Wristband Authentication with Apple Watch

Pedro Macedo
Faculty of Computer and
Information Science
Ljubljana, Slovenia
Email: pf06708@student.uni-lj.si

***Abstract*—Smart devices allows us to have more data about a person. For example, the person's location in a specific time, perform a person's ECG and a lot other features. Nonetheless, due to this rapidly growing of a person's information, the data security is more compromised, and needs to be saved using different types of authentication. That's why the society invented the current authentication methods. However, this authentication methods have become more tiresome with the increasing of person's personal smart devices. In consequence, people need to pass, for each of their smart devices, the respective security method. They lack the method of logging-in one time on a device, and log on the other person's personal devices without the need to pass throught the authentication process in each smart device. This paper will present an improved authentication method, which will take a smartwatch to work as a token to unlock the other personal devices.**

I. Introduction

Today, we are surrounded by an ever-growing number of smart devices, ranging from smartphones and tablets to wearable and IoT gadgets. This devices have become a part of our daily life, collecting and generating an unprecedented volume of data that must be securely managed and accessed. As the number of interconnected devices increase, so does the complexity of authenticating and securing access to them. Current authentication methods, such as PINs, biometrics (e.g., fingerprint and facial recognition) are design to handle only individual devices. This methods becomes exhausting when user want to authenticate in several smart devices simultaneously. As so, this approach becomes inefficient and introduces a poor user experience. Moreover, many existing authentication methods, such as password authentication possesses high security risks, because passwords can be stolen. Furthermore, biometrics, while convenient to use, have crisis when dealing in different types of environments, such as, poor lightning for facial recognition or dirty hands for fingerprint recognition. Hence, this project aims to to address this limitations of current authentication methods by proposing a more efficient solution: a wristband-based authentication system using an *Apple Watch*.

A. Related work

Our approach was deliberated in some studies, such as *WearLock* [2] and *"Flick me once and I know it's you!"* authentication approach [3]. *WearLock* uses "acoustic tones as tokens to automate the unlocking securely" [2], while the other authentication method analysis the behaviour of the user's wrist, and tries to collect the the users flicking wrist movement in two different scenarios, sitting and standing [3].

II. Application core

In this section, we will present the purpose of our application. The main goal is unlocking the user's iPhone using his Apple Watch. The main challenge will be to match the user's watch sensor with the readings taken from the non-authenticated iPhone, and, in case the two sensors patterns match, authenticate the iPhone.

A. Motion data collection

Our plan is to use the accelerometer and gyroscope sensors to capture real-time motion data. Both the Apple Watch and the iPhone are equipped with accelerometers, which measures acceleration along the three axis x, y and z axes, and gyroscope which measures angular velocity along theses axes. Combining the data of these two sensors gives a detailed picture of the movement and orientation of each device. In order to have the best and correct measurements to perform this experience, we will collect data from two different sensors of both the iPhone and the Apple Watch: the accelerometer and the gyroscope. To collect and process these sensors data, we will use the *CMMotionManager* Swift class [4]. In the initial phase, the system will require the user to press a button on the app to initiate the unlocking process. Upon clicking the button, the user will make a movement with the Apple Watch and the iPhone simultaneously (this is the part where the sensing starts and it will collect data for 5 seconds after the button is pressed), which will trigger the iPhone to unlock. However, our goal is to evolve this process into a fully automated system. Later, we aim to remove the need of the user to press a button ans instead implement a method that continuously monitors the movement of the Apple Watch. This way, when the correct movement pattern is detected, the iPhone will automatically unlock without any manual input. This approach will improve user accessibility by making the authentication process seamless. Moreover, in order to prevent all types of users, in this study, it will be conducted experiments to determine the best sampling rate value, which needs to be capable of detecting correctly the device's motion rapid movements.

B. iPhone unlocking process

In order for the user to unlock the iPhone, the system needs to check, at least, two constraints: (i) **proximity** and **motion**. The *Apple Watch* must be near the iPhone (no more than 3 meters of distance from each other [5]). The model will be able to calculate the proximity by using *Bluetooth*, more specifically the *Received Signal Strength Indicator (RSSI)* [5]. Additionally, the model will calculate the motion of both devices and try to find if they are synced. The model will need to apply a basic transformation for each machine axis, in order to align the both of them (this is, in order for both to have the same axis reference), as in Figure 1.

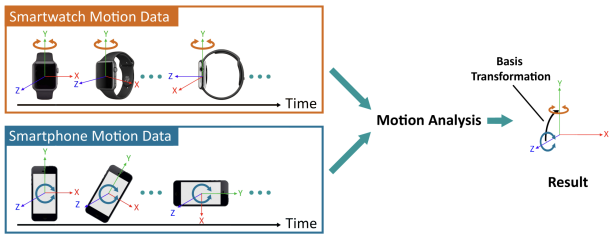

Figure 1: Motion Detection between devices

Once satisfying both the proximity and motion conditions, the systems sends an **unlock signal** to the iPhone. In this process, the Apple Watch will act as a secure token. Once both devices movements match, it triggers a secure *handoff* of the authentication signal from the Apple Watch to the iPhone via Bluetooth. The Apple Watch will act as an authentication proxy, allowing the iPhone to trust the Apple watch's credentials.

C. Technology

As mention before, we will use the **Swift** programming language, since it's the primary language for developing applications for *iOS*, *watchOS* and *macOS*. *Swift* is integrated with useful Apple's frameworks, such as *CoreBluetooth*, *CoreMotion* and *WatchConnectivity*. We will also use **SwiftUI** framework to create the user interface for both the iPhone and Apple Watch, making it easier to providing feedback the user and visualize data.

III. Work plan

Week 2 corresponds to the week from 07.10.2024 to 11.10.2024.

Week	Work
2	Project initiation and requirements gathering
3	Research on motion analysis and device co-authentication
4	System design, defining motion patterns for device interaction
5	Data collection for device motion analysis
6	Preprocessing of motion data
7	Develop initial authentication modules + Implement and train the Machine Learning model
8	mid semester presentation + discussion over feedback
9	Model validation and performance tuning
10	Develop device communication protocols
11	Test authentication mechanisms in various scenarios
12	Fine-tune motion analysis for better accuracy + Integration and system testing
13	Final touches and applications refinement + Presentation preparation
14	Demo and final presentation

Table I: Work plan per week

IV. Application evaluation

In order to evaluate and test our application, we will collect *feedback* from a group of users and analyse their behaviours and how the application behaves, and also the overall user-friendliness (ease of use design). The *feedback* will include a questionnaire to express the user's satisfaction with the overall experience from a scale from 1 to 10. Additionally, we will compare the performance using the follwing evaluation metrics:

- **True-Positive Rate (TPR)**: Measures the success rate of the cases that were predicted correctly has a positive case.
- **False-Positive Rate (FPR)**: Measures the error rate of the cases that were predicted as positive cases when were actually negative cases.
- **True-Negative Rate (TNR)**: Measures the error rate of the cases that were predicted correctly as negative cases.
- **False-Negative Rate (FNR)**: Measures the error rate of the cases that were predicted as negative cases when were actually positive cases.
- **Equal Error Rate(EER)**: Reflects the method's accuracy:

$$EER = \frac{FPR + FNR}{2}$$

Effectively, by calculating the accuracy metric, we aim to refine the application and enhance both usability and reliability. Moreover, we will also conduct a few tests at various sampling rates (20Hz, 50Hz and 100Hz) in order to test the effect on data quality, evaluation metrics and overall user experience. The sampling rate value will be adjusted based on the conclusions taken from the tests that will be performed.

V. Conclusion

In conclusion, ensuring the security and privacy of the transmitted motion data is critical to ensure the success of the propose *Wristband Authentication Apple Watch* system. To prevent unauthorized access, all data exchange between the *Apple Watch* and the *iPhone* must be encrypted using secure protocols such as TLS and SSL. These encryption methods will ensure that the motion data is protected during transmission, minimizing the risk of data leaks.

References

[1] Apple, *Take an ECG with the ECG app on Apple Watch*, Apple company, September 03, 2024. Available at: https://support.apple.com/en-us/120278

[2] S. Yi, Z. Qin, N. Carter and Q. Li, "WearLock: Unlocking Your Phone via Acoustics Using Smartwatch," 2017 IEEE 37th International Conference on Distributed Computing Systems (ICDCS), Atlanta, GA, USA, 2017, pp. 469-479, doi: 10.1109/ICDCS.2017.183. Available at: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=arnumber=7979992

[3] Y. Li, F. Ferreira and M. Xie, "Flick me once and I know it's you! Flicking-based Implicit Authentication for Smartwatch," 2023 International Conference on Computing, Networking and Communications (ICNC), Honolulu, HI, USA, 2023, pp. 318-323, doi: 10.1109/ICNC57223.2023.10074521. Available at: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=arnumber=10074521

[4] Apple Company, CMMotionManager, Available at: https://developer.apple.com/documentation/coremotion/cmmotionmanager

[5] UbiComp'13, BlueEye – A System for Proximity Detection Using Bluetooth on Mobile Phones, Available at: https://dl.acm.org/doi/pdf/10.1145/2494091.2499771

List of Figures