in time
t1 = scipy.arange(0, secs1, Timessample1)
FFT1 = (scipy.fft(signal1))

FFT_side1 = FFT1[(range(N1//2))]  # one side FFT range
freqs1 = scipy.fftpack.fftfreq(signal1.size, t1[1]-t1[0])
fft_freqs1 = np.array(freqs1)
freqs_side1 = freqs1[range(N1//2)]  # one side frequency range
fft_freqs_side1 = np.array(freqs_side1)
X = min(N, N1)
mse = mean_squared_error(FFT[:X], FFT1[:X])
MSE = int(abs(mse/100000))
return MSE
```

Calculating distance between signals using Dynamic Time Warping

```
y1, source1 = librosa.load('/path/to/file')
y2, source2 = librosa.load('/path/to/file')
mfcc1 = librosa.feature.mfcc(y1, source1)
mfcc2 = librosa.feature.mfcc(y2, source2)
# print(mfcc1.shape)
```

```
dist, cost, acc_cost, path = dtw(mfcc1.T, mfcc2.T, dist=lambda x,
    y: norm(x - y, ord=1))
DTW = (int(dist))
#print("DTW: ", DTW3)
return DTW
```