Observation-Resistant Multifactor Multimodal Authentication

Aleksander Furnes Mallasvik

Masteroppgave
Master i informasjonssikkerhet
30 ECTS
Avdeling for informatikk og medieteknikk
Høgskolen i Gjøvik, 2010

Avdeling for
informatikk og medieteknikk
Høgskolen i Gjøvik
Postboks 191
2802 Gjøvik

Department of Computer Science
and Media Technology
Gjøvik University College
Box 191
N-2802 Gjøvik
Norway

Observation-Resistant Multifactor Multimodal Authentication

Aleksander Furnes Mallasvik

30th June 2010

Abstract

This thesis investigates the use of hand gestures as an additional modality in authentication schemes, to thwart the risk of observation (shoulder surfing) attacks. We used the accelerometer already embedded in the iPod Touch to gather accelerometer signals, which were used to conduct experiments on how accurately we could recognize and differentiate different gestures. We restricted ourselves to a pre-defined set of gestures, and achieved an EER of 5% on the controlled wrist movements, and 8% after including two circular motions. The algorithms we used were tailored to fit the limited computational power of the iPod Touch, as we needed a recognition module that could be used in real time for our authentication schemes. After assessing the characteristics of the different hand gestures, we developed two unique authentication schemes that incorporate hand gestures as an additional modality for authentication. We developed suitable attack scenarios, and found that both schemes adds additional entropy to the scheme, as well as a significant amount of shoulder surfing resistance.

Sammendrag

Denne oppgaven ser på bruken av håndbevegelser som en ekstra modalitet i autentiseringsskjemaer, for å minske risikoen for observasjonsangrep. Vi utviklet et program for iPod Touch som bruker den innebygde akselerasjonsmåleren til å samle akselerometerdata. For å undersøke hvor nøyaktig vi kan gjenkjenne og skille mellom forskjellige håndbevegelser, ble dette programmet brukt til å samle data om 6 forskjellige bevegelser fra totalt 38 deltagere. Vi begrenset oss til et forhåndsdefinert sett av bevegelser, og oppnådde en EER på 5% på de kontrollerte håndleddsbevegelsene, og en EER på 8% da vi inkluderte to sirkulære bevegelser. Algoritmene vi brukte var skreddersydd for å passe den begrensede regnekraften i en iPod Touch, da det var viktig at gjenkjenningsmodulen kunne brukes av våre autentiseringsskjemaer i sanntid. Etter å ha oppnådd tilfredsstillende feilrater utviklet vi to unike autentiseringsskjemaer som bruker håndbevegelser i autentiseringen, og det ble utviklet passende angrepsscenarier for å verifisere styrken av disse. Sikkerheten til autentiseringsskjemaene ble vurdert, og vi fant at begge skjemaene øker sikkerheten og resistansen mot observajonsangrep betraktelig.

Acknowledgments

First of all I would like to thank all the participants that participated in my experiments. Without them, it would have been impossible to conduct this project. Secondly, I would like to thank my supervisor, Stephen D. Wolthusen, whom during the entire period have been of great help and assistance. He have throughout the project period shown great interest in my work, and always been available for questions and motivational talks. I would also like to thank my co-supervisor Patrick Bours, whom have been a perfect sparring partner throughout the project period. My good friend Magnus Mustorp, along with Rune at Gjøvik Filmverksted deserves a special thank, as they set aside a whole day for helping me conduct the experiments on my challenge-response scheme. Last but not least, i would also like to thank my classmates for adding a social aspect into all the hours we spent at the master lab.

Aleksander Furnes Mallasvik, 30th June 2010

Contents

Abstract . . . . **iii**
Sammendrag . . . . **v**
Acknowledgments . . . . **vii**
Contents . . . . **ix**
List of Figures . . . . **xiii**
List of Tables . . . . **xv**
List of Algorithms . . . . **xvii**
1 Introduction **1**
1.1 Topics covered by the project 1
1.2 Keywords 1
1.3 Problem description 1
1.4 Justification, motivation and benefits 2
1.5 Research questions 2
1.6 Contributions 3
1.7 Choice of methods 3
1.8 Chapter overview 3
2 Background **5**
2.1 A brief introduction to the field of authentication 5
2.1.1 Authentication 5
2.2 An introduction to accelerometers 9
2.2.1 Sensor details 10
2.3 Human considerations 11
2.4 Recognition specific algorithms and methods 12
2.4.1 Dynamic Time Warping 12
2.4.2 Synthetic recognition 13
3 Related Work **15**
3.1 Gesture recognition 15
3.2 Observation resistant protocols 19
4 Device capability and usability **23**
4.1 Experiment details 23
4.1.1 Experiment goal 23
4.1.2 Experiment setup 23
4.2 Experiment protocol 25
4.3 Acquisition program 26
4.4 Participants 27
5 Template extraction and generation **29**
5.1 Template creation 29
5.1.1 General template creation 32
5.1.2 Our median implementation 34
5.2 Gesture characteristics 34

5.2.1 General description 35
5.2.2 Right and left flip 35
5.2.3 Front and Back flip 37
5.2.4 Circular motions 39
5.3 Discussion of results 40
5.3.1 Personalization of templates 40
6 Recognition performance 41
6.1 Distance metric notes 41
6.1.1 Costs 41
6.1.2 Combining the three vectors 42
6.2 Gesture distinctiveness 42
6.2.1 Comparison details 43
6.3 Using the general templates for comparison 44
6.3.1 Verification dataset details 46
6.4 Additional notes 46
6.4.1 Implementation and complexity details 47
7 Authentication schemes 49
7.1 Gesture and PIN-code based authentication 49
7.1.1 Validation experiment 50
7.2 Challenge-response scheme 52
7.2.1 Protocol flow 53
7.2.2 Theoretical security assessment 55
7.2.3 Attack experiment 58
8 Analysis and results 63
8.1 Gesture and PIN-code based authentication 63
8.1.1 Simple login scenario 63
8.1.2 Random login scenario 64
8.1.3 Qualitative participant feedback 64
8.1.4 Security assessment 65
8.2 Challenge-response scheme 66
8.2.1 Achieving adequate video quality 66
8.2.2 General information about the security assessment 67
8.2.3 Results from the attack experiment 69
8.2.4 Deducing the PIN-code 70
9 Conclusion 73
10 Future Work 75
Bibliography 77
A Signal graphs 83
A.1 Right and left flip 83
A.2 Front and back flip 86
A.3 Circular motions 90
B Flow charts for our authentication schemes 93
B.1 Gesture and PIN-code based authentication scheme 93
B.2 Challenge-response scheme 94
C Participant agreement declaration 95
D Experiment protocol description - Gesture-Placement based scheme 97

D.1 Detailed information about the experiment protocol 97
D.2 Experiment description for the participants 98

List of Figures

1 An example of a linear accelerometer 9
2 The orientation of the accelerometer in the device 10
3 DTW path alignment 13
4 Wrist rotations and degree of rotation possible along each axis 17
5 Authentication protocol developed by Patel *et al.* 19
6 Illustrates the device's start position 23
7 Shows the views in the acquisition program 27
8 Shows the age distribution in the dataset 27
9 Illustrates how a general template is created 32
10 Cross-plot of a number of different *left flip* personal templates. 36
11 Shows the general template for a *left* and *right flip*. 37
12 Shows the general template for a *front* and *back flip* 38
13 Cross-plot of the left and right circles general templates 39
14 Illustrates the benefits with using general templates 46
15 Shows the 2-tuple input sequence 50
16 An attackers chance of successfully attacking the scheme 58
17 Example of a challenge presented by the scheme 61
18 Setup used to achieve adequate video footage 67
19 Shows the valuable areas of a gesture signal 75
20 Cross plot of different right flip personal-templates 83
21 Cross plot of different left flip personal-templates 84
22 Cross plot of the general templates from a left and right flip 85
23 Cross plot of different back flip personal-templates 86
24 Cross plot of different front flip personal-templates 87
25 Cross plot of the general templates from a front and back flip 88
26 Cross plot of different left circle personal-templates 90
27 Cross plot of different right circle personal-templates 91
28 Cross plot of the general templates from a right and left circle 92
29 Flow chart for our first authentication scheme 93
30 Flow chart for our challenge-response scheme 94

List of Tables

1 DTW distances between personal templates and raw sequences 31
2 DTW distances between personal and general templates 33
3 Error rate table for the all-to-all comparison 44
4 Error rates when using general templates 45
5 Shows the gesture-placement relationship 50
6 An example of a gesture-color association 53
7 Distribution generated from the PRNG 54
8 Illustrates the gesture-color and PIN-code associations 68
9 An adversaries notes after observing an unprotected login 68
10 An adversaries notes after observing a protected login 69
11 Example: user reveals gesture-color associations 69
12 Protecting the gesture-color associations 70

List of Algorithms

1 Template creation algorithm 34
2 Algorithm for the *all-to-all* comparison 43
3 EER calculation with general templates 45
4 Algorithm used to perform identification and recognition 48
5 Algorithm for the *one-to-one* recognition module 48

1 Introduction

1.1 Topics covered by the project

As mobile devices are increasingly being used for high security applications like stock trading, confidential e-mails and SMS banking, there is a great need for robust authentication mechanisms. The main topic of this project is the incorporation and development of a set of novel, unobtrusive authentication mechanisms, which utilizes hand gestures in conjunction with PIN codes in real time, to thwart the threat of observation attacks.

Hand gestures can range from a simple tilt to more complex motions like circles, and it was our goal to recognize and measure such motions by the usage of accelerometers in mobile devices. Although accelerometers provide us with a limited set of data, we have shown that we can recognize and distinguish between different hand gestures. Recognizing hand gestures allows us to, in the simplest case, include gestures as a part of an authenticator (e.g., we can have the user move his device in a predefined way between entering digits of a PIN-code), providing additional entropy depending on the sensitivity and resolution of the accelerometers used. Investigating how accurately we could model gestures by the usage of accelerometers was therefore a vital part of the thesis.

A problem with using behavioral features as apposed to passwords for authentication, is that they are not either right or wrong. A person will never present the exact same gesture twice, and this needed to be taken into consideration. Since we did not use the hand gestures for biometric purposes, but rather as an additional modality in our authentication schemes, we developed modules that not only differentiate one gesture from another, but also recognizes specific gestures, even though they are conducted by different people. For this reason, we had to restrict ourselves to a set of predefined hand gestures.

Adding additional features to protocols where PIN-codes and tokens are the only present authentication factors, allow us to mitigate the risk of observability (shoulder surfing). We developed two authentication schemes that incorporate hand gestures as a second modality, and we experimentally validated their resilience against shoulder surfing attacks. Suitable experimental protocols were designed for this purpose.

1.2 Keywords

Hand gestures, Accelerometers, Modulation of gestures, Multi-modal authentication, Challenge response protocols, PIN-codes, Authentication protocols, Observation attacks

1.3 Problem description

Identity theft and bank accounts being emptied by thieves is an ever increasing problem. The fact is, that the security of an application relies completely on the authentication mechanism used, and it is highly undesirable that a thief should be able to withdraw money from a victims bank account after simply observing his PIN-code and stealing his bank card. Similarly, an increasing amount of companies store sensitive data on portable devices which their employees use outside the office. Should such information fall into the wrong hands, this could have serious effects on the businesses revenue and financial situation. Therefore, applying stronger security mechanisms on these devices should be seen as crucial by all companies. This thesis investigates how we can incorporate hand gestures as a part of a multimodal challenge-response authentication process, to increase the security of the device and thwart the risk of shoulder surfing attacks.

Although this project is not about bank card security, but rather the general aspect of including hand gestures as an authentication factor in mobile devices, it is a good example of the problem area. To clarify, it is not this thesis goal to use hand gestures as biometric features, but to be able to recognize different gestures and use these as an additional modality in authentication mechanisms in real time. We faced numerous research problems, but the foremost important one was to investigate whether we could identify a way to analyze the accelerometer data produced by an iPod Touch in a way that allowed us to recognize pre-defined hand gestures in real time, and use these extra parameters in an authentication process (e.g., via a challenge-response protocol).

1.4 Justification, motivation and benefits

We have shown that we can, by including hand gestures as an additional modality in challenge-response protocols, significantly increase the workload for an attacker wanting to gain unauthorized access to a PIN-code protected device. Since we can recognize and measure hand gestures precisely enough for authentication usage, we can implement them as features in multimodal authentication mechanisms, which alone improves the authentication schemes entropy significantly. We have found that our schemes introduces a significant amount of shoulder surfing resistance, even under rigid attack scenarios.

Although we did not look at the biometric aspect of hand gestures, we performed a thorough analysis of the characteristics describing each gesture, in order to achieve good recognition rates. Our research can therefore be used as a building block for future research, as we investigated all aspects that could affect our results. Also, since hand gestures have not been precisely modeled by the usage of accelerometers alone before, our aim was at recognizing hand gestures, not the owner of one. Since we utilized already embedded accelerometers, we showed that we can drastically improve security without adding extra cost for devices such as smart phones.

1.5 Research questions

The following research questions will be addressed throughout this thesis:

1. How detailed is the information derived from the accelerometer in an iPod Touch, and how precisely can this information be used to recognize and differentiate gestures?
   - When information content is constrained by time and resolution of devices?

2. To what degree does the inclusion of hand gestures in multimodal challenge-response schemes increase the degree of difficulty for an attacker wanting to perform observation attacks?
   - How does the performance of the device, in terms of speed, affect the degree of observability obtained by an attacker?

3. Which combination of modalities and protocols, within the constrains imposed by both device and usability, yields the most observation resistance?

1.6 Contributions

The main contribution of this master thesis was the development of a set of novel authentication mechanisms based on the combination of accelerometer derived gestures with other modalities in direct, and challenge-response combinations. A software kit for the iPhone platform was developed, which can record and recognize hand gestures, and also, more importantly, use hand gestures as a part of multimodal challenge-response authentication schemes. We also investigated how reproducible hand gestures are, and how accurately we could measure, recognize and distinguish them. Based on the information derived from the related work study, we restricted ourselves to a predefined repository of six gestures, which was thoroughly analyzed. We have shown that by including hand gestures as a part of a multimodal authentication scheme, we can mitigate the risk of shoulder surfing attacks and increase the overall robustness of the authentication scheme significantly.

We have developed and experimentally verified the robustness of two unique challenge-response schemes for the iPhone/iPod platform that utilizes a multimodal approach with both PIN-code and hand gesture as parameters for authentication.

1.7 Choice of methods

As answering the research questions required to draw on several research areas and methods, we do as a result describe them in the chapters where they are applied.

The methods used to gather hand gesture samples are described in Chapter 4, while a signal analysis, along with the methods used to recognize and distinguish between different hand gestures, is presented in Chapter 5. The method used to calculate error rates is described in Chapter 6, and the protocols and methods used to create and assess our authentication schemes in Chapter 7.

1.8 Chapter overview

This section presents a brief summary of the content in the different chapters in this thesis.

Chapter 2

Introduces topics that are important for the accomplishment of this thesis, as well as other non-trivial details that increases the readers chance of understanding the discussions and prerequisites taken.

Chapter 3

Gives an overview of related work that has been conducted in the fields of gesture recognition and observation resistant authentication protocols.

Chapter 4

Describes the methods used to perform the data acquisition experiment. The data gathered in this experiment formed the basis for both the signal analysis and the template generation described in Chapter 5. Further on, the dataset was also used in the distinctiveness experiment in Chapter 6.

Chapter 5

Presents both the analysis of the accelerometer signals produced by our hand gestures, as well as a description of the template creation method used.

Chapter 6

Presents the analysis performed on the distinctiveness of hand gestures, and the recognition rates obtained by our recognition modules.

Chapter 7

Presents both the anatomy and a theoretical security evaluation of the two authentication schemes we developed. It also describes the experiments that we ran on the schemes in order to test their security properties.

Chapter 8

Presents an analysis of the results from the experiments described in Chapter 7, as well as an overall discussion of the security and usability of the two authentication schemes.

Chapter 9

Presents a conclusion which summarizes and highlights the most important findings in this thesis.

Chapter 10

Presents a number of topics that would be interesting to investigate in future research.

2 Background

This chapter introduces topics that are important for the accomplishment of this thesis, as well as other related non-trivial details.

Section 2.1 gives a brief introduction to the field of authentication, while Section 2.2 gives an introduction to accelerometers in general. Some hardware specific information about the accelerometer utilized in the experiments is also presented in this section. Section 2.3 describes the human aspects that had to be taken into account when modeling hand gestures, and Section 2.4.1 presents a brief description of *Dynamic Time Warping*, which is the recognition algorithm we utilized in our experiments.

2.1 A brief introduction to the field of authentication

As authentication is one of the core topics of this dissertation, this section gives a brief introduction to the field. We will not go into details, but explain the key terms and refer to sources for further reading.

2.1.1 Authentication

Authentication is a wide area of research due to its many applications. In todays community, where almost all information is stored in computer systems, the need for secure and usable authentication mechanisms grows rapidly.

Authentication is the process of verifying a claimed identity, in contrast to identification, where we establish an identity. The topic of this master thesis is authentication rather than identification. In order to claim an identity we need to present something to the authentication system, where the easiest example is a username. After having presented this unique username, the authentication system will expect proof that the user is who he says he is, by, e.g, prompting the user for the password that matches the entered username. There are many ways of authenticating a person, but when we generalize, all factors fall into one of the following categories [1]:

- Something you *know*, like a PIN-code.
- Something you *have*, e.g, a smart-card.
- Something you *are*, e.g, a biometric property.

Knowledge

This is the oldest and most used authentication factor, and it includes PIN-codes, passwords, secret phrases etc. Although such factors are very user friendly, the main problem is the human aspect. Today people are forced to keep a high number of passwords and PIN-codes to access different services, and they can only remember a limited number of them. Also, when we enforce strict rules on the characters they can have in their passwords (to prevent attackers from guessing or cracking their passwords), people will eventually start writing them down or reusing them, which clearly is a security concern.

Possession

This factor implies that we authenticate ourselves by presenting something we possess. Examples are keys, smart cards and passports. Although we avoid the problems of people having to remember long, obscured passwords, we have other concerns when using this factor. Intuitively, it is very easy for an impostor to pose as another if a smart card is the only authentication factor needed. If an attacker simply steals or skims the card, then he have in theory gained the required details to pose as another. For these reasons, this factor is almost never used alone, but is often combined with other factors to increase security.

Intrinsic properties

Explaining this factor means moving on to the domain of biometrics. Biometrics properties are either *physiological* or *behavioral* [2]. The physiological properties describe static properties of our body, like for instance a fingerprint, while the behavioral properties focus more on the dynamics on how one person performs a certain action, an example is gait recognition.

To assess the strength of a biometric trait, we evaluate it against a wide range of properties which are divided into those that make the measurement practical (1-4), and those that make it possible to distinguish one person from another (5-7), hence making the authentication system secure. The properties are listed below as defined in [2].

1. Universality: Everyone should have the characteristic.
2. Distinctiveness: Any two persons should be sufficiently different in terms of the characteristic.
3. Permanence: The characteristic should be sufficiently invariant (with respect to the matching criterion) over a period of time.
4. Collectability: The characteristic can be measured quantitatively.
5. Performance: Refers to the achievable recognition accuracy and speed, the resources required to achieve the desired recognition accuracy and speed, as well as the operational and environmental factors that affect the accuracy and speed;
6. Acceptability: Indicates the extent to which people are willing to accept the use of a particular biometric identifier (characteristic) in their daily lives;
7. Circumvention: Reflects how easily the system can be fooled using fraudulent methods

The main benefit by using biometric features for authentication is that it is "impossible" for an attacker to steal another persons biometric property (at least for the physiological properties), and it is also hard to learn/copy another persons unique way of performing an action, as utilized in behavioral authentication. Using biometrics also removes the need for remembering passwords or carrying keys, as your body is the authentication factor. Even though we did not directly assess the strength of hand gestures as a biometric trait in this thesis, we did perform many of the same experiments and analyses as would be done in such a thesis. The only difference is that we, instead of looking at the uniqueness of a gesture for an individual person, recognize gestures and use these as additional factors in our authentication schemes. It is important to remember that many of the properties above is vital for non-biometric approaches as well, as they assess the usability and strength of the protocol. As this thesis aims to develop a set of observation resistant authentication schemes without relying on the biometric distinctiveness, *observation resistance* should be seen as the most important property when assessing the strength of our schemes.

The paper by Zhang *et al.* [2] contains more information on biometrics in general, the characteristics and popular methods used in the authentication process.

Result assessment

In the context of knowledge based authentication, a password/PIN-code entrance is always either 100% correct or wrong. In biometrics however, this is not the case. In the field of biometrics, no two biometric samples (even from the same person) are identical, and there is therefore always a risk that the system falsely accepts another persons biometric sample. Related to this, two terms emerge; *intra-class* and *inter-class* variance. Intra-class variance describes the variance between two samples of the same person. In biometrics, although a persons biometric property is unique, the measurement of it is never identical from time to time. It is therefore crucial to model and keep the intra-class variance as low as possible. Inter-class variance on the other hand, refers to the difference between samples from different persons. Intuitively, one wants to have this as high as possible, to minimize the *false match rate*.

In the context of this thesis we introduce two related terms; *intra-gesture* and *inter-gesture* variance. As we are not investigating the biometric distinctiveness, but the general distinctiveness of hand gestures, these terms represent the variances within the same gesture from different people, and between different gestures, respectively.

To compare two sequences, we utilize *distance metrics*, which calculates the distance from one sample to another, i.e. how similar they are. The more similar they are, the more certain we can be that the template and the probe descends from the same person. These metrics can be as simple as taking the absolute distance between two points, however, they are generally a bit more elaborate. Also, to make the best possible "reference" for each individual user (that is, build up the best possible template), every biometric system has an *enrollment phase*, where each user presents his biometric trait multiple times, from which a template is generated. Although it was not our goal to make a biometric system, we did generate templates describing how each user performs a specific gesture. These per-person templates was used to verify that the principal components describing each gesture was the same for all users.

When assessing biometric systems, the two most important errors are the *False Match Rate (FMR)* and the *False Non-Match Rate (FNMR)*. These rates deviates from the *False Acceptance Rate (FAR)* and the *False Rejection Rate (FRR)*, in the sense that they focus only on the analyzing algorithm, while FAR and FRR includes the *Failure to Capture* and *Failure to Acquire* rate. These rates incorporates error cases where users cannot enroll/produce a probe[1] sample due to for instance a physiological problem, like for instance if a user is unable to present his fingerprint due to an accident.

FMR does, as shown below, describe the rate to which impostors are wrongfully accepted [2]. This leads to cases where an impostor for instance gains unauthorized access to a restricted area. The FNMR on the other hand, deals with cases where a genuine

[1] A probe is in this context a gesture presented by a user to the scheme, which is compared to the template for that particular gesture.

attempt is rejected, i.e. that a legitimate user is wrongfully denied access.

$$\mathrm{FMR} = \frac{\text{Number of accepted impostor attempts}}{\text{Total number of impostor attempts}} \tag{2.1}$$

$$\mathrm{FNMR} = \frac{\text{Number of rejected genuine attempts}}{\text{Total number of genuine attempts}} \tag{2.2}$$

When tuning a biometric system, one strives to get the lowest possible FNMR and FMR, and this typically includes trying out numerous distance metrics and comparison algorithms to see what yields the best results. However, there is a tradeoff between FNMR and FMR. In any authentication system, one decides upon a value which is set as the *threshold* for a match or a non-match. This means that (depending on the implementation) if the total distance between two samples is below the threshold, then it is said to be a match, and if it is above; a non-match. This gives the researchers the possibility of tweaking the system in such a way that one can either set the threshold low and have a more secure system (this will increase the FNMR), or set a higher threshold and have a more user-friendly system with a lower FNMR, but also a higher FMR. This decision has to be taken based on what the system is going to be used for; if we are in a high security facility, we would employ a low threshold, while if it guards the access to our personal computer, a higher threshold might be preferable. The rate where the FNMR equals the FMR is referred to as the *equal error rate (EER)*, and is often used to evaluate how well a particular system performs.

Multi-modal authentication

In many cases, especially in *knowledge* and *possession* based authentication schemes, more than one factor is used to improve security. By combining for instance *knowledge* and *possession*, we can significantly increase the workload for an attacker. As an example, the attacker now has to steal both the victims access card and observe the PIN-code to gain access.

Fusion of multiple modalities in an authentication scheme can be done in many different ways, and how to combine the individual results from the PIN-code and gesture comparison modules was investigated extensively during this master thesis. An important aspect is that of correctness; since a PIN-code is either completely correct or wrong, the PIN-code should always be correct. Even though the gestures match perfectly, a wrong PIN-code entry should in no case lead to an acceptance. This is because the gestures can be seen as the weak modality among the two, and are implemented solely to thwart shoulder surfing attacks.

Challenge-response schemes

Challenge response in the context of authentication schemes, is in [3], defined as:

> A *challenge-response* authentication system is one in which *S* sends a random message *m* (the challenge) to *U*, and *U* replies with the transformation *r= f(m)* (the response). *S* validates *r* by computing it separately.

To simplify; Challenge response can in computer security be defined as protocols where authentication is based on the expected *response* from one entity to a specific action (the challenge) sent from the authenticating entity. In the simplest case, challenge response can be as easy as the ATM asking you for a PIN-code that matches the credentials in your bank card. In this thesis however, we explore more elaborate observation resistant schemes, as described in Chapter 7. There has been done a lot of research on these types of schemes in the last years, as discussed in Section 3.2.

2.2 An introduction to accelerometers

Since we in this thesis use accelerometers to measure gestures, this section introduces some background theory on the subject.

In principle accelerometers are electromagnetic devices that measure acceleration forces. This means that an accelerometer can measure both static (like for instance gravity) and dynamical (movement) forces, which means that we can figure out both how the device is tilted, and also in what direction it is moving. These capabilities have led to a broad usage area for accelerometers, and their application range from giving users improved HCI on their mobile phones while playing games, to measuring vibrations and other forces during for instance cargo transfer.

There exists different ways of measuring acceleration, but in this project we are going to use capacitive sensors which measure acceleration by measuring capacitance (the ability of a structure to store electric charges). We are not going to go into details of the physics, but capacitive sensors utilize two known rules of physics in order to measure acceleration; the first, Newtons second law, states that *force equals mass times acceleration*, and the second, Hooke's Law, states that the extension of a spring is proportional to the amount of load added to the spring itself (in our case the acceleration). When combined in practice, these laws work in such a way that the mass within moves when subjected to acceleration, which leads to a displacement of the capacitor holders. In order to measure the acceleration at a specific time, the displacement is translated to acceleration by the system, following specified rules. Figure 1 shows a simplified example. The movement in one direction (dislocation due to mass movement), will be the direct opposite in the two capacitors, thus giving us the possibility to determine the direction of the acceleration as well.

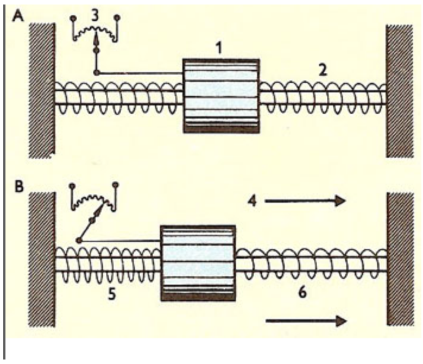


Figure 1: An easy example of an linear accelerometer. *If the device is moving at constant velocity, the mass (1), supported on a bar by springs (2), remains static and an intermediate reading is registered on the potentiometer (3). On acceleration in the direction of the bar (4), i.e., along the accelerometer's sensitive axis, inertia causes the mass to lag behind, compressing the spring behind it (5) and stretching the spring ahead of it (6): a high voltage is registered. On deceleration, inertia causes the mass to compress the spring ahead of it and stretch that behind it, and thus a low voltage is registered on the potentiometer.* Caption from [4].

Apple uses a LIS302DL [5] 3-axis accelerometer with +− 2g capability, which should give us a satisfactory description of the gestures, as described in Section 3.1. One of the biggest drawbacks of accelerometers is that they in theory only can measure linear motions. When the device containing the accelerometer is rotated, the acceleration due to gravity is mistaken for linear motion, and thus we cannot rely on accelerometers for accurately measuring horizontal rotations. To measure rotations we would need to use gyroscopes, which then again does not respond to linear movement. Therefore, to measure complex motions we would have had to combine the output from a gyroscope and a accelerometer. In newer devices such as the iPhone 4G, a gyroscope will be embedded, which will enable future research to investigate the effect of combining these. From a theoretical perspective, combining accelerometers and gyroscopes, should give a more descriptive image of the gestures.

For this thesis though, where we will mostly use tilting motions, recording linear motions will be satisfactory. Figure 2 shows the orientation of the accelerometer in the device, as well as explaining the basic details about the relationship between the device orientation, and the x, y and z coordinates. It is important for us to understand these aspects, as they directly affect the outcome of the gesture recognition module.

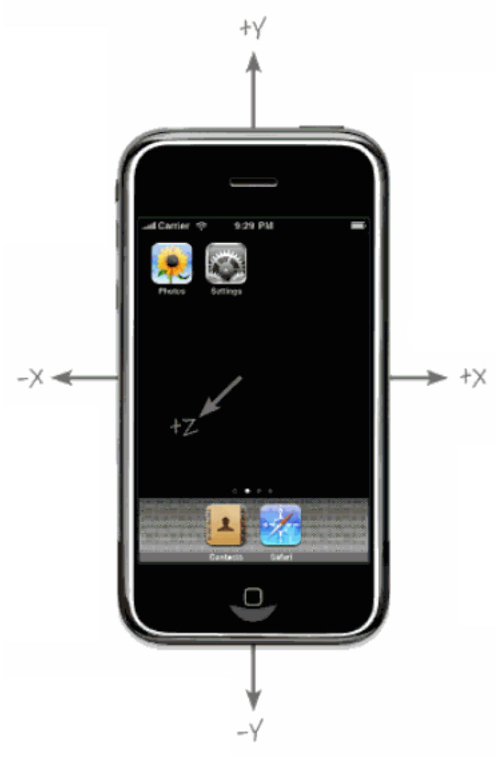


Figure 2: Description of the orientation of the accelerometer in the iPhone. Illustration from [6].

2.2.1 Sensor details

The LIS302DL is an ultra small 3-axis accelerometer which contains a free fall detector, embedded self testing and a high pass filter [5]. It also has the capability of surviving up to 10000g high shocks, which makes it fairly robust. Besides these features, the chip from

STMicroelectronics contains a highly programmable interrupt generator, which makes it ideal for manufacturers of devices where for instance a "wake up on shake" mechanism is of interest. LIS302DL has a user selectable capability of +−2/+−8g, and can provide an output data rate of either 100 or 400HZ. Apple informs that the LIS302DL has a +−2g capability in the iPhone/iPod Touch, and we must therefore assume that they have selected this setting after an evaluation of the usage area for the accelerometer.

When it comes to error rates, STMicroelectronics inform that the precision of the output data rate is related to the internal oscillator and the external clock precision, and an error margin of +− 10% is expected. This is clearly something which would need careful consideration in eventual biometric approaches.

In the device brochure [5], STMicroelectronics states that the LIS302DL is ideal for a range of applications; free fall detection, motion activated functions, gaming and vibration monitoring and compensation. For more thorough information, block schemes and other figures, we refer the reader to the device brochure [5].

2.3 Human considerations

When conducting the signal analysis, there were a few important aspects that needed to be taken into consideration. As accelerometers combine gravity and linear movement, the noise from a users shivering or other involuntary movements while holding the device gets amplified. Since our aim was at creating a baseline for how a *general* gesture looks like, and not investigate the biometric distinctiveness, investigating the personal information in the gestures was not as interesting for us as it would be in a biometric approach. This is because personal characteristics can, if somewhat stable, be used as features to gather detailed information about how one person performs a certain gesture. Having stated this, we had to consider how shaking affected our general templates.

However, making general templates that are representative for all users, does not come without challenges. The human capability for wrists movements is different from person to person, and this was something that was taken into consideration when creating the *general templates*, as described in Chapter 5. How one performs a certain gesture is affected by each user's shivering, quickness, stiffness of joints, amount of experience with such devices, diseases and so on. It is for instance expected that a young computer engineer which is familiar with holding and operating such devices will be quicker in performing the gestures, than an older, more inexperienced person. Since the time used when performing a gesture differs from person to person, it was important that the general template took this into consideration. Aspects such as the speed of execution also had to be considered, and there exists methods that can be used to mitigate the effect of such factors. As an example, *time interpolation* transforms two sequences of unequal length into two of equal length. Another aspect is that different people use different amounts of time from pressing the start button to starting the actual gesture. As this introduces a varying amount of delay, *sliding window algorithms* can be used to mitigate such factors.

The algorithms we utilized in our recognition modules had to be of low time and space complexity, as we operated on a fairly restricted platform. As we needed to conduct recognition in real time in our authentication schemes, we focused on getting good results while keeping the pre-processing steps to minimum.

Since we used DTW, an input sequence that is much longer than the template will get a higher distance score than a sequence that is more in line with the templates length.

This is due to the fact that DTW uses *insertion* and *deletion* costs for comparing sequences of unequal length, as described in Section 2.4.1. Time interpolation on the sequences before comparison can be used to mitigate high distance scores between samples of the same gesture due to difference in timing. Another possible solution to these problems is to use a synthetic recognition algorithm where we look at the principal components describing each gesture, as discussed in Section 2.4.2.

2.4 Recognition specific algorithms and methods

2.4.1 Dynamic Time Warping

The algorithm known as *Dynamic Time Warping(DTW)* was first introduced by Bellmann *et al.* [7] in 1959. Initially meant for speech recognition [8], DTW has in recent years been applied to a number of different areas like for example gait[9] and handwriting recognition [10]. DTW has its primary advantage in being extremely efficient in comparing sequences of unequal length, which makes it very suitable for signal processing of behavioral (biometric) sequences, where the length will vary even though they come from the same person. To simplify, DTW can be seen as a similarity-measurement algorithm, where it uses different *cost* or *distance* algorithms to calculate how much it "costs" to transform one sequence into another.

To give a simple example [1], consider two sequences; *misses* and *mystery*, where the first is the probe and the latter is the template. By utilizing substitutions, insertions and deletions, we can find out the number of operations needed to transform *misses* into *mystery*. As we can see, letters number 2,4,6 (misses) are different from the corresponding letters in *mystery*. Furthermore, *misses* is one letter shorter, so we will need to *insert* a letter after we have made the 3 first substitutions. This example illustrates how we can look at the letters as sequences, and transform the first sequence (misses) into the second sequence (mystery) by using 4 operations. Of course, there are many ways of making this transformation in 4 steps, but that is not the main point here. There also exist other algorithms that one can use in order to determine the *edit distance*[2] between two sequences, such as the Hamming distance [11] or the Levenshtein distance [12].

In a real implementation, DTW's *cost*-function also separates between these 3 operations (insertions,deletions, and substitutions) [13], where each of them have individual costs. The costs are up to the author to decide, and typically depends on the type of signals that is going to be compared. The algorithm starts by building a so called *distance matrix*, which is a two dimensional matrix with lengths of the two sequences. This matrix contains(after all distance calculations are completed) all the pairwise distances between the two sequences, and is used to determine the cheapest way to transform sequence **A** into sequence **B**. What is important is that the distance functions produce small distances for similar entrances, and high distances for entries that are more apart. This makes sure that a sequence that is only slightly different (for instance only shifted in time) gets a lower score than a less similar one.

DTW can not only be used to calculate a *distance score* (which we are going to focus on in this thesis), representing the number of insertions, substitutions and deletions required, but also to find the cheapest *path*. Actually, if we look at the path as an image, as shown in Figure 3, we see that the more the two sequences differ, the more the path

[2] The edit distance between two sequences of characters is the number of operations needed to transform A into B.

deviates from the diagonal line. Had we ran two identical sequences through DTW, we would have seen a perfect straight line along the diagonal, as no operations were needed.

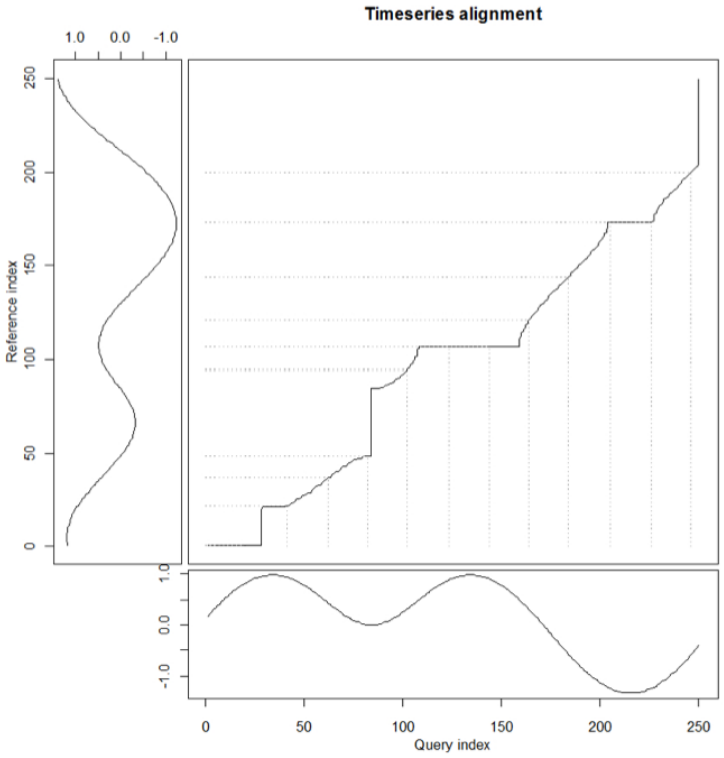


Figure 3: Illustration showing how the path alignment is affected by the similarity of the compared sequences [13].

A more elaborate description of the cost functions, and the DTW algorithm in general can be found in [13], and we will also describe our DTW implementation in more detail in Chapter 6.

2.4.2 Synthetic recognition

By synthetic recognition we mean a recognition module where each gesture is modeled by its principal components. In the case of gestures, this would involve looking at which x, y and z values one could expect for a certain gesture. Our investigation of the hand gestures leaves us with little doubt that for the controlled wrist movements, a synthetic approach would have given us good results. This is based on the fact that each gesture have separate and algorithmically describable characteristics, as discussed in Chapter 5.

The main disadvantage with a synthetic recognition module is that all gestures has to be modeled beforehand. Our approach allows us to add unconstrained gestures to our vocabulary, and we have shown that we can, to a certain degree, separate between arbitrary gestures as well. The more exotic gestures we include in our vocabulary, the harder it will be to model and make synthetic descriptions of the gestures. Although using a synthetic recognition module would have given us good results on the constrained movements, we wanted a recognition module that could be used on all gestures, not only the controlled ones.

3 Related Work

Since our project draws on two separate fields, this chapter serves to describe the work that has been done in the fields of gesture recognition and challenge-response protocols. We will focus on the different algorithms and methods that have been used to achieve good results in both fields, and focus especially on the aspects of the fields that are relevant to us.

3.1 Gesture recognition

The human body has a rich repository of gestures with meaningful relations, and by recognizing these we can improve the effectiveness of human-computer interaction. There are many options when it comes to detecting body or device movement and responding to this movement. Gesture recognition have for this reason been investigated extensively, albeit mainly focused on cameras and specialized devices.

In the late 1990s many gloves based systems were developed and Sturman *et al.* [14] did in 1999 perform a survey of glove based input to computer systems. Although the accuracy and gesture recognition algorithms were in its infancy the studies conducted proved positive results when it comes to improving effectiveness.

An early review on visual interpretation of hand gestures for human-computer interaction (HCI) was made by Pavlovic *et al.* [15]. They observed that the most effective HCI gestures take the characteristics of normal gestures into account. They therefore proposed a method which use both spatial and dynamic information in order to recognize a gesture. Similarly, Wu *et al.* [16] performed a more general review on the subject focusing on temporal gesture recognition. The observations made by Pavlovic *et al.* supports our assumption that we will have to restrict ourselves to some pre-defined gestures for our experiments.

Typically, many of the early HCI implementations focused on adding extra devices to control the computer, like for instance gloves. Therefore, the paper by Harrison *et al.* [17] from 1998 is very interesting in our context since they were the first to investigate the usage of non-conventional interaction mechanisms for mobile devices. They focused on situations where the physical manipulation were directly integrated into the device that were to be controlled. They implemented simple gestures, as flicking the input pen on the corner of a document to change pages, and to focus on a selection of the document by performing a predefined movement over the specified area.

More recently, device capabilities have brought these within reach of commercial off-the-shelf components. This is illustrated by the improvements in both sensor and computational capacity, and the inclusion of integrated mobile cameras allowed Wang *et al.* to develop a computer vision-based software module for gesture recognition suitable for mobile phone cameras [18]. Although their program *TinyMotion* had a very limited feature space, it recognized hand gestures by utilizing the built in camera. *TinyMotion* allowed for handwriting capture and gesture based games such as controlling the blocks in a Tetris game by moving the phone left and right.

Similarly, the inclusion of inertial devices such as accelerometers in mobile devices gave room for new ways of performing HCI on mobile devices. Angesleva *et al.* presented a study [19] on the possibilities of associating gestures with body part movement, and the possibilities of making application based triggers using body mnemonics respectively. The usage of accelerometers for such purposes have been tested in [20, 21, 22, 23, 24]. Although the accelerometers used in most of these approaches were less precise than what we have now, they proved encouraging results when considering the limited amount of samples per second they had at their hands. Also, they do not have the problems with illumination which is typical for computer-vision based approaches like the one proposed by Wang *et al.* [18]. Also, vision based approaches does not work when line of sight is obstructed.

A problem that was present in many of the above proposals was the accuracy of reproduced trajectories. To increase the accuracy, one could impose constraints such as having the users stop the motion before and after the gesture was compared. This was tried in most of the above work, and gave their algorithms good baselines, which let their algorithms separate and classify one gesture from another more consistently.

More recently, Choi *et al.* proposed a gesture-based interaction method [25] using a tri-axis(x,y,z) accelerometer to identify numbers written in the air. Unlike the earlier methods which used trajectory detection in order to recognize a gesture, they chose to use the raw signals from the accelerometer. This gave them a 97.01% average recognition rate in their experimental study. To mitigate the user inconvenience of having to stop before and after a motion, Choi *et al.* implemented an algorithm that could recognize shaking as an indication that a motion should start/stop.

The placement of accelerometers on different body locations for gesture recognition was discussed by Guerreiro *et al.* [26]. They found that by placing numerous accelerometers on the body, they could utilize the body's built in repository of gestures to improve the HCI by making the user perform actions which he initially relates to a specific action. They developed one position-based and one feature-based prototype and found that the feature-based prototype was the most suited for advanced gestures like for example rotations. They achieved an average recognition rate of 97%, which is promising results considering the fact that we will utilize many of the same methods in our recognition modules.

Similarly, in the context of gait recognition, Gafurov *et al.* investigated how the placement of wearable sensors on different body parts affected the acceleration signals, and the error-rates [27]. They found that the placement of the sensor have a great impact on the EER rates. This tells us something about how fragile the accelerometer signals are, and increases our opinion on the fact that we should restrict ourselves to a fixed repository of predefined gestures. Gafurov *et al.* have made a significant contribution to the field of gait recognition by using accelerometers as the source of signals [28], and we can, since gesture signals are not so different (although gait signals are cyclic), learn from their research when it comes to the statistical analysis.

Although this type of recognition is interesting in our context, the more constrained cases of wrist-based motions might be more typical for the multimodal interactions considered in our project. Rahman *et al.* recently conducted both a survey on the range and accuracy of motions and techniques for achieving reliable resolution for typical gestures [29]. They focused their research on analyzing the level of control possible with wrist based gestures. This is exactly the kind of motions that we seek, since having small flicks and turns is much more convenient and stealthy in a challenge-response protocol than writing letters or making circles in the air. Their findings might prove to be very useful when we are to determine which gestures to incorporate in our schemes.

Wrist based gestures/tilting have in early HCI literature been refereed to as the pronation and supination [30] of the human wrist, and as an extension including the ulnar and radial movements. The classification of possible controlled wrist movements following this classification was used by Rahman when investigating the tilt interaction possibilities, and is illustrated in Figure 4.

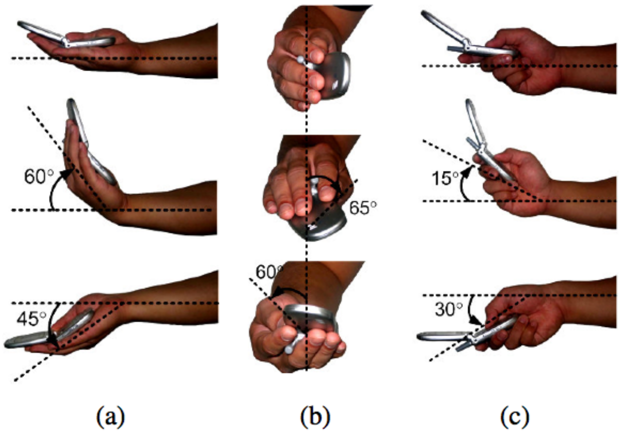


Figure 4: Wrist rotations and degree of rotation possible along each axis of rotation as classified by Grandjean [30].

Related to this, tilt-based interaction can be grouped into two main categories; *precision grip tilting* and *force grip tilting*. Precision grip tilting is when the device is held mostly by the fingers. This employs an entirely new set of possible movements along the three axises, as our fingers are much more flexible than our wrists. Force grip tilting, which is the more controlled type, is when all fingers are used to hold the device as for example shown in Figure 4. The force grip is the one that most of us use while holding mobile devices, because of the control yields, and was utilized in [21, 29].

A study on the human performance in tilt control tasks was performed by Crossan *et al.* [31], where they focused on the usage of accelerometers. They found that there is a great difference in the variability of reproduced gestures when one is moving upwards or downwards, from a horizontal starting position. They showed that people are more stable when moving downwards from center instead of upwards. Also, the variability was greater when moving in the x direction than in the y direction. This shows that we might expect higher variabilities in gestures that have significant movement in the x direction, compared to those with mostly acceleration in the y direction. These findings might prove to be very valuable when we are to define our predefined gestures, and also, when it comes to analyzing and tuning our system to gain the best possible results.

Related to this, Mantyla *et al.* found in their study on the subject [32], that there are numerous factors which highly affects the performance of such systems. Dynamic and temporal differences in how the gesture is performed along with the initial, intermediate and final position of the device was all found to be aspects which lead to false non match cases. Also the physical dimensions of the user (e.g., how tall he is, or how long his arms are), and the standing pose while performing a gesture, proved to have an affect on the recognition rate. This is also why we enforced a static starting position when we gathered gesture samples.

In recent years the utilization of tilt input to perform actions on mobile devices such as twisting the display to better watch images or surf the internet have grown extensively. Also, the newest iPhone contains a shaking gesture feature that allows one to erase written text. Apple even filed a patent application [33] for a specific method of selecting input values based on sensed motions in August 2008. Although this is not exactly the same as we will investigate in our project, it shows that human-computer interactions are increasing in popularity and accuracy.

One of the most recent works on accelerometer based gesture recognition was performed by Lui *et al.* where they presented uWave [24], an algorithm for recognizing personalized gestures based on accelerometer data. They achieved very good results without having to use training samples. Their approach does however not seem to be very resistant to observation attacks as they report an EER rate on 10% on the experiments where and adversary can observe the users modality. Their approach does however provide excellent recognition rates, and are highly usable for other applications such as HCI.

Due to the variations mentioned above, and the fact that we will never see two identical accelerometer signals even though they are made by the same person and visually is the same gesture, we need to investigate how to compare two accelerometer signals with unequal length and characteristics. The template signal for a gesture needs to be compared to the input probe signal in such a way that similar (within a certain threshold) signals should be accepted. Accelerometer signals are similar to the signal from gait and voice recognition in this context due to the dynamic variations in length.

Hidden Markov Models (HMM), which is the most popular method for performing voice recognition, was used by Mantyjarvi *et al.* to perform gesture recognition [34]. Since HMM methods require big training sets, they are not very well suited for our application. Mantyjarvi *et al.* notified this problem and tried to convert two samples into a big set of training samples by adding random Gaussian noise to the manufactured training samples. A problem with this method is that by using gaussian noise they classify the variation in hand gestures to be Gaussian.

A more suitable approach for us is to use Dynamic Time Warping (DTW), which we discussed in Section 2.4.1. DTW has been used extensively in authentication systems that needs to compare two signals of unequal length. Dynamic time warping was also used by Liu *et al.* [24] in their uWave algorithm, which have proven to be the most accurate implementation yet. For this reason, looking at DTW for comparing the gesture parts of our signals will most likely be the best approach. Further on, accelerometers are inertial and therefore subject to both external and hardware noise. In many of the early attempts, filtering algorithms was applied to smooth the signals. Typical methods used was sliding window smoothing, averaging and time interpolation mechanisms. The latter can be used to generate a representable template from a range of signals varying in length. This has been used in a variety of voice, e.g., [35], gait, e.g., [36] and various behavioral mouse-movement recognition systems.

When it comes to multimodal authentication, this have been considered by many authors; Patel *et al.* proposed a mechanism for authenticating to a public terminal based on simple gestures (i.e. shaking and the absence of shaking) in accelerometers [37]. Patel *et al.* wanted to address the problem of the amount of user interaction required to authenticate to a public terminal. By moving the authentication factor from *knowledge* over to *possession*, they can as illustrated in Figure 5, reduce user interaction drastically and improve user friendliness. It is however important to notice that their protocol does not worry about theft. In order for such a protocol to be safe, they would have to include password schemes to mitigate the risk of attackers stealing the users devices. This protocol is however interesting, since if combined with a password, or simply by making the gesture for each device secret, one can add additional entropy to todays authentication scheme.

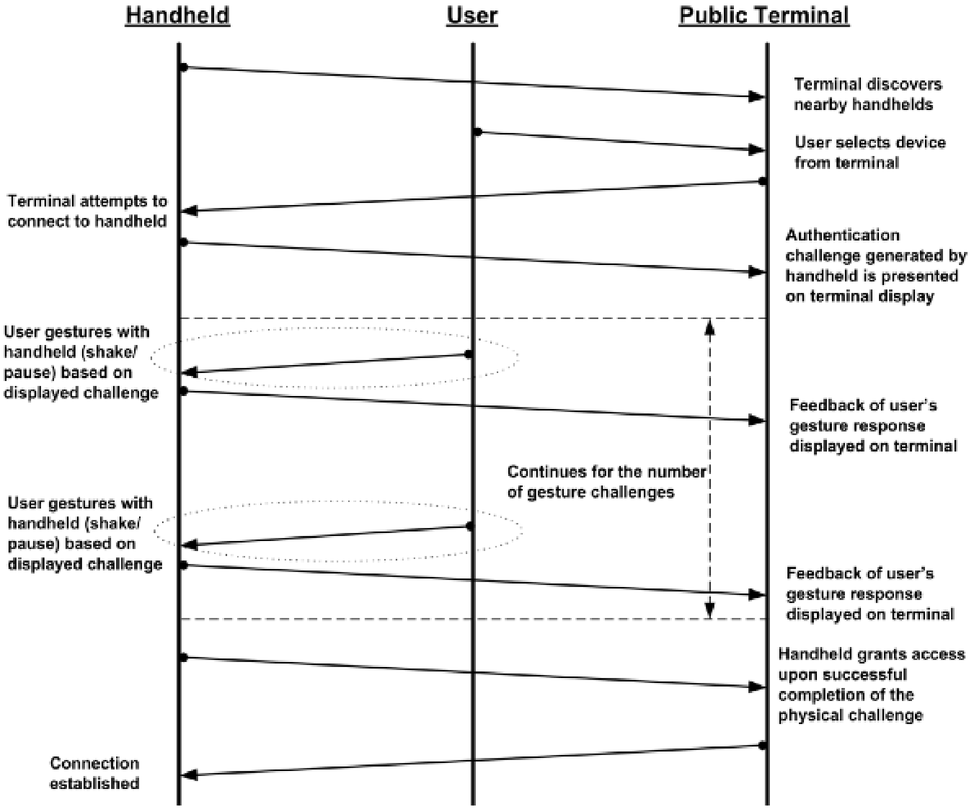

Handheld
User
Public Terminal
Terminal discovers nearby handhelds
User selects device from terminal
Terminal attempts to connect to handheld
Authentication challenge generated by handheld is presented on terminal display
User gestures with handheld (shake/ pause) based on displayed challenge
Feedback of user's gesture response displayed on terminal
Continues for the number of gesture challenges
User gestures with handheld (shake/ pause) based on displayed challenge
Feedback of user's gesture response displayed on terminal
Handheld grants access upon successful completion of the physical challenge
Connection established

Figure 5: This figure illustrates the gesture based authentication protocol developed by Patel *et al.* [37].

3.2 Observation resistant protocols

Having talked about gesture recognition, we will in this section move on to the other research domain of this thesis; to create an observation (shoulder surfing) resistant authentication protocol. Such methods have been discussed by many authors which all acknowledge how easily magnetic stripe cards are skimmed or stolen, and PIN-codes are obtained by means of shoulder surfing attacks. As an analogy, shoulder surfing is not the only problem with using PIN-codes as the only authentication factor; if we say that all passwords are equally likely, then the number of possible combinations of 4 digit PIN-codes are 10^4. Since we typically have three login attempts before being locked out, we got a $\frac{3}{10^4}$ chance of guessing the correct PIN-code in three attempts. Although this is a good security property, we have to consider the other factors that can influence and weaken the "security" of the standard PIN-entry scheme. The fact that many people will choose PIN-codes that are easy to remember or have a relation to them personally (like date of birth etc), will significantly affect the entropy of the system.

There has been done extensive work on making observation resistant authentication schemes and password protection schemes [38, 39, 40, 41], and Hoanca *et al.* have also proposed a theoretical framework for the assessment of eavesdropping resistant authentication schemes [42]. In this framework they describe the necessities for making an observation resistant and user friendly authentication interface, which can be interesting to take into account when we assess our own schemes. Similarly Lei *et al.* [43] propose a virtual password scheme for protecting passwords. In this context, Roth *et al.* propose the usage of dynamic virtual keyboards and probabilistic entry methods to increase the difficulty of observation and replay attacks [44]. Their scheme works in such a way that the system challenges the user with three or four questions for each digit in the PIN. These challenges are communicated to the user by a visual color coding of the digits on the PIN pad. Since different challenges are presented for each session, an observer cannot replay the session. Also, the user never points/pushes/clicks directly at the item which forms his password.

In their experiments they proved that even when the attacker video recorded the login sequence, he would have great troubles replaying the cognitive PIN-code. However, if an attacker can record multiple logins, the attacker can deduce information about the challenges and their responses and eventually determine the "secret" password.

Similarly Wiedenbeck *et al.* proposes an interactive game-like graphical password scheme [45] where a user chooses a number of *pass icons* as his "secret" identifier. At login, the user is presented with several rounds of challenge-response authentication. Wiedenbeck *et al.* define their scheme as a *Convex hull click scheme*, where the main idea is that for each challenge-response the user has to locate three or more of his chosen icons on the screen. After having located them, he needs to click on the convex surface that is formed by the located pass icons, where a convex hull is defined as the edges joining a set of three or more pass face icons. This approach is very interesting because, as with Roth *et al.* attempt, the user never clicks on the pass icons themselves, and the "password" is never the same between two sessions or two challenge-response rounds.

The general principles of observation-resistant virtual keyboards was discussed by Tan *et al.* [46], where they also presented a novel approach for designing keyboards for entering sensitive text on public terminals. The usage of graphical passwords instead of PIN-codes to increase both usability and security have been a popular area of research over the last years, and the results are very promising [47, 48, 49]. However, a common disadvantage with such schemes is that people often chose images or passes that in some way relate to them. For instance in the Passface scheme [47], a man with the preference of girls with dark hair is more likely to pick passfaces fitting this description. By utilizing this knowledge, attackers can make educated guesses and significantly lower the entropy of the system. The problems with visual and picture based passwords was investigated in more detail by Komandui *et al.* in [50].

Perkovic *et al.* did in [51], look at three different methods for observation resistant PIN-code entry, based on the user performing very simple mathematical operations, or simple table lookups, designed for the partially observable attack model (where the attacker only partially can observe the input and output). They found that by using for instance earphones to include the challenge-response possibility, they can derive very good observation resistance along with a minimal overhead when it comes to login time, and error rates. Similarly, Perkovic *et al.* have also proposed another challenge-response method called Shoulder Surfing Safe Login (SSSL) [52], which proved to be both user friendly and cost efficient.

De Luca *et al.* propose a very simple interactive protocol where the users are prompted with challenges in form of vibrations in the mobile device, based on sharing secret information between the terminal and the device [53]. The authors argue that this method is resilient to observation attacks and has the potential to replace current PIN-code entry methods. In earlier research, De Luca *et al.* investigated shape-based mechanisms for authentication, and the cognitive load imposed by such approaches [54]. Their findings proved that people tend to support their memory when recalling PIN-codes with an imaginary line over the num pad, in other words, they doesn't necessarily know the PIN-code numbers, but where to push on the num pad. For this reason they argue that people might more easily remember shapes instead of complex numbers. This information is interesting in our context, since we also will investigate how easily people can remember a sequence of gestures, as a part of the authentication protocol.

In a more recent survey, Kratz and Ballagas describe the cognitive complexity of gestures and strategies for feedback mechanisms [55], where they found that using seamless feedback to the user significantly improves the recognition rate. They also found that this significantly lowers the standard deviation of the recognition rate.

Nali *et al.* did in 2008 present CROO [56], a *universal infrastructure and protocol to detect identify fraud*, which they claim to be capture resilient in the sense that their protocol can notice unauthorized usage by an attacker when the one-time password generator is stolen and used for authentication. Although this is outside the scope of our task, it is a related topic which might help many people store they passwords safely instead of writing them down.

Along with the development of more advanced technologies, more exotic schemes have been developed for the purpose of secure authentication. Although these technologies are prototypes, and not yet meant for wild usage, they provide an interesting aspect to this discussion. As an example De Luca *et al.* propose a method [57] which utilizes what they refer to as "eye gestures". In this scheme, the user performs different eye movements to form gestures, and thereby his password. Similarly Kumar *et al.* [58], propose a scheme where the orientation of the pupils form the password. Here, the attacker would need to know exactly what the user is looking at, in order to replay the password. Even more exotically, Thorpe *et al.* discuss the possibilities and benefits of using brain-computer interfaces for authentication [59].

4 Device capability and usability

This chapter describes the data acquisition experiment, where the main goal was the collection of the hand gestures dataset used throughout this thesis. As described in Chapter 5, this dataset was used to create the general templates that describe each gesture. Further on, the methods described in this chapter was also used to obtain a second dataset, which was used to derive the recognition rates presented in Section 6.3. The experiment protocol and the results of this experiment is therefore used directly throughout the entire thesis.

4.1 Experiment details

This section describes the experiment goal and setup, and also presents a detailed description of the gestures in our vocabulary.

4.1.1 Experiment goal

The main goal of this experiment was to determine the principal components describing each gesture, in order to optimize our recognition modules. We gathered accelerometer signals from our participants while they performed six different gestures. The gestures should be performed in a somewhat equal manner, and the users were carefully instructed in the experiment protocol before we started the experiment. The signals gathered in this experiment were used to generate both personal and general templates for each specific gesture. Analyzing the results from this experiment also gave us a good indication of how stable the signals were, and what problems we could expect in the generation of the general templates. It was important to study the curves produced by each gesture, as these gave us a good indication of the characteristics describing each gesture.

The following sections contains a description of the experiment setup and protocol, the population, and also the acquisition program used in the experiment.

4.1.2 Experiment setup

To get a similar starting point for all the gestures, all participants where instructed to start their gestures while holding the device in a flat, horizontal position, as shown in Figure 6. This gives x and y values at 0, and z at approximately -1. This was done because starting all the gestures in somewhat identical positions, gives us a good reference point when analyzing the signals.

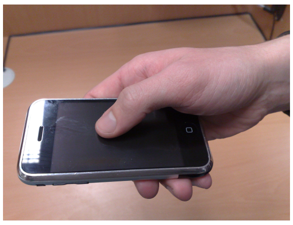


Figure 6: Illustrates approximately how the device should be held at the start and end of an input.

It is important to specify that we used a 2nd generation iPod Touch for this experiment as the weight and weight distribution of the device, along with its shape, can affect the the way our participants perform the gestures. When it comes to the choice of gestures, we chose to follow the recommendation by Grandjean [30], and included the following gestures (which also is shown in Figure 4) in our vocabulary:

- Left flip (LF): a controlled wrist flip where the participant holds the device in the start position illustrated in Figure 6, and from there tilts the device to the left so the device is in a vertical position with the screen facing left. Due to the accelerometers orientation in the device (shown in Figure 2), a left flip should only be concerned with the x and z directions. The y-direction is theoretically not included in this movement, and the data produced in this direction, for both the right and left flip, will mostly be noise caused by shivering. The x values should range from ~ 0 to ~ -1, while the z direction should produce values from ~ -1 to ~ 0.
- Right flip (RF): This gesture is the exact opposite of left flip (the screen is tilted to the right instead of left). We should see x values from ~ 0 to ~ 1, and z values from ~ -1 to ~ 0.
- Back flip (BF): A back flip is when the participant flips the device from the start position up in the air, so that the device is standing in the air, with the screen horizontally facing the participant. Since we in the back and front flips are moving in the y-direction instead of the x-direction, the x direction is the one that theoretically only should produce noise. A back flip should produce y values from ~ 0 to ~ -1, and z values from ~ -1 to ~ 0.
- Front flip (FF): Here the participant flips the device downwards instead of upwards, otherwise its completely the same as the back flip. This should produce y values from ~ 0 to ~ 1 and z values from ~ 1 to ~ 0. According to Grandjeans findings [30], the human wrist can only be tilted ~66 degrees in this direction, something which means that we should not expect z values lower than ~ 0.6.

The gestures described above are controlled wrist movements that should be reproducible from a human motorics perspective. We also wanted to investigate signals from more complex motions, and we therefore included two arbitrary gestures;

- Circular motions: We included two circular motions; one *starting left - going right (CL)* and another *starting right - going left (CR)*. When it comes to arbitrary gestures, there is the concern that these are less reproducible than the more controlled ones, and that we for this reason should expect higher intra-gesture differences. However, since recognizing arbitrary gestures gives us more gestures to use in the challenge response scheme, they are included.

  When it comes to the expected x, y and z values, these cannot be modeled fully due to the nature of the movement, and it is expected that the particular signal curve will vary significantly from person to person, at least when it comes to the amount of acceleration produced. From a theoretical perspective, the gestures should produce quite similar signals and the greatest difference should be in the x direction, where we expect opposite curves. A more thorough discussion on this matter is presented in Chapter 5, where the circular motions are discussed.

From the description above it is clear that most of the gestures have one axis which in theory only produces noise. This noise can introduce additional error cases, but it can also be used to distinguish between different gestures.

4.2 Experiment protocol

The methodology used to perform this experiment is quite straightforward. Using the acquisition program especially tailored for this experiment (which is discussed more thoroughly in the following section), 5 sequences from each gesture, was gathered from each participant. The reason why we limited ourselves to 5 samples per gesture, per participant, was because we were not going to look at the biometric intra-class distances of the gestures. This is because our main goal was to create a general description of each gestures, not to investigate the intra-class distance. Having 5 samples per gestures, per participant, gives us 100 unique samples to describe each gesture, and it is this number that should be considered as the statistical baseline.

The samples were all gathered in one round, which means that there was only one session per participant. Since our focus was on looking at the signals produced when performing these gestures, not the intra-class distances, we did not have to spread the samples out over multiple sessions, as we got the diversity we needed from the 20 unique participants. This does however mean that we cannot include any consideration when it comes to gesture fluctuations across several sessions. However, since each general template consist of 100 samples gathered from 20 different persons, this provided us with the fluctuation we needed in the dataset.

The experiment is location independent in the sense that it can be conducted anywhere, the only constraint was that the room used were quiet and gave the participants the possibility to concentrate while we explained the details of each gesture. As mentioned, we used six different gestures, and each gesture was demonstrated and carefully described before we started the acquisition. We also instructed the participants in how they should hold the device before and after completing a gesture. The participants conducted so called *dry runs*, giving them the possibility of getting familiar with the gestures before we started the experiment. The latter was to get as clean samples as possible, and to save time by having to restart the experiment due to participants performing a gesture in a wrong manner. These dry runs were used to ensure that the participants understood the difference between the different gestures.

After having gone through the details of the acquisition process, we presented the participants with the device, and let them use the developed acquisition program to record their gestures. The program works in such a way that we can easily, after having acquired a gesture, bind each sample to a certain gesture and participant. In order to maintain the anonymity of our participants, we used user numbers instead of names or initials, to separate between participants. This was to ensure that the only way one can bind a specific participant to his samples, is by looking at the participant sheet that contains contact information, gender, age and so on.

The acquisition program works by having the participants click a button located in the middle of the screen to begin the recording, and then, after having completed a gesture, they click the same button to finish the sample. This was repeated 5 times for each gesture, which gave us a total of $5 * 20 = 100$ samples per gesture and a total of 600 samples. After each participant had completed his session, we gathered the data files generated on the device, and stored them locally. The files were stored read-only, so that the raw data files were kept intact.

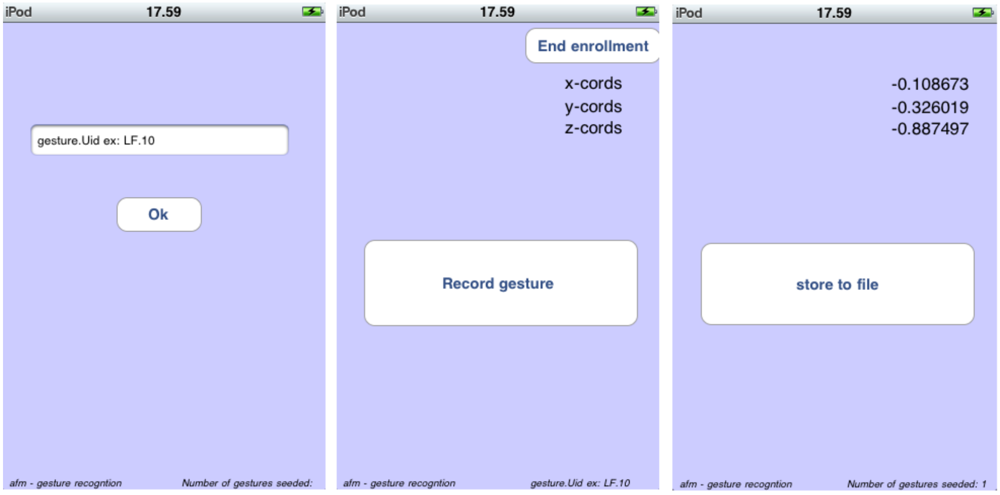
4.3 Acquisition program

The acquisition program has three main views, as shown in Figure 7. In order to keep track of which participant is currently performing a gesture, and the type of gesture, the first view is intended for the test personnel. Here, the user-id (*uid*) of the participant, along with the gesture the participant is going to perform is specified. This information is used by the file creation module of the program, which produce one file for each of the samples conducted, along with a template based on the median values (described in detail in algorithm 1). The filenames are logically named as follows; <*gesture.uid.samplenumber*>. The template created here is per participant, which means that it is based on 5 samples per gesture. The personal templates was used in an analysis where we investigated if there was a big difference between user templates, and the general templates.

After having typed in the needed information, the program continues to the second view, which is the view that the participants utilize while performing their gestures. The participants start and stop recording by pushing the centralized button. After completing one gesture (i.e. having performed 5 samples of the same gesture), we clicked the *end enrollment* button, which generates the personal templates. Then finally, we proceed to the final view where we, by pushing the *to file* button, store samples and template to file.

The last two steps could easily have been automated, but we chose to implement it in such a way that we easier could handle errors in acquisition. If a participant for instance forgets to push the stop button, or makes a wrong gesture, we can simply restart the procedure without having problems with multiple files with the same name. The extra time it took to click these two buttons were seen as a small problem in comparison with having bad samples and templates.

The personal templates was directly used to compare the same gesture from multiple persons, in order to see if there was any significant vectors that we could use to recognize the gestures more accurately.



```
-0.108673
-0.326019
-0.887497
```

Figure 7: Shows the views in the acquisition program. View 1 is the leftmost one, and they follow in chronological order.

4.4 Participants

The participants in this experiment were mainly students and employees at HiG. We had a total of 20 participants (15 male and 5 female). None of the participants had any significant illnesses or other conditions that could affect the execution of the gestures. We had one left-handed participant, and we expected that his performance of a *right flip* was more descriptive than the right handers *right flips*, since this in his case is a *left flip*. We had people from different cultures, and also from different faculties. When conducting the gestures, the participants could choose whether they wanted to stand or sit, whatever made them feel more relaxed. The only requirement we enforced on the environment was that they should start and stop the gesture in the predefined position. The age for the female participants ranged from 23-55 (μ=31 and σ=13.72953), while the males age ranged from 22-42 (μ=26.13 and σ=5.22). In total we had an average age of 27.35 years, and a standard deviation of 8.028. Figure 8 shows the age and gender distribution of the participants.

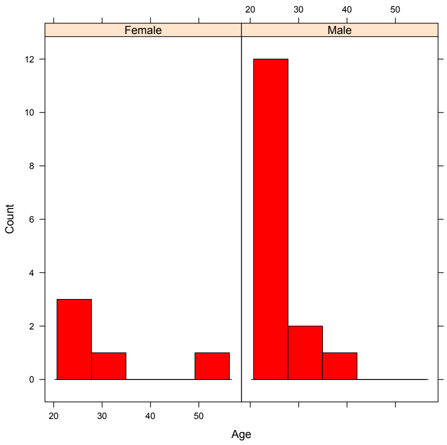


Figure 8: Shows the age and gender distribution in the data acquisition experiment

5 Template extraction and generation

As specified in Chapter 3, we did, for the purpose of our experiments, restrict ourselves to a set of predefined hand gestures. The main reason why we chose to do this is because the accelerometer signals produced by hand gestures have been investigated little at best, and we wanted to have as much reproducibility and control over the dataset as possible. Also, the findings discussed in Chapter 3, support our hypothesis that hand gestures are quite variable, and that the gestures which gives us the best possibility for recognition is the controlled wrist movements as classified by Grandjean [30].

This chapter presents an analysis of the accelerometer signals produced by the gestures, which our participants performed in the experiment described in Chapter 4. It also includes an analysis of the template generation method used. Due to the fact that we were looking for ways to recognize and differentiate between different gestures, we based the analysis on the personal templates generated from the participants, as shown in Figure 9. As this chapter will focus on analyzing the signals, we refer the reader to Chapter 4 for a description of the gestures. The findings in this chapter formed the basis for the recognition modules that is described and analyzed in Chapter 6.

Since this is not a biometric project, but a project where the recognition of gestures is a subpart, we acknowledge the fact that we have not explored as many distance metrics and pre-processing steps as possible. The reason for this is because our aim was not only to obtain good recognition rates, but also to design and develop authentication schemes utilizing these. It is also important to keep in mind that the processing power of the iPod Touch is very limited, and it was for this reason very important to enforce as simple and effective algorithms as possible.

5.1 Template creation

Before heading into the analysis of the different gestures, their signals and properties, this section introduces the methodology used in the generation of the templates.

We generated templates to minimize errors. The most important aspect in achieving this, was to make sure the templates were representative for all the users in the dataset, for each specific gesture. The reason why we chose to generate one template for each gesture, which should be representative for all users, instead of using the already generated personal templates, was because we wanted to generalize. Also, the *general templates* gave us sufficient accuracy without requiring the use of personalization, as shown in Section 6.3. It is clear that using a training data set for personalization will provide improved results, and these can be obtained in the course of regular use, avoiding the need for explicit training sessions whilst also allowing for conceptual drift in the training. However, as described in Section 5.3.1, this will be considered in future work. Also, including personalization would mean having the same participants in all experiments, something which we did not have the luxury of.

In the generation of the *general templates*, we had to consider all the factors that could affect the strength of the templates. We focused on maintaining the statistical properties of the sequences, in order to achieve unbiased recognition rates. The human considerations discussed in Section 2.3 was taken into account, as it was important that the choice of fitting function used for the template generation, reflected the challenges in the population and in the signals produced.

Our dataset may contain outliers in the sense that some people might have completely different ways of performing a gesture, in comparison with the rest of the population, as discussed in Section 5.3.1. As our templates should be representative for all users, we had to make sure that both the length of the sequence, the data point values, and the curvature was representative. Due to the difference in acceleration we experienced when different people performed the same gesture, we expected high variances between different points in the datasets, something that could lead to skewed curves. It is important to remember that the most important aspect in recognizing a gesture is that the curves produced from the template sequences are representative in such a way that they maintain the principal components describing each gesture, as this enables us to create more accurate recognition algorithms. The time used to perform a gesture will, as discussed in Section 2.3, affect the length and skewness of the graphs produced. To handle these variations, and to make our templates as representative as possible, we investigated a number of different fitting functions.

The *arithmetic mean* is a well known fitting function, but since we expected high variances in the data sets data points, we decided that this would skew the curves to a point where the characteristics of the template would not be representative for the entire population, or the gesture. If for instance one person produce more acceleration than all the others, his way of performing the gesture will get a higher impact on the template than desirable. The *mean without outliers*, was also considered, but after testing, this was also found unqualified, as it produced curves which did not maintain the principal components. This is because it was very difficult to identify the outliers in the dataset, since people have different ways of performing a gesture, with different amounts of acceleration and speed. We also discussed the possibility of implementing more sophisticated fitting functions like *polynomial fitting*, but we concluded that this would be an overkill, when considering the type of sequences we have to play with. After investigating and performing small range experiments with different fitting functions, we chose to use *median* as our fitting function. Using median gives us templates which represent the "general" way of performing a certain gesture without having to worry about extreme cases having a tremendous affect on the template. Median will also give us smooth lines in the templates which makes it easier to maintain the principal components describing each gesture. Another argument for using simple fitting functions is that the gestures are restrained by physiological properties. Having restrained gestures provides us with clearly distinguishable and stable signals, which does not require sophisticated fitting functions, as the data values will not vary significantly from the expected values. This also means that the principal components describing each gesture will be constant across different datasets.

To investigate how representative templates we got by using median, we ran tests using DTW to generate distances from the samples the template was generated from, to the template itself. It is important to notice that the numbers themselves does not mean very much, as the costs for insertion, deletion and substitution affects these, what is important notice is the standard deviation. This tells us something about how representative the generated template is for each of the underlying samples. Table 1 shows the distances generated from 5 of our users to the personal template generated from those 5 samples. What is important to notice is that most of the distances are close together, with a low standard deviation, which indicates that the template is representative for all of them. There are however some outliers, and we can see that those typically come in the first sample for each user. This means that the users perform the gesture a bit odd the first time, and then become more stable. This is the reason why we have two lines showing different standard deviations in Table 1. The first one represent the whole set, while we in the second have taken out the first sample for each user, and recalculated the standard deviation.

The DTW implementation used to generate these distances is the same implementation that is described in Chapter 6. The only difference is that we chose to eliminate the non-contributing axes for each particular movement, as they only introduce noise. Eliminating the noise allowed us to focus solely on determining whether median was the best fitted function.

Table 1: Shows the DTW distances generated when comparing each sample of 5 randomly picked users against the template generated from those samples. This particular example represents 5 users *back flips*, where each column represents one user's distances to his own template. The first line of standard deviation shows the standard deviation of the whole set, while the second to last line shows the standard deviation after removing the first sample, which was seen as a training sample.

1	**2**	**3**	**4**	**5**
41.18	29.36	14.29	31.67	30.57
19.73	27.09	17.81	21.35	24.57
17.84	22.72	20.94	17.74	16.29
21.17	27.79	21.69	18.89	26.03
22.42	18.71	21.64	14.25	32.56
$\sigma = 9.49$	$\sigma = 4.35$	$\sigma = 3.20$	$\sigma = 6.60$	$\sigma = 6.32$
$\sigma = 1.96$	$\sigma = 4.22$	$\sigma = 1.83$	$\sigma = 2.95$	$\sigma = 6.68$

The same calculation as shown in Table 1, was performed for all gestures, with similar results. The intra-class distances (by this we mean the distances from one persons samples to his own personal template) are very close, which indicates that the templates are representative. What is important to remember is that although two sequences get the exact same distance score, this does not mean that they are identical. This only means that the same amount of operations was needed to transform those sequences into the template. Also, since hand gestures are very variable, even from the same person, the main purpose of a template is to be as close to all samples presented by that user. Having this feature allows us to not only recognize a gesture when the user performs it perfectly, but also when he performs it a bit oddly. This is also why a low standard deviation indicate that the templates are representative.

The fact that circular motions have a higher standard deviation than the rest was expected, as they are harder to reproduce, thus introducing an even higher amount of intra-gesture variance. For a visual representation of the template generation, see Figure 9.

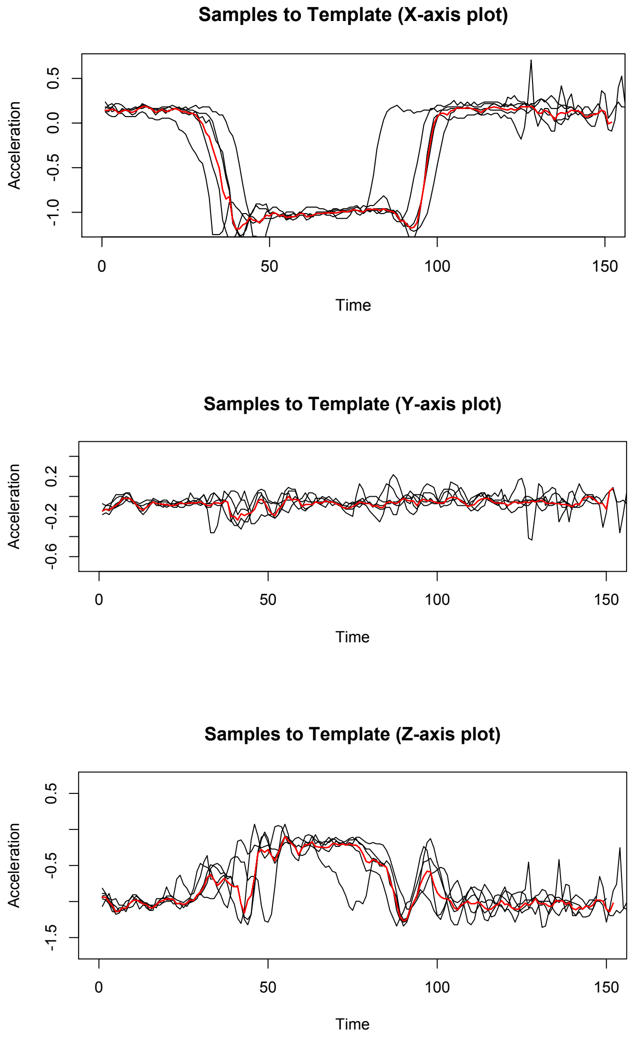


Figure 9: The red line represents the template which is generated from the 5 other samples (the black lines). The data in these graphs represent a *left flip*.

5.1.1 General template creation

When it comes to the creation of the *general templates*, which represents the general way of performing a certain gesture based on input from all users, we used the same methodology as for the personal templates. The only difference is that we took the templates from all users for a specific gesture as input. Table 2 shows the distances when comparing each users personal template against the general template for that particular gesture. The reason why CR and CL get higher distances is because we included all three directions, as all of the axes are involved when conducting a circular motion. This will also cause a higher standard deviation, as the distances grow proportionally.

The reason why the standard deviation is generally higher in Table 2, than in Table 1, is because we compare each user's personal template to the general template that should be representative for everyone. This indicates that a biometric approach might be applicable in controlled environments, as the users are generally closer to their own template than to the general template. The reason why some users produce higher distance scores than others is because they have a very different way of performing that particular gesture than what is considered "normal" in our dataset. This affects the standard deviation for the whole dataset, as one user's high distance will increase the standard deviation for the entire dataset. This is also the reason why we chose to include a second line of standard deviations in Table 2, where we excluded one user that had abnormally high distance scores (the second to last user). After investigating why this participant consequently got higher distances than the others, we found that the participant in question used more than twice as much time to complete each gesture. This clearly shows the importance of tempo and speed in completing the gestures, something which indicates that using time interpolation could improve our recognition rates.

Since the templates provides us with relatively close distance scores to most of our users, this indicates that they should be usable for recognition purposes. As the general templates job is to reflect the "normal" way of performing a gesture, it should be as close as possible to the personal templates. It should however, not be affected by outliers in the sense that one outlier increases the distance for all. We would rather have that 19 out of 20 personal templates match accurately, than having all 20 match somewhat. This is also the reason why ended up with median as the fitting function. In order to achieve good recognition rates, it was important to set thresholds and cost variables that reflected the fluctuations in the dataset. This analysis is presented in Chapter 6.

Table 2: DTW distances produced when comparing each user's template to the general templates. The lines marked "Alt" shows the calculations after excluding one problematic user.

LF	**RF**	**BF**	**FF**	**CR**	**CL**
33.907	22.628	33.105	35.297	67.460	64.2166
56.414	54.892	56.527	32.839	89.965	78.3554
35.071	19.653	15.957	30.552	88.0144	93.6099
50.987	44.707	41.457	40.892	101.6427	98.6015
47.869	22.141	35.697	64.119	135.6198	127.383
38.993	39.178	40.648	33.575	86.7814	91.1293
65.538	74.869	25.975	32.027	90.0870	66.6860
40.307	47.511	35.214	35.347	65.8085	62.0754
55.027	43.380	32.915	35.886	75.2030	94.9251
42.494	50.209	53.863	55.210	109.381	106.340
38.625	58.137	51.944	48.432	160.056	152.534
36.529	44.342	52.790	44.205	91.8093	83.4760
53.839	50.863	45.603	42.307	75.2340	87.2440
41.031	33.657	39.208	41.244	89.9226	69.4960
29.741	24.541	29.771	31.438	49.1099	79.5474
49.908	59.875	47.644	47.077	86.0605	92.6215
43.574	60.262	53.479	60.270	85.2900	78.2280
62.177	54.229	37.650	34.331	74.7037	82.4424
68.067	110.16	100.69	87.015	162.056	138.379
35.384	31.521	25.427	47.186	106.097	94.8507
$\sigma = 11.086$	$\sigma = 20.94$	$\sigma = 17.49$	$\sigma = 14.011$	$\sigma = 29.051$	$\sigma = 23.917$
$\mu = 46.27$	$\mu = 47.33$	$\mu = 42.77$	$\mu = 43.96$	$\mu = 94.51$	$\mu = 92.10$
Alt. $\sigma = 10.097$	$\sigma = 15.24$	$\sigma = 11.25$	$\sigma = 9.941$	$\sigma = 24.98$	$\sigma = 21.877$
Alt. $\mu = 45.12$	$\mu = 44.03$	$\mu = 39.73$	$\mu = 41.69$	$\mu = 90.96$	$\mu = 89.67$

5.1.2 Our median implementation

In the creation of the templates, one sample is represented as a class-instance that has three arrays; one for x, one for y and one for z-values. To create the templates we therefore used the algorithm outlined in algorithm 1, where the length of the template x, y, z arrays is determined by the average length of all the sample x, y, z arrays.

Further on, to populate the template arrays, we calculated the median value of each index by extracting for instance the first x, y, z values from all the samples, storing these in separate arrays (separated by direction), and then take the median of these temporary arrays. We then fed the calculated median value into the template array for that particular index (in this example at the beginning). By using this methodology, we make sure that the generated templates are representative for the samples that were gathered, at each data point.

Algorithm 1 Pseudocode for the template creation.

```
All samples have three arrays containing $x, y$ and $z$ values.
for i=0; i<avg length of arrays do
    for all sample arrays do
        fetch $x, y, z$ values at index i
        store these into temporary $x, y, z$ arrays
        calculate median of these arrays, and put this value into template at index i
    end for
end for
```

When it comes to deciding the length of the general templates, this was explored in detail since the length of the 100 sequences varied significantly. We found that all participants introduced a delay between conducting the gesture and pressing the stop button, something which gives us noise at the end of the signal. The same delay was also found in the beginning of the signal. After investigating the sequences manually we found that most users used approximately the same amount of time to complete the gestures, and that it was the delay that varied the most between the participants. For this reason, we chose to take the average length of all of the sequences for one gesture as the length of the general template. We experimented with other, more elaborate ways of calculating where the end should be, like for instance checking the relationship between acceleration and time. However, we found that using the average length was the most suitable and stable approach. We also ran an experiment where it was found that none of the 100 sequences had any noticeable acceleration after this point in time. This means that we only removed noise at the end, which wont have a big affect when it comes to the DTW distances, as it will simply enforce a few deletions or insertions, should a sequence be longer or shorter than the template.

5.2 Gesture characteristics

This section includes a description of the principal components that make up each gesture. We will also discuss the principal characteristics of each gesture, when it comes to recognizing and separating one gesture from another. It is important to have general templates that are close to all users, but it is even more important to be able to separate one gesture from another. Without that feature, an authentication scheme using gestures would be inapplicable.

The same analysis was performed on every gesture, but we will only present one full analysis (on RF and LF), since it is a repetitive process. For the remaining gestures we will focus on describing the key aspects of each gesture. While reading this section, we refer the reader to Section 4.1.2, where the gestures are described. The gestures will be discussed in pairs, as for instance a *right flip* is an inverted *left flip*, and should produce very similar signals. As we only show selected graphs in this section, we refer the user to Appendix A, where all the signal graphs are shown in more detail.

5.2.1 General description

Before we start describing each gesture's signals, it is important to clarify some points that count for all of the gestures. As we in our experiment instructed the participants to: Click a button, perform the gesture, and then click the button again, this introduced a delay. This delay is the time between the participant clicks the start button, to he performs the gesture, and also the time after finishing the gesture to pushing the stop button. The amount of delay varies from person to person, and we can see it in the graph as the time where the acceleration in the active directions stay around 0 before picking up noticeable acceleration. Figure 11 serves as an example, where we clearly can see the delay before and after the gesture is conducted. The effect this delay had on our recognition modules was considered during the recognition experiments.

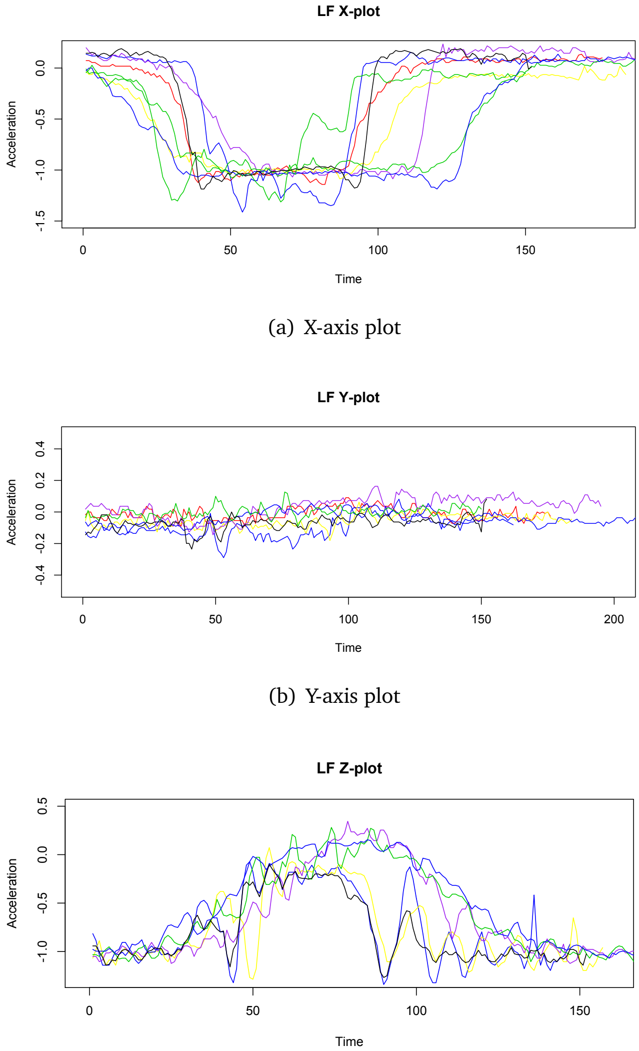
5.2.2 Right and left flip

Right and left flips are, as described in Section 4.1.2, when a user tilts the device from its starting position to the right and left, respectively. Figure 10 shows a number of different left flip templates plotted against each other. As we can see, there is a repetitive pattern in the x-direction, where the users start and end up around 0 in acceleration, while they at the minimum point reach -1. For a right flip, as shown in Figure 11(a), the x direction is the inverse of the left flips curve, starting at 0, going up to maximum of 1. This is in line with our expectations, and gives us a good view of the principal components describing the x direction.

In both a left and right flip, the y-direction should only produce noise. It should in theory be situated around 0 in acceleration, because the direction should not be involved in these particular movements. As we can see from the signals in Figure 10(b), there is, however, some acceleration present in this direction, which most likely comes from shivering. It can however, also indicate an abnormal way of performing the gesture. If one for instance tilts the device forward or backward while conducting the gesture, the y direction will get acceleration. Such an execution of a left flip would most likely have made a high impact on the distance scores to the general template if we included this direction for left and right flips, as was discussed in Section 5.1.1. However, since we are using median as the fitting function, we do not have to worry about outliers affecting the overall template.

From a biometric perspective, the y direction might be very interesting, as it can contain user specific information. An investigation on whether the shivering was constant for the different users, could have added extra bits to the gesture, which could increase the inter-class distance.

Figure 10(c), shows that the z direction is very much involved in the execution of a left and right flip. This is also in line with our expectancies. Although the peaks are skewed from user to user, the pattern remains the same. The reason why some have higher peaks than others, is due to the amount of acceleration they produce while conducting the gesture. This can vary from person to person, but as we are only interested in recognizing a general gesture, the pattern is more important than where the peaks are. The z direction provides us with almost identical signals for both a right and left flip, as shown in Figure 11(c).



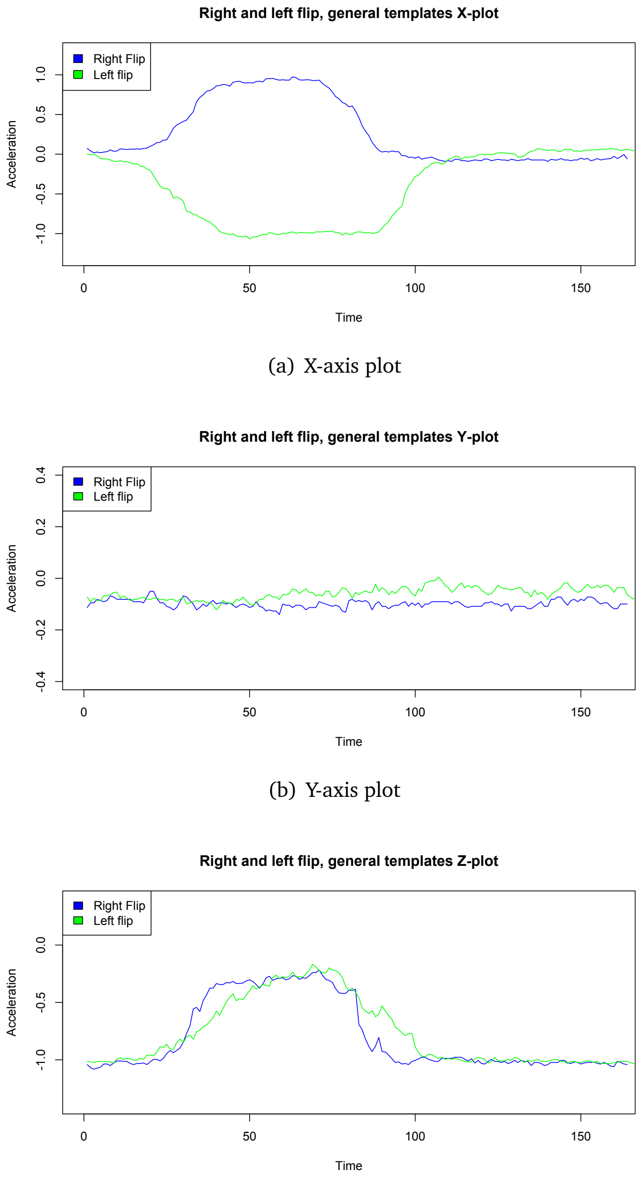
(a) X-axis plot

(b) Y-axis plot

(c) Z-axis plot

Figure 10: Cross-plot of a number of different *left flip* personal templates.

Figure 11 shows the general templates generated from all of the participants personal templates for left and right flips plotted together. What is important to notice is that the only real deviation between a right and a left flip comes in the x direction. In the y and z direction they are almost identical. This means that we will have to focus on the x direction in order to separate them.



(a) X-axis plot

(b) Y-axis plot

(c) Z-axis plot

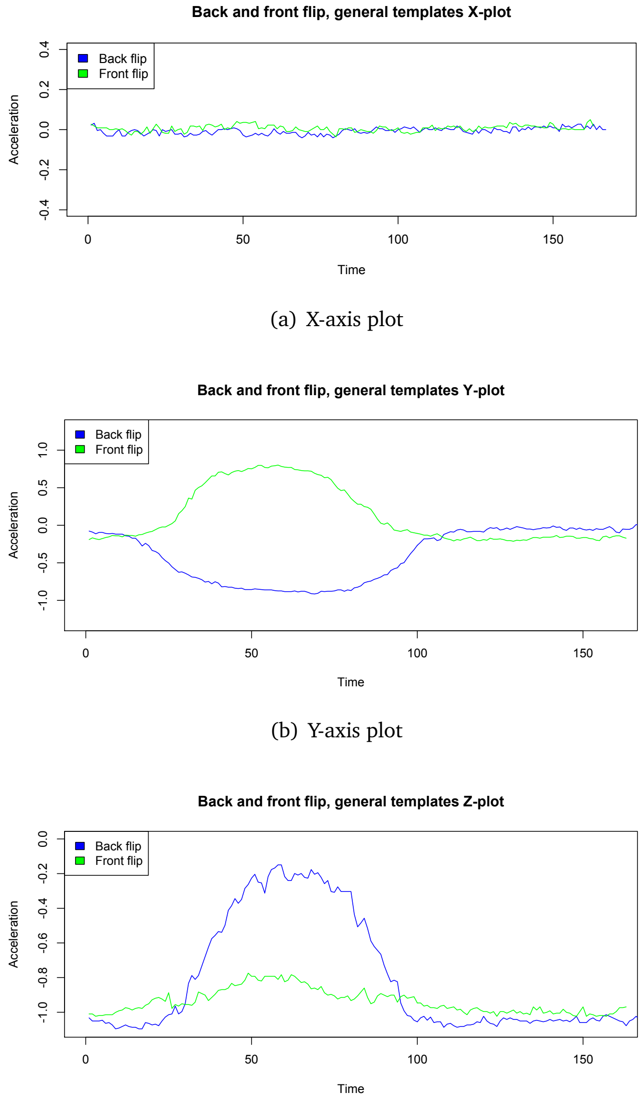
Figure 11: Shows the general template for a *left* and *right flip*.

For more detailed graphs describing each gesture and their templates, we refer the reader to Appendix A.

5.2.3 Front and Back flip

Our next pair is the front and back flip gestures. As with the left and right flips, the back and front flip are in theory, identical, but inverted gestures. In this case, it is the x direction that is the uninteresting axis, as this in theory should only consist of noise. Figure 12 supports this hypothesis, as we can see that the general templates for a front and back flips x axis is situated around 0 in acceleration. Similar to what we saw in Figure 10, we do experience some variances in the noise direction, for the same reasons as expressed earlier. This can be seen in Appendix A.

The y axis is, as shown in Figure 12, the axis where a front and back flip differs the most. We clearly see that the signals are pretty much identical, but inverted. The most interesting axis in the front and back flip gestures is the y axis, as this contains the most acceleration. As discussed in Sections 2.3 and 4.1.2, the capability of our wrists restrict our front flips more than our back flips. This is clearly seen in Figure 12(c), where the front flip produce significantly less acceleration in the z direction. This may of course vary from person to person, which is shown in the more elaborate graphs in Appendix A, but for the average participant, our wrist are easier to bend backwards than forwards. In a biometric approach, the front flip might prove to be the most distinctive movement, as physical restrictions make it harder to reproduce another person's trait.



(a) X-axis plot

(b) Y-axis plot

(c) Z-axis plot

Figure 12: Shows the general template for a *front* and *back flip*.

5.2.4 Circular motions

Although the signals produced from the circular motions are very restricted, we clearly see a repetitive pattern. Figure 13 also shows that the two circular motions are, as expected, identical but inverted. All three axes produce wave-like signals, but with very little acceleration. The amount of acceleration produced, depends on the speed and size of the circle, as shown in the more detailed figures in Appendix A. We can however clearly see that the fluctuations in values are much less than in the other gestures, and that all three axes are involved in the movement. To get a more detailed description of the gesture, we would have needed a gyroscope (since these can measure translations, such as horizontal movement without acceleration) in combination with our accelerometer.

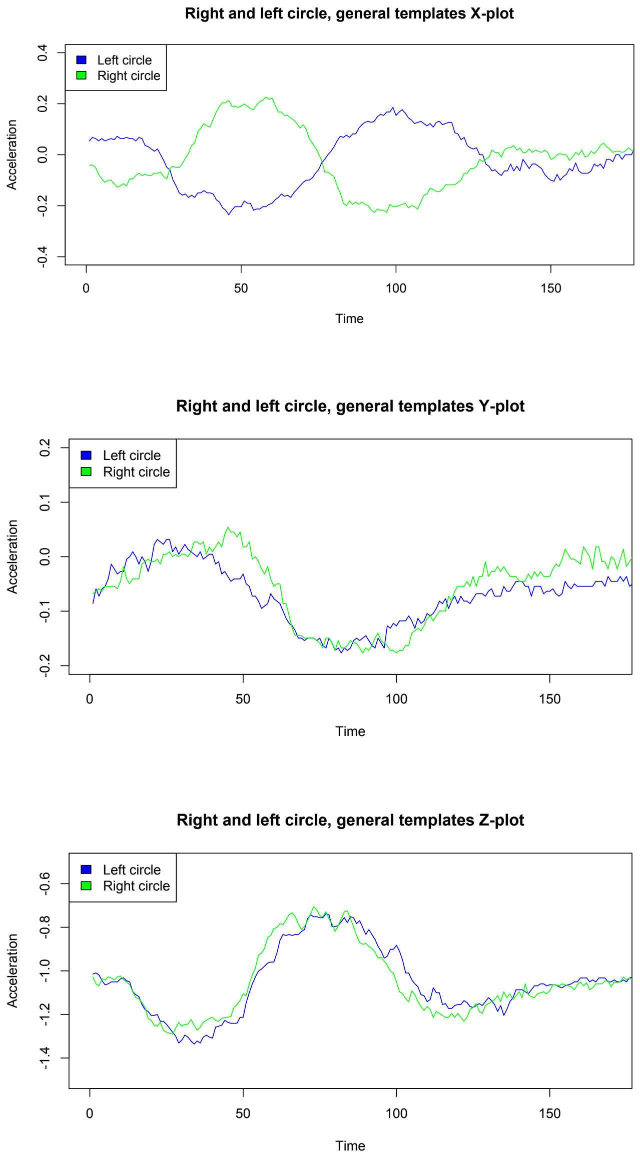


Figure 13: Shows the general templates for *left circles* and *right circles* plotted against each other.

5.3 Discussion of results

Throughout the analysis of the signals we have seen that the general templates describe and maintain the principal components of the gestures. We have also seen that some people deviate in the execution of the movements from the principal components, by including the so called noise axes. Whether or not we should include these noise axes when making our recognition modules, is discussed in Chapter 6. On one side, these axes does not contain any descriptive information, but they might indicate that two gestures are not the same (e.g., if the probe sequence has significant acceleration in a direction which, in the general template, should be noise). Including the noise axes in the comparisons might therefore help us to differentiate one gesture from another, as a sequence which have acceleration where the template expects noise, will get a significant addition to the total distance.

When we calculated distances from each participants personal templates to the general templates, we saw that most of our participants had very similar distances to the general templates. In this particular experiment we excluded the noise axes, as we wanted to focus on the important axes in order to determine the principal components. We experienced that some persons produced higher distances than others, which also increased the standard deviation significantly. If we compare the standard deviation of the distances between one participants samples and personal template, to the distances from the personal template to the general template, this support our hypothesis that the principal components are the same for all users for the same gestures, and that our template creation method have maintained this in such a way that it is representative for everyone.

Since all samples was gathered in one round, we cannot deduct any explicit reasoning whether or not people get more stable in conducting the gestures over time. We do however, believe that people will become more and more similar to the general templates over time, as these explicitly represent the principal components describing each gesture.

5.3.1 Personalization of templates

Since we wanted to focus on building our authentication schemes and test these, we did not implement any method for template personalization. We acknowledge the possibility of adapting the general templates over time, to improve the recognition rates. The basic idea is that every time a person performs a gesture which is accepted, that particular sequence should have a personification affect on the general template. This allows us not only to improve the recognition rates, but also to include some sort of personal information in the general templates, which will make it harder for an attacker to reply the login sequence. There have been presented different methods for template adaption, but perhaps the most promising one is provided by Liu *et al.*[24]. In their paper they acknowledge the same fact that we have proven throughout our analyses in Chapter 6; that there is a big variance in how different people perform the same gesture, and that this calls for templates. This is also the reasons why we experimented with personal templates, as shown in Table 1. We showed that we by using personal templates got significantly lower intra-class distances than if we simply used one of the raw samples as the template. This is for the same reason as discussed for the general templates.

As mentioned in Chapter 10, this is something that would be interesting to look into in future research.

6 Recognition performance

This chapter presents the analysis performed on the distinctiveness of hand gestures, and the recognition rates obtained by using our recognition modules. We have performed different analyses, and this chapter describes the results and prerequisites of each of them. Section 6.2 describes an experiment where the goal was to investigate how distinctive the raw hand gesture sequences were, while Section 6.3 describes a more elaborate recognition experiment, where we utilize the general templates, generated in Chapter 5, for recognition purposes.

In the forthcoming sections, the following equations form the basis for how we calculated the false match (FMR), and false non-match rate (FNMR):

$$\mathrm{FMR} = \frac{\text{Number of accepted impostor attempts}}{\text{Total number of impostor attempts}} \tag{6.1}$$

$$\mathrm{FNMR} = \frac{\text{Number of rejected genuine attempts}}{\text{Total number of genuine attempts}} \tag{6.2}$$

It is important to remember that, since we are operating on restricted platforms, our aim was not only to achieve good recognition rates, but also to keep the computational costs to a minimum. The reason for this is that the code assessed in this chapter forms the basis for the recognition modules incorporated in our authentication schemes, as described in Chapter 7.

6.1 Distance metric notes

Before heading into the results of our experiments, this section describes the distance metric used, its parameters and modifications. The DTW algorithm allows for tuning by altering the costs for insertions, deletions and substitutions. We can also tune our recognition algorithm by experimenting with how we combine the three individual distance scores (x, y, z), that we get from each comparison.

6.1.1 Costs

By adjusting the costs for insertions, deletions and substitutions, we directly affect the distances produced between two samples. It is important to find a relationship between these three cost-factors that generates high distances between inter-gesture sequences, but keeps the intra-gesture distances as low as possible. Having too high costs produces undesirable false non-match cases, as small fluctuations between two sequences will produce high distances. Also, due to the different characteristics of each gesture, we needed to find costs that work for all gestures, something which means they may not be optimized for each individual gesture, but for the whole vocabulary of gestures.

In the related work study we investigated work on both hand gestures and similar signals (e.g, from gait and speech), where all reported different cost values. However, as all were situated between $0.1 - 1$ for insertions and deletions, and as the optimal cost for each operation varies from dataset to dataset, we decided to test every value in this interval. We found that using 0.4 gave us the best error rates for the total set. The reason why one value is better than another is due to the fluctuations, and the average intra-gesture distances in the dataset [13].

When it comes to the cost for substitutions, we found that using the absolute value of the distance between the two vectors, divided by the maximum set distance, gave us a suitable relationship between insertion, deletion and substitution costs. This means that it should be cheaper to perform a substitution than an insertion or deletion if the data points are very close, and vice versa.

$$sub = \left| \frac{(x - y)}{\text{Maximum set distance}} \right| \tag{6.3}$$

6.1.2 Combining the three vectors

Since we in our case have three axes (x, y and z) that, when combined, describe one gesture, we will for each comparison get three unique distances. Earlier, the possibility of eliminating the noise axis was discussed, but since we need one dynamic recognition module that allows for many-to-one recognition (both identification and recognition) in our authentication schemes, all axes needed to be included. We experimented with using only the "important" axes for each gesture, but found that although the noise axis does not contain valuable information for describing a gesture, it helps us to separate gestures; if we see acceleration in a direction where the template has none, then we can be quite sure that these sequences does not descend from the same type of gesture. Also, the error rates produced when using only the important axes had more false match cases, something which supports this hypothesis. This is the reason why we chose to use all axes in the comparisons, regardless of which gesture we are comparing against. Further on, since we wanted to put an equal amount of weight on each axis (since we developed one universal recognition module), the total distance from one gesture to another is the sum of the three distances.

$$Distance[A, B] = DTW[A[x], B[x]] + DTW[A[y], B[y]] + DTW[A[z], B[z]] \tag{6.4}$$

If we were to develop a *one-to-one* recognition module, then we probably could have increased the recognition rates by focusing on the important axes for each gesture. Such recognition modules are however not usable in challenge-response schemes.

6.2 Gesture distinctiveness

In order to get a picture of how distinctive the different gestures were, we conducted an experiment on the dataset collected in the data acquisition experiment, described in Chapter 4. Even though we in Chapter 5 saw that the principal components describing each gesture are significantly different, we needed to investigate how different they were in terms of DTW distances. To achieve this, we conducted an experiment where every sample sequence was compared against all others. This allowed us to see see how close all sequences from the same gestures were, and how far away they were from the other gestures, in terms of DTW-distances. For this particular analysis, we focused on investigating the more controlled wrist gestures RF,LF,BF and FF. The algorithm used in this experiment is shown in Algorithm 2.

Algorithm 2 Pseudocode for the all-to-all comparison. All sequences are read from file in such a way that we know which "type" they are.

```
for all sequences $i$ do
  for all sequences $y$ except $i$ do
    Calculate distance between $i$ and $y$
    if i.type == y.type && distance > threshold then
      false non match
    else if i.type != y.type && distance <= threshold then
      false match
    end if
  end for
end for
```

Algorithm 2 is, by using the *type* parameter, aware of what kind of gesture i and y are. As everything is counted twice, we get the following number of genuine and impostor attempts:

$$\begin{aligned}\textsc{Genuine attempts} &= (((\text{samples per participant} \times \text{participants}) \\ &\quad \times \text{participants} - 1) * \text{number of gestures}) \\ &= ((5 * 20) * 99) * 4 = 39600 \qquad (6.5)\\ \textsc{Impostor attempts} &= (((\text{samples per participant} \times \text{participants}) \\ &\quad \times \text{number of gestures} - 1) * \text{number of gestures}) \\ &= ((5 * 20) * 300) * 4 = 120000 \qquad (6.6)\end{aligned}$$

6.2.1 Comparison details

The fact that there are 4 different gestures considered in this experiment had to be taken into account when generating error rates. As each gesture has different characteristics, with different intra-gesture distances, we had to specify unique thresholds for each of them, in order to obtain optimal results. Also, we focused on the total error rates of the four gestures combined, which is why the error rates in Table 3 is varying when it comes to the FMR and FNMR for each specific gesture.

As shown in Table 3, we got a FMR of 30.58%, and a FNMR of 25.27% in this experiment. This clearly shows that the raw sample sequences themselves are not very distinctive, something which is caused by high intra-gesture variances, even between the same participant. The rates from the individual gestures can, although not perfectly distributed, indicate the distinctiveness of each particular gesture. We see that BF and FF is by far the most distinctive gestures, and produces the lowest error rates. This has two main reasons; (1) they are the easiest to conduct, in the sense of repeatability and wrist movements, and (2) they produce less acceleration than the others. The latter makes them easier to compare and distinguish, because the possible fluctuations between participants is limited by the gesture itself. Section 6.3 contains an analysis when using the general templates for recognition, where we achieved significantly better error rates.

Table 3: Shows the false match and false non-match rates when performing an all-to-all comparison, where every sequence is used as a template for comparison.

Gesture	**FMR**	**FNMR**
RF	42.33%	24.14%
LF	44.80%	21.9%
BF	20.01%	26.44%
FF	15.17%	28.61%
Total	**30.58**%	**25.27**%

6.3 Using the general templates for comparison

This section contains a description of the experiment where we utilized the general templates created in Chapter 5. We used the same methodology as described in Section 6.2, and did not utilize any preprocessing algorithms since we operate on a platform with restricted computational power.

One could argue that since these experiments are conducted offline, we could have implemented preprocessing steps to optimize performance. However, since the goal of this project is to develop a solution for live recognition in real time, we focused on this particular problem. All algorithms and methods was developed with the intent of live recognition, and with the aim of being usable as recognition modules in our authentication schemes. In this experiment we also included the arbitrary, circular motions. Since we in the creation of the general templates used the complete dataset gathered from the data acquisition experiment, we acquired an additional dataset to avoid any statistical dependency which would have biased our results. The dataset collected for this experiment is described in detail in Section 6.3.1.

The algorithm used to produce error rates in this experiment (Algorithm 3), uses the general templates as static references, which gives us the following number of impostor and genuine attempts:

$$\text{G\textsc{enuine attempts}} = (\text{samples per participant} \times \text{participants}) = 18 \times 5 = 90 \tag{6.7}$$

$$\begin{aligned}\text{I\textsc{mpostor attempts}} = (&(\text{samples per participant} \times \text{participants}) \\ &\times \text{number of gestures} - 1) \\ &= (18 \times 5) \times (6 - 1) = 450\end{aligned} \tag{6.8}$$

Since we now have significantly fewer attempts, each sample counts more in terms of statistical calculations, and our calculations are therefore more susceptible for outliers.

Table 4 shows the error rates obtained in this experiment. Since we in this experiment had less genuine and impostor attempts, we focused on getting both the individual error rates and the combined error rate as close to an equal error rate as possible. By comparing these results with the ones in Table 3, we see that they are not only lower (FMR=5.17% and 8.11%, FNMR=4.72% and 8.15%), but also consecutive when it comes to the fact that "back flip" is the most stable and distinctive gesture. As indicated in the signal analysis, a BF is very easy to perform, and this is also most likely why it produces the best results. The other controlled wrist gestures FF, LF and RF are more rigid gestures, which means

Algorithm 3 Pseudocode for using the general templates as the reference in the error rate calculation.

```
for All gestures general template $i$ do
  for all sequences $y$ do
    Calculate distance between $i$ and $y$
    if i.type == y.type && distance > threshold then
      false non match
    else if i.type != y.type && distance <= threshold then
      false match
    end if
  end for
end for
```

that there are more room for personalization of the gesture between participants, which can be seen in the error rates.

The circular motions were included in this experiment, and we clearly see that they produce significantly higher error rates than the controlled wrist movements. This is, however, as expected, as circles introduces additional vectors when it comes to the execution of the gesture, like for example speed and size of the circle, along with the orientation of the device while conducting the gesture. These are all factors which will significantly affect the distance from the template, to the probe sequence.

Table 4: Shows the false match rates and false non-match rates when using the general template for each gesture as the reference.

Gesture	**FMR**	**FNMR**
RF	6.44%	5.56%
LF	5.33%	4.44%
BF	2.89%	2.22%
FF	6.00%	6.67%
CR	14.22%	15.56%
CL	13.78%	14.44%
Total without CR,CL	**5.17%**	**4.72%**
Total w/CR,CL	**8.11%**	**8.15%**

The reason why these error rates are lower than the ones in Section 6.2 is due to the distance metric used to create the general templates. As each template is an artificially generated sequence based on a median calculation of 100 raw samples descending from 20 unique participants, it is situated somewhere in the *middle* of all of the raw sequences. As a result of the distance metric used, the template has an averagely lower distance to all of the raw samples, making it less vulnerable against the intra-gesture variances. Figure 14 illustrates a visual example, where the red dot indicates the general template, and the black dots are raw sequences descending from the same gesture.

In the experiment leading to the results in Table 3, all raw samples were used as references, and compared to all other samples from the same gesture. As illustrated in Figure 14, having raw samples as templates gives us a higher average intra-gesture distance, than when using the calculated templates. Since the general templates lowered the intra-gesture distances, this allowed us to set stricter thresholds with the effect of eliminating false match and false non-match cases and increasing our recognition rates.

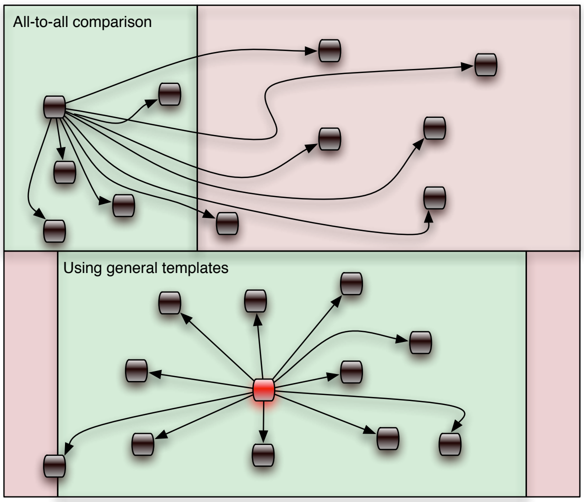


Figure 14: Illustrates the benefits with using general templates. In the all-to-all comparison, each and every node gets compared to all others, while when we use general templates, these are the only references. Since the general template is based on a big number of sequences, it is located somewhere in the middle of the other samples, hence giving us lower intra-gesture variances and better results. The green and red windows indicate a sequence within and outside the threshold, respectively.

6.3.1 Verification dataset details

Acquisition

The acquisition of this dataset followed the exact same restrictions as imposed in the data acquisition experiment outlined in Chapter 4. The methodology, setup and circumstances were identical, the only separating factor was the participants.

Participants

The participants in the verification set were mainly students and employees at HiG. We had a total of 18 participants (15 male and 3 female). Nine of the participants also contributed in the first dataset, but we gathered new data to eliminate any dependencies against the general templates. As in the first dataset, none of the participants had any significant illnesses that could affect the execution of the gestures. We had also here people from different cultures, age groups and faculties. When conducting the gestures, the participants could choose whether they wanted to stand or sit. The only enforced requirement on the environment is that they should start and stop the gesture in the predefined position. The age for the female participants where closely coupled this time, and ranged from 22-24 (μ=23 and σ=1), while the males age ranged from 21-35 (μ=25.06 and σ=3.86). In total we had an average age of 24.72 years and a standard deviation of 3.61.

6.4 Additional notes

It is important to remember that, in order to achieve good recognition rates in the authentication schemes, the participants will need to follow the same restrictions when it comes to gesture execution. By following these prerequisites, we have shown both through the signal analysis, and in our experiments, that we can distinguish rigidly between different gestures even without any preprocessing steps. However, when we incorporated our recognition modules in the authentication schemes, we had to consider the fact that the users are not only performing gestures, but also interacting with the scheme. This might lead to more unbalanced ways of performing the gestures, something that was taken into account when we incorporated the recognition modules in the authentication schemes, by slightly heightening the thresholds for each specific gesture.

6.4.1 Implementation and complexity details

Our recognition module allows for both *many-to-one* and *one-to-one* recognition. Many-to-one recognition includes both identification and recognition, while one-to-one recognition includes only the latter. The two algorithms are outlined in algorithm 4 and 5.

The algorithms themselves are quite intuitive, however, there are a few things that should be specified; The first being that each of the general templates have their own thresholds, derived from the experiments conducted in the previous chapter. In the case of an *one-to-one* comparison (algorithm 5), the algorithm matches a probe sequence against the expected gestures template. This is the simplest case of recognition, and limits us to usage areas where we know what kind of gesture to expect, something which is clearly undesirable. Algorithm 4, which allows for both identification and recognition (many-to-one), is a bit more complicated. To identify which gesture was performed, the algorithm computes the DTW-distances from each template to the probe sequence. The algorithm then picks the closest template, in terms of DTW-distance, and investigates whether or not the probe sequence matches within the identified gestures threshold. When we get a *non-match* in this algorithm, this means that the user performed a gesture which did not match any of the general templates within their individual thresholds.

The reason why we generate distances to all templates before checking against the individual thresholds (and take the decision of a match or non match), is because we wanted our recognition to be as stable as possible. Also, to eliminate error cases where for instance a *left flip* gets a score below the *right flip* templates threshold (a FMR case), we compute all distances on beforehand. This allows us to pick the closest template, and the odds are great that the conducted *left flip* will match the general template for a *left flip* even more accurately than the *right flip* template. We have also seen throughout our analyses that even though a sequence is declared as a non-match, it is in most cases closest to the *correct* general template. The drawback with the many-to-one comparison module is that the device has to conduct x number of DTW-comparisons, where x is the number of gestures enrolled in the system, while only $\frac{1}{x}$ comparisons in an one-to-one comparison.

The mode of operation does not affect the error-rates, but it does directly affect the computational cost required to make a comparison. We implemented both methods, and found that the many-to-one module uses roughly 1.5 seconds to identify and recognize a gesture (when including all gestures), on the 2nd generation iPod Touch. This is of course an undesirable overhead in any authentication scenario, and it shows how important it is for us to keep the complexity of the recognition module to a minimum. As we are not using any post- or pre-processing steps, the complexity lies in the DTW algorithm. Since we utilize a standard DTW approach, the complexity evaluation in [60] holds for this discussion; As each cell in the matrix is filled once to determine the best possible way to transform **A** into **B**, we get a time and space complexity of $O(N^2)$, if $N = |A| = |B|$. In this thesis where $|A| \neq |B|$ in most cases, we utilize *insertion* and *deletion* operations to deal with sequences of unequal lengths. This means that the complexity in our case is $O(N^2 + (|A| - |B|))$. As we have a matrix that grows quadratically, comparing long sequences will enforce significantly higher computational costs, than comparing shorter ones. This means that more descriptive gestures will require more computational time than the more controlled wrist movements. Also, in newer devices, N might increase as a result of more descriptive sensors or the inclusion of gyroscopes.

Having discussed the complexity of the recognition module, we must take into consideration that these schemes are mostly aimed at more sophisticated devices like the iPhone 3GS/4G or the HTC Desire, where the computational power is significantly greater than in our device. As an example, the HTC Desire has a 1GHz processor and 576 MB of RAM, while our device has merely a 412 MHz processor and 128 MB of RAM. We can for this reason tolerate some time-overhead in our schemes, as this is for experimentally purposes only. This discussion also strengthens our hypothesis that we cannot include any elaborate pre-processing steps, or utilize more intelligent distance metrics, while keeping computational cost to a minimum.

Algorithm 4 Pseudocode for the algorithm used to perform identification and recognition (many-to-one recognition).

```
Record probe gesture $y$
for All general templates $i$ do
  Calculate distance between $i$ and $y$ by using DTW
end for
Sort distances and find the closest template $i$ to $y$.
Set $i =$ the closest template
if The distance between $y$ and $i <= i's$ threshold then
  We have identified and recognized the gesture $y$
else
  $y$ does not match any of our templates within the threshold
end if
```

Algorithm 5 Pseudocode for the *one-to-one* recognition module, which means that we can expect a certain type of gesture as input.

```
Decide which gesture we are expecting, and choose the corresponding template $i$
Record probe gesture $y$
Calculate distance between $i$ and $y$ by using DTW
if The distance between $y$ and $i <= i's$ threshold then
  We have identified and recognized the gesture $y$
else
  $y$ does not match $i$
end if
```

7 Authentication schemes

This chapter presents the two authentication schemes we have developed. We will throughout this chapter describe the anatomy of the schemes, and what they add in terms of security. We will also describe the different experiments we have performed to validate them. The experiment descriptions contains detailed information about the experiment protocol, the experiment goal, and also theoretical security assessments.

Section 7.1 presents a scheme where we use gestures and PIN-codes in combination to allow for obscured logins, while we in Section 7.2 present a more elaborate scheme resistant to replay and shoulder surfing attacks.

Definitions

The following terminology is used in the description of the authentication schemes and the experiments:

Victim: The victim is the person operating the device, hence it is also he who utilizes the authentication scheme.

Attacker: The attacker is the person trying to deduce the PIN code of the victim. He can either do this by watching (experiment 7.1), or by analyzing video footage (experiment 7.2). An attacker can also be referred to as an *observer*.

Experimenter: The experimenter is the person responsible for the execution of the experiment. He is responsible for describing the experiment protocol and the scheme to the participants. The participants can take the role of both an attacker and a victim, depending on which scheme we are testing.

Safe and decoy colors: In the challenge-response scheme described in Section 7.2, we refer to *safe* and *decoy* colors. In the context of this scheme, the *safe* colors are the ones that the user protects, while the *decoy* colors are the ones he willingly reveals throughout the protocol, in order to protect his PIN-code.

7.1 Gesture and PIN-code based authentication

This scheme utilizes a combination of PIN-code and gestures to improve the overall security of the authentication scheme. What separates, and makes this protocol more secure than normal PIN entry schemes, is that it incorporates gestures to mitigate against shoulder surfing attacks, by introducing obfuscation in the PIN entry. The gestures also increase the total entropy of the scheme.

The basic flow of the protocol is shown in Appendix B.1, Figure 29. The principal idea behind the protocol is to use the gestures as a challenge to the user, by forcing him to not only enter the correct PIN digits, but also by having him place them in the correct order, by using different gestures. This means that instead of entering 4 digits in one go (in the correct order), our protocol expects a 2-tuple $C(P, G)$, consisting of a digit and a gesture, as shown in Figure 15. The type of gesture in the tuple decides the placement of the PIN digit, as shown in Table 5.

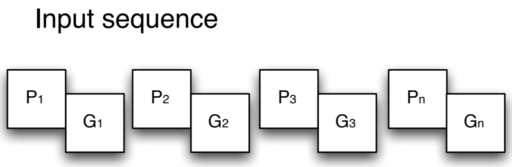


Figure 15: Shows the 2-tuple input sequence, consisting of both a number and a gesture.

As the gestures plays the role of "placing" the entered digit in the right relative position, each gesture is preassigned to a particular placement, as shown in Table 5. This allows the user to enter his PIN-code in an obscured manner, since he does not have to enter the digits in a particular order.

Table 5: Shows the fixed relationship between a gesture and digit placement

Gesture	**Corresponding placement of digit**
Front flip	1
Left flip	2
Right Flip	3
Back flip	4

Another feature is the ability to replace or overwrite previously entered PIN digits. Although the scheme use a 4-digit PIN-code, we leave the "authentication game" open for as long as the user wants. This means that the user can enter digits (decoys) which are not even in the PIN-code, and then replace them with the correct digit later, in an attempt to throw off eventual observers. We expected that the ability to overwrite previously entered digits, would make it harder for an observer to deduce the PIN-code.

7.1.1 Validation experiment

Experiment goal

This experiments aim to validate the following hypothesis:

> Combining hand gestures and PIN-code will increase the authentication schemes resilience against shoulder surfing attacks.

This scheme was developed for the purpose of being intuitive and easy to use. We believe that its features will mitigate human shoulder surfing attacks, and thus increase security. We also wanted to investigate the effectiveness of the obscurity mechanism included in this scheme.

Experiment constraints and prerequisites

Before conducting the experiment we had two constraints to take into account; our time schedule, and the participant resources. Optimally we would have liked to perform a thorough assessment of both the security and the user friendliness of the scheme. However, since we had a limited amount of time, and for the reason that we wanted to assess two different schemes, we chose to focus on the observation resistance. Since we already had verified the recognition rates, and the usability of our gesture recognition modules throughout the experiments in Chapter 6, focusing on the observation resistance does not introduce unforeseen dependencies when it comes to error rates, as these are known from previous experiments.

Experiment protocol

Although the experiment protocol is influenced by the aforementioned constraints, the integrity of the results is maintained and all results are statistically sound. As the methodology needed to be time-effective we had one static victim, while the participants acted as attackers. The attack scenario is one-one (in the sense that we had one attacker and one victim per session), which means that we get;

$$\text{impostor attempts} = \text{participants} * \text{number of tries}. \tag{7.1}$$

Since this scheme is dynamical in the sense that the victim can login differently from time to time, we chose to separate between two types of login scenarios to see if how the victim utilizes the features of the scheme, affects the observation resistance;

1. The simple login, where the user enters the PIN-code in an obscured manner, without using the overwrite mechanism. The goal of this sub-experiment was to see if the attackers managed to connect a PIN-digit and a movement. We expected that this type of login should be less resilient against observation attacks than the random login sequence.
2. The random login sequence, where the victim enforces the overwrite mechanism by entering fake digits and replacing them at a later stage.

The reason why we only had one victim was because we saved time by not having to instruct multiple victims in how to use the scheme. Like in all "exotic" authentication schemes, there is a training period where the users needs to get familiar with the method, and we minimized this by keeping the same victim throughout all sessions. Since we had already verified the gesture recognition modules utilized in this scheme, we could focus on the observation resistance. We did however, record error cases where the victim got a wrong login, either due to faulty execution of a gesture, or if he simply had made a mistake in the scheme. These errors were seen as the same, as we did not want to give eventual attackers any feedback from the protocol when it comes to what went wrong in the user's login attempt. In order to calculate separate error rates for each scenario, we separated the two sub-experiments throughout the execution of the experiment.

Attack scenario

We decided to keep a completely open attack scenario. This means that all attackers were thoroughly instructed in how the scheme operates, and which gesture enforced which placement of the corresponding digit, as shown in Table 5. Before the experiment started, each participant was instructed to read and understand the experiment description shown in Appendix D.2.

Experiment execution

Having described the attack scenario, we can now describe the actual execution of the experiment. As stated earlier, we had one static victim and a prearranged fixture between gestures and placements, which was known to both the victim and the attacker. Since we wanted to focus on the observation resistance of the scheme, we followed a few simple steps to identify whether or not the attacker had managed to deduce the actual PIN-code, from watching the login sequence performed by the victim. We had one session for every attacker (participant). Each session consisted of 2 simple login attempts, and 2 random login sequences. The reason why we chose such a low number of login attempts per participants was due to the available participant resources. For more detailed information about the experiment procedure, please see Appendix D.1.

The experiment protocol is quite straightforward; we let the victim utilize the scheme to authenticate himself, and gave the attacker full observability. By full observability we mean that the only constraint on the attackers was that they could not interfere with the login sequence, or disrupt the victim in any way. The attacker could watch the login sequence from the angle he preferred. To check whether or not the attacker had managed to deduce the victims PIN-code, we had the attacker tell the experimenter what he believed was the correct PIN-code, after the victim has completed his login. The reason why we had the attackers orally inform the experimenter the secret, instead of instructing him to replay the login sequence, is because if the attacker had managed to deduce the secrets, then he should also be able to login, presuming he doesn't encounter any errors while conducting the gestures. As we wanted to eliminate error cases where the attacker faulty performs a gesture and gets denied access, even though he have deduced the victims PIN, we excluded such error cases as they would only have given us a biased image of the schemes observation resistance. By having the attacker simply tell the experimenter what he believed was the PIN-code, we could ignore such error cases and focus solely on the observation resistance.

Additional notes

We acknowledge that if an attacker can remember all the stages taken in this scheme, it can be directly replayed. This is why we have implemented a more elaborate scheme (described in Section 7.2), where challenge-response is included to mitigate such threats. The goal of this scheme was to see if including gestures adds enough extra bits to the entropy of the scheme, so that the attackers have a hard time physically remembering the login sequence, and replaying it.

An obvious extension to this scheme is to allow each victim to bind individual gestures to placements. This would force the attacker to remember all stages taken, since he cannot deduce the PIN-code itself. This means that the only attack possible on this scheme is a direct replay attack, which most likely requires either an eidetic memory, or a video camera.

Participants

The participants in this experiment were mainly students at HiG. We had a total of 20 participants (18 male, and 2 female). All of the participants understood the protocol before we started the experiment, and they all read the experiment description in Appendix D.1. The age of the two female participants were 23 and 56 ($\mu = 39$ $\sigma = 24.04$), while the age of the male participants ranged from 21 to 56 ($\mu = 25.61111$ $\sigma = 8.161163$). In total we had an average age of 27 years, and a standard deviation of 10.32116.

7.2 Challenge-response scheme

Although the previously described scheme increases security significantly, it is vulnerable against more elaborate shoulder surfing attacks, where the attackers utilize cameras to videotape the entire login sequence. For this reason, this section proposes a method which is more resilient against such attacks. In contrast to the previously described scheme, this scheme introduces randomness and challenge response from both the device and the user. The protocol flow is illustrated in Appendix B.2, Figure 30.

Introducing new elements of randomness allowed us to fully mitigate the threat of shoulder surfing and replay attacks. In addition to the device randomness, this scheme introduces two secrets, which is shared between the user and the device:

- The users PIN-code, which in this scheme consists of 8 digits.
- The association between gestures and index regulation (gesture-color associations), which will be described in detail later.

As stated, the goal was to develop a scheme that is difficult to observe and which will be different for each protocol run, thus the introduction of randomness. Although the scheme does not depend on the number of digits in the PIN, the security properties does.

7.2.1 Protocol flow

This section describes the basic flow of the protocol, as illustrated in Appendix B.2, Figure 30.

1. We assume that the device is a personalized device, so that each user has his own PIN-code and color association. The PIN is treated as an array consisting of n digits, with indices ranging from 1 to n.
2. Using a (pseudo-) random number generator ((P)RNG), the device chooses two indices i and j, from the users full PIN. These are visually displayed on the screen, and presents the user with the challenge of choosing one of the two, by using gestures. The device also shows two colors on each side of the screen, which associates colors with the gestures the user is going to use to choose between the two indices. As this is a personalized device, each user selects both a PIN and the gesture-color associations. This means that the user has two secrets, thus reducing the possibility of shoulder surfing attacks, and also significantly improve the entropy. An example of such a challenge can be seen in Figure 17.
3. The user utilize his gesture-color associations decided on beforehand to deduce which gestures are legitimate, and uses these gestures to choose if he wants to enter the PIN digit corresponding to the $i'th$ or $j'th$ index of his full PIN. As an example lets say that the users PIN is 41243176, and his gesture-color associations as follows:

   Table 6: Shows an example of a gesture-color association

Color	Corresponding gesture
Black	Front flip
Blue	Left flip
Red	Right Flip
Green	Back flip

   Now, if the device generates and displays index number 3 and 1, and displays red on the lefthand side of the device, and blue on the righthand side, the user knows that to select index number 3, he will have to perform the gesture corresponding to his red color (RF), and to choose index number 1, he will have to perform a LF (blue). Lets say that the user chooses to perform a RF, thereby choosing index 3 from his password, he will then have to enter the digit 2 as the response to the challenge, as this corresponds to index 3 of his full PIN-code.

4. After this is completed, the user have entered his first digit, and the device will restart the procedure from point (2). When the user have answered four such challenges (thus having entered 4 digits), the device will check if the 4 PIN-codes and their corresponding indices match those in the users PIN-code. The fact that we are only gathering a subset of the users PIN-code in a pseudo-random way for each login sequence, also mitigates the threat of replay attacks. To check whether the user have entered a valid login sequence or not, the scheme compares the entered digits to the digits in the corresponding indices chosen during the protocol run. This is done after four challenges has been answered, and the scheme does therefore not give away any information to eventual observers during the protocol run.

Additional notes and prerequisites

A few prerequisites needs to be described to fully describe the anatomy of the scheme:

- The same color cannot appear twice in the same challenge. This means that we will get two unique colors in each challenge.
- Similarly to above, the same index cannot appear twice in one challenge. If this was possible, the attacker would at once deduce the digit-index association.
- A user can choose to enter the same digit multiple times. By this we mean that if the same index appears in different challenges, the user can choose to present it twice.

Pseudo-random number generator

For this experiment we used the `arc4random` [61] number generator made available for the Cocoa framework by Apple. `arc4random` uses $8 * 8$, 8 bit S-Boxes, which can have a total of 2^{1700} states. It is a cryptographic hash function where the seed mechanism is not documented thoroughly by Apple. Apple only states that the function is self-seeding. As we require only basic statistical properties, we employed a simple statistical test where we generated 10^8 numbers between 1 and 4, which gave the following statistical distribution:

Table 7: Shows the distribution generated from the pseudo-random number generator

Number	**Was generated x times**
1	25003864
2	24998805
3	25002511
4	24994820

As we can see from the distribution in Table 7, this generator works sufficiently for our purpose.

7.2.2 Theoretical security assessment

The mathematical calculations presented in this section holds under the attack scenario described in Section 7.2.3. In a real life scenario an attacker would almost certainly not have gotten access to such footage, for the reasons expressed in Section 8.2.1.

The possibility of including a "ignore next digit" movement in step 3 above was thoroughly considered. By performing a random gesture which did not match any of the two gesture-color associations displayed in an challenge, the user could signalize to the protocol that the following digit should be ignored. This would serve much of the same purpose as the possibility of overwriting previously entered digits in the first scheme, which is described in Section 7.1. This feature was however, excluded as it can be used by an adversary to gain access. As an example, consider a scenario where the attacker videotapes 4 separate login sequences of the same user. The attacker has full knowledge of the protocol, and the only thing keeping him from logging into the device is the users secret PIN and gesture-color associations. After carefully investigating the login procedure, the attacker can, by looking at the gesture-color-index-input relationship, partially deduce the gesture-color associations, and for example deduce 4 of the total 8 index-PIN code relationships. Now, if we had included the "ignore next digit" movement, the attacker could simply perform this movement whenever he gets a challenge he does not know the answer to. For this reason, we chose to enforce a static 4-challenge login sequence. By doing this, the attacker has to rely on luck to get exactly the four index challenges that he knows the answer to. This adds an additional layer of protection when it comes to full observation resistance on video, which is the attack scenario we investigated in this experiment.

In mathematical terms, this schemes direct replay resistance can be calculated as shown below, where n is the number of possible placements of the PIN digit, and k is the number of colors.

$$\left(\frac{1}{(k*n)*(n-1*k-1)}\right)^4 = \left(\frac{1}{4*8*7*3}\right)^4 = \left(\frac{1}{672}\right)^4 \approx 5*10^{-12} \tag{7.2}$$

As we can see from the calculation above, there is an extremely slim chance of conducting a successful replay attack. As mentioned earlier, the only way of breaking this scheme is to deduce the secret gesture-color associations. The calculations of how many challenges (we display 4 per login session) needed to break the gesture-color associations is virtually impossible to estimate, as this includes modeling both human and device randomness. There are also certain steps a user can utilize to increase the resistance of the scheme. By choosing two colors which he wants to protect (safe colors), a user can make sure that an attacker can never deduce more than two gesture-color associations, regardless of how many login attempts he gains access to. In order to achieve this, the user simply needs to remember the following rule of thumb:

> Whenever two *safe colors* occurs together, the user can choose either of the two. This also counts when the two *decoy* colors appear simultaneously. Whenever one *safe* and one *decoy* color appear together, perform the gesture corresponding to the *decoy* color.

If the user is aware of this protection mechanism, he can whenever one of the two revealed associations (decoy colors) occurs along with one of the safe colors, choose to perform the gesture corresponding to the decoy color. This effectively keeps the adversary from learning anything new. Should the user see a challenge with both of his safe colors, he can do whatever he wants because the attacker does not learn anything new in either case. This is because the only way an attacker can decode a new gesture-color association, is to find a challenge where he knows one of the gesture-color associations displayed, in which the user performs the gesture corresponding to the unknown color.

An important note is that although the attacker does not know two of the associations, he can still make educated guesses. Since there are only two gesture-color associations unknown to him, he has a 50% chance of getting the gesture right. However, he can never be sure whether he guesses right or wrong, as he will also have to input the PIN digit corresponding to the chosen index. Since he does not know the PIN-code nor the index, it will be significantly more difficult for him succeed. For such an attack to work, the attacker will have to get the exact same challenges multiple times, and as we have shown, there is a very slim chance of a replay attack working on this scheme.

Even though we have stated that an adversary can never deduce more than two gesture-color associations, consider the case where an adversary has decoded all of them. Now, depending on how many challenges he has seen, and the nature of them, he will have access to certain index-digit combinations. What is the chance of the attacker getting the challenges he needs in order to successfully attack the protocol? As an example lets say that the attacker have deduced 4 of the total 8 index-PIN digits. Then, the probability of an attacker getting the challenges he needs, in order to successfully attack the scheme is:

$$P = \left(1 - \frac{3}{14}\right)^4 = \left(\frac{11}{14}\right)^4 \approx 38.1\%. \tag{7.3}$$

Of course, the probability of getting the challenges he needs increases with the number of PIN digit-index relationships he reveals, but for this particular example, the attacker has a 38% chance of successfully attacking the protocol. In the calculation above, the prerequisite is that the attacker gives up whenever he gets a challenge he does not know the response to. An adversary will probably not give up so easily. By including the case where the attacker make an educated guess whenever he gets indices he does not know the corresponding digit to, we get:

$$\begin{aligned} P = \left(\frac{11}{14}\right)^4 + \left(\left(1 - \left(\frac{11}{14}\right)^4\right) * \frac{1}{10}\right) &= 0.381 + \left(0.619 * \frac{1}{10}\right) \\ &= 0.381 + 0.0619 \approx 44.29\% \end{aligned} \tag{7.4}$$

As we can see, the attackers odds increase by roughly 6% by making a guess on one of the digits. Since three login attempts is normally allowed before blocking a user out, the adversary can also (depending that he gets the same unknown challenge in every run), increase the percentage to 45.8% as follows:

$$\begin{aligned} P = \left(\frac{11}{14}\right)^4 + \left(\left(1 - \left(\frac{11}{14}\right)^4\right) * \frac{1}{8}\right) &= 0.381 + \left(0.619 * \frac{1}{8}\right) \\ &= 0.381 + 0.077 \approx 45.8\% \end{aligned} \tag{7.5}$$

As can be derived from this discussion, the main security of the scheme lies in the gesture-color associations. When an adversary has broken these, it is just a matter of acquiring/observing enough challenges. As stated earlier, a user can easily protect two of his gesture-color associations, something which will decrease the attackers chances of gaining access significantly. As shown below, there exist 336 unique challenges in this scheme, when we have 4 colors and 8 PIN-digits (represented by C and D respectively, in the formula below).

$$\frac{(D * C) * ((D-1) * (C-1))}{2} = \frac{(8 * 4) * (7 * 3)}{2} = 336 \tag{7.6}$$

In order to determine how many of these challenges an attacker knowing two gesture-color associations and 4 PIN-index combinations, can answer, we have to investigate the anatomy of the challenges. As we have 4 different colors, there exist 6 unique ways of combining these; *(A,B)*, *(A,C)*, *(A,D)*, *(B,C)*, *(B,D)*, *(C,D)*. Each of these pairs can have $7 * 8 = 56$ different challenges, which is verified by the fact that $56 * 6 = 336$. Intuitively, the attacker can in 4 of the pairs (where he knows one of the colors) answer 50% of the challenges, as he knows the gestures corresponding to half of the colors, as well as half of the PIN-index relationships. Moving on, there is one pair in which the attacker cannot answer any challenges, as both colors are unknown to him. Lastly, there is also one pair in which he knows the gestures corresponding to both of the colors in the challenge. In this case, the attacker can answer all challenges expect the ones where both PIN-index challenges are unknown to him. In order to determine how many crackable challenges there are, we can use the equation below, where n is the number of colors unknown to the attacker, and N is the number of digits in the PIN.

$$\text{Crackable challenges} = N * (N-1) - n * (n-1) + 4 * (N-1) * (N-n) \tag{7.7}$$

The lefthand side of formula represents the pair where the attacker knows both colors, while the righthand side represents the 4 pairs in which he only knows one of the colors. Continuing our earlier example, we get the following number of crackable challenges (assuming that the knows 4 PIN-index relationships and 2 gestures-color associations):

$$(7 * 8) - (4 * 3) + 4 * (7 * 4) = 156 \tag{7.8}$$

We can now calculate an attackers chance of successfully attacking the scheme under the aforementioned circumstances, as follows:

$$\left(\frac{\text{Number of crackable challenges}}{\text{Total number of challenges}}\right)^4 = \left(\frac{156}{366}\right)^4 = 0.033 = 3.3\% \tag{7.9}$$

By applying the simple security mechanism which protects our gesture-color associations we have effectively lowered the attackers chance of successfully attacking the protocol from 38.1% to 3.3%, something which is especially promising when we take the extremely open attack scenario into consideration. Also, as stated in Section 8.2.1, it is highly unlikely that an attacker can acquired video footage that clearly shows all the challenges and gestures performed, something which increases the observation resistance additionally. As described earlier, an attackers chance of successfully attacking the protocol increases with the number of decoded PIN-index relationships. Figure 16 illustrates the relationship between decoded associations and the corresponding probability of a successful attack.

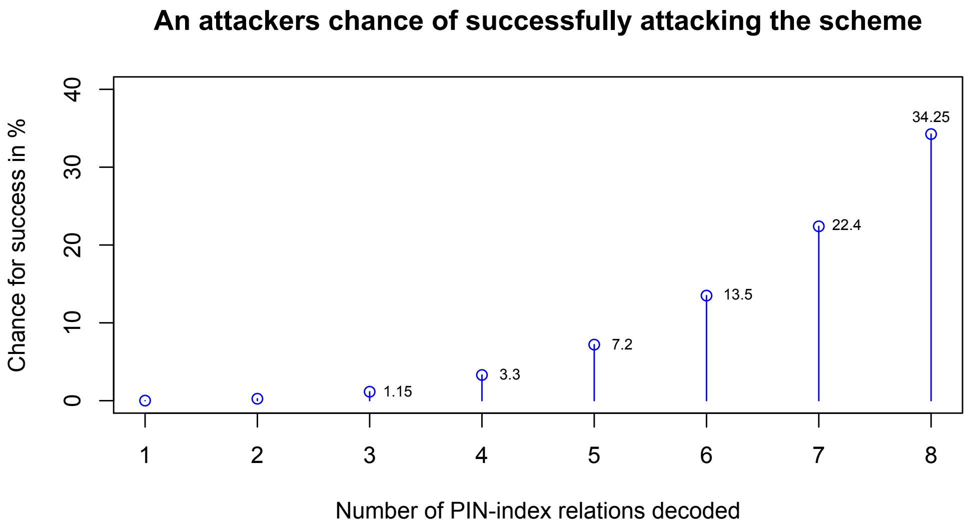


Figure 16: Shows the attackers chance of successfully attacking scheme. The percentages in this graph is viable when the attacker have decoded two of the gesture-color associations (the decoy colors).

What is interesting to see is that even though an attacker have decoded all PIN-index relationships, he still has a lower chance of succeeding (34.2%), than when he had decoded only 4 PIN-index relationships, but all of the gesture-color associations (38.1%). This clearly confirms that the gesture-color associations are more important than the PIN-index relationships.

7.2.3 Attack experiment

In this scheme we gave the attackers full observability on video. This includes the possibility for repeatability (watching each login sequence several times) and a perfect angle which clearly shows all challenges and gestures performed during the login attempts.

Experiment goal

The goal of this experiment was to investigate how resilient our scheme is against full observability (video-taping) attacks. We expected that an attacker would be able to deduce the decoy gesture-color association at one point, but we were interested in seeing how much effort it required, and also how many login sequences an attacker needs to gain access to, in order to decode the secret relations. We have therefore outlined two research questions that will be answered throughout this experiment:

1. How challenging is it for an attacker to obtain video footage, which clearly shows all the challenges and gestures performed?
2. How resilient is the scheme once the attacker has access to perfect video footage?

The second research question was answered in Section 7.2.2, where it was found that the number of decoded PIN-index relationships is the variable that decides an attackers chance of successfully attacking the scheme, as shown in Figure 16. An analysis of how many challenges an attacker needs to video tape in order to deduce x number of PIN-index secrets is presented in Chapter 8.

Experiment constraints and prerequisites

Because the flow of the protocol is random, and never identical from one login sequence to another, the results we get when it comes to how much effort and how many sequences we need to crack the scheme, will never be entirely accurate. However, they will, along with the theoretical evaluation in Section 7.2.2, give us an approximation of the scheme's resistance.

Experiment protocol

Since the attacks in this experiment had to be done offline, and requires a significant amount of time, we will act as the attackers. We videotaped two participants while they performed 5 login attempts. In the context of this scheme, we did not worry about the number of impostor and genuine attempts, but rather how many login sequences of the same user an adversary needed to obtain, in order to deduce the victims secrets. This is because direct replay attacks are theoretically and practically impossible (within the error margins of the PRNG). Also, human observation attacks are highly unlikely, as the decoding of the gesture-color associations needs an extensive investigation over time.

For this reason, we chose to use the full observation resistance scenario, with videotaping as the adversaries tool to crack the scheme. The main goal of this experiment reflects this decision as it focuses on resistance against high-end observation attacks on video. Of course, the resistance against replay and (human) observation attacks come at a price, and we acknowledge that this scheme might be harder to use than the previous one. This is because the user needs to remember both the 8 digit PIN-code, and the gesture-color associations. However, as we wanted to achieve full human observation resistance, and also resilience against video observation, this is something that we considered as a prerequisite.

Attack scenario

Although we have more or less described the attack scenario above, there are a few things that need to be described in more detail. In this scheme, observation is in the form of video footage with full observability. This means that the attacker has access to perfect images, clearly showing the whole login procedure.

The main reason for testing the scheme under such severe conditions is that we also wanted to get a baseline on how hard it is to obtain this kind of footage. Although we know that our scheme can be cracked under such circumstances, we can deduce information about how hard it is to crack this scheme under "normal" observation circumstances, by using the worst case scenario as a baseline. It is also hard to describe a scientifically sound attack scenario with "normal" observation, as this introduces many unforeseen variables. It is our belief that this scheme is "breakable" under perfect conditions if an attacker can videotape enough login attempts, but that such conditions are hard to obtain.

Further on, when it comes to protocol knowledge, the attacker is us (the developers of the scheme), meaning that the attackers have full knowledge of the protocol and which secrets needs to be decoded. In a real life scenario, an attacker would never have the perfect vision we get in our tapes, but we chose to test the strength of the protocol under these conditions nevertheless. The fact that this attack scenario presents the attackers with abnormally good vision and protocol knowledge, was taken into account when assessing the strength of the scheme in Chapter 8.

Experiment execution

The execution of the experiment follows the prerequisites and the attack scenario described earlier. We made an assumption that an adversary seldom have access to more than $2 - 3$ sessions from the same victim. Due to the fact that the challenges presented by the scheme is pseudo-random (this is of course limited by the PRNG), there should in principle not be any need for more than 1 participant. However, we decided to use two participants, which each conducted 5 login attempts. This way we could also learn something about the scheme's usability.

We recorded the device with full visibility, meaning that the images on the device and the gestures performed, clearly shows on the video. The camera was placed behind/above the device for maximum disclosure. Similarly to the other scheme, a failure in the gesture recognition enforced a repetition of the login sequence, as this is what would have happened in a real life scenario.

Attack protocol

This section describes the method used to deduce the PIN-index and gesture-color relationship in our security assessment. Assessing the strength of this protocol requires more rigid work than with the previous protocol, where the "attackers" (which was our participants), either knew the PIN at the end of the login sequence or not. In this protocol, we did as described above, act as the attackers, and tried to break the protocol by investigating a number of login sessions. To clarify, the following secrets needs to be decoded by the attackers:

- The PIN digit-index relationship. The goal for the attackers is to deduce as many of these as possible by using the available (videotaped) sequences.
- The gesture-color associations.

As stated earlier, the first step in attacking this protocol is to decode as many gesture-color associations as possible. This can be achieved by looking at a number of challenges in combination, where the same color and gesture appear with different opposing colors. Section 7.2.1 describes the anatomy of the challenges in more detail, while an example is given in Figure 17.

Figure 17 illustrates an example of how one of the four challenges would look like. In this example, the user can pick between entering the PIN digit corresponding to index 3 or 7 of his full PIN-code. Should the user choose to enter the digit corresponding to index 3, he would have to perform the gesture corresponding to the red color. As mentioned before, the user chooses the gesture-color association along with his PIN-code. Since these challenges are random, direct replay attacks are out of the question. What the attacker needs to do is to look at a number of challenges in combination, and see if he can deduce the relationship between a gesture and a color. Should the user perform a back flip and enter the digit 1 as the answer to the challenge presented in Figure 17, the

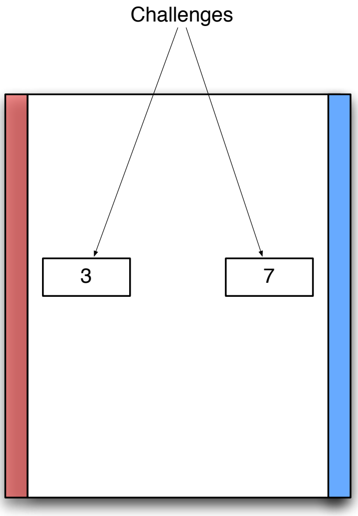


Figure 17: An example of how the user sees the challenges presented by the scheme

attacker have learned that the digit 1 corresponds to either index 3 or 7 from the users full PIN. He have also learned that a back flip either corresponds to the red or blue color (since these were the colors in the challenge). The attacker now needs to remember these variables and look at other challenges where either the color red or blue occur. Should the attacker for example see a challenge that has the color combination red/black, in which the user performs a back flip, the attacker have learned that red corresponds to a back flip, since red is the only color present in both challenges. He have also then, by back-tracing the sequence, learned that the digit 1 corresponds to index 3 of the user full PIN-code, since this was entered by the user in the previous challenge.

By using this method, an attacker can eventually decode the decoy parts of the gesture-color associations. Having done this, the attackers next job is to obtain enough challenges to decode the PIN-code, in order to increase his chance of success. How many challenges an attacker needs, and how much effort it requires of the attacker to break the protocol, is investigated in Section 8.2.

Participants

Due to the fact that most of the security evaluation of this scheme is conducted offline, we only used two participants. Each of the participants performed 5 login sequences using their own specified PIN and gesture-color associations. The reason why we did not need any more participants is that we only needed people to use the scheme live while we investigated how easy it is to videotape. We used a theoretical approach to conduct the remaining security assessment.

The participants were both male, both 23 years old, and students at HiG. Both were familiar with the Apple platform and authentication mechanisms in general.

8 Analysis and results

This chapter presents an analysis of the results gathered from the experiments described in Chapter 7. We will assess the shoulder surfing resistance of both authentication schemes, and the results will form the basis for the following discussion. As this chapter mostly presents an analysis of results, we refer the reader to Chapter 7 for more information about the experiments in general. It is also important to keep the extremely open attack scenarios in mind while reading the results.

8.1 Gesture and PIN-code based authentication

This section presents the results from the experiments we conducted on the scheme outlined in Section 7.1. The explicit protocol details of the two attack scenarios is more thoroughly described in Appendix D.1.

An additional point to keep in mind while reading this section is that when we applied the identical attack scenario on the normal PIN entry scheme, our participants were able to deduce the full PIN-code in 98.75% of the attempts.

8.1.1 Simple login scenario

The simple login scenario is where we do not apply the overwrite mechanism. What is important to keep in mind while reading these results is that if a participant deduced the PIN on the first attempt, we changed the PIN and performed another login attempt. However, if he did not, we performed the identical login procedure again (with the same PIN-code). As stated in the experiment description, we had 20 participants which acted as attackers, and we made sure that all participants fully understood the protocol before initiating the experiment.

Out of these 20, 3 managed to deduce the PIN-code on the first attack attempt, while 3 more deduced the PIN in their second attempt (they had then seen two identical login sequences). As stated in the experiment protocol, we changed the PIN for the 3 that deduced the PIN on their first attempt, and ran another login attempt. In this case, only 1 out of the 3 was able to repeat their success. This gives us the following odds that an attacker breaks the protocol in the first round (when we do not apply the overwrite mechanism):

$$\frac{\text{Number of successful attacks in first attempt}}{\text{Total number of first time attempts}} = \frac{4}{23} = 17.39\% \qquad (8.1)$$

Further on, we had 3 cases where an attacker managed to deduce the PIN on their second attack attempt, after observing two identical login sequences. Since 3 participants made it on the first try, this leaves us with 17 attempts where the participant did not deduce the PIN on their first try. This means that $3/(20-3) = 17.5\%$ of the participants that did not manage to deduce the PIN on the first attempt, managed it after watching two identical login sequences. By combining these error rates we get a total attack success rate of 17.5% throughout the entire simple login scenario, as shown in equation 8.2.

$$\frac{\text{Total number of successful attacks}}{\text{Total number of attempts}} = \frac{4+3}{23+17} = 17.5\% \tag{8.2}$$

8.1.2 Random login scenario

As described in Section 7.1, and Appendix D.1, this scenario is where we employed the overwrite mechanism to throw off the attackers. We had the same 20 participants, and more or less the same methodology. In the first login attempt, we performed one overwrite, and then x overwrites in the second attempt[1].

From the 20 first attack attempts, 2 participants were able to deduce the PIN code when we applied the overwrite mechanism once. Following the experiment protocol; we changed the PIN code and increased the number of overwrites, before conducting the second attack attempt for these participants. In this case, none of them managed to successfully deduce the PIN. For the participants that were unable to deduce the PIN-code in their first attempt, we made another attempt where the PIN remained the same, and the number of overwrites was either 1 or 2. The victim randomly chose this. After watching another login attempt, with the same PIN code, 4 additional participants were able to deduce the PIN-code.

To summarize; $2/20 = 10\%$ of the participants deduced the PIN code on their first try. Further on, none of these two managed to repeat their success when we changed the PIN and ran another login attempt. However, 4 new participants managed to deduce the PIN-code after watching the same PIN-code being entered two times (with different input sequences). This gives us an total attack success rate of 15%, as shown below;

$$\frac{\text{Total number of successful attacks}}{\text{Total number of attempts}} = \frac{2+4}{40} = 15\% \tag{8.3}$$

8.1.3 Qualitative participant feedback

We asked the participants a few questions about the scheme and how they considered its security properties. The participants found the scheme fun and interesting to use. When it comes to the security, they stated that the time it took for the gesture recognition to finish, gave them the seconds they needed to "place" the digits in their minds. They said that it would have been much harder to deduce the PIN-code if the gesture recognition had gone faster. This is also something that we believed could happen, however, as we did not have access to a faster device, this will simply have to be taken into consideration when assessing the strength of the protocol. Many of the participants also found the overwrite mechanism confusing, and they stated that it was only due to the waiting time enforced by the gesture recognition, that they were able to deduce the PIN with overwrites.

Further on, many of the participants stated that the attack scenario (with full observation) was unrealistic, and that it would be much harder to deduce the PIN in a more realistic scenario. They also stated that in a real life scenario, where one does not have full observation, one still have a chance of gaining access to a normal PIN entry scheme even though they only were able to observe 3 of the 4 PIN digits, if one consider guessing. In our scheme though, where the gestures decides the placement, and we

[1] In this attack scenario, it was as described in the experiment protocol, up to the victim to decide how many overwrites he wanted to perform

have the possibility of overwriting digits, deducing the PIN without observing the entire login sequence perfectly, would be more challenging. These statements are in line with our assumptions, and they allow us to draw the conclusion that this scheme does indeed add a significant amount of entropy to the scheme, without affecting the usability of the scheme tremendously.

Usability

Five of our participants took the time to try and use the scheme. They found the protocol easy to remember and utilize, and also quite user friendly. There were cases where some of our participants gestures did not match the templates, but these cases were good within the EER of 5%. All of them stated that they would indeed be willing to use this protocol as their authentication mechanism, to gain the extra security. They also agreed with us in the case that if we had implemented the protocol on a newer device, this would not only make it harder to deduce the PIN for an attacker, but it would also make the login time insignificant, as the time used for the gesture recognition would be minimized.

8.1.4 Security assessment

When assessing the results gathered from our experiments, it is important to take into account both the experiment protocol and the attack scenario. Since our aim was to make a scheme, which is more resilient to observation attacks than the standard PIN entry method, we asked our participants to observe a standard PIN login under the same observation scenario. In this case, we had only 1 login out of 80 login attempts, where the attacker was not able to deduce the PIN code. It should also be mentioned that this participant claimed he was not ready, and made it two times in a row after the first failure. The reason why we did not conduct a more elaborate baseline experiment is because the vulnerability of the normal PIN-code scheme is well known, and acknowledged in all relevant references in Chapter 3.

For this reason, it is safe to say that our scheme is significantly more resilient to shoulder surfing attacks than the normal PIN entry scheme. In the same test, our scheme was in 85% of the attempts resilient to shoulder surfing attacks, while the standard PIN entry method was only secure in 1.25% of the cases. Although this is a significant improvement, there are two aspects that we believe will increase the schemes resilience additionally:

- As stated earlier, implementing the scheme on a newer device will minimize the time used for gesture recognition. This will make it harder for an attacker to observe and deduce the PIN code, as he is not granted a few seconds of thinking time between each PIN digit entry.
- In a real life scenario, an attacker will almost never see the complete login perfectly. In a normal PIN entry scheme, an attacker has a 10% chance of guessing the missing PIN digit. In our scheme, missing a PIN entry or a placement gesture, provides the attacker with a significantly more challenging problem, as he does not know if we performed an overwrite, nor where the digit was placed.

Another important point is that although it may not show very clearly from the results, the overwrite mechanism made it harder for the attackers to deduce the PIN. In the simple scenario there were many cases where the participants deduced 2 or 3 PIN digits. However, when we enforced the overwrite mechanism, our participants were generally further away from deducing the PIN. Many of them failed to "see" the overwrite in the sense that they believed that the decoy digit was a part of the victims PIN. One thing that should be mentioned is that an attacker will most likely get more skilled in deducing the PIN over time. However, we believe that the points discussed in the list above will thwart this risk.

8.2 Challenge-response scheme

In this section we describe the methods used, and the results gathered from the attack experiment on our challenge-response scheme. Although a subpart of the second research question outlined for this experiment was answered in Section 7.2.2, this section investigates how many login sequences of the same person an attacker needs to obtain in order to decode x number of PIN-index relationships.

8.2.1 Achieving adequate video quality

As described in Section 7.2.3, we wanted to test this scheme under the most challenging conditions. This means that an adversary has access to video footage clearly showing all of the challenges displayed on the screen, throughout the whole login sequence. One of the research questions we wanted to answer by conducting this experiment was to find out how hard it is to obtain such video footage.

We therefore searched help at Gjøvik Movie Workshop which is located in Ungdommens Hus at Gjøvik. We got help by a trained professional when it comes to choosing cameras to use, and to setup the shooting environment. Even though we tried everything from handheld HD cameras to high-end DV camcorders, we experienced great troubles when it comes to obtaining footage that could be used to crack the scheme.

We started with the most realistic tool for an attacker; handheld cameras, and tried to shoulder surf the login sequence. However, we soon realized that this was virtually impossible as we could not even get the hand held cameras to obtain footage which clearly showed all challenges displayed on the screen. In fact, the only information we could retrieve from these footages was the color challenges. Also, the focus on these hand held cameras was very vulnerable to shaking by both the person holding the camera, and the participant performing the gestures. Once the participant moved a bit, the camera had to be refocused. The need for focus is because the iPods screen is reflective, which means that protective screen filters might improve the attackers chance of acquiring adequate video footage.

After a couple of hours of testing, we concluded that we had to employ an even more restricted environment. We found that by using an external light source, and placing a Sony pd150 [62] with a close range lens on a tripod 10 centimeters from the device, we started getting closer to achieving adequate footage. Figure 18 illustrates the environment we ended up with.

In order to get clear footage of both the screen and the gestures performed, we got help by the staff at the movie workshop to tune the cameras colors and focus. We also made an effort in setting the hertz on the camera to the one of the iPod Touch screen. As can be seen in Figure 18, we ended up with placing the camera almost vertically above the device, as this was the only way we could get clear footage of the screen,

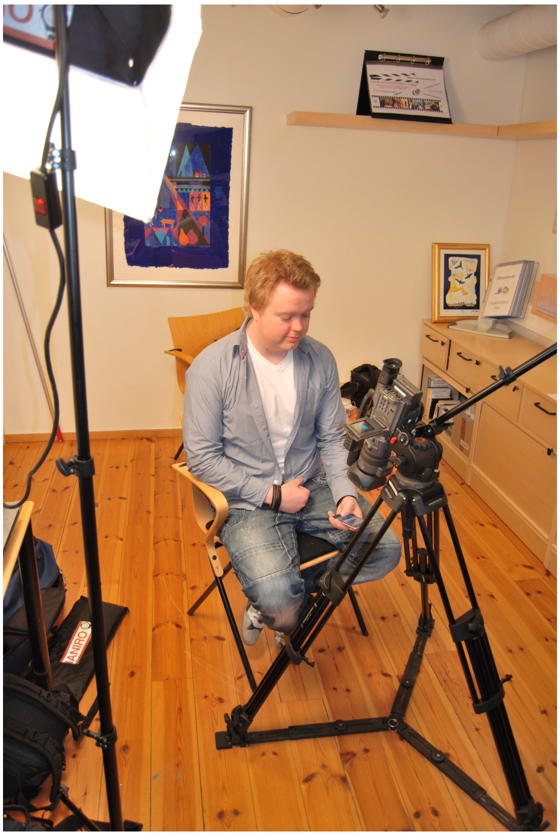


Figure 18: Shows the setup used to obtain full observability in our video recordings

while still being able to see the gestures. The device was held approximately $10cm$ below the camera, and it was very important that we held the device in approximately the same position, for each taped login sequence. If we moved the device further away, the focus had shifted, and we could not deduce the index challenges. Due to the problems with getting the camera's focus perfect, in order to see both the screen and the gestures performed, we used around a hour to get the environment right.

Even after enforcing an optimal scenario for the attacker, we still had to employ different post-processing techniques to get the index-challenges to clearly stand out on the video. We had to lower the contrast and lightning of the movie so that the black index numbers on the light-blue background, clearly stood out.

8.2.2 General information about the security assessment

As mentioned earlier, the scheme's resistance against observation attacks, when an adversary has full observability, can only be proven probabilistically to a certain degree due to the human factor. Since the strength of the scheme is affected by how the user answers the challenges, we will provide an assessment of the scheme using two different scenarios; one where the user is unaware of the possibility of protecting himself, as discussed in Section 7.2.2, and one where he is aware. As the main objective of this analysis was to assess how easily, and how many login sequences an attacker needs to decode x number of PIN-index relationships, we recorded a number of login sequences which is used in the following analysis.

We also wanted to assess how many login attempts an adversary would need to crack the gesture-color associations. We have $5 * 2$ login sequences, with two different PIN-codes and gesture-color associations, as shown in Table 8. We are aware that this is not a high number when it comes to statistical significance, however, they are only used as an example of how an attacker would attack this protocol, as the strength of the protocol was mathematically assessed in Section 7.2.2. We can, by using the recorded login sequences derive $\frac{5^2-5}{2} = 10$ unique combinations where an attacker has access to two unique login sessions, for each PIN/color-gesture setup. This gives us a total of 20 examples where an attacker has two unique login sequences at his disposal, in order to attack the gesture-color associations.

The reason why chose to use two sequences as a tuple, is because we believe that it is highly unlikely that an attacker can obtain more than two login attempts from the same person under these conditions, considering how difficult it was for us to acquire adequate video footage of the login sequence. In fact, our experiment shows that it is highly unlikely that an adversary can even get a hold of one login attempt with this kind of quality.

Table 8: Describes the gesture-color associations and the PIN-code in our analysis.

Setup1: PIN: 22347815		**Setup2:** PIN: 41953618	
Gesture	**Color**	**Gesture**	**Color**
RF	Red	RF	Blue
LF	Blue	LF	Yellow
BF	Green	BF	Red
FF	Yellow	FF	Black

Attack model

Before heading into the analysis of the scheme, we will present an example of the method used to deduce the secret associations. The first and most important aspect is getting video footage which provides adequate vision. Then, by looking at this video, an adversary can after a number of challenges, decode the decoy gesture-color associations. This can be achieved by looking at the colors displayed in each challenge in combination with the gesture performed by the victim. If the attacker writes this relationship down, he should end up with a list like the one presented in Table 9. To make this example we used a randomly picked sequence videotaped from setup 2, shown in Table 8. Note that the user conducting this particular login attempt is unaware of how he can protect himself.

Table 9: An adversary notes after watching a login attempt of an user unaware of how to protect himself.

Left color	Right Color	Gesture performed
Yellow	Black	LF
Blue	Red	BF
Yellow	Red	LF
Red	Black	BF

After obtaining this list, the attacker can treat this as a linear equation system with 4 unknown relationships. The attacker must choose one and one color and try to deduce information about which gesture it is associated with. The important thing is to look after the same gesture being performed in different challenges. If we look at challenge 1 and 3, the user performs a left flip on both occasions. Since yellow is included in both challenges, with two different opposing colors, we learn that yellow has to correspond to a left flip. Using the same methodology we learn that the red color has to correspond to a back flip, by looking at challenge 2 and 4. The attacker has now deduced 2 out of 4 gesture-color associations by using one login sequence. Some might argue that the user was unlucky with the challenges, but the most important thing is how easily he could have avoided giving away such information, by using the simple technique described in Section 7.2.2. To clarify, we will give an example of how the user could have utilized this technique to protect himself. Using the same challenges as shown in Table 9, and the same setup (Setup 2, in Table 8), the user could have protected his gesture-color associations by using the gestures shown in Table 10 as responses to the challenges. In this example, black and yellow was chosen as *safe colors*.

Table 10: An adversary notes after watching a login attempt of a user aware of how to protect himself.

Left color	Right Color	Gesture performed
Yellow	Black	LF
Blue	Red	BF
Yellow	Red	BF
Red	Black	BF

As we can see from the attackers notes in Table 10, the attacker can now only deduce that red is associated with a back flip.

8.2.3 Results from the attack experiment

As stated above, we have 20 unique sets, consisting of two login attempts that form the basis for this analysis. The method used is thoroughly described in 8.2.2. We performed two separate analysis, one where the simple protection mechanism is employed, and one where it is not.

Results - User does not employ the protection mechanism

It was generated tables equal to the one shown in Table 9 for all of the login attempts.

Attacker has access to one login attempt

The first investigation we made was the case where the attacker only has access to one login attempt. Following the methodology above, we were able to deduce the full gesture-color association in 1 out of 10 logins. This login sequence is shown in Table 11.

Table 11: An example of a single login attempt which allows an attacker to deduce the full gesture-color association. Sequence from Setup 1 in Table 8.

Left color	Right Color	Gesture performed
green	yellow	BF
blue	green	BF
blue	green	LF
green	red	RF

As we can see, the user revealed more than necessary in this run. Had the user employed the protection mechanism, by for instance choosing green and yellow as his safe colors, he would not have revealed more than maximally two gesture-color associations. Table 12 shows an example of how the user could have answered the protocol using the protection mechanism.

Table 12: Shows how easy the user could have avoided revealing his entire gesture-color association in one attempt.

Left color	**Right Color**	**Gesture performed**
green	yellow	BF
blue	green	LF
blue	green	LF
green	red	RF

The modified answers shown in Table 12, effectively turns this login attempt into strong one, where the user only reveals one secret; that red corresponds to a right flip. An attacker can in this case only deduce one PIN-index relationship, which would present him with extremely slim chances of successfully attacking the scheme, as shown in Figure 16.

Attacker has access to two login attempts

As stated earlier, it was our hypothesis that an attacker should be able to deduce the gesture-color associations in most cases, if he is given access to two successful login attempts, where the user does not apply protection mechanisms. This hypothesis holds as we in our analysis was able to deduce the gesture-color associations 16 out of 20 times.

The reason why this happens is that the number of challenges doubles, and therefore so does the odds of revealing the associations. When it comes to the 4 times where we could not deduce the gesture-color associations, these were broken once we added a third login attempt to the equation.

Results - User employs the protection mechanism

The purpose of this analysis was to confirm the statements made in Section 7.2.2, which states that an attacker never can deduce more than two gesture-color associations, if the user employs the simple protection mechanism. To verify this, we played the game of the victim with the same challenges as presented above, and effectively protected the associations in all cases. This is as expected, as it is mathematically impossible to deduce more than 2 of the 4 gesture-color associations if the user employs the protection mechanism described in Section 7.2.2, regardless of the amount of login sequences he gains access to.

Therefore, in order to protect this scheme against even the most severe observation attacks like the one employed in this experiment, all the user has to do is to employ an extremely easy to remember protection mechanism. Of course, an adversary can, although highly unlikely, deduce the color-gesture relationship by guessing, but this is outside the scope of our experiment.

8.2.4 Deducing the PIN-code

As found in Section 7.2.2, it is the number of decoded PIN-index relationships that decides the attackers odds of attacking the scheme. Even though an attacker cannot deduce more than 2 of the 4 gesture-color associations, he can still try to attack the scheme. He can, as described earlier, by looking at the videotaped login sequences, deduce one index-PIN relationship each time the user performs one of the two gesture-color associations known to the attacker. Since all challenges are random, it is not possible to determine exactly how many login sequences an attacker would need to deduce for instance 4 PIN-index associations. We can however, make an educated estimate based on the observations we have made, and a probabilistic argument;

Consider the case where an adversary has access to two login sequences ($4 * 2 = 8$ color-gesture challenges, which corresponds to 12 possible combinations of colors, since the same color cannot appear on both sides of the challenge), where the colors appear in perfect randomness (highly unlikely as the number of challenges is to small to gain a perfectly distributed sample set). In this case, the attacker sees known colors in 6 of the challenges, and he can in theory always deduce a PIN-index relationship in 2 out of these 12 challenges (the challenges where both decoy colors appear simultaneously). The attacker can also deduce a PIN-index relationship if the user performs the gesture that corresponds to the decoy color, in 4 of the other challenges. This means that if all of these prerequisites occur, and the scheme during these 8 challenges never display two identical challenges (which is quite unlikely), the attacker can deduce 4 PIN-index relationships by looking at two sequences, which gives him a 3.3% chance of getting the challenges he needs to attack the scheme.

In reality, the scheme will never produce such a perfectly distributed set of challenges, which means that an attacker in some cases will need to gain access to more than 2 login sequences to deduce 4 PIN-index relationships. Should however an attacker manage to obtain enough login sequences to decode for instance 6 PIN-index relationships, this still only gives him a 10.8% chance of successfully attacking the protocol, as shown in Figure 16. We can therefore safely conclude that since an attacker is highly unlikely to achieve adequate video footage of more than 1 or 2 login sequences of the same victim, an attacker will almost certainly not be able to deduce more than 4, maximally 5, PIN-index relationships. As this leaves him with a 3.3% and 7.2% chance of success respectively, we can clearly conclude that this scheme offers a significant amount of observation resistance, even against video observation attacks.

9 Conclusion

As the main goal of this thesis was to incorporate hand gestures as an additional modality in mobile authentication schemes to mitigate the risk of shoulder surfing attacks, we have worked with a number of different topics throughout this thesis. We have developed and experimentally verified a gesture recognition module for mobile devices, that effectively have been incorporated into two unique authentication schemes, which both offer a significant amount of resistance against observation attacks.

Our first contribution was the *development of a dynamical many-to-one recognition module for hand gestures*. As stated throughout Section 3.1, there has been done a lot of work in the field of hand gestures, though none have proven to be very successful. We therefore decided to conduct a thorough analysis of the accelerometer signals produced by our gestures. To limit the scope of the thesis, we restricted ourselves to 6 different gestures, in which each of is thoroughly described in Chapter 4. We developed a data acquisition program for the iPod Touch which was, as described in Chapter 4, used to acquire a total of 600 hand gesture samples, divided on 20 participants and 6 distinct gestures. Further on, we performed a distinctiveness experiment, which is described in Section 6.2, with the goal of investigating how distinctive raw hand gesture samples were, and how accurately we could recognize and separate between different gestures. In this experiment we achieved an EER of $27 - 28\%$.

As our goal was to implement hand gestures in authentication schemes, we had to lower our error rates significantly. We found it reasonable from both a user friendliness, and a theoretical perspective, to generate general templates for each gesture, which should be representative for all participants. In order to generate such templates we had to investigate different mathematical operations that could be used to achieve representative templates. After having analyzed the signals, we found that using a median calculation, as described in Chapter 5, gave us the most representative templates[1]. Since the data from the data acquisition experiment was used in the generation of these templates, we gathered new samples from 18 participants that were used to calculate error rates. *By using the templates we effectively lowered our EER to 8% and 5%*, with and without arbitrary gestures respectively. These error rates were achieved without any significant preprocessing steps, and by using DTW as our distance metric. After having conducted several analyses and recognition experiments on our hand gesture dataset, it is safe to say that the signals derived from the accelerometer provides us with accurate and well defined data, suitable for authentication purposes.

As it was our aim to develop recognition modules that should be used directly in our authentication schemes, achieving good error rates, while keeping the computational time and cost low, was crucial. As we achieved a 5% EER on the constrained gestures without performing any elaborate preprocessing steps, we believe that in newer and faster devices, one can probably lower the error rates additionally, as discussed in

[1] Meaning that the general templates are as close as possible, in terms of DTW distance, to all participants for one specific gesture.

Chapter 10. Although getting good recognition rates was important, it is not the gestures themselves that add entropy to our authentication schemes. In our case, the gestures are merely used as a second modality in the authentication schemes.

Our next major contribution was the *development and verification of a set of observation resistant multi-modal authentication schemes*. As stated in our research questions in Section 1.5, our aim was to mitigate the threat of shoulder surfing attacks without affecting the usability of the scheme to much. Since shoulder surfing scenarios can range from a person watching, to more elaborate attacks utilizing cameras, we developed one scheme for each of these scenarios. In all our attack experiments, full observability was enforced as we wanted to test the worst case scenarios.

In our first scheme (described in Section 7.1), the *gestures were used to place the PIN digits in the correct order*, giving a user the possibility of entering his PIN digits in an obscured manner. Although not fully robust, the scheme allowed us to investigate how much extra security including gestures gave us, and how effective the implemented obfuscation mechanism were. In the attack experiment described in Section 8.1, we utilized two different sub-scenarios; one where the user applied the overwrite mechanism, and one where he did not. In these scenarios, we experienced that an attacker could deduce the PIN after watching two login attempts in 15% and 17.5% of the cases, respectively. When considering the attack scenario, and the fact that the attackers was given time to think between each PIN entry (due to the time our gesture recognition modules use to recognize a gesture), these rates are very promising. Also, this scheme is very easy to use, something which our participants agreed upon, as described in Section 8.1.3.

Since our first scheme is vulnerable against video observation attacks, we *developed a challenge-response scheme with the aim of being more resilient against such attacks*. This scheme is presented in Section 7.2, and analyzed in Section 8.2. We decided that due to the nature of the scheme, it would not be possible for a human observer to crack the scheme (unless he has an eidetic memory), and we therefore enforced the worst attack scenario an authentication mechanism could possibly face; full observability on video. In this attack experiment we found, as described in Section 8.2.1, that it is extremely difficult for an attacker to obtain video footage that clearly shows all the challenges and gestures performed. Since such footage is essential for cracking the scheme, this adds another layer of shoulder surfing resistance. The schemes resilience against these kinds of attacks was mathematically assessed throughout Section 8.2 and 7.2.2, where it was found that it is the number of decoded index-PIN relationships that affects the attackers chance of successfully attacking the protocol, as shown in Figure 16. Even though an attacker manages to decode 4 PIN-index relationships, he still only has a 3.3% chance of getting the challenges he needs, in order to successfully attack the protocol. This allows us to clearly conclude that this scheme offers a significant amount of resilience against even the worst kinds of observation attacks. Although this scheme might demand more from the user, we find it reasonable to state that it is applicable for high risk applications, like for instance a mobile password vault.

The experiments we conducted on our authentication schemes *clearly shows that including hand gestures adds additional entropy to our authentication schemes, and also increases the shoulder surfing resistance significantly*. We also found that hand gestures can be modeled and recognized quite accurately by the usage of accelerometer signals, and we believe that this area of research can be of great interest in the future.

10 Future Work

Since the recognition of hand gestures is a relatively new area of research, there are many aspects that deserve further scrutiny. Newer, more powerful devices, permit the inclusion of different pre-processing steps that can be used in an attempt to lower the EER. After investigating the signals, we believe that the most viable approach would be to implement a method which extracts and focuses on the areas of the signal where the most acceleration occurs. This can eliminate errors cases where signals from the same gesture produce high distances even though they are only shifted in time. Figure 19 shows an example where we have marked the area where the most significant acceleration occurs. Such an algorithm could be implemented by using a *sliding window* technique, which look for acceleration over and below a certain threshold to indicate the start and end of a gesture.

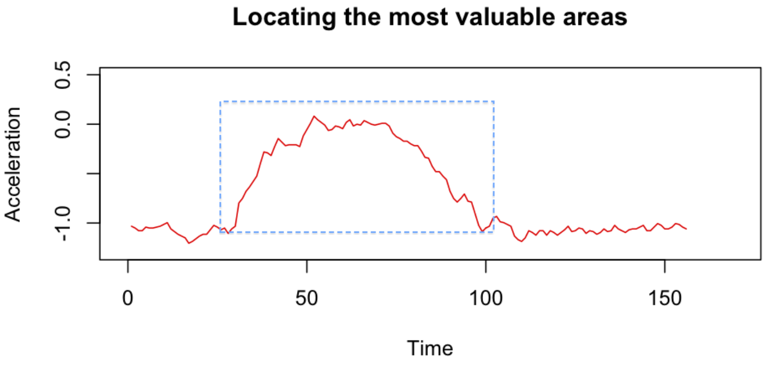


Figure 19: Shows the most valuable areas of the signal, when focusing on acceleration.

Also, implementing a curve fitting function that normalizes all sequences to a certain length, could eliminate error cases where a user performs a gesture too slowly or too fast. Besides this, it would, as described in Section 5.3.1, be interesting to investigate template personalization methods. This would allow us to personalize the general templates over time and thereby not only decrease the error rates, but also make the templates somewhat personal in the sense that an attacker would have difficulties replaying a user's gesture.

Although we have not directly looked at the biometric aspect of hand gestures, this is an area of research that certainly would be interesting to study in more detail. We do however, believe that since the principal components describing each gesture is constant for all users, a biometric approach will have to focus on the individual persons shivering and other personal characteristics. In Figure 19, this would mean emphasizing on the areas not marked as *valuable*, as these are likely to contain user specific information like for example shivering. We have throughout this thesis highlighted many problem areas which can be taken into account in future research, and it is especially the human considerations discussed in Section 2.3, that we believe will be crucial to look into in eventual biometric approaches. Further on, a biometric approach will most likely be more susceptible to hardware noise and sensor errors than we were in our approach. If, for instance, a biometric approach would use shivering as a distinguishable factor, then such factors would need to be modeled extensively in order to clearly separate personal information from different kinds of noise. The inclusion of more accurate accelerometers and gyroscopes in newer devices, might also further enable biometric approaches to clearly separate these factors. Related to this, it would also be interesting to investigate the effect from both better sampling (higher density), and the length and complexity of gestures.

Regarding the authentication schemes, we believe that it would be harder for an attacker to attack our schemes if the gesture recognition took less time (especially in the scheme where the gestures place the digits), and it would therefore be interesting to assess the strength of the schemes on more powerful devices. Additionally, it would be interesting to test our schemes under more realistic attack scenarios, as this could prove that missing out on parts of the login presents the attacker with a more difficult problem in our schemes, than in normal PIN entry schemes. In this context one could also, although more general, conduct a formal analysis investigating how much entropy/information one can add to a password while still keeping the password and protocol easy to remember and utilize. As we suspect that people might be more comfortable with gestures that they use on a daily basis, investigating whether some types of combinations of modalities is easier to remember than others, might also be an interesting research topic in this context.

Aside from accelerometer derived gestures, mobile devices has additional modalities that can be used for authentication purposes. As an example, future research can utilize touch screens to include additional entropy and challenge response functions into the schemes. These could be used in conjunction with gestures, by for instance having the user move his finger on the screen in response to a particular challenge, while performing a gesture.

Bibliography

[1] Bours, P. August 2009. Authentication course, IMT4721. Gjøvik University College, Reader for Authentication course.

[2] Jain, A., Ross, A., & Prabhakar, S. Jan. 2004. An introduction to biometric recognition. *IEEE Transactions on Circuits and Systems for Video Technology*, 14(1), 4–20.

[3] Bishop, M. A. 2002. *The Art and Science of Computer Security*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA.

[4] The Internet Encyclopedia of Science. Linear Accelerometer, `http://www.daviddarling.info/encyclopedia/A/accelerometer.html`. Last visited: 10.06.2010.

[5] STMicroelectronics. October 2008. LIS302DL MEMS motion sensor 3-axis - ± 2g/± 8g smart digital output "piccolo" accelerometer, Product Description. ST research labs.

[6] Dan, P. & Tracey, P. October 2009. *Head First iPhone Development*. O'REILLY Media, 1 edition.

[7] Bellman, R. & Kalaba, R. Nov 1959. On adaptive control processes. *IRE Transactions on Automatic Control*, 4(2), 1–9.

[8] Myers, C., Rabiner, L., & Rosenberg, A. Dec 1980. Performance tradeoffs in dynamic time warping algorithms for isolated word recognition. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 28(6), 623–635.

[9] Holien, K. Gait recognition under non-standard circumstances. Master's thesis, Gjøvik University College, Box 191, 2802, Gjøvik, Norway, June 2008.

[10] Efrat, A., Fan, Q., & Venkatasubramanian, S. 2007. Curve Matching, Time Warping, and Light Fields: New Algorithms for Computing Similarity between Curves. *J. Math. Imaging Vis.*, 27(3), 203–216.

[11] Bookstein, A., Kulyukin, V. A., & Raita, T. 2002. Generalized hamming distance. *Inf. Retr.*, 5(4), 353–375.

[12] Yujian, L. & Bo, L. 2007. A normalized levenshtein distance metric. *IEEE Trans. Pattern Anal. Mach. Intell.*, 29(6), 1091–1095.

[13] Senin, P. Dynamic Time Warping Algorithm Review. Technical report, Department of Information and Computer Sciences, University of Hawaii, Honolulu, Hawaii 96822, December 2008.

[14] Sturman, D. & Zeltzer, D. Jan 1994. A survey of glove-based input. *Computer Graphics and Applications, IEEE*, 14(1), 30–39.

[15] Pavlovic, V. I., Sharma, R., & Huang, T. S. 1997. Visual Interpretation of Hand Gestures for Human-Computer Interaction: A Review. *IEEE Trans. Pattern Anal. Mach. Intell.*, 19(7), 677–695.

[16] Wu, Y. & Huang, T. S. 1999. Vision-Based Gesture Recognition: A Review. In *GW '99: Proceedings of the International Gesture Workshop on Gesture-Based Communication in Human-Computer Interaction*, 103–115, London, UK. Springer-Verlag.

[17] Harrison, B. L., Fishkin, K. P., Gujar, A., Mochon, C., & Want, R. 1998. Squeeze me, hold me, tilt me! An exploration of manipulative user interfaces. In *CHI '98: Proceedings of the SIGCHI conference on Human factors in computing systems*, 17–24, New York, NY, USA. ACM Press/Addison-Wesley Publishing Co.

[18] Wang, J., Zhai, S., & Canny, J. 2006. Camera phone based motion sensing: interaction techniques, applications and performance study. In *UIST '06: Proceedings of the 19th annual ACM symposium on User interface software and technology*, 101–110, New York, NY, USA. ACM.

[19] Ängeslevä, J., Oakley, I., Hughes, S., & O'Modhrain, S. Jan 2003. Body Mnemonics Portable device interaction design concept. *Proceedings of UIST'03, ACM*.

[20] Wigdor, D. & Balakrishnan, R. 2003. TiltText: using tilt for text input to mobile phones. In *UIST '03: Proceedings of the 16th annual ACM symposium on User interface software and technology*, 81–90, New York, NY, USA. ACM.

[21] Rekimoto, J. 1996. Tilting operations for small screen interfaces. In *UIST '96: Proceedings of the 9th annual ACM symposium on User interface software and technology*, 167–168, New York, NY, USA. ACM.

[22] Partridge, K., Chatterjee, S., Sazawal, V., Borriello, G., & Want, R. 2002. TiltType: accelerometer-supported text entry for very small devices. In *UIST '02: Proceedings of the 15th annual ACM symposium on User interface software and technology*, 201–204, New York, NY, USA. ACM.

[23] Hinckley, K., Pierce, J., Sinclair, M., & Horvitz, E. 2000. Sensing techniques for mobile interaction. In *UIST '00: Proceedings of the 13th annual ACM symposium on User interface software and technology*, 91–100, New York, NY, USA. ACM.

[24] Liu, J., Zhong, L., Wickramasuriya, J., & Vasudevan, V. Dec 2009. uWave: Accelerometer-Based Personalized Gesture Recognition and its Applications. *Pervasive and Mobile Computing*, 5(6), 657–675.

[25] Choi, E.-S., Bang, W.-C., Cho, S.-J., Yang, J., Kim, D.-Y., & Kim, S.-R. Dec. 2005. Beatbox music phone: gesture-based interactive mobile phone using a tri-axis accelerometer. In *ICIT 2005. IEEE International Conference on Industrial Technology*, 97–102.

[26] Guerreiro, T., Gamboa, R., & Jorge, J. 2009. Mnemonical body shortcuts for interacting with mobile devices. In *GW 2007: 7th International Gesture Workshop on Gesture-Based Human-Computer Interaction and Simulation, Lisbon, Portugal, May 23-25, 2007, Revised Selected Papers*, 261–271, Berlin, Heidelberg. Springer-Verlag.

[27] Gafurov, D. & Snekkenes, E. 2009. Gait recognition using wearable motion recording sensors. *EURASIP J. Adv. Signal Process*, 2009, 1–16.

[28] Gafurov, D. *Performance and security analysis of gait-based user authentication*. PhD thesis, University of Oslo, 2008.

[29] Rahman, M., Gustafson, S., Irani, P., & Subramanian, S. 2009. Tilt techniques: investigating the dexterity of wrist-based input. In *CHI '09: Proceedings of the 27th international conference on Human factors in computing systems*, 1943–1952, New York, NY, USA. ACM.

[30] Grandjean, E. Jan 1980. Fitting the Task to the Man: An Ergonomic Approach. *London UK: TAYLOR & FRANCIS LTD.*

[31] Crossan, A. & Murray-Smith, R. 2004. Variability in Wrist-Tilt Accelerometer Based Gesture Interfaces. In *MobileHCI 2004: 6th International Symposium*, Lecture Notes In Computer Science, 144–155.

[32] Mantyla, V.-M., Mantyjarvi, J., Seppanen, T., & Tuulari, E. Jan 2000. Hand gesture recognition of a mobile device user. In *ICME 2000. IEEE International Conference on Multimedia and Expo*, volume 1, 281–284 vol.1.

[33] Apple. February 18 2010. Motion based input selection, Pub. No.: US 2010/0042954 A1. United States Patent Application Publication.

[34] Mäntyjärvi, J., Kela, J., Korpipää, P., & Kallio, S. 2004. Enabling fast and effortless customisation in accelerometer based gesture interaction. In *MUM '04: Proceedings of the 3rd international conference on Mobile and ubiquitous multimedia*, 25–31, New York, NY, USA. ACM.

[35] Axelrod, S. & Maison, B. May 2004. Combination of hidden Markov models with dynamic time warping for speech recognition. In *IEEE International Conference on Acoustics, Speech, and Signal Processing, 2004. Proceedings. (ICASSP '04).*, volume 1, I–173–6 vol.1.

[36] Tamviruzzaman, M., Ahamed, S. I., Hasan, C. S., & O'brien, C. 2009. ePet: when cellular phone learns to recognize its owner. In *SafeConfig '09: Proceedings of the 2nd ACM workshop on Assurable and usable security configuration*, 13–18, New York, NY, USA. ACM.

[37] Patel, S. N., Pierce, J. S., & Abowd, G. D. 2004. A gesture-based authentication scheme for untrusted public terminals. In *UIST '04: Proceedings of the 17th annual ACM symposium on User interface software and technology*, 157–160, New York, NY, USA. ACM.

[38] Shi, P., Zhu, B., & Youssef, A. 2009. A PIN entry scheme resistant to recording-based shoulder-surfing. In *SECURWARE '09: Proceedings of the 2009 Third International Conference on Emerging Security Information, Systems and Technologies*, 237–241, Washington, DC, USA. IEEE Computer Society.

[39] Sasamoto, H., Christin, N., & Hayashi, E. 2008. Undercover: Authentication usable in front of prying eyes. In *CHI '08: Proceeding of the twenty-sixth annual SIGCHI conference on Human factors in computing systems*, 183–192, New York, NY, USA. ACM.

[40] Sabzevar, A. P. & Stavrou, A. 2008. Universal multi-factor authentication using graphical passwords. In *SITIS '08: Proceedings of the 2008 IEEE International Conference on Signal Image Technology and Internet Based Systems*, 625–632, Washington, DC, USA. IEEE Computer Society.

[41] Xiao, Y., Li, C. ., Lei, M., & Vrbsky, S. V. 2008. Secret little functions and codebooks for protecting users from password theft. In *Proceedings of IEEE ICC 2008*, 1525–1529, Washington, DC, USA. IEEE Computer Society.

[42] Hoanca, B. & Mock, K. 2009. A Theoretical Framework for Assessing Eavesdropping-Resistant Authentication Interfaces. In *HICSS '09. 42nd Hawaii International Conference on System Sciences*, 1–10.

[43] Lei, M., Xiao, Y., Vrbsky, S. V., Li, C., & Liu, L. 2008. A virtual password scheme to protect passwords. In *Proceedings of IEEE ICC*, 1536–1540, Washington, DC, USA. IEEE Computer Society.

[44] Roth, V., Richter, K., & Freidinger, R. 2004. A PIN-entry method resilient against shoulder surfing. In *CCS '04: Proceedings of the 11th ACM conference on Computer and communications security*, 236–245, New York, NY, USA. ACM.

[45] Wiedenbeck, S., Waters, J., Sobrado, L., & Birget, J.-C. 2006. Design and evaluation of a shoulder-surfing resistant graphical password scheme. In *AVI '06: Proceedings of the working conference on Advanced visual interfaces*, 177–184, New York, NY, USA. ACM.

[46] Tan, D. S., Keyani, P., & Czerwinski, M. 2005. Spy-resistant keyboard: more secure password entry on public touch screen displays. In *OZCHI '05: Proceedings of the 17th Australia conference on Computer-Human Interaction*, 1–10, Narrabundah, Australia. Computer-Human Interaction Special Interest Group (CHISIG) of Australia.

[47] Brostoff, S. & Sasse, M. Jan 2000. Are Passfaces more usable than passwords: A field trial investigation. *HCI 2000: Proceedings of People and Computers XIV - Usability or Else!*, 405–424.

[48] Dhamija, R. & Perrig, A. 2000. Déjà Vu: a user study using images for authentication. In *SSYM'00: Proceedings of the 9th conference on USENIX Security Symposium*, 4–4, Berkeley, CA, USA. USENIX Association.

[49] Catuogno, L. & Galdi, C. 2008. A Graphical PIN Authentication Mechanism with Applications to Smart Cards and Low-Cost Devices. In *WISTP*, Onieva, J. A., Sauveron, D., Chaumette, S., Gollmann, D., & Markantonakis, C., eds, volume 5019 of *Lecture Notes in Computer Science*, 16–35. Springer.

[50] Komanduri, S. & Hutchings, D. R. 2008. Order and entropy in picture passwords. In *GI '08: Proceedings of graphics interface 2008*, 115–122, Toronto, Ont., Canada, Canada. Canadian Information Processing Society.

[51] Perković, T. 2010. Shoulder Surfing Safe Login in a Partially Observable Attacker Model. *The 14th International Conference on Financial Cryptography and Data Security (Financial Cryptography 2010 - FC10)*.

[52] Toni Perković, Mario Čagalj, N. R. September 2009. SSSL: Shoulder Surfing Safe Login. *Proceedings of the SoftCOM 2009 (International Conference on Software, Telecommunication and Computer Networks), co-sponsored by the IEEE Computer Society (IEEE-CS)*.

[53] De Luca, A., Von Zezschwitz, E., & Hu, H. 2009. Vibrapass: secure authentication based on shared lies. In *CHI '09: Proceedings of the 27th international conference on Human factors in computing systems*, 913–916, New York, NY, USA. ACM.

[54] De Luca, A., Weiss, R., & Hussmann, H. 2007. PassShape: stroke based shape passwords. In *OZCHI '07: Proceedings of the 19th Australasian conference on Computer-Human Interaction*, 239–240, New York, NY, USA. ACM.

[55] Kratz, S. & Ballagas, R. 2009. Unravelling seams: Improving mobile gesture recognition with visual feedback techniques. In *CHI '09: Proceedings of the 27th international conference on Human factors in computing systems*, 937–940, New York, NY, USA. ACM.

[56] Nali, D. & Oorschot, P. C. 2008. CROO: A Universal Infrastructure and Protocol to Detect Identity Fraud. In *ESORICS '08: Proceedings of the 13th European Symposium on Research in Computer Security*, volume 5283 LNCS, 130–145, Berlin, Heidelberg. Springer-Verlag.

[57] De Luca, A., Denzel, M., & Hussmann, H. 2009. Look into my eyes! Can you guess my password? In *SOUPS '09: Proceedings of the 5th Symposium on Usable Privacy and Security*, New York, NY, USA. ACM.

[58] Kumar, M., Garfinkel, T., Boneh, D., & Winograd, T. 2007. Reducing shoulder-surfing by using gaze-based password entry. In *SOUPS '07: Proceedings of the 3rd symposium on Usable privacy and security*, volume 229, 13–19, New York, NY, USA. ACM.

[59] Thorpe, J., Van Oorschot, P. C., & Somayaji, A. 2006. Pass-thoughts: Authenticating with our minds. In *NSPW '05: Proceedings of the 2005 workshop on New security paradigms*, 45–56, New York, NY, USA. ACM.

[60] Salvador, S. & Chan, P. 2007. Toward accurate dynamic time warping in linear time and space. *Intell. Data Anal.*, 11(5), 561–580.

[61] Apple. April 1997. arc4random pseudo-random number generator. MAC OS X Reference Library.

[62] Sony. 2000. DSR-PD150 Professional Digital Camcorder - Device Brochure.

A Signal graphs

A.1 Right and left flip

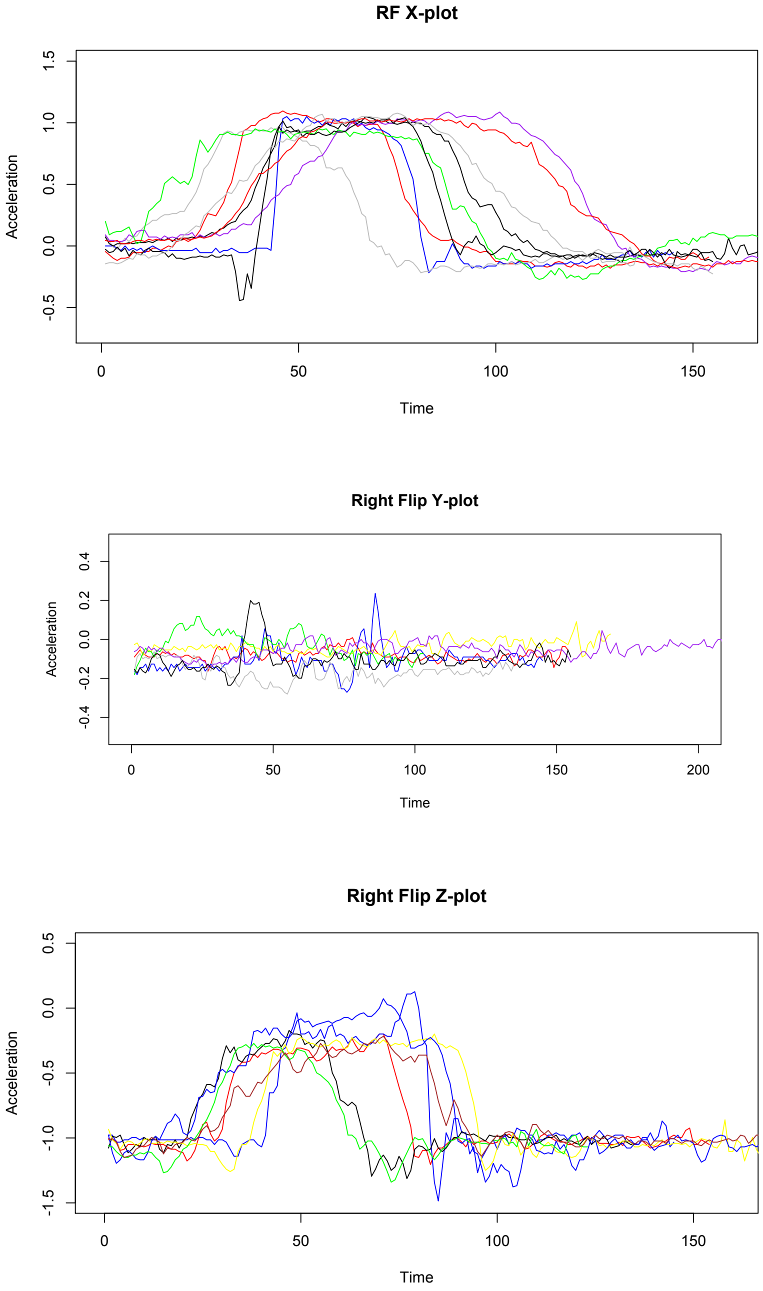


Figure 20: Shows a number of different *right flip* templates plotted against each other. Each color represents a unique persons gesture execution.

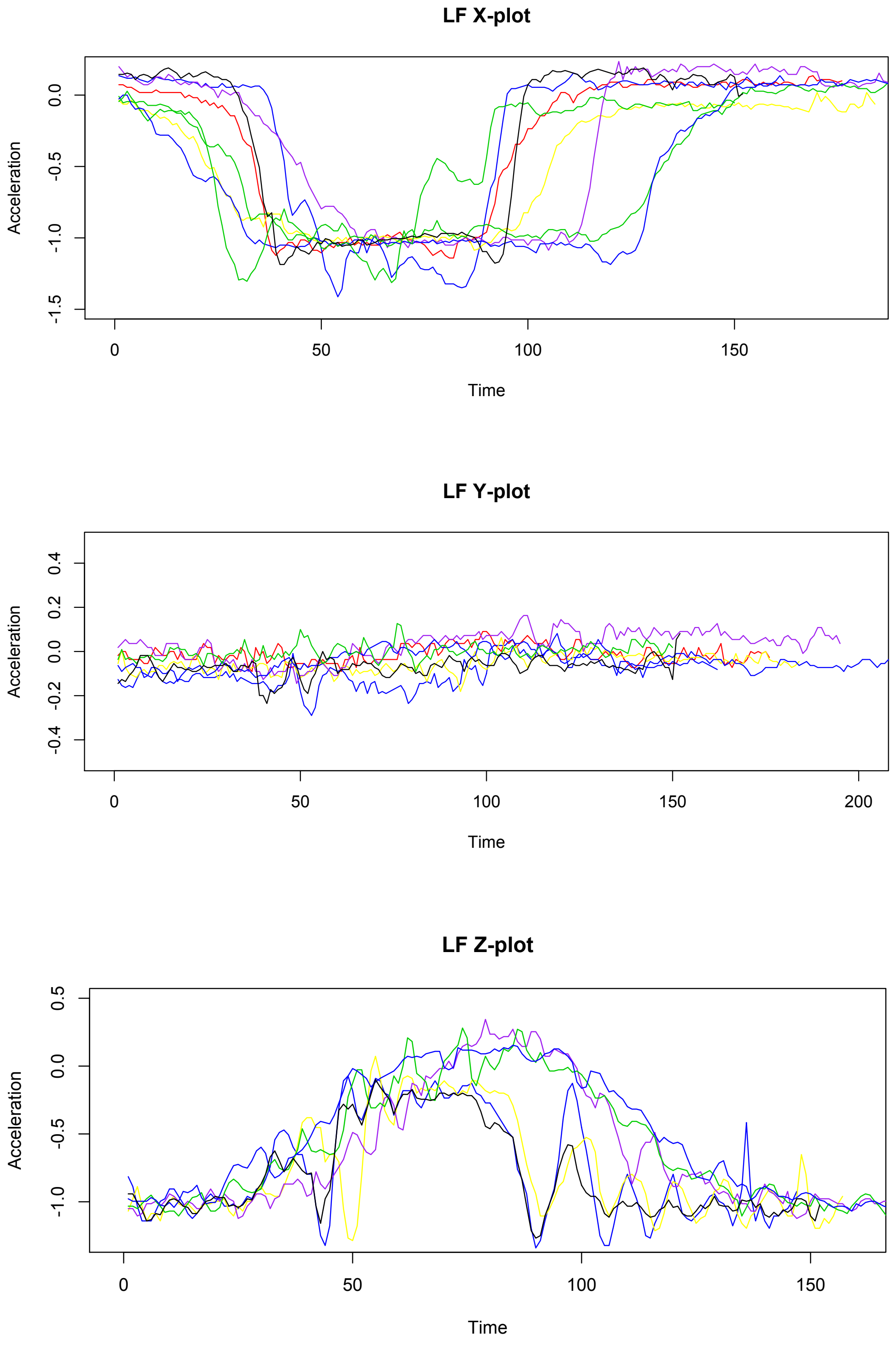
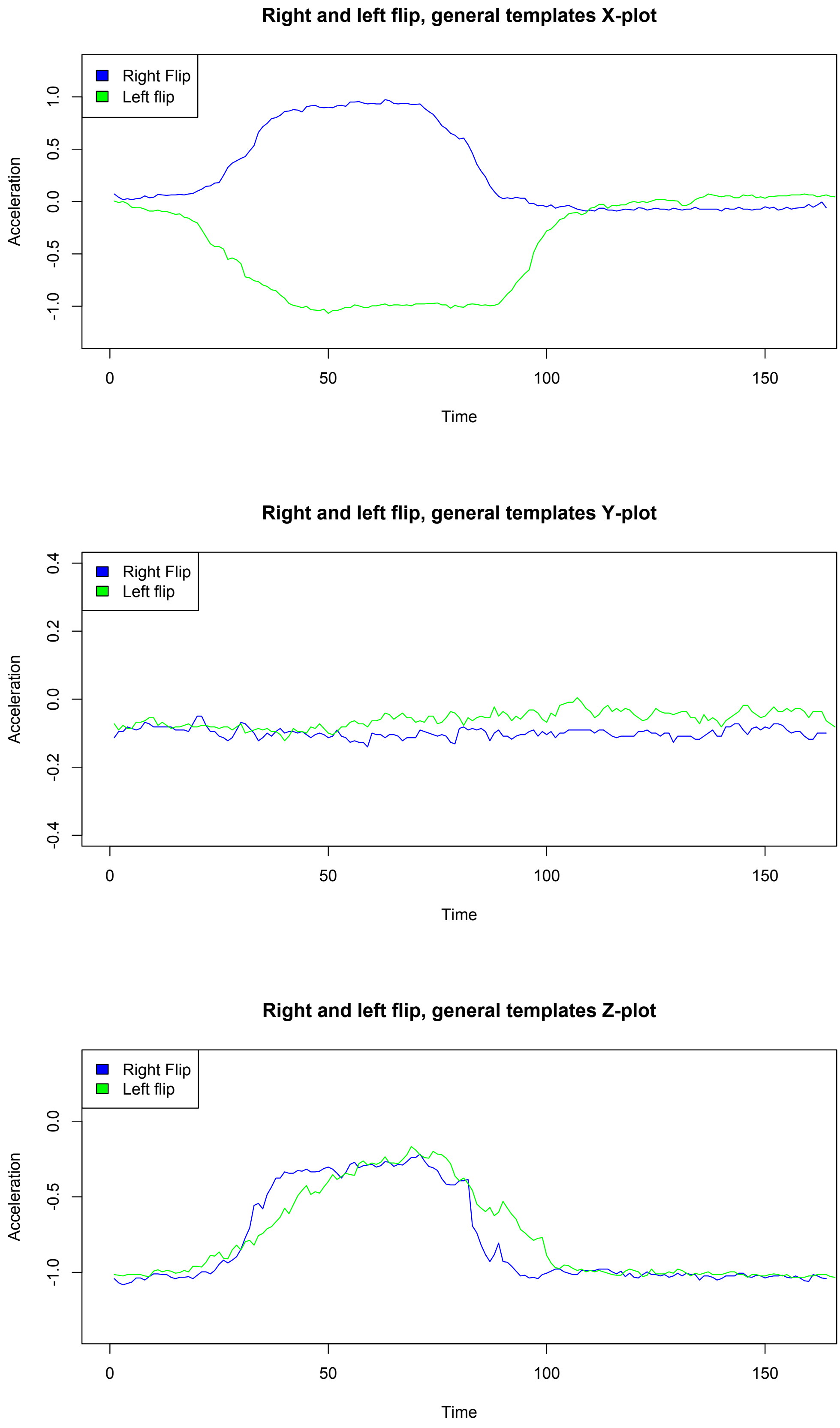


Figure 21: Shows a number of different *left flip* templates plotted against each other. Each color represents a unique persons gesture execution.



Right and left flip, general templates X-plot

Right and left flip, general templates Y-plot

Right and left flip, general templates Z-plot

Figure 22: Shows the general templates for a *left* and *right flip* plotted against each other.

A.2 Front and back flip

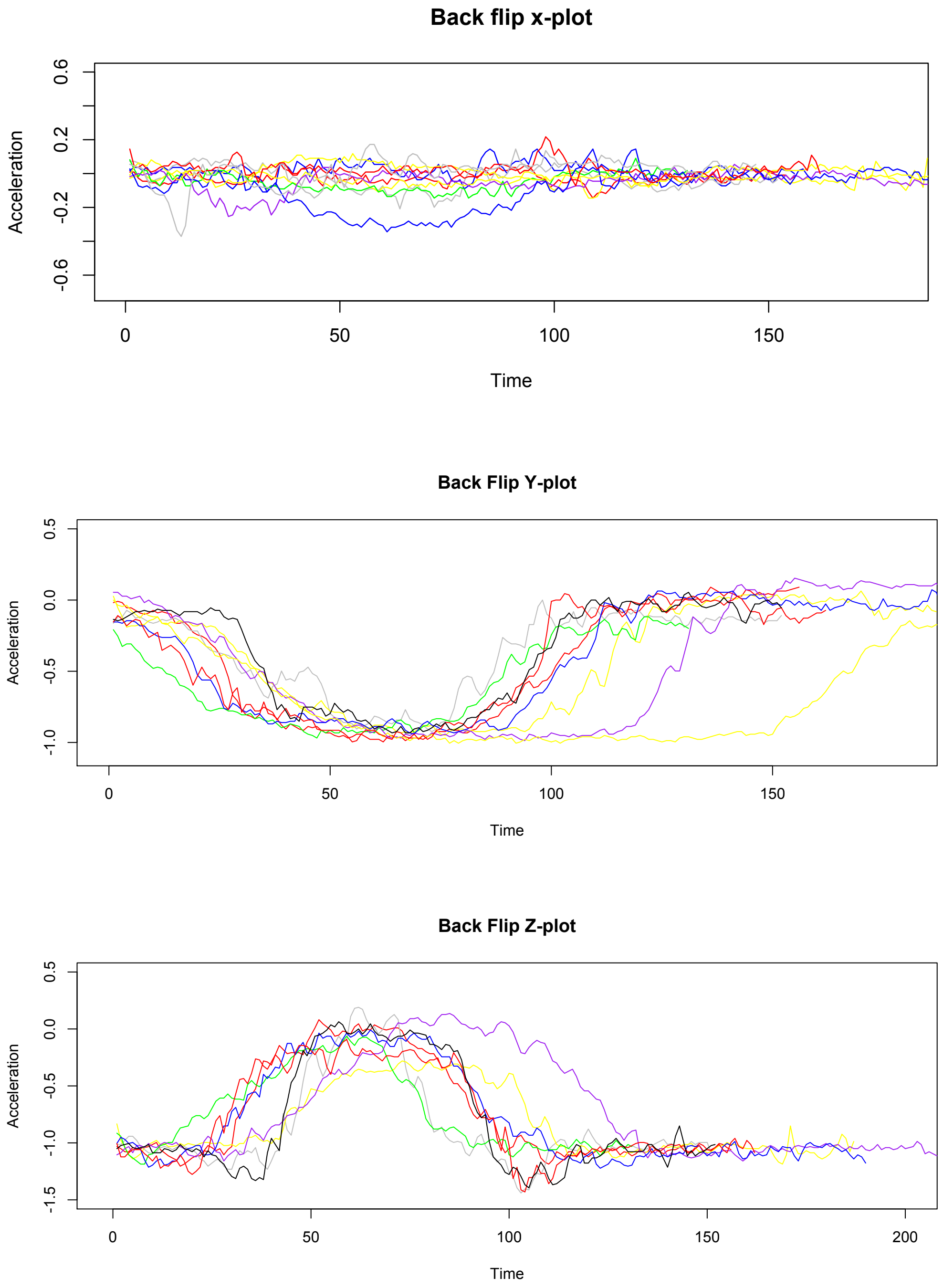


Figure 23: Shows a number of different *back flip* templates plotted against each other. Each color represents a unique persons gesture execution.

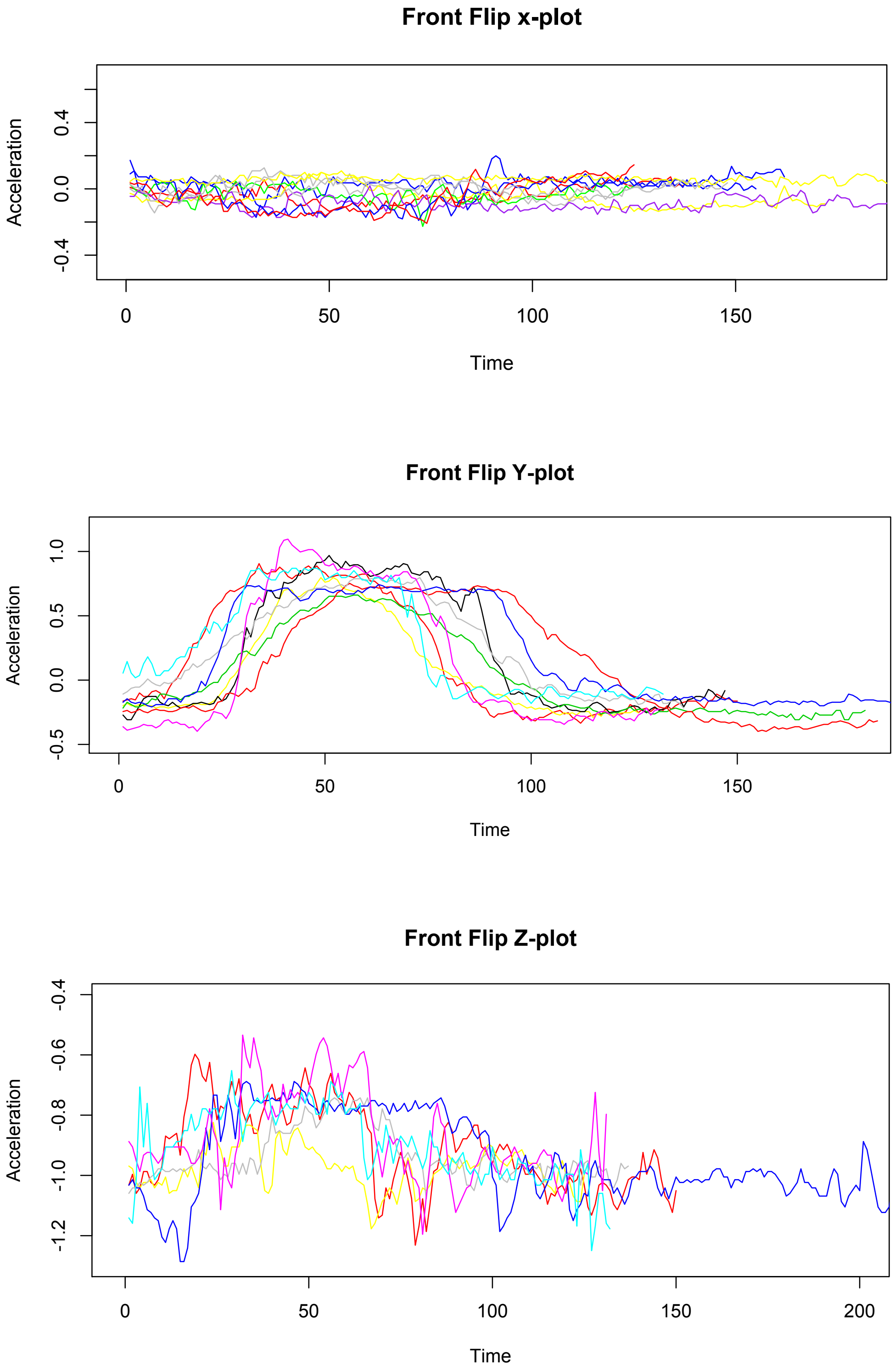


Figure 24: Shows a number of different *front flip* templates plotted against each other. Each color represents a unique persons gesture execution.

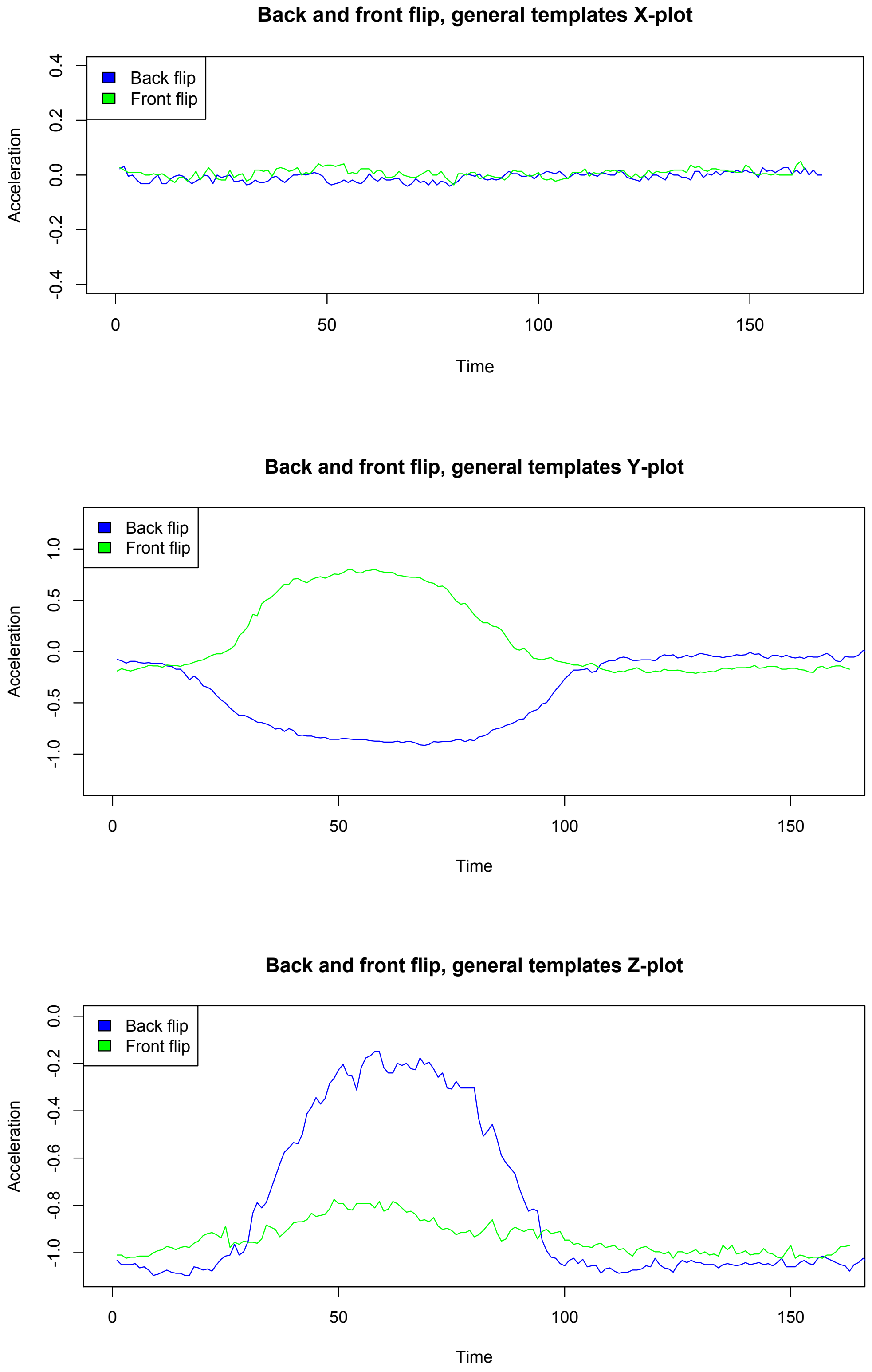


Figure 25: Shows the general templates for a *front* and *back flip* plotted against each other.

A.3 Circular motions

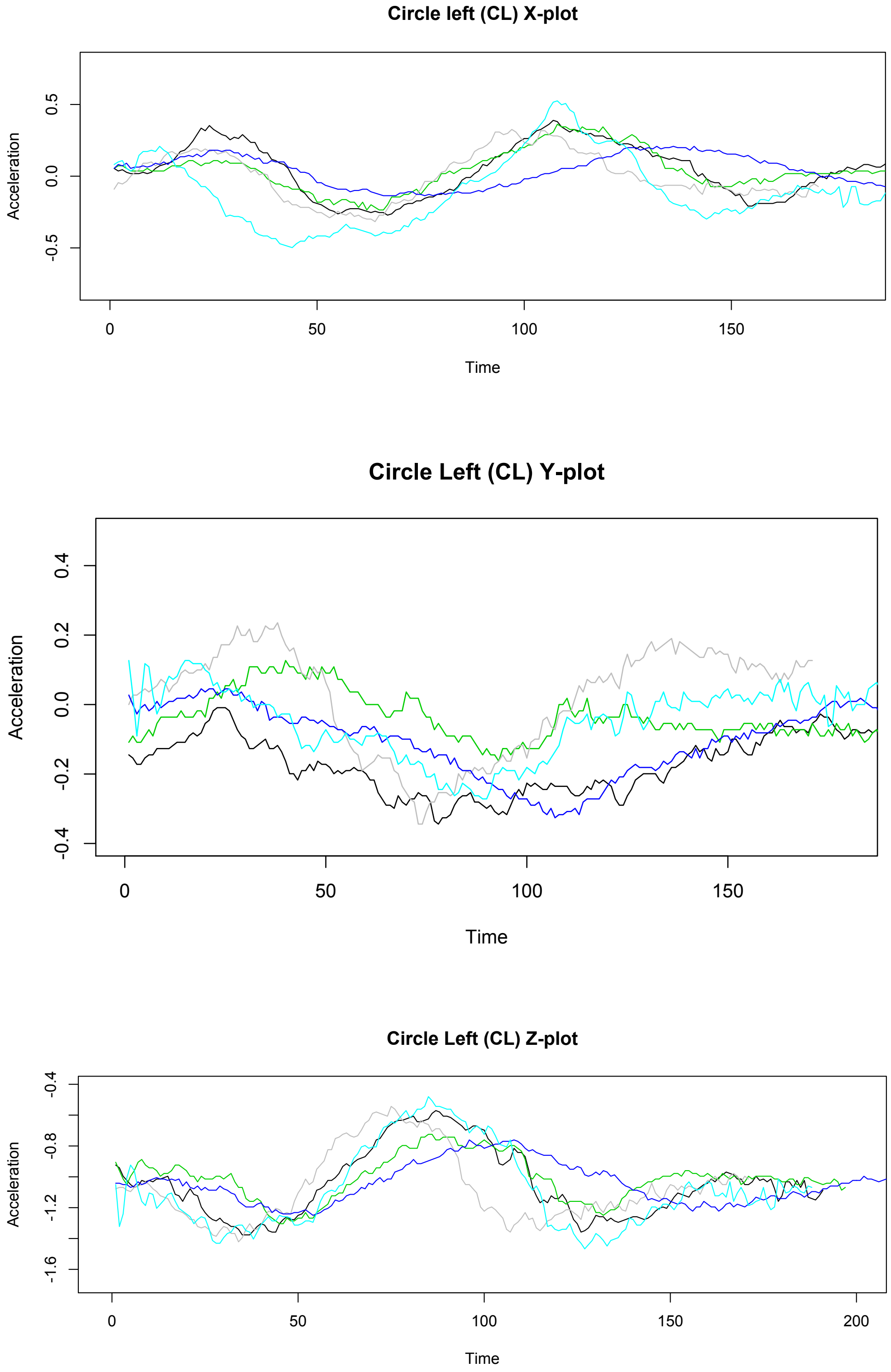
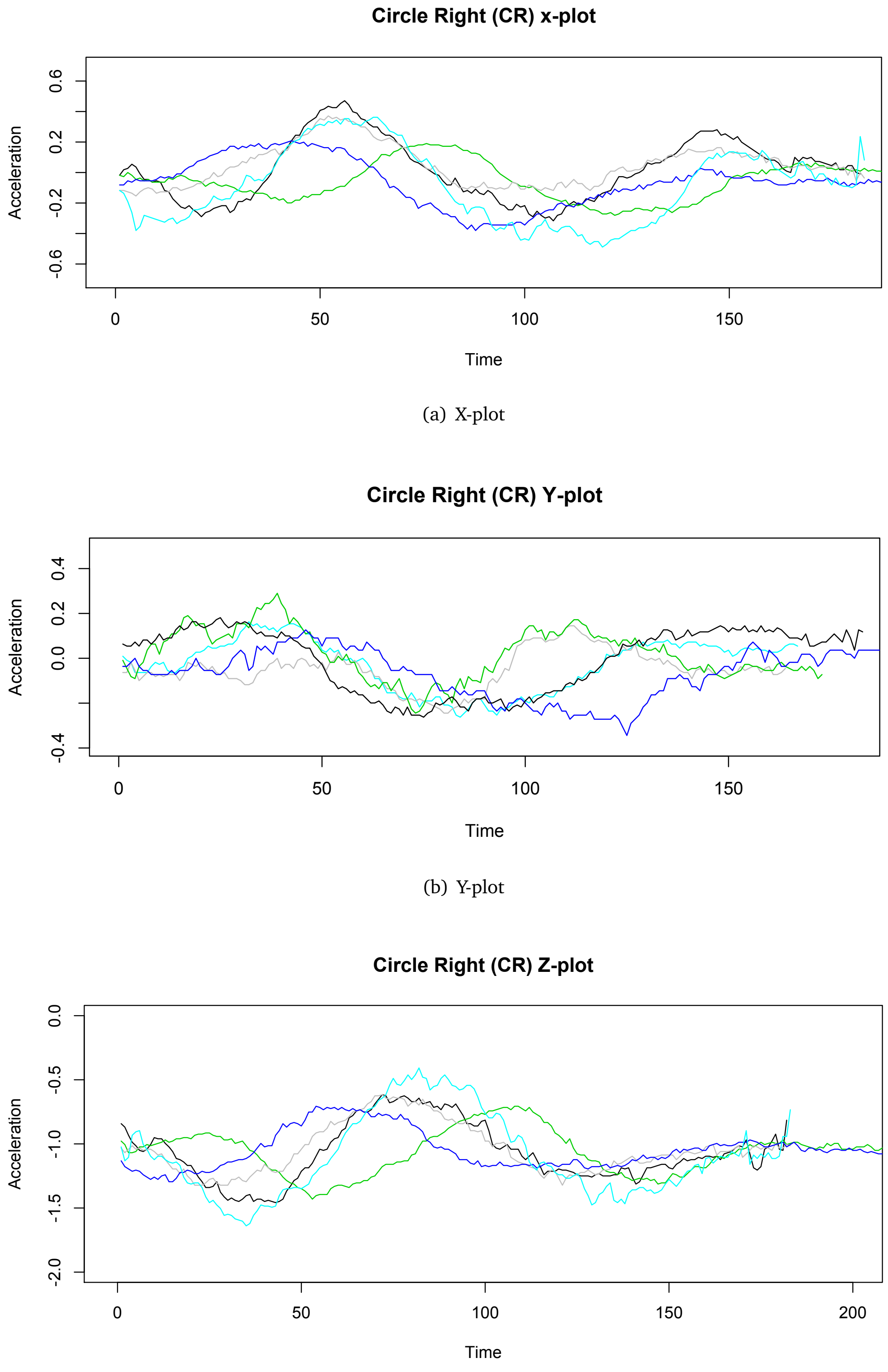


Figure 26: Shows a number of different *left circle* templates plotted against each other. Each color represents a unique persons gesture execution.



Circle Right (CR) x-plot

(a) X-plot

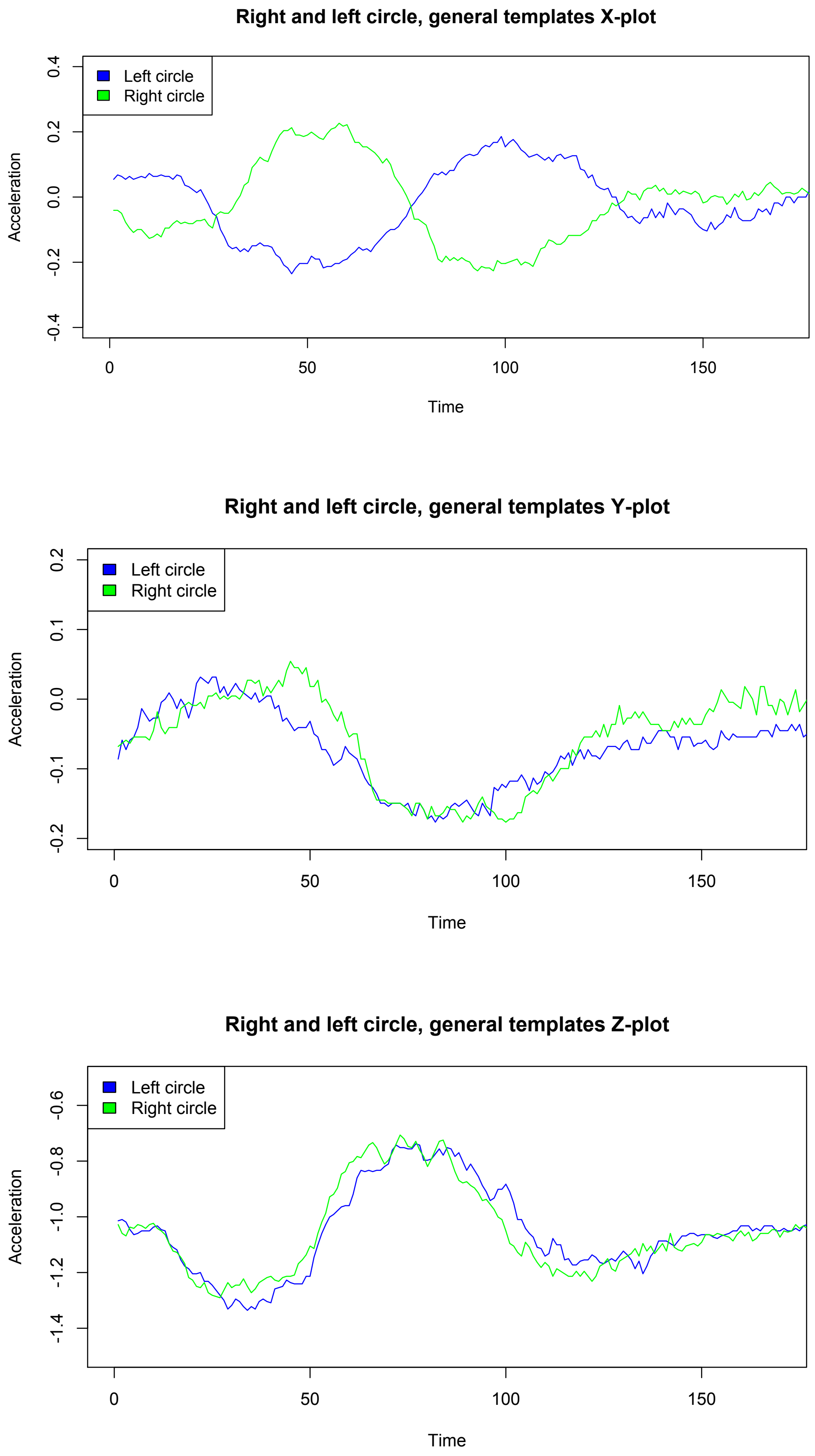
Circle Right (CR) Y-plot

(b) Y-plot

Circle Right (CR) Z-plot

(c) Z-plot

Figure 27: Shows a number of different *right circle* templates plotted against each other. Each color represents a unique persons gesture execution.



Right and left circle, general templates X-plot

Right and left circle, general templates Y-plot

Right and left circle, general templates Z-plot

Figure 28: Shows the general templates for a *left* and *right circle* plotted against each other.

B Flow charts for our authentication schemes

B.1 Gesture and PIN-code based authentication scheme

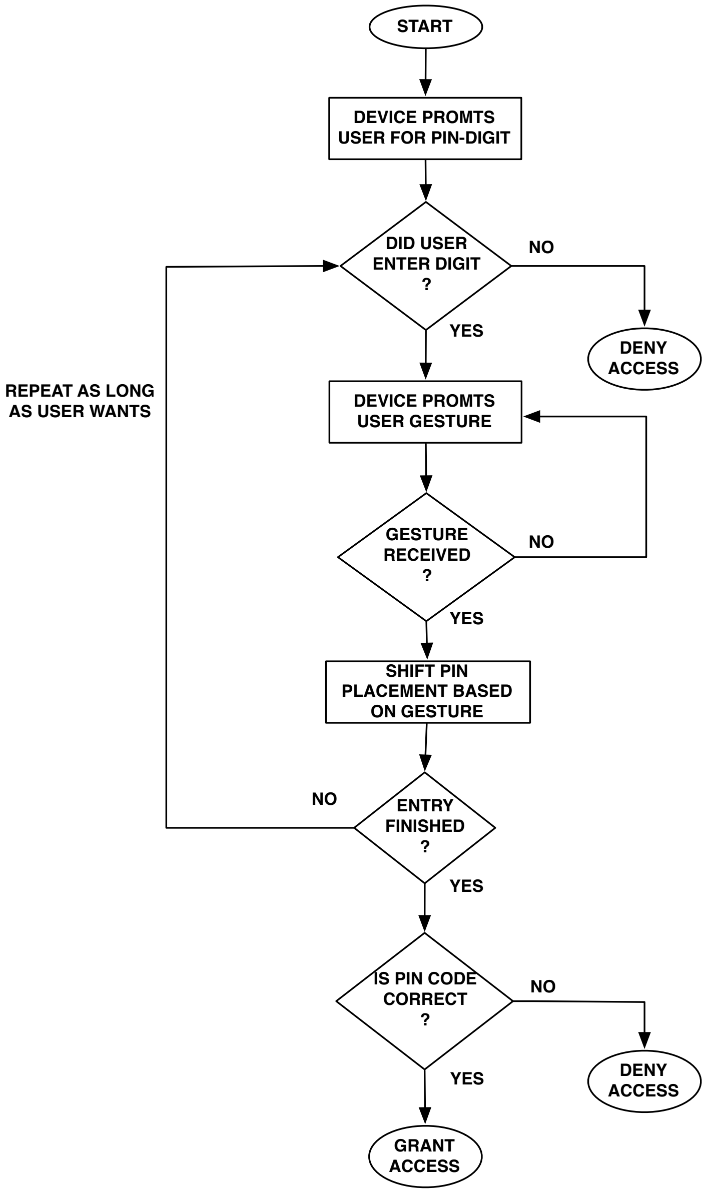


Figure 29: Shows our authentication scheme where the user uses gestures to place PIN-digits into the correct order.

B.2 Challenge-response scheme

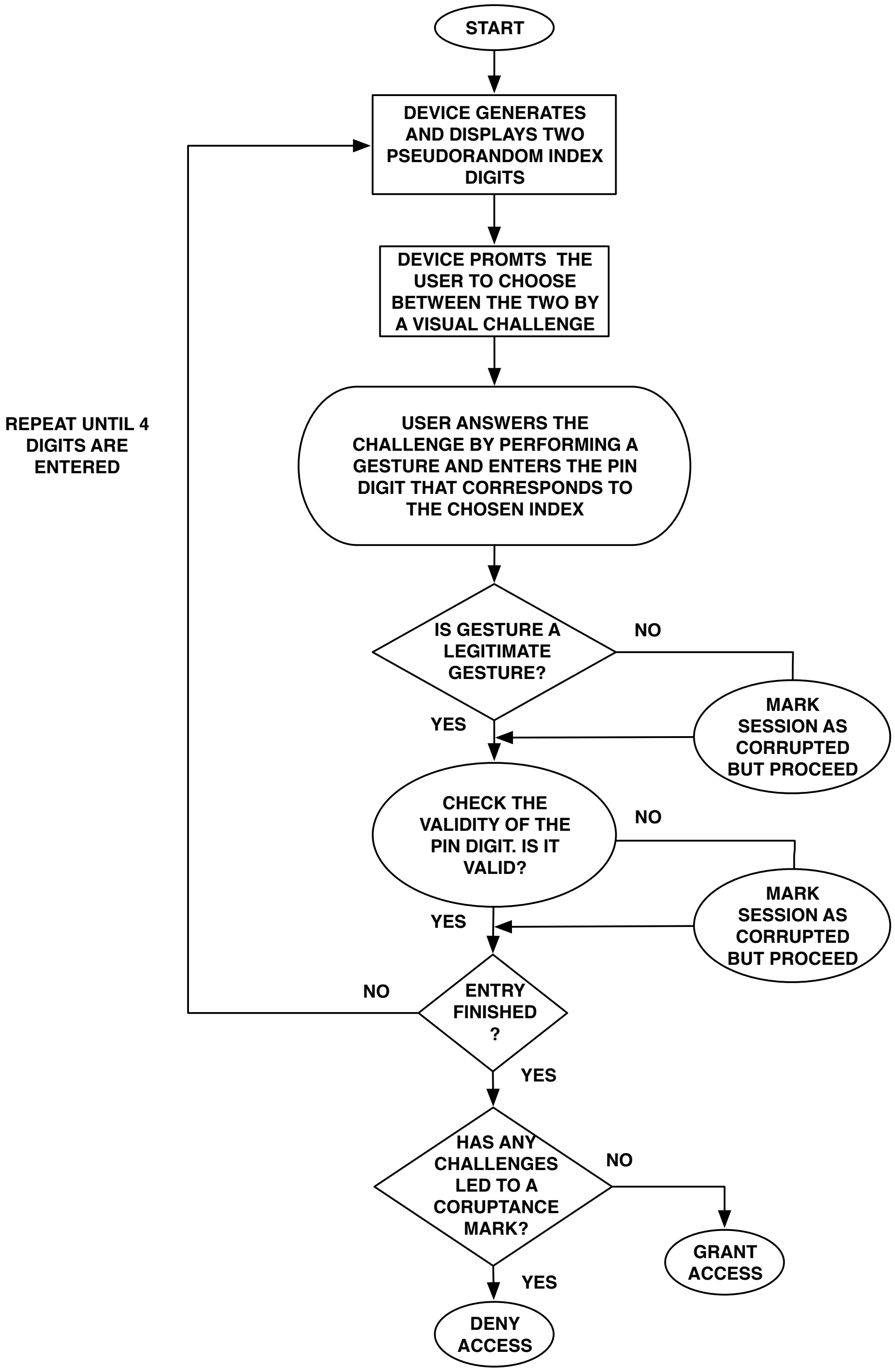

START
DEVICE GENERATES AND DISPLAYS TWO PSEUDORANDOM INDEX DIGITS
DEVICE PROMTS THE USER TO CHOOSE BETWEEN THE TWO BY A VISUAL CHALLENGE
REPEAT UNTIL 4 DIGITS ARE ENTERED
USER ANSWERS THE CHALLENGE BY PERFORMING A GESTURE AND ENTERS THE PIN DIGIT THAT CORRESPONDS TO THE CHOSEN INDEX
IS GESTURE A LEGITIMATE GESTURE?
NO
MARK SESSION AS CORRUPTED BUT PROCEED
YES
CHECK THE VALIDITY OF THE PIN DIGIT. IS IT VALID?
NO
MARK SESSION AS CORRUPTED BUT PROCEED
YES
NO
ENTRY FINISHED ?
YES
HAS ANY CHALLENGES LED TO A CORUPTANCE MARK?
NO
GRANT ACCESS
YES
DENY ACCESS

Figure 30: Shows the basic flow our second challenge-response authentication scheme.

C Participant agreement declaration

Participation in hand gesture based authentication scheme experiment

I am voluntarily participating in this experiment. The information about my hand gestures are gathered to fulfill the purpose of this thesis. The person responsible for the data processing is Aleksander Furnes Mallasvik, who is doing this experiment as a part of his master thesis. He will take care that the recorded data is solely used for research purposes.

With my signature I confirm the following:

1. I have been informed in oral and written form about the content and purpose of the collected data that is in relation to my person.
2. My data will only be used to serve this purpose. The detailed description of the purpose is documented on the back side of this sheet.
3. I allow that information about my hand gestures are recorded.
4. The data will not be displayed in possible future publications on this experiment.
5. I have been informed that I can reject to sign the agreement.
6. I have been informed that I can request to receive insight in the collected data before such data is used for teaching and research purposes.
7. I know that I can withdraw my participation anytime I want without giving any explanation and all data collected from we will be deleted permanently.
8. I accept that the data can be used for future research at HiG. All data will be deleted, the link between the data and my name will be destroyed when it is no longer necessary to maintain it. This will happen as the research experiment has been completed.

First name - family name ________________________

Gjøvik, date ________________________ Signature ________________________

D Experiment protocol description - Gesture-Placement based scheme

This appendix gives a more detailed overview on how we conducted the experiments outlined in Section 7.1. It also contains the protocol description which was given to our participants prior to the experiment D.2. As described in Section 7.1, the participants acts as attackers in the the following attack scenarios. To prove whether or not an attacker had successfully decoded the PIN-code, he orally informed the experimenter about what he believed was the PIN-code after each protocol run. Each participant participated in two attack scenarios; one where we overwrote digits, and one where did not.

D.1 Detailed information about the experiment protocol

Simple attack scenario

- PIN: 3968
- Login procedure;
  1. User enters 8, performs backflip
  2. User enters 9, performs left flip
  3. User enters 6 perform right flip
  4. User enters 3 perform front flip

The procedure above was repeated twice for each participant. We asked the user about what the PIN-code is after each run. Should the participant decode the PIN after the first run, we will use another PIN for the next run. This gives us the possibility to withdraw a number of different statistical properties, as can be seen in Section 8.1.

Random attack scenario

- PIN: 4148
- In this experiment, the user (the victim) can employ a different login sequence each time. The only restriction in this scenario, is that the victim has to employ at least one overwrite (decoy) gesture.
- The login sequence changes from time to time, meaning that the victim does not enter the same login sequence twice.

As in the simple attack scenario, The PIN will only change if it is deduced on the first run. In this case, the PIN will be changed, and we will increase the number of overwrites we apply.

D.2 Experiment description for the participants

In this experiment you will take on the role of an attacker. You will witness 4 separate login attempts, where your goal is to decode the PIN code by using the information provided to you in this document. You are granted full observation, meaning that you can watch the device from any angle you wish. The idea is that you should be able to see everything that is happening on the device, and also the gestures performed by the victim.

Unlike in a normal PIN-code authentication system, this scheme utilizes gestures to "place" the digits which are entered throughout the protocol. If we consider a PIN code as an array of four digits, then we have four different places where a PIN digit can be placed. In this scheme, we utilize four unique gestures to enforce this placement mechanism. Each gesture is statically linked to a position in the PIN as follows;

- Front flip - digit is placed at index 1.
- Left flip - digit is placed at index 2
- Right flip - digit is placed at index 3
- Back flip - digit is placed at index 4

A PIN entry in this scheme is therefore not just a digit, but also a gesture. This means that before you are finished entering a PIN digit, you will both have to enter the digit and perform the placement gesture. As an example consider the case where an user has the following correspondence with the authentication scheme;

1. User enters digit 1, performs a backflip
2. User enters digit 3, performs a left flip
3. User enters digit 5, performs a right flip
4. User enters digit 4, performs a front flip.

By utilizing our knowledge about the statical placement-gestures, we can decode that the users PIN-code is; 4351. What is important to notice is that the user can place the digits in any order, and he also has the possibility of overwriting a digit. This is done by performing the same gesture twice in one protocol run. In this case, the digit corresponding to the first execution of that gesture is replaced by the last digit corresponding to the same placement gesture.

To decode the PIN-code, one therefore has to remember the PIN digit, and the corresponding placement gesture.