Examensarbete

Device Sensor Fingerprinting
Mobile Device Sensor Fingerprinting With A Biometric Approach

Examensarbete utfört i säkra system
vid Tekniska högskolan vid Linköpings universitet
av

Anna Karlsson

LiTH-ISY-EX--15/4838--SE
Linköping 2015
Device Sensor Fingerprinting
Mobile Device Sensor Fingerprinting With A Biometric Approach

Examensarbete utfört i säkra system
vid Tekniska högskolan vid Linköpings universitet
av

Anna Karlsson

LiTH-ISY-EX--15/4838--SE

Handledare: Jonathan Jogenfors, PhD student
ISY, Linköping university
Engineer Philip Engström
Cybercom AB

Examinator: Jan-Åke Larsson, Ph.D
ISY, Linköping university

Linköping, 10 juni 2015
Fingeravtryck i Mobila Enheter

Device Sensor Fingerprinting

Anna Karlsson

Fingeravtryck i Mobila Enheter

Device Sensor Fingerprinting

Anna Karlsson

The number of connected devices connected to the Internet is growing rapidly. When talking about devices it also covers the ones not having any contact with humans. This type of devices are the ones that are expected to increase the most. That is why the field of device fingerprinting is an area that requires further investigation. This thesis measures and evaluates the accelerometer, camera and gyroscope sensor of a mobile device to the use as device fingerprinting. The method used is based on previous research in sensor identification together with methods used for designing a biometric system. The combination with long-proven methods in the biometric area with new research of sensor identification is a new approach of looking at device fingerprinting.
Sammanfattning

Abstract

The number of connected devices connected to the Internet is growing rapidly. When talking about devices it also covers the ones not having any contact with humans. This type of devices are the ones that are expected to increase the most. That is why the field of device fingerprinting is an area that requires further investigation. This thesis measures and evaluates the accelerometer, camera and gyroscope sensor of a mobile device to the use as device fingerprinting. The method used is based on previous research in sensor identification together with methods used for designing a biometric system. The combination with long-proven methods in the biometric area with new research of sensor identification is a new approach of looking at device fingerprinting.
Acknowledgments

I would like to thank everyone that sending in the sensor data to my thesis, without you it would not be much to write about. Also many thanks to all employees at Cybercom, it has been fun working with the thesis when you were showing so much interest. Especially thanks to Dan Rosenqvist for all the good thoughts, ideas and that I had the opportunity to work with this interesting topic.

Other persons that have made my education to alot more fun than I expected is my classmates, 720 13-14 and LiU AIF IBK, love to all of you.

The biggest acknowledgements is of course to my parents who have always been there for me, without them I would never have made it through the education. Then of course my brother Johan who is always with me.

Linköping, June 2015
Anna Karlsson
Contents

List of Figures xi
List of Tables xiii
Notation xv

1 INTRODUCTION 1
1.1 Background . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 1
1.2 Aims & Objectives . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 2
1.3 Thesis Outline . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 3

2 COMMUNICATION & AUTHENTICATION 5
2.1 Two factor authentication . . . . . . . . . . . . . . . . . . . . . . . . . . . 5
2.2 Challenge-Response authentication . . . . . . . . . . . . . . . . . . . . 6
2.3 M2M - Machine to machine . . . . . . . . . . . . . . . . . . . . . . . . . . . 7
  2.3.1 Difference between M2M and IoT . . . . . . . . . . . . . . . . . . . . . 8
  2.3.2 M2M authentication . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 8
2.4 The biometric process . . . . . . . . . . . . . . . . . . . . . . . . . . . . 9
  2.4.1 Recognition . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 9
  2.4.2 Biometric systems . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 9
  2.4.3 Biometric authentication . . . . . . . . . . . . . . . . . . . . . . . . . . 10
  2.4.4 Biometric measurements . . . . . . . . . . . . . . . . . . . . . . . . . . 10
  2.4.5 The design of a biometric system . . . . . . . . . . . . . . . . . . . . . 10

3 CHARACTERISTICS OF A MOBILE DEVICE 13
3.1 Accelerometer . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 14
  3.1.1 Fingerprinting characteristics . . . . . . . . . . . . . . . . . . . . . . . 14
3.2 Gyroscope . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 14
  3.2.1 Fingerprinting characteristics . . . . . . . . . . . . . . . . . . . . . . . 14
3.3 Camera . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 15
  3.3.1 Fingerprinting characteristics . . . . . . . . . . . . . . . . . . . . . . . 15
3.4 Allan variance . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 15
3.5 Previous work of device sensor fingerprinting . . . . . . . . . . . . . . . . . . 16
List of Figures

2.1 Challenge-response authentication with bank card reader .......... 7
2.2 The design cycle of a biometric system ............................ 12

3.1 The pyramid of features in a mobile device that can be used for fingerprinting.[9] ................................................. 13

4.1 The coordinate system used in JavaScript[11] .......................... 20
4.2 The device rotation axes for the JavaScript DeviceOrientation ...... 21
4.3 Screen-shots of web-page during accelerometer measurements in test I ................................................................. 22
4.4 Motion sensor measurements II on a Google Nexus 7 ............... 23
4.5 Sensor measurements on a Google Nexus 7 ........................... 24
4.6 the MATLAB medfilt2 outputs the median of each pixel by its 3-by-3 neighbors ...................................................... 24

5.1 Scatter-plot on accelerometer recordings of 6 Apple devices ...... 27
5.2 Diversity of device brand sampled in measurements I ............... 28
5.3 Most common devices models in measurements I .................... 28
5.4 Bias from twelve Sony Xperia devices measured with JavaScripts acceleration ............................................................. 29
5.5 Bias from twelve Sony Xperia devices measured with JavaScripts accelerationIncludingGravity .............................................. 29
5.6 Accelerometer readings of x-axes on a Sony Xperia Z1 Compact and a Google Nexus 7 over 50 days ............................... 30
5.7 Accelerometer readings of y-axes on a Sony Xperia Z1 Compact and a Google Nexus 7 over 50 days ............................... 31
5.8 Accelerometer readings of z-axes on a Sony Xperia Z1 Compact and a Google Nexus 7 over 50 days ............................... 31
5.9 Scatter-plot of accelerometer readings Sony Xperia-device, one of them with measurements performed on the same device over 50 days ...................................................... 32
5.10 Calculations of statistical accelerometer features. From [10, p.6] .. 32

A.1 Scatter-plot of mean values of 12 Sony Xperia Z-devices including one device with readings over a period of 50 days .............. 51
A.2 Scatter-plot of standard deviation values of 12 Sony Xperia Z-devices including one device with readings over a period of 50 days . . . . . . . 52
A.3 Scatter-plot of average deviation values of 12 Sony Xperia Z-devices including one device with readings over a period of 50 days . . . . . . . 52
A.4 Scatter-plot of skewness value of 12 Sony Xperia Z-devices including one device with readings over a period of 50 days . . . . . . . 53
A.5 Scatter-plot of kurtosis values of 12 Sony Xperia Z-devices including one device with readings over a period of 50 days . . . . . . . 53
A.6 Scatter-plot of RMS values of 12 Sony Xperia Z-devices including one device with readings over a period of 50 days . . . . . . . 54
A.7 Scatter-plot of min values of 12 Sony Xperia Z-devices including one device with readings over a period of 50 days . . . . . . . 54
A.8 Scatter-plot of max value of 12 Sony Xperia Z-devices including one device with readings over a period of 50 days . . . . . . . 55
List of Tables

3.1 Comparing studies of accelerometer fingerprinting .... 16
3.2 Comparing studies of camera fingerprinting ........... 17

5.1 Comparing distance between values of statistical features for the accelerometer. ................................. 33
5.2 Comparing distance between values of statistical features for the gyroscope. ........................................... 34
5.3 The Allan variance differences between measurements of all devices and same devices (Z1Comp & Nexus7) . . . . . . 35
5.4 The FAR and FRR of the MATLAB simulation when changing threshold values th1 and th2, the code can be found in appendix B . . . 36
5.5 FRR and time taken to compare PRNU of camera images. 38

7.1 Conclusions of the factors of choosing fingerprint sensor. 47
# Notation

## Notation

<table>
<thead>
<tr>
<th>Notation</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>$G$</td>
<td>G-force</td>
</tr>
<tr>
<td>$\epsilon$</td>
<td>Bias</td>
</tr>
<tr>
<td>$F_C$</td>
<td>Coriolis force</td>
</tr>
</tbody>
</table>

## Abbreviations

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>FAR</td>
<td>False acceptance rate</td>
</tr>
<tr>
<td>FRR</td>
<td>False rejection rate</td>
</tr>
<tr>
<td>FTE</td>
<td>Failure to enroll</td>
</tr>
<tr>
<td>IoT</td>
<td>Internet of Things</td>
</tr>
<tr>
<td>M2M</td>
<td>Machine-to-machine</td>
</tr>
<tr>
<td>MEMS</td>
<td>Micro-electromechanical System</td>
</tr>
<tr>
<td>OS</td>
<td>Operating System</td>
</tr>
<tr>
<td>PRNU</td>
<td>Photo-Response Non-Uniformity noise</td>
</tr>
<tr>
<td>RMS</td>
<td>Root Mean Square</td>
</tr>
<tr>
<td>SVM</td>
<td>Support Vector Machine</td>
</tr>
</tbody>
</table>
This paper is the report for my master thesis in Computer Science and the last part of my education of becoming an engineer in information-technology in the field of secure systems. This thesis was performed at Cybercom AB in Linköping. This chapter of introduction will give an overview of the work together with background and aims and objectives that is used as the basis for the work presented in this thesis.

1.1 Background

Cars, locks, birds, stoves, refrigerators, coffee makers, watches, cat feeders, sewing machines… The world of connected devices is growing rapidly. This world is known under the term ‘Internet of Things’. To make these things connect to each other secure authentication methods is needed. To be sure that the device are connecting to the device it is suppose to and not anything or anyone else.

Two-factor authentication is something we humans use basically every day when accessing buildings, part of networks, our bank and so on. When talking about two factor authentication we usually use a combination of either three things; something you know like a password, something you have like a passport or something you are like your fingerprint. (More about those in chapter 2.)

Something you know or have are things that can be copied, stolen or modified fairly easy and without knowing all that much about the person or thing you try to authenticate as. This compared to something you are requires much more effort and time since you only can focus at one person a time. Machines do not have those attributes as us human, they are build upon hardware parts.
The aim of this thesis is to explore the possibility of a machine to have a fingerprint that can be used to more securely authenticate them. This can be applied in several areas. An example is the new smart homes where fridges, stoves, coffee makers and doors shall communicate with each other. Another example could be when you want to limit the access to your bank account to your phone only to avoid that a malicious user accessing your account.

1.2 Aims & Objectives

Today most of the solutions for machine-to-machine (M2M) authentication involves a certificate, token, UUID etc. This is something the machine knows or has. The area of device fingerprinting is more investigated in line with the world of connected devices, which is called IoT (Internet-of-things) is growing. The aim of this thesis is to look into if the fingerprinting methods found today can be used as something the machine are for two factor authentication between them. The problems this thesis aims to solve are:

- Is it possible to create a device fingerprint by using the sensor characteristics in a mobile device?

- Could the device fingerprint be used as a second factor to identify the device?

The problems above state a mobile device and not a general machine, which is one of the limitations in the thesis. The focus is also identification as a biometric process where you are able to collect a set of data from the device in a database in an enrollment phase. This means that new devices in the system first have to be checked by collecting sensor data from your device, just like the police has to collect fingerprint from the suspect to compare with the fingerprints from the crime scene. As written in the background the devices building stone are hardware, thus something the devices are that is the point of view of the thesis. This is similar to biometric authentication of us humans.

The objectives of this work can be summed up to:

- Explore different sensor characteristics of a mobile device
  Mobile devices today are equipped with a lot of sensors. The sensors as hardware in general contains manufacturing defects that may cause bias. The bias that may be unique enough to differ from another device of the same model. Measurements from the gyroscope-, accelerometer- and camera-sensor will be collected and valuated like biometric fingerprints.

- Combining M2M, two factor and biometrics
  Biometric authentication has methods of identify fingerprints and designing such systems. These will be used to compare the characteristics of the sensors and evaluate the possibility of two factor authentication between the devices.
1.3 Thesis Outline

This chapter includes background, aims and objectives that gives a quick view of what the thesis is about. The chapters that follows are divided into different parts that map to the different objectives listed above.

Ch.2: Theory-chapter about how authentication is made today between machines, two factor, the challenge-response protocol and in biometrics.

Ch.3: Theory-chapter about the different hardware characteristics of a mobile device. Together with previous work in the area of the thesis.

Ch.4: The method used when doing measurements of the characteristics described in chapter 3.

Ch.5: Result of measurements.

Ch.6: Discussion about the result and method used. Followed by another discussion about the work in a wider context.

Ch.7: Conclusions that refers back to the aims and objectives and also includes further work in the area if the thesis.
Since about all devices that are connected to a network are one way or another
connected to the Internet you can bet that they find themselves in an untenanted
or malicious environment. Everything connected to the Internet is very likely to
be hacked. Thus, authentication is needed for remote sensing devices to commu-
nicate. [24]
This chapter presents ways of authentication (two factor, M2M and biometric).
The section about biometrics is included in the thesis because it has methods of
measuring strength of a biometric trait. These methods will be used when com-
paring strength of characteristic bias in the mobile device.

2.1 Two factor authentication

There are more ways to authenticate users than the use of passwords, however
it is the most common. The types of authentication is often divided into three
categories;

- Something the authenticator has like a tag, key, credit card or passport.
- Something the authenticator knows like a password.
- Something the authenticator is, biometrics such as fingerprint or iris pat-
tern

[2, p. 31]
Authentication in two factors means a combination of two of the three types of
authentication above. An example can be the use of a credit card (that you have)
in combination with a PIN-code (that you know) to collect the money from an
ATM. Something the authenticator has and knows is the most common combination. The biggest reason that biometrics is not that common yet is due to costs. [2, p. 47]

2.2 Challenge-Response authentication

The challenge-response protocol is built upon the idea that the user of a system first must complete a challenge decided by the system in order to access the system. An example is a modern car key when trying to start the engine, the engine controller gives the key a challenge consisting of a random $n$-bit number. The key encrypts the challenge and responds.

The problem challenge-response protocols faces is often to achieve good randomness, thus if the challenge is not random enough there is a risk for a malicious user to calculate the $n$-bit number.

There are other applications than locks, like the HTTP Digest Authentication. That uses the authentication process where a web server challenges a client or a proxy with the common secret of a password. The server send nonce to the client or proxy, that hash the nonce with the password and the requested URI. (Nonce is an arbitrary number that only can be used once, often generated as random or pseudo-random.) This authentication mechanism is not vulnerable to password snooping and is used in cases like client-server-authentication in SIP or the protocol for Voice-Over-IP telephony.

A more common use of challenge-response is in two-factor authentication. An example is if you have a bank card reader when accessing your bank on the Internet. When you want to log in there is a random set of $n$ numbers displayed in the screen. You put these numbers together with a PIN into your bank card reader. The reader encrypts these numbers (pin + $n$ numbers) using a secret key shared with the server of the bank. The first $n$ numbers of the encryption is displayed on the card reader and you enter this in the login screen as a password.
2.3 M2M - Machine to machine

Information that is exchanged via a communication network between machines has to establish conditions for doing so, that is where M2M is used. M2M is often a short synonym for M2M communication, meaning the communication conditions between devices. M2M communication is only the communication made between machines without any human behind it. A mobile phone interacting with a call center application is not M2M, because there is a human behind the mobile device calling. The reason for using mobile devices in this thesis, that is controlled by a human, is that they contain many sensors. These sensors can be found in other simpler devices where M2M communication can be applied. M2M often involves similar devices in a M2M area network interacting with an application. This makes it possible for devices to access public networks as well, via a gateway or router. An example is the heating system in smart homes. M2M

Figure 2.1: Challenge-response authentication with bank card reader

Description of figure 2.1:

1. Bank sends challenge XXXX XXX to the requesting address.

2. User enters PIN and XXX XXX in the bank card reader.

3. The reader encrypts the PIN and number with a secret key shared with the bank. The first numbers of the encryption are displayed on the reader. \((YYYYYYY = XXXXXXX, PIN_k)\)

4. The user enters the encrypted numbers YYYY YYY on the log in screen and sends it as a password to the bank.

[2, ch.3]
is important to make devices talk without a human behind. This affects the requirements on the applications and networks dealing with the devices. Characteristics of these devices are listed below:

- **Multitude** - The part of IoT that is not directly interacting with humans is the part growing the most. The part is soon expected to be significant more than the one which interacts direct with humans. This will put more pressure on application and networks dealing with all devices.

- **Variety** - The connected devices have requirements such as data exchange rate, form factor, computing and communication capabilities. M2M applications have to be built in order to define and develop common enabling capabilities.

- **Invisibility** - The device has virtually zero human control. The more invisible the lower the probability of errors caused by humans.

- **Criticality** - Devices that can harm people because of electrical failure and such. Therefore reliability is an important factor.

- **Intrusiveness** - Many of the increasing connected devices raise the privacy question like refrigerators, stoves, doors, etc.

All these devices with no human control are very different. But many of them have some characteristics such that the functionality is limited, low-powered, embedded and have long life cycles. The fact that they often are embedded makes it hard to separate machine-to-machine, machine-to-human and human-to-human communication. [7, p. 2-4]

### 2.3.1 Difference between M2M and IoT

The term Internet-of-things, means everything that is connected to the Internet. IoT is now in its starting pits and ready to explode. Machine-to-machine communication is a part of that, but it also covers other areas that IoT does not and vice versa. The common denominator is according to Polsonetti the remote device access. Where the embedded hardware modules in a machine that communicate wireless or not is M2M applications. Remote device access for IoT has a wider perspective that is not only including same device communication. But also communication between passive and other low-power sensors, that not can be motivated as a M2M hardware module. [23]

### 2.3.2 M2M authentication

There is no standardized way of authentication in M2M, but effort is done in the area. An example is authentication based on what a machines knows or have. This consist of a hardware message of a computer, such serial number of CPU, MAC address of network card, machine ID etc. [13]

These things have through the years been proven to be pretty easy to spoof. There are hundreds of blog-articles and forum topics of how to do that for many platforms of mobile devices.
2.4 The biometric process

“A biometric system measures one or more behavioral characteristics...information of an individual to determine or verify his identity.”

[14, p. 3]

2.4.1 Recognition

The person showing a biometric identifier (fingerprint, iris, DNA, etc.) to the biometric system, is seen as a user of the system. The strength of biometrics is also the fact that it knows if a user is known to the system even if the user denies it. [14, ch. 1]

2.4.2 Biometric systems

There are blocks for building a biometric systems which can measures characteristics of a user. In biometrics these characteristics are called traits, indicators, identifiers, or modalities. In thesis it will still be called characteristics.

The first step of biometric authentication is to collect biometric data and store it in a database together with the user’s identity. The recognition is then done by again collecting biometric data from the user and compare it to the database. This is the so called enrollment and recognition phase. The raw biometric data is often destroyed after the enrollment and the recognition is all about pattern matching. This matching is done in four steps;

1. Sensor - Collects the raw biometric samples, which can be an image, amplitude signal, online signature, odour or chemical-based.

2. Feature extractor - Makes the raw biometric samples comparable, which is most of the time done in three pre-process operations;
   - Quality assessment - Checks if the sample is good enough.
   - Segmentation - Removes background noise from sample.
   - Enhancement - Uses an algorithm to improve characteristic features of the sample.

3. Database - Contains the data from the enrollment phase together with some identity data (like name or ID). The database should have an access control mechanism for security reasons.

4. Matcher - The sample from the enrollment is compared with the sample in recognition, to see if it is a match or not. This is done by having a match score to decide how close the enrolled and recognition sample is. The score is calculated in different ways depending on the characteristics that is used.

[14, ch. 1]
2.4.3 Biometric authentication

Biometric authentication, is sometimes also called verification which answers the question Are you the one you say you are?. There is also biometric identification which answers Are you someone known to the system? The practical difference is that in authentication the user has to give the system some kind of information (username, passport, email etc.) of who they claim to be. For identification the user just gives the sample to the system, which then checks if the user is known to the system or not. The identification look-up takes longer time since it compares the biometric input with all samples in the database. Authentication only compare sample with the sample of claimed identity. [14, ch. 1]

2.4.4 Biometric measurements

Biometric measurements is more difficult than in a password-based system, where the answer just is match or not match. The accuracy of the biometric system must be considered when choosing characteristics. This is measured by two FRR (False rejection rate) that is the probability that two samples from the same user is not a match and FAR (False acceptance rate) is the probability that two samples from different users is a match. There is a threshold \( \eta \) that is used to decide the FRR and FAR. The proportion of authentic scores \( (\omega_1) \) that are less than \( \eta \) is defined as FRR and the impostor score \( (\omega_0) \) that are greater than or equal to \( \eta \) is FAR. The rates can be described mathematical as;

\[
\text{FAR}(\eta) = p(s \geq \eta|\omega_0) = \int_{\eta}^{\infty} p(s|\omega_0)ds,
\]

\[
\text{FRR}(\eta) = p(s \leq \eta|\omega_1) = \int_{-\infty}^{\eta} p(s|\omega_1)ds,
\]

where \( p(s \leq \eta|\omega_x) \) us the probability density function of the authentic respective impostor score. [14, p. 18]

2.4.5 The design of a biometric system

When designing a biometric system it is done in an activity cycle of five steps. Depending on the outcome of one activity, the next step could be forward or re-doing earlier activity. These five steps are explained below followed by a flow-chart of the design cycle. Figure 2.2

Understand the nature of application

Deciding functionality upon type and classification based on how well the system fits different behaviours; cooperative, overt, habituated users, attended, un-tenanted operation, controlled operation and open system. The first is if the user will be cooperative or not, like if the user wants to access something it is likely to cooperate. Overt is if the user knows that it is object for biometric recognition. If the user interacts with the system a lot it is likely that the user will be habituated. The enrollment and recognition operations can either
be attended by a human or not. The environment of the operations may have to be controlled in terms of temperature, pressure, etc. in order to work. Last there is the question of if the system will be closed or open, such if the database of biometric data will be shared between applications or be in one closed application.) This chapter and the next that includes theory, can be compared to this part of the biometric design cycle.

**Choose biometric characteristics**

The choice is based on seven different factors. The disadvantages of biometrics is that it will never be completely solid, therefore factors will have different significance in different systems.

1. *Universality*, the trait should be possessed by the ones authenticated to the system. The fail-to-enrollment (FTE) rate should be low.
2. The *uniqueness* of the characteristics is high the rate of FAR will be low.
3. The characteristic should be high in terms of *permanence* and not be changing significantly over time.
4. *Measurability* from the user perspective in terms of collecting characteristics should be convenient.
5. The time of the authentication is the factor of *performance*.
6. User should have a high *acceptability* when presenting their characteristics to the system.
7. *Circumvention*, in terms of how easy it is to maliciously fake the characteristics.

**Collect biometric data**

As the name implies this step is about the choice of how to collect the biometric data. The choice also includes factors of time, cost and size of the equipment.

**Choose features and matching algorithm**

This is critical step since this is the heart of the system and has to be done with a great deal of knowledge of the selected characteristics and the data extracted from it.

**Evaluate the biometric system**

There is no framework or standardisation for doing the evaluation and it has to account different perspectives that require experts of different fields such psychology, business, computer science and statistics. The proposed method is divided into three evaluation-stages technology, scenario and operational. [14]
Figure 2.2: The design cycle of a biometric system
3

CHARACTERISTICS OF A MOBILE DEVICE

Compared to the biometric design cycle is this a part of understand nature of application.

In the hardware of a device there are features that can be used to distinguish devices from each other. In most cases its not called features rather error sources, noise or bias. Device fingerprinting is the term used for this feature characteristics and the pyramid seen in figure 3.1 shows the different types of sources of device fingerprinting. This thesis focuses on the top of that pyramid that is the sensors.

Figure 3.1: The pyramid of features in a mobile device that can be used for fingerprinting.[9]

As seen in figure 3.1 are sensors an untapped source of fingerprints in mobile devices and example of sensors are microphone, accelerometer, barometer, speakers and gyroscope. The sensors investigated in this work are the accelerometer, gyroscope, and camera sensors. All of them are common sensors in most of the
3.1 Accelerometer

The accelerometer is the sensor that detects movement of a mobile device, like when you change orientation on your device. Acceleration is measured by sensing how much force is applied to the device. The type of accelerometer sensor found in a mobile device are a micro-electromechanical systems known as MEMS. [25]

3.1.1 Fingerprinting characteristics

Measures the characteristics from the accelerometer is done by taking the long term average of the output when the accelerometer is in rest. Which is the biggest error source in the accelerometer and grows quadratic over time. When the accelerometer is in rest the error $\epsilon$ can be calculated as a function of time $t$:

$$s(t) = \epsilon \ast \frac{t^2}{2} \quad (3.1)$$

[25]

3.2 Gyroscope

The gyroscope senses how the device is moving in terms of angles, for measure the orientation. This is originally a mechanical system based on the principle of conservation of angular momentum. The most popular Gyroscope for devices today is MEMS that uses silicon micro-mechanical techniques. Coriolis effect is measured with vibrating elements in the MEMS gyroscope. Coriolis effect is the change of moving objects direction when looking at it from a rotating reference system. The equations of Coriolis force:

$$F_C = -2 \, m \, (\omega \ast v)$$

Where $m$ is the mass of the particle, $\omega$ the angular velocity and $v$ the velocity of the particle in the rotating system. [27]

3.2.1 Fingerprinting characteristics

The gyroscope has some error characteristics like constant bias, white noise, bias instability, calibration error and temperature effects. One of these characteristics that can be tested by reading the output from a gyroscope in rest is the constant bias. Which is bias of the gyroscope output when the gyroscope is still. This constant error $\epsilon$ of the bias over time $t$ leads to an angular error that grows linear;

$$\theta(t) = \epsilon \ast t \quad (3.2)$$

If take the long term average output from the gyro in rest, the constant bias of the gyroscope can be estimated.[25]
3.3 Camera

Note that normally bias in a camera sensor is called noise but for uniformity reason of this report it will be referenced to bias.

The digital camera of a mobile device also includes sensors and other hardware that can be used as fingerprinting characteristics. The basics is that light travels through the lens and hits the imaging sensor which contains pixels that has a filter array in front. The filter gives each pixel a detected color. The pixels are then added together to a resulting signal which is send to some final post processing (color correction, white balance, etc.) steps before the image is written to the memory card. In this process there are different types of bias that effects the picture:

- **Shot noise** - the amount of photons hitting the sensor and each pixel varies a random amount.
- **Fixed pattern noise** - a small electric current that leaks from photo-diodes in each pixel, which is caused by dark current.
- **Photo-response non-uniformity noise (PRNU)** - when manufacturing sensors the silicon gets imperfection which causes that pixels are not equally sensitive to light. This is the main source of pattern bias and makes it unlikely for two cameras to have the same pattern. This bias is not affected by temperature or humidity.

The three types of bias can be described as a mathematical model for getting the output of the sensor $y_{ij}$:

$$y_{ij} = f_{ij}(x_{ij} + \eta_{ij}) + c_{ij} + \epsilon_{ij}$$

where $f_{ij}$ is a multiple factor close to one that captures PRNU, $x_{ij}$ is the number of photons hitting the sensor, $\eta_{ij}$ the shot noise, $c_{ij}$ the dark current and $\epsilon_{ij}$ the additive random bias. The key for a unique fingerprint of the camera (in the mobile device) is to find $f$. [15]

3.3.1 Fingerprinting characteristics

In this work the PRNU will be used as bias as in the research by [15]. PRNU is the average of multiple pictures used and substantially an approximation of $f$. The first step is to remove the pictures-content which leaves the noise, which is done using a denoising filter.

3.4 Allan variance

In clocks, oscillators and amplifiers there is a measures of stability known as Allan variance. This variance is an estimation of bias processed and not imperfections as temperature effects and frequency drift. [1]

This is also a common variance to use when calibrating gyroscope. [26] [18]
The mathematical term of Allan variance is $\sigma^2_y(\tau)$ and the square root of Allan variance is called Allan deviation, that mathematically becomes $\sigma_y(\tau)$.

Allan Variance:

$$\sigma^2_y(\tau) = \frac{1}{2} \langle (\bar{y}_{n+1} - \bar{y}_n)^2 \rangle = \frac{1}{2\tau^2} \langle (x_{n+2} - 2x_{n+1} + x_n)^2 \rangle$$

Allan Deviation:

$$\sigma_y(\tau) = \sqrt{\sigma^2_y(\tau)}$$

[1]

### 3.5 Previous work of device sensor fingerprinting

Accelerometer fingerprinting is a recent field of studies compared to the camera fingerprint that has been around for a longer time. The camera has for a long time been object of identification in forensic purposes and therefore research has been made and is applied today. Most of them uses advanced algorithms to extract the fingerprint and time of identifying has not been a concern. However in the use of this thesis time is an important factor, since accessing a system is a process expected to be fast. In table 3.1 and table 3.2 previous studies is presented in brief, followed by a longer presentation. Studies of gyroscope fingerprinting have not been found. The majority of recent studies regarding the gyroscope have been about speech recognition. [20]

<table>
<thead>
<tr>
<th>Year</th>
<th>Devices</th>
<th>Purpose</th>
<th>Fingerprint</th>
<th>Ref.</th>
</tr>
</thead>
<tbody>
<tr>
<td>2014</td>
<td>107</td>
<td>Identification</td>
<td>Statistics</td>
<td>[10]</td>
</tr>
<tr>
<td>2014</td>
<td>3583</td>
<td>Tracking</td>
<td>Bias offset</td>
<td>[5]</td>
</tr>
<tr>
<td>2015</td>
<td>60-100</td>
<td>Identification</td>
<td>Statistics</td>
<td>this thesis</td>
</tr>
</tbody>
</table>

*Table 3.1: Comparing studies of accelerometer fingerprinting*

**AccelPrint: Imperfections of Accelerometers Make Smartphones Trackable**

The research shows that the accelerometer in a mobile device can be used for identification and tracking purposes. Tests are performed on android devices with an application and on standalone accelerometer chips. Their fingerprint consists of statistics values such mean, standard deviation, skewness, min and max-values in both time and frequency domain. The research make recordings with and without vibrations and in different circumstances; in car, running, walking and standing still. Their test environment machine learning that uses the statistics to build a fingerprint is used.

The result has an accuracy on 98% when having alien devices among the already known devices which. Alien devices means that they are not previously known
to the system.

The research also states that the time needed of identifying a device is 30 seconds and that a CPU-load less than 40% is not affecting the result. Another important thing to notice that since they also used standalone accelerometer in different OS that rules out the possibility of an OS affecting the output from the accelerometer. [10]

**Mobile Device Identification via Sensor Fingerprinting**

The research has a much larger scale experiments of 3583 devices. Experiments is performed using JavaScript in a web-page. The fingerprint consists of calculating the bias offset on the accelerometer data. The result however are not as good as the previous, with successful identification on 15.1%. To improve the result UserAgent-data were added and success rate goes up to 58.7%. But UserAgent is software-based identification that more easily can be modified at the client side. [5]

Since the researches are of such different size they are difficult to compare. It may be the case that AccelPrint gets similar success rate if scaling it up and vice versa.

**Camera**

<table>
<thead>
<tr>
<th>Year</th>
<th>Devices</th>
<th>Purpose</th>
<th>Fingerprint</th>
<th>Ref.</th>
</tr>
</thead>
<tbody>
<tr>
<td>2008</td>
<td>16</td>
<td>Identification</td>
<td>Probabilistic SVM classifier</td>
<td>[8]</td>
</tr>
<tr>
<td>2009</td>
<td>150</td>
<td>Identification</td>
<td>PRNU correlation</td>
<td>[15]</td>
</tr>
<tr>
<td>2015</td>
<td>10</td>
<td>Identification</td>
<td>PRNU correlation</td>
<td>this thesis</td>
</tr>
</tbody>
</table>

*Table 3.2: Comparing studies of camera fingerprinting*

**Blind Identification of Source Cell-Phone Model**

Using a probabilistic SVM (support vector machine) classifier based on different features they manage to get good results (success rate on 95.1%) even on images that is manipulated such cropped, resized or rotated. This however are a small scale experiment with more advanced technique that cannot be applied in authentication purposes rather in forensics. The thing to notice here is that the experiment is performed on cell-phones from 2008 when the pictures had lower quality than today’s smart-phones. [8]

**Digital Camera Identification**

One of the experiments performed in this research included 150 devices with images that had random motives, zooming and other post-processing. The finger-
print consisted of the PRNU correlation and resulted in a FRR of 2.4% and a FAR of 0.043%. The difference to this work is the use of camera of a mobile device instead of a digital camera. [15]
As the title implies this is the part of collect biometric data compared to the biometric design cycle. It can also be seen as a part of choose biometric characteristics.

Overview of the tests performed:

**Measurement I - Motion:** Collect accelerometer and gyroscope data by the use of a JavaScript web-page. With purpose to find out which of accelerationIncludingGravity and acceleration is better in purpose of extract unique device characteristics.

**Measurement II - Motion:** Collect accelerometer and gyroscope data by the use of a JavaScript web-page. With purpose to find unique device characteristics from the sensors.

**Measurement II - Camera:** Collect one video from each device and extract pictures frames from the video. Calculate and compare the PRNU of the extracted pictures. The videos collected by the same process as motion measurement II above

**Measurement III - Camera:** Collected ten pictures instead of a video from the device.

### 4.1 Measurements of motion sensors in JavaScript

Measurements of sensors from mobile devices can be gather in different ways. In the work of this thesis a browser application in JavaScript is used for the data collection.
JavaScript have since the use of mobile devices adapted a lot of new features, which makes it possible to access a lot of hardware features in the devices. No permission is needed to access the gyroscope and accelerometer-data, thus the user do not have to know that the sensors are measured.

![Figure 4.1: The coordinate system used in JavaScript](image)

### 4.1.1 Accelerometer in JavaScript

To get measurements from the accelerometer an event listener called `devicemotion` is used. The output from measurements is the acceleration of the device in \(m/s^2\) according to x-, y- and z-axes (figure 4.1).

There are two types of accelerometer output in JavaScript `accelerationIncludingGravity` and `acceleration`. The acceleration including gravity is acceleration made by the device. In context to `acceleration` not depending on influence of gravity only by the acceleration made on the device. What this actually means is that if a device lies still with the screen facing upwards the `acceleration` output will be zero in x, y and z-axes but the `accelerationIncludingGravity` will be zero along x and y-axes, the z-axis will be equal to G. If you put the device in free fall with the screen facing upwards the acceleration is zero in in all axes with `accelerationIncludingGravity` and x=0, y=0 and z=-G for the `acceleration`. [4]

The rotation rate of the device is also available from the `devicemotion`, that is the acceleration around the axes as seen in figure 4.2.

The JavaScript for measuring the accelerometer:

```javascript
if(window.DeviceMotionEvent) {
    window.addEventListener('devicemotion', function(event) {
        x = event.acceleration.x;
        y = event.acceleration.y;
        z = event.acceleration.z;
        r = event.acceleration.rotationRate;
    });
}
```

[11]
4.1.2 Gyroscope in JavaScript

A listener is implemented in the same way as for the accelerometer. This listener is called `deviceorientation`. The output from this listener is given in degrees of the rotation angle. JavaScript has named these rotations as the figure 4.2.

![Image showing device rotation axes for the JavaScript DeviceOrientation]

**Figure 4.2:** The device rotation axes for the JavaScript DeviceOrientation

Alpha is measured in the range of $0^\circ$ to $360^\circ$ around the z-axis, beta in the range of $-180^\circ$ to $180^\circ$ around x-axis and gamma in the range of $-90^\circ$ to $90^\circ$ around y-axis.[11]

```javascript
if (window.DeviceOrientationEvent) {
    window.addEventListener('deviceorientation', function (event) {
        alpha = event.alpha;
        beta = event.beta;
        gamma = event.gamma;
    }, false);
}
```

**Listing 4.1:** JavaScript measurement of the gyroscope

4.2 Measurement I - Motion

The purpose of the first measurement was to analyse the accelerometer with and without the impact of gravity. To evaluate if any of them was a better choice in terms of characteristics uniqueness in the devices.

The data was collected by developing a JavaScript web-page that used the listeners described in section 4.1.1. The test was completely diverse in sense of device platform and only required a browser installed and Internet connection.

The measurements required that the device was still on a flat surface, then started by pressed a button. It gathered 1000 samples of accelerometer data that where saved as a CSV-file for further analyzing. It also collected gyroscope data as well for possible future analyzing purposes. The screen-shots (figure 4.3) shows
the web-page while measuring and the when the measurements are finished and ready to send.

![Screen-shots of web-page during accelerometer measurements in test I](image)

**Figure 4.3:** Screen-shots of web-page during accelerometer measurements in test I

### 4.3 Measurement II - Motion

The second measurements were also performed from a web-page using JavaScript to collect gyroscope and accelerometer data with an additional step to collect measurements from the camera of the device. As of the result in last test there were a few changes made to improve the accuracy of the measurements and to collect sensor samples from the gyroscope and camera:

1. Adding time-stamp to every recording sample to know exactly recording frequency to enable further analyzing.

2. Time based recording on 30 seconds instead of taking 1000 samples as in the first measurement.

3. It is also sampling at a lower rate of at least 10 ms instead of as fast as it could before to reduce the effect of other processes that may are in use on the device.

4. The accelerometer readings used is only accelerationIncludingGravity, due to results described in section 5.2.

5. Added a readings of the gyroscope

6. Collecting camera sensor data by a five seconds black video, section 4.4.
4.4 Camera measurements

The research found on identifying a camera based on pictures has been in forensic purposes. The difference with forensics and the use in authentication of a system or application is that there are harder time-limits. Integrity is also a factor that comes into play to the system to be socially acceptable. That is why some limitations has been made in these measurements. The black motive is used due to integrity, thus no information that could reveal the environment surrounding the camera is sent. Because of having a socially acceptable system there are limited number of pictures that can be taken in an enrollment phase.

To measure the camera two measurements were gathered. In both cases was the device put on a flat surface which makes the camera result black. Both of the measurements are analysed by the PRNU-method used in [15] described in section 5.4.

Collecting I - Black video:
The recommended number of pictures for camera fingerprinting is 50 [15]. Which is not convenient in gathering purposes, thus not many users would send 50 pictures in order to access a system or application. That is why the first test asked to recording a 5 seconds video-recording with the camera towards a flat surface. This video is then shuttered into picture frames, 5 seconds generate 100-200 pictures depending on the recording rate of fps (frames per second).

Collecting II - 10 black pictures:
Taking 10 pictures from a device, also with the camera pointing down on a flat surface. Since [15] were using pictures of diverse motives this aims to investigate
if it may be enough with 10 pictures when the motive is the same.

Screen-shots from the camera-page of the second measurements:

![Image of Nexus 7 sensor measurements]

**Figure 4.5:** Sensor measurements on a Google Nexus 7

For calculating the bias the MATLAB `medfilt2` are used, which is a 2-D median filtering that outputs the median value of each pixel by its 3-by-3 neighbors.

![Image of MATLAB median filter]

**Figure 4.6:** the MATLAB `medfilt2` outputs the median of each pixel by its 3-by-3 neighbors

From the `medfilt2` a picture is gained without bias which is subtracted from the original. In this case the picture without bias is removed from the original to obtain the bias. This technique works best if there are no feature added to the pictures such auto-fix, black and white etc. The more images used for the average value the more accurate the bias gets and more of the random bias is removed. For calculating the PRNU there is a recommendation minimum of 50 pictures. This is then seen as the reference pattern used for correlating the noise from another picture. This correlation is calculated like:

\[
\text{corr}(n, r) = \frac{(n - \bar{n})(r - \bar{r})}{\|n - \bar{n}\|\|r - \bar{r}\|}
\]
where $n$ is the reference pattern and $r$ the noise from another picture. [15]
The chapter is seen as one part of the *choose biometric characteristics* and a part of *choose features and matching algorithm* steps of the biometric design cycle. The chapter covers the results of measurements described in chapter 4.

### 5.1 Pre-measurements

To get a hint if accelerometer is a possible fingerprinting candidate pre-measurements were performed. This was in the early state of the development of the web-page used in measurements I and II. Measurements performed on six different iPhones showed in figure 5.1 indicates that the accelerometer is a sensor that could be used in fingerprinting purpose.

![Figure 5.1: Scatter-plot on accelerometer recordings of 6 Apple devices](image-url)
5.2 Result of measurements I - Motion

The data was gathered as described in section 4.2 from the web-page (figure 4.3). This resulted in over a hundred recordings with an FTE of 5% and had diversity in platforms, brands and models (figure 5.3).

The purpose of this measurement was to identify if there was differences in terms of bias characteristics between the JavaScripts two accelerometer readings. The result of the measurements can be showed by making scatter-plots of the output acceleration of the devices. As seen in the figure 5.3 the Sony Xperia devices represents more than a fifth of the total devices in the measurement.

![Figure 5.2: Diversity of device brand sampled in measurements I](image)

![Figure 5.3: Most common devices models in measurements I](image)
5.2 Result of measurements I - Motion

Figure 5.4: Bias from twelve Sony Xperia devices measured with JavaScripts acceleration

Figure 5.5: Bias from twelve Sony Xperia devices measured with JavaScripts acceleration including gravity
5.3 Result of measurements II - Motion

The result is of the gyroscope and accelerometer data collected from 60 devices with an FTE of 2% by an improved version of the JavaScript web-page used in measurements I. The changes that were made is described in section 4.3 to improve the analyze of the data. The diversity of the devices brands in the measurement is have not changed significant compared to measurements.

5.3.1 Permanence of accelerometer

When choosing biometric trait one of the factors to considred is permanence described in section 2.4.5, that is the trait not changing over time. To test permanence measurement II were performed on a Sony Xperia Z1 Compact over a period of 50 days. The choice of device was based on that Sony Xperia devices is 30% of the devices in measurements II. The same test were also made on a Google Nexus 7 tablet. The graphs below shows the difference of accelerometer readings over time. To get an perspective of this measurements among devices the scatter-plot

![Graph](image)

**Figure 5.6:** Accelerometer readings of x-axes on a Sony Xperia Z1 Compact and a Google Nexus 7 over 50 days

in figure 5.9 that include the same measurements from Sony Xperia Z1 Compact as in figure 5.6, figure 5.7 and figure 5.8.
5.3 Result of measurements II - Motion

Figure 5.7: Accelerometer readings of y-axes on a Sony Xperia Z1 Compact and a Google Nexus 7 over 50 days

Figure 5.8: Accelerometer readings of z-axes on a Sony Xperia Z1 Compact and a Google Nexus 7 over 50 days
5.3.2 Features of accelerometer data

As [10] statistical features calculated by the time domain. The features used is calculated as followed:

<table>
<thead>
<tr>
<th>Feature Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mean</td>
<td>$\bar{x} = \frac{1}{N} \sum_{i=1}^{N} x(i)$</td>
</tr>
<tr>
<td>Std-Dev</td>
<td>$\sigma = \sqrt{\frac{1}{N-1} \sum_{i=1}^{N} (x(i) - \bar{x})^2}$</td>
</tr>
<tr>
<td>Average Deviation</td>
<td>$D_{\bar{x}} = \frac{1}{N} \sum_{i=1}^{N}</td>
</tr>
<tr>
<td>Skewness</td>
<td>$\gamma = \frac{1}{N} \sum_{i=1}^{N} \left(\frac{x(i) - \bar{x}}{\sigma}\right)^3$</td>
</tr>
<tr>
<td>Kurtosis</td>
<td>$\beta = \frac{1}{N} \sum_{i=1}^{N} \left(\frac{x(i) - \bar{x}}{\sigma}\right)^4 - 3$</td>
</tr>
<tr>
<td>RMS Amplitude</td>
<td>$A = \sqrt{\frac{1}{N} \sum_{i=1}^{N} (x(i))^2}$</td>
</tr>
<tr>
<td>Lowest Value</td>
<td>$L = (\text{Min}(x(i))</td>
</tr>
<tr>
<td>Highest Value</td>
<td>$H = (\text{Max}(x(i))</td>
</tr>
</tbody>
</table>

*Figure 5.10: Calculations of statistical accelerometer features. From [10, p.6]*
To compare these features and get a picture of if any of them are good for fingerprinting plots of devices were made. Those can be found in appendix A. The chosen devices for the plots are twelve Sony Xperia Z-devices including the Sony Xperia Z1 Compact that contain measurements over 50 days. In the graphs the medium, min, max and the RMS is plotted. The Sony Xperia Z1 Compact measurements still are quite gathered compared to the other device. Standard deviation looks to differentiate a bit more and kurtosis, and skewness means deviation can can no pattern be seen.

In order to compare which properties that is best, the distance between these points for all the 60 units were calculated. A point contain the x-, y- and z-coordinates of the feature and the distance is the Euclidean distance. The minimum and median distance from all the sample points calculated into features to compare with the same values calculated from only one unit (Sony Xperia Z1 Compact or Google Nexus 7) over time. The choice to use the median and not average value because it could be outliers in the measurements. As seen in table 5.1 the values proves the result read from appendix A.

<table>
<thead>
<tr>
<th>Minimum distance</th>
<th>Mean</th>
<th>RMS</th>
<th>Std.dev.</th>
<th>Min</th>
<th>Max</th>
<th>Median</th>
</tr>
</thead>
<tbody>
<tr>
<td>All</td>
<td>0,018</td>
<td>0,0193</td>
<td>0,0001</td>
<td>0,0287</td>
<td>0,0365</td>
<td>0</td>
</tr>
<tr>
<td>Z1Comp</td>
<td>0,0171</td>
<td>0,0171</td>
<td>0,0002</td>
<td>0,0224</td>
<td>0,0144</td>
<td>0,0175</td>
</tr>
<tr>
<td>95%</td>
<td>89%</td>
<td>200%</td>
<td>78%</td>
<td>39%</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Nexus7</td>
<td>0,0237</td>
<td>0,0182</td>
<td>0,0008</td>
<td>0,0267</td>
<td>0,0119</td>
<td>0,0225</td>
</tr>
<tr>
<td>132%</td>
<td>94%</td>
<td>4%</td>
<td>93%</td>
<td>33%</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Minimum distance</th>
<th>Mean</th>
<th>RMS</th>
<th>Std.dev.</th>
<th>Min</th>
<th>Max</th>
<th>Median</th>
</tr>
</thead>
<tbody>
<tr>
<td>All</td>
<td>0,7934</td>
<td>0,3925</td>
<td>0,0202</td>
<td>0,89</td>
<td>0,9199</td>
<td>0,7953</td>
</tr>
<tr>
<td>Z1Comp</td>
<td>0,0519</td>
<td>0,0519</td>
<td>0,0009</td>
<td>0,0447</td>
<td>0,054</td>
<td>0,0575</td>
</tr>
<tr>
<td>7%</td>
<td>13%</td>
<td>690%</td>
<td>5%</td>
<td>6%</td>
<td>7%</td>
<td></td>
</tr>
<tr>
<td>Nexus7</td>
<td>0,0285</td>
<td>0,0275</td>
<td>0,0019</td>
<td>0,0361</td>
<td>0,0302</td>
<td>0,0283</td>
</tr>
<tr>
<td>4%</td>
<td>7%</td>
<td>10%</td>
<td>4%</td>
<td>3%</td>
<td>4%</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Median distance</th>
<th>Mean</th>
<th>RMS</th>
<th>Std.dev.</th>
<th>Min</th>
<th>Max</th>
<th>Median</th>
</tr>
</thead>
<tbody>
<tr>
<td>All</td>
<td>0,7934</td>
<td>0,3925</td>
<td>0,0202</td>
<td>0,89</td>
<td>0,9199</td>
<td>0,7953</td>
</tr>
<tr>
<td>Z1Comp</td>
<td>0,0519</td>
<td>0,0519</td>
<td>0,0009</td>
<td>0,0447</td>
<td>0,054</td>
<td>0,0575</td>
</tr>
<tr>
<td>7%</td>
<td>13%</td>
<td>690%</td>
<td>5%</td>
<td>6%</td>
<td>7%</td>
<td></td>
</tr>
<tr>
<td>Nexus7</td>
<td>0,0285</td>
<td>0,0275</td>
<td>0,0019</td>
<td>0,0361</td>
<td>0,0302</td>
<td>0,0283</td>
</tr>
<tr>
<td>4%</td>
<td>7%</td>
<td>10%</td>
<td>4%</td>
<td>3%</td>
<td>4%</td>
<td></td>
</tr>
</tbody>
</table>

_table 5.1: Comparing distance between values of statistical features for the accelerometer. Z1Comp and Nexus7 is the devices that have been measured over 50 days. (Z1Comp=Sony Xperia Z1 Compact & Nexus7=Google Nexus 7)

5.3.3 Gyroscope

The same calculation and plots of the measurements as for the accelerometer has been done with the gyroscope. Since the output of the measurements is in degrees and as written in section 4.1.2 the alpha value goes from 0 to 360 degrees, beta from -180 to 180 degrees and gamma from -90 to 90 degrees. To get rid of the case when the values in measurement readings switch from 0 to 360 or
-90 to 90 degrees. The output is calculated through sinus, cosine and tangent, 
\( (\alpha = \sin(\text{alpha}), \beta = \cos(\text{beta}), \gamma = \tan(\text{gamma})) \). As the measurements is in degrees the measurements is only the same if the device is rotated in the exactly same angular-values of the axes as last time. Constant bias cannot be calculated in the same way as for the accelerometer were the measurements should be zero without bias. 

The constant bias from the gyroscope is calculated as the distance between the vectors \( (v = \{\alpha, \beta, \gamma\}) \) of the measurements, because that value would be the same in an ideal sensor with zero bias. That however did not result in the same stability in permanence as seen in table 5.2.

<table>
<thead>
<tr>
<th></th>
<th>Mean</th>
<th>Std.dev.</th>
<th>RMS</th>
<th>Min</th>
<th>Max</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Minimum distance</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>All</td>
<td>0,000188</td>
<td>1,31E-05</td>
<td>0,000112</td>
<td>2,63E-05</td>
<td>0</td>
</tr>
<tr>
<td>Z1Comp</td>
<td>0,00924</td>
<td>0,001157</td>
<td>0,00896</td>
<td>0,009478</td>
<td>0,001348</td>
</tr>
<tr>
<td>Z1Comp/all</td>
<td>«100% «100% «100% «100%</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Nexus7</td>
<td>0,006013</td>
<td>0,003204</td>
<td>0,006512</td>
<td>0,000738</td>
<td>0,000126</td>
</tr>
<tr>
<td>Nexus7/all</td>
<td>«100% «100% «100% «100%</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>Mean</th>
<th>Std.dev.</th>
<th>RMS</th>
<th>Min</th>
<th>Max</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Median distance</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>All</td>
<td>0,019079</td>
<td>0,005938</td>
<td>0,016074</td>
<td>0,012646</td>
<td>0,007945</td>
</tr>
<tr>
<td>Z1Comp</td>
<td>0,00924</td>
<td>0,001157</td>
<td>0,00896</td>
<td>0,009478</td>
<td>0,001348</td>
</tr>
<tr>
<td>Z1Comp/all</td>
<td>48% 19% 56% 75% 17%</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Nexus7</td>
<td>0,006013</td>
<td>0,003204</td>
<td>0,006512</td>
<td>0,000738</td>
<td>0,000126</td>
</tr>
<tr>
<td>Nexus7/all</td>
<td>32% 54% 41% 6% 2%</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 5.2: Comparing distance between values of statistical features for the gyroscope. Z1Comp and Nexus7 is the devices that have been measured over 50 days. (Z1Comp=Sony Xperia Z1 Compact & Nexus7=Google Nexus 7)

If the gyroscope values in table 5.2 are compared to the accelerometer values in 5.1, is the accelerometer much more stable over time. The percentage of the gyroscope distances is much higher than the accelerometer percentage.

## 5.3.4 Allan variance

As described in section 3.4 the Allan variance is used to calibrate sensors. The Allan variance calculated from all sixty devices compared in table 5.3. If the variance stays the same between measurements for each device it would be a good fingerprinting feature. 

As read in the table 5.3 is the Allan variance not the same between measurements of the same device. Thus the variance between all the 60 devices is smaller than the variance between the variance of one device measured over time. This result is not making the Allan variance to a candidate of a fingerprinting feature of the gyroscope.
5.4 Result Camera-measurements

<table>
<thead>
<tr>
<th>Minimum distance</th>
<th>All</th>
<th>Z1Comp</th>
<th>Nexus7</th>
<th>Z1C./All</th>
<th>Nex./All</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accelerometer</td>
<td>2,28E-14</td>
<td>9,06E-14</td>
<td>1,02E-12</td>
<td>«100%</td>
<td>«100%</td>
</tr>
<tr>
<td>Gyroscope</td>
<td>1,91E-19</td>
<td>2,85E-17</td>
<td>2,57E-17</td>
<td>«100%</td>
<td>«100%</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Median distance</th>
<th>All</th>
<th>Z1Comp</th>
<th>Nexus7</th>
<th>Z1C./All</th>
<th>Nex./All</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accelerometer</td>
<td>3,64E-12</td>
<td>3,57E-13</td>
<td>4,96E-12</td>
<td>10%</td>
<td>&lt; 100%</td>
</tr>
<tr>
<td>Gyroscope</td>
<td>1,68E-16</td>
<td>4,17E-17</td>
<td>1,44E-16</td>
<td>25%</td>
<td>86%</td>
</tr>
</tbody>
</table>

Table 5.3: The Allan variance differences between measurements of all devices and same devices (Z1Comp & Nexus7)

5.3.5 Simulate authentication of motion sensors in MATLAB

To test the time features of the accelerometer a simulation were performed in MATLAB. In the simulation fingerprints of all devices is calculated. It contains the features described in section 5.3.2 that resulted in the most stable values over time; min, max, mean and RMS. The code of the simulation can be found in appendix B.

When a new measurement is to the simulation, features are calculated and compared to the already known devices. The comparing is done by an algorithm that calculates the point distance between all points of the input device and a known device. Point distance is the distance between two points. In this case all points of the input device is compared to all points in a known device.

The min, max, mean and RMS is then calculated between the distances. The smaller values the closer to the input device. The features is then used to decide if there is a match or not, by sorting out the lowest values. Since the percentage of features median distance for the accelerometer is around a twentieth a threshold of the 5% the devices of each feature is chosen. If the most common device among the devices in the output is the input device there is a match.

As in biometric system the threshold decides how far from a device in the database an input can be and sill be a match. This threshold creates a rate of error in the system called FRR and FAR (see section 2.4.4). There are two values that can be changed in the simulation that affects the FAR and FRR that is \( \text{th1} \) and \( \text{th2} \). The result of these changed values is presented in table 5.4.

5.4 Result Camera-measurements

To get result of the camera sensor the PRNU value is calculated as an approximation of the algorithm described in section 4.4 and also used by [15].
Table 5.4: The FAR and FRR of the MATLAB simulation when changing threshold values \( th_1 \) and \( th_2 \), the code can be found in appendix B

5.4.1 Camera measurement I

Since this thesis compared to earlier work (section 3.5) has the purpose of authentication and not forensics, is convenience of the collecting and measurability factors to take into account. That is why the first experiment is asked the users to record a five seconds video-clip with the device camera facing down on a flat object, like a table. Instead of making the user take 50 pictures or more which require a lot more time.

The video is then shuttled into images (100-200 from a 5 seconds video depending on fps on recording camera) that is used for calculating the PRNU.

```matlab
%% Make images from video frames
shuttleVideo = VideoReader(filename);
i = 1;
while hasFrame(shuttleVideo)
    img = readFrame(shuttleVideo);
    fn = [sprintf([filename '_%03d'],i) '.jpg'];
    imwrite(img,fn); % Write to a JPEG file
    i = i+1;
end

%% Calculate PRNU from images
imagefiles = dir([filename '.*.jpg']);
for ii=1:nbr_of_images
    currentfilename = imagefiles(ii).name;
    currentimage = imread(currentfilename);
    img = im2double(currentimage);
    filtImg = medfilt2(img);
    noise = noise + ( img - filtImg );    % add noise from current image
end
prnu = noise / nbr_of_images; % get average noise
```
% width and height is saved for comparing correlation with images of different size
save(filename, ’prnu’);

**Listing 5.1:** Shutter a video into picture, calculating the PRNU of the pictures in MATLAB

To compare a pictures between all collected PRNU the same calculation is done. Then the noise from the reference pictures is compared to all collected PRNU and correlation is calculated like in listing 5.1.

```matlab
load(prnu_mat);
% Make it a flat vector instead than a matrix
prnu_vector = reshape(prnu, 1, numel(prnu));
% Calculate the mean PRNU value
p = prnu_vector - mean(prnu_vector);

ref_img = im2double(imread(imgname));
noise = ref_img - medfilt2(ref_img); % get noise by remove denosied image scene
img_vector = reshape(noise, 1, numel(ref_img)); % reshap to get same lenght as prnu
i = img_vector - mean(img_vector);

% calculate correlation between PRNU and reference image
correlation = (i * (p')) / (sqrt(i * i') * sqrt(p * p'))
```

**Listing 5.2:** Comparing the PRNU of an input picture with already known PRNU in MATLAB

Identify an input PRNU with the PRNU from already known devices reached a high value of FRR with only six devices, only two of them were correctly identified. Since [15] made better result than this, that the bad result may occurred due to the use of video instead of pictures. Thus the decision to redo the measurements but with picture instead of videos for calculating the PRNU.

### 5.4.2 Camera measurement II

Since the bad result in camera measurement I the new test consist of 10 pictures from every device. The recommendation from [15] to use at least 50 images is here compensated by using black pictures (picture taken with the camera facing down). Since the motives always is the same the idea is that the noise removal will be better in fewer images. The same code is used as in measurements I with the differences that the video-to-image part is removed. The sizes of the images is larger since the camera on the mobile devices has higher resolution when taking a picture then when recording.
The result of the measurements started out well with no FRR with five devices, but FRR increased rapidly as seen in table 5.5. As the value grew that quickly no more samples from devices were gathered.

<table>
<thead>
<tr>
<th>Devices</th>
<th>FRR</th>
<th>Time [s]</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>0%</td>
<td>15-20s</td>
</tr>
<tr>
<td>7</td>
<td>50%</td>
<td>17-26s</td>
</tr>
<tr>
<td>10</td>
<td>67%</td>
<td>25-46s</td>
</tr>
</tbody>
</table>

*Table 5.5: FRR and time taken to compare PRNU of camera images.*
This chapter interweaves the theory and method with the result. Discuss the difference between the theory and result is and why. The limitations of the method used is also discussed. This chapter together with the next chapter that includes conclusions is be seen as choose features and matching algorithm of the biometric design cycle.

6.1 Accelerometer

6.1.1 Result

The result of measurements I resulted in some unexpected result, the JavaScripts output without gravity does not seem to have any constant bias at all. The reason could be that some software calibration of the sensor data is done. The recommendation from MEMS accelerometer manufacturers is to calibrate the sensors. [17] Doing some research on Android sensors reviled that their SensorEvent also has two types of accelerometer sensors that can be used:

- TYPE_ACCELEROMETER is the hardware measurements that measures the force of acceleration including the force of gravity with the SI unit \( \text{m/s}^2 \). This sensor is stated as only containing hardware sensor output. But there have been some bias removal from the sensor such bias from different temperature.
- TYPE_LINEAR_ACCELERATION is without gravity and a combination of hardware and software sensor. [3]

It would be a reasonable assumption that JavaScripts acceleration without gravity gets sensor data from Androids TYPE_ACCELEROMETER and JavaScripts acceleration including gravity gets data from TYPE_LINEAR_ACCELERATION. Thus software calibrations or calculations have been done on the output event from
the acceleration including gravity. This however is not anything that is public in any specifications such as W3C or Mozilla. [4] [22]
As a result of motion measurements I the used measurement of the accelerometer in motion measurements II is the one including gravity.

As seen in the figures 5.6, 5.7 and 5.8 the Google Nexus 7 has not changed much over the 50 days compared to the Sony Xperia Z1 Compact that especially has changed on the y-axis. The reason for the difference of accelerometer change over time may be due to the Google Nexus 7 being in the same place during those 50 days, and therefore only used when the tests were performed. Unlike the Sony Xperia device that was used daily and might have been dropped at some point. An additional fact about the measurements is that both devices have changed OS version between measurements 2 and 3, from Android version 4.4.4 to 5.1.1 and that different browser were used (Opera, Chrome and Firefox). Only the Sony Xperia device had changed and not the Google Nexus, which leads to the conclusion that OS version or browser does not matter and that the use of the device may affect the accelerometer.

When comparing distances between the time features there are some values to discuss. The percentage that is calculated in table 5.1 is the percentage calculated to compare if the the distances of features between all 60 devices is larger than the distances between measurements of one device. If the min-distance has a percentage larger than 100% that means that there are different devices that have closer feature-distance than the ones between one device, thus not a good candidate for fingerprinting. Average deviation, kurtosis and skewness were excluded from the table since their percentage were all too high (the min-distance in percent were higher than 100%). The median distance of the features gives a value of the normal case of the measurements. Example the median mean-distance between all devices is ten times longer than the median mean-distance between the measurements of Nexus7. The lower percentage the lower risk of that another device has similar values. From this point of view the Mean, Maximum, Minimum and Median makes the best features of fingerprinting.

6.1.2 Method

As discussed in the beginning of the section above the JavaScript or Android/iOS is doing some calibration with the sensor that effects the results if not dealing with raw data. As mentioned is the data used in measurements II probably as raw data as you can get without reading from the accelerometer alone. To read directly from the accelerometer would however be hard since manual calibration of noise caused by temperature had to be done.
6.2 Gyroscope

6.2.1 Result

The first method used to compare the measurements was based on research of the accelerometer since there were no earlier research on the gyroscope. This may affect the outcome since there may be other features that would have given better results.

The other method where calculating the Allan variance is used for calibration of gyroscope, did not give the expected results. Since the variance is used for gyroscope calibration it may be the case that it already is calculated and compensated for in the device.

The gyroscope seems much more sensitive in measurements than the accelerometer and therefore it is harder to extract the constant bias. The fact that Android or JavaScript does not reveal information on what bias compensation and calibration that has been done before the developer get the measurement data, makes it harder to analyse the noise. The gyroscope seems to be much more sensitive than the accelerometer is drawn by reading from table 5.2 were the *Sony Xperia Z1 Compact* device changed the min median distance with 75% and the *Google Nexus 7* only with 6%. The *Sony Xperia* has been used over the fifty days of measurements, compared to the *Nexus* that were only used at the time of the measurements.

A thing to take to account before the constant bias from the gyroscope is ruled out, is if the sensor data gotten from JavaScript contains software calibrations or the output data coming raw from the sensor.

Android developer page about sensor events state that they make factory calibration and temperature compensation even on their uncalibrated sensor events (magnetometer and gyroscope). Which is relativity new feature added in Android 4.3 Jelly Bean (API level 18 [19]) from 2013, the original once used since Android 1.5 Cupcake (API level 3 [12]) from 2009 makes more noise compensation and calibration. What kind of compensation and calibration done is not public. [3]

Since the output of both the calibrated and uncalibrated sensor is in \( \text{rad/s} \) implies that it could be some software calibration in the data from JavaScripts that is in degrees.

6.2.2 Method

The choise of using JavaScripts listener to collect the data seems to work as expected. The question to ask is the same as for the accelerometer, how much calibration and compensation of bias and drifts already done before the software developers get the output from the gyroscope. The positive thing about using JavaScript instead of developing an application is that the diversity of the collected devices is much better. It also gets easier to collect measurements since it is a web-page is much easier to spread and no installation is needed, in context to
an application that has to be installed. The gain of using an Android application when measuring the gyroscope would be that Android provides an uncalibrated version of the gyroscope since 2013 [19]. This rawer data may result in better feature values in time domain or Allan variance.

6.3 Camera

6.3.1 Result

The result of the camera were not as good as expected or as good as by [15] were PRNU also was used. The significant differences is the use of a mobile device camera instead of a digital camera. Although the high level specification seems to be comparable with the digital cameras from 2009, since they had around 11 mega-pixels, an images size of around 4000x3000 pixels, and digital zoom of 4 times and had HD video recording width 30 fps. [6] This is about the average smart-phone today, but some other specifications may also impact as ISO, optical zoom etc.

6.3.2 Method

The two methods used for collecting picture features had different advantages and disadvantages. The video-collecting done in connection to the second measurement were good in terms of measurability since it was easy to record a video of five seconds and send. It generated 100-200 which made enough pictures for a trait. On the other hand that lead to worse result in terms of uniqueness. The second way of collecting data was not as good in terms of measurability but it got slightly more uniqueness, but far from good enough.

6.4 The work in a wider context

There is a lot of issues to discuss in terms of privacy and integrity when dealing with the sensors of devices. To begin neither of the motion sensors require any permission to read when visiting a web-page. If there is an easy way of identifying a device by a sensor the days of using cookies will be long gone. Which of course can has advantage in terms of ease of use. But the value of your personal information today of the commercial and advertising industry. It is hard to set a price for something that could identify you everywhere on the Internet. The tracking possibilities are enormous and have to be concerned if this type of identification can be done.

There are of course some good things in the view of ethical and social aspects. If sensor-data is used as aimed in this thesis it gain privacy and integrity since the provided possibility of more secure authentication both between human and machine and M2M. Because, you want to know that it really is one of your heat sensors that sending signals to your thermostat, or that only your mobile device
that can unlock the front door.

The point here is that fingerprinting features of your device should be treated in the same way as your biometric trait. This means that you want to have control over where your biometric trait is used. Most of us think that it is legit that an Authority would use our fingerprint if it resulted in a more secure society. On the other hand, most of us do not want our fingerprints to be used in commercial purposes. This concept should be considered when fingerprinting a device as well. The accelerometer data may be applicable to use by banks, to your door or your car. But you may not want as a login feature to a commercial site that may sell that information.
This chapter reconnects to the aim and objectives of the thesis. In comparison to designing a biometric system this would be the part of *choosing feature and matching algorithm*.

### 7.1 Choise of characteristics

In the selection of characteristics, there are seven different factors that must be considered (section 2.4.5). The sensors in the thesis are compared to decide which sensor is best suitable as a second factor in authentication between devices.

**Universality**

The first factor regarding universality. The FTE of the accelerometer and gyroscope is quite low, around 4% which could be lower if more tests and adjustments are done in the code. Since one of the conditions when doing the measurements was that the device should lay still on a flat surface there are conditions to decide if the device is still or not ([16]). The conditions together with some additional checks for valid sensor-measurements should lower the FTE. The camera is also good in terms of universality because it is almost impossible to find a mobile device without camera today.

**Uniqueness**

As shown in the result the accelerometer is the best choice since the FAR is zero when both threshold values is set to one. The FRR were as high on the other
sensor that no calculations on FAR were made. But there are other methods used of identify the camera as in previous research that shows good uniqueness. [8]

**Permanence**

What was also shown in the result is that the permanence of the accelerometer is better than compared to the gyroscope. If considered using accelerometer in a system were an authentication is done more than once a month, but further testing is recommended. Also some kind of machine learning on the drift of bias would be preferable as used by [10].

The permanence of the camera was not tested but it seems likely that it has a good value of permanence since the result in previous research has tested a random pick of images from portfolios that had been taken at different time and various environments. [15]

**Measurability**

When it comes to measurability, the accelerometer and gyroscope are good choices since they seems to work quite well when only 600 samples are used as in the MATLAB simulation which is just a few seconds depending on the device and sampling rate. Furthermore is it quite quick since the data to send is about 57 KB. To take a picture and send takes longer time considered the size of a picture of a mobile device is between 0.5 and 1.3 MB.

**Performance**

The time to authenticate the accelerometer is just a fraction compared to the camera authentication method. The accelerometer simulation in MATLAB takes around five seconds with sixty devices and the camera took 25-46 seconds when only ten devices were compared.

**Acceptability**

The ethical aspects discussed in section 6.4 regarding information of sensors noise is a part of the acceptability. Today does not many of us care sending sensor information, since we do not think it is (or can) be used to anything else than what the application aims to do (e.g. rotating the device or uploading a picture to a social media site). Today is a gray area for this type of sensor reading, especially when you read research as with the title:

"Gyrophone: Recognizing Speech From Gyroscope Signals"

This is a Stanford security research showing that it is possible to do exactly as the title implies, gyroscopes in smart phones are capable of measuring acoustic signals that can recognize speech. [21]

The conclusion is that it is acceptable of the majority of people today but may not be the case with more knowledge in the area.
Since the number of uploaded pictures today in social media etc. is growing, it is hard to believe the use of pictures in an authentication system would not be acceptable.

To conclude this, all the sensors are probably socially accepted to use for authentication today. The question is what happens in the near future when the sensor data could be used as speech recognition or tracking.

Circumvention

Circumvention is not in the area of the thesis since this is a question of how to implement the authentication system and the security in it. The reason for having a section on challenge-response (section 2.2) in the authentication chapter is that it would be a protocol to consider that would make it harder to malicious fake sensor noise. Ways to do this with the accelerometer is discussed later in this chapter (section 7.2).

Summary of characteristics

The table 7.1 summarizes the conclusions made about the different characteristics to make a summarized answer to the question asked in the aims and objectives of the thesis (section 1.2).

<table>
<thead>
<tr>
<th>Characteristics</th>
<th>Accelerometer</th>
<th>Gyroscope</th>
<th>Camera</th>
</tr>
</thead>
<tbody>
<tr>
<td>Universality</td>
<td>Good</td>
<td>Good</td>
<td>Good</td>
</tr>
<tr>
<td>Uniqueness</td>
<td>Good</td>
<td>Bad</td>
<td>*</td>
</tr>
<tr>
<td>Permanence</td>
<td>Good</td>
<td>Bad</td>
<td>Good</td>
</tr>
<tr>
<td>Measurability</td>
<td>Good</td>
<td>Good</td>
<td>Bad</td>
</tr>
<tr>
<td>Performance</td>
<td>Good</td>
<td>Good</td>
<td>Bad</td>
</tr>
<tr>
<td>Acceptability</td>
<td>*</td>
<td>*</td>
<td>Good</td>
</tr>
<tr>
<td>Circumvention</td>
<td>Good</td>
<td>Good</td>
<td>Good</td>
</tr>
</tbody>
</table>

*Table 7.1: Conclusions of the factors of choosing fingerprint sensor. (Factors from biometric characteristics see section 2.4.5)*

*See explanation respective title above.*

7.2 Further work

Taking this work to the next step that could be to further evaluate and test the accelerometer since that is the only one of the sensors that seems like a promising second factor to use in M2M authentication. That work would make measurements on more devices, and therefore check the scalability of using an ac-
Celerometer. What is the maximum number of devices to have in this kind of authentication before the FAR and FRR grows to unacceptable numbers. Another thing to explore is the possibility of including the challenge-response protocol in the authentication to make it harder of a malicious device to authenticate. With malicious device meaning a malicious human using a device or pretend to be a device. The challenge could be things like vibrating a pattern or moving the accelerometer in a certain way. If continuing with the accelerometer other features of extracting the constant bias would be an area to explore and evaluate if they have lower rates of FAR and FRR or are more scalable in the number of devices that can be used.

Another thing to explore is other sensors than the one presented in this thesis as the microphone, speaker, magnetometer or even the barometer. The most important factors to explore is the scalability and uniqueness because without neither of them the sensor would not be suitable in the aim as characteristics in a M2M authentication system. Also before saying that the gyroscope has bad uniqueness and permanence the data could be collected from an application were the data may be less calibrated.
Appendix
Motion measurements II: Feature plots

In the result of motion measurements II (section 5.3, plots were scattered to analyze which features that are most suitable for device fingerprinting. This appendix includes these plots that are used in section 5.3 and discussed in section 7.1.

Scatter-plot of mean values

![Scatter-plot of mean values](image)

*Figure A.1: Scatter-plot of mean values of 12 Sony Xperia Z-devices including one device with readings over a period of 50 days*
Scatter-plot of standard deviation values

Figure A.2: Scatter-plot of standard deviation values of 12 Sony Xperia Z-devices including one device with readings over a period of 50 days

Scatter-plot of average deviation values

Figure A.3: Scatter-plot of average deviation values of 12 Sony Xperia Z-devices including one device with readings over a period of 50 days
Scatter-plot of skewness values

Figure A.4: Scatter-plot of skewness value of 12 Sony Xperia Z-devices including one device with readings over a period of 50 days

Scatter-plot of kurtosis values

Figure A.5: Scatter-plot of kurtosis values of 12 Sony Xperia Z-devices including one device with readings over a period of 50 days
Scatter-plot of RMS values

Figure A.6: Scatter-plot of RMS values of 12 Sony Xperia Z-devices including one device with readings over a period of 50 days

Scatter-plot of min values

Figure A.7: Scatter-plot of min values of 12 Sony Xperia Z-devices including one device with readings over a period of 50 days
Scatter-plot of max values

Figure A.8: Scatter-plot of max value of 12 Sony Xperia Z-devices including one device with readings over a period of 50 days
function fingerprint_calc(device_id)
%FINGERPRINT_CALC receive the device id and save fingerprint in a mat-file
% The CSV-file is received and being extracted to a fingerprint

file = ['recordning-' device_id '.csv'];
if exist(file, 'file')
    file = importdata(file);
    t = file.data(:,1) - file.data(1,1); % timestamps
    acc = file.data(:,5:7); % accelerometer data
    f_acc = [min(acc);
             mean(acc);
             median(acc);
             max(acc)];
    id = device_id;
    mat_name = ['db/' device_id '.mat'];
    if exist(mat_name, 'file')
        disp('Not saved, %s already exists', device_id);
    else
        save(mat_name, 'id','t','acc','f_acc'); % save to database
    end
end
end

Listing B.1: Making a fingerprint file from a CSV-file in MATLAB

function [match] = fingerprint_matcher(inputfile)
%FINGERPRINT_MATCHER The matcher of an accelerometer data input
% The input file is a CSV-file with accelerometer data in
  column 5-7

th1 = 1; %threshold number 1
th2 = 1; %threshold number 2
nbrOfDeviceIDinSystem = 140;
nbrOfDevicesInSystem = 59;

foundDevices = 0;
labels = cell(1,nbrOfDevicesInSystem);
ansAcc(4,nbrOfDevicesInSystem) = 0;

inputData = importdata(inputfile);
in_acc = inputData.data(:,5:7); % Acc data is in column 5-7

compSamples = 600; %number of sample used to compare
for iii = 1:nbrOfDeviceIDinSystem
  if iii<10
    name = ['00' num2str(iii)];
  elseif iii<100
    name = ['0' num2str(iii)];
  else
    name = num2str(iii);
  end
  file_out = ['db/' name '.mat'];
  if exist(file_out, 'file')
    foundDevices = foundDevices +1;
    mat = importdata(file_out);
    labels(foundDevices) = mat.name;
    diff_acc =
      pdist2(in_acc(1:compSamples,:),mat.acc(1:compSamples,:));
    ansAcc(1,foundDevices) = mean2(diff_acc);
    ansAcc(2,foundDevices) = max(diff_acc(:));
    ansAcc(3,foundDevices) = min(diff_acc(:));
    ansAcc(4,foundDevices) = median(diff_acc(:));
  end
end
% sort the distances, the shortest distance is the one matching
[sort_acc, ind_mean] = sort(ansAcc(1,:));
[sort_acc, ind_max] = sort(ansAcc(2,:));
[sort_acc, ind_min] = sort(ansAcc(3,:));
[sort_acc, ind_med] = sort(ansAcc(4,:));

%take the threshold 2 number of best matches of each feature
out =
  [ind_mean(1:th2);ind_max(1:th2);ind_min(1:th2);ind_med(1:th2)];
%Counts which device_id that is most common
[M,F] = mode(out(:));

if(F>th1 && ~isempty(labels{M}))
    %MATCH, sending back deviceID of device with best match
    match = labels{M};
else
    %NO MATCH
    match = 0;
end
end

Listing B.2: Simulation of authenticating a new CSV-input against already known fingerprint
Example of CSV-file of measuring accelerometer and gyroscope

This is an example of the first row of an csv-file that were made when recording measurements from the web-page. The decimal in the table are decreased to five since the limit of page with. The the real CSV is the output of a sample like:

time = 1427124966085
alpha = 286.42725394605435
beta = 0.9896375362002724
gamma = -7.288607417105047
x = 1.22528076171875
y = 0.1465606689453125
z = 9.65521240234375

As said before is time in milliseconds, alpha, beta, gamma in degrees and x,y,z in \(m/s^2\).
<table>
<thead>
<tr>
<th>time</th>
<th>alpha</th>
<th>beta</th>
<th>gamma</th>
<th>x</th>
<th>y</th>
<th>z</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.4271E+12</td>
<td>286,4273</td>
<td>0.9896</td>
<td>-7,2886</td>
<td>1,2253</td>
<td>0.1466</td>
<td>9,6552</td>
</tr>
<tr>
<td>1.4271E+12</td>
<td>286,4273</td>
<td>0.9896</td>
<td>-7,2886</td>
<td>1,2322</td>
<td>0.1473</td>
<td>9,6552</td>
</tr>
<tr>
<td>1.4271E+12</td>
<td>286,4273</td>
<td>0.9896</td>
<td>-7,2886</td>
<td>1,2322</td>
<td>0.1473</td>
<td>9,6552</td>
</tr>
<tr>
<td>1.4271E+12</td>
<td>286,4273</td>
<td>0.9896</td>
<td>-7,2886</td>
<td>1,2322</td>
<td>0.1473</td>
<td>9,6552</td>
</tr>
<tr>
<td>1.4271E+12</td>
<td>286,4273</td>
<td>0.9896</td>
<td>-7,2886</td>
<td>1,2444</td>
<td>0.1791</td>
<td>9,6552</td>
</tr>
<tr>
<td>1.4271E+12</td>
<td>286,4273</td>
<td>0.9896</td>
<td>-7,2886</td>
<td>1,2444</td>
<td>0.1791</td>
<td>9,6552</td>
</tr>
<tr>
<td>1.4271E+12</td>
<td>286,4273</td>
<td>0.9896</td>
<td>-7,2886</td>
<td>1,2444</td>
<td>0.1791</td>
<td>9,6552</td>
</tr>
<tr>
<td>1.4271E+12</td>
<td>286,4273</td>
<td>0.9896</td>
<td>-7,2886</td>
<td>1,2444</td>
<td>0.1791</td>
<td>9,6552</td>
</tr>
<tr>
<td>1.4271E+12</td>
<td>286,4273</td>
<td>0.9896</td>
<td>-7,2886</td>
<td>1,2444</td>
<td>0.1791</td>
<td>9,6552</td>
</tr>
<tr>
<td>1.4271E+12</td>
<td>286,4273</td>
<td>0.9896</td>
<td>-7,2886</td>
<td>1,2444</td>
<td>0.1791</td>
<td>9,6552</td>
</tr>
<tr>
<td>1.4271E+12</td>
<td>286,4273</td>
<td>0.9896</td>
<td>-7,2886</td>
<td>1,2444</td>
<td>0.1791</td>
<td>9,6552</td>
</tr>
<tr>
<td>1.4271E+12</td>
<td>286,4273</td>
<td>0.9896</td>
<td>-7,2886</td>
<td>1,2444</td>
<td>0.1791</td>
<td>9,6552</td>
</tr>
<tr>
<td>1.4271E+12</td>
<td>286,4273</td>
<td>0.9896</td>
<td>-7,2886</td>
<td>1,2444</td>
<td>0.1791</td>
<td>9,6552</td>
</tr>
<tr>
<td>1.4271E+12</td>
<td>286,4273</td>
<td>0.9896</td>
<td>-7,2886</td>
<td>1,2444</td>
<td>0.1791</td>
<td>9,6552</td>
</tr>
<tr>
<td>1.4271E+12</td>
<td>286,4273</td>
<td>0.9896</td>
<td>-7,2886</td>
<td>1,2444</td>
<td>0.1791</td>
<td>9,6552</td>
</tr>
<tr>
<td>1.4271E+12</td>
<td>286,4273</td>
<td>0.9896</td>
<td>-7,2886</td>
<td>1,2444</td>
<td>0.1791</td>
<td>9,6552</td>
</tr>
<tr>
<td>1.4271E+12</td>
<td>286,4273</td>
<td>0.9896</td>
<td>-7,2886</td>
<td>1,2444</td>
<td>0.1791</td>
<td>9,6552</td>
</tr>
<tr>
<td>1.4271E+12</td>
<td>286,4273</td>
<td>0.9896</td>
<td>-7,2886</td>
<td>1,2444</td>
<td>0.1791</td>
<td>9,6552</td>
</tr>
<tr>
<td>1.4271E+12</td>
<td>286,4273</td>
<td>0.9896</td>
<td>-7,2886</td>
<td>1,2444</td>
<td>0.1791</td>
<td>9,6552</td>
</tr>
<tr>
<td>1.4271E+12</td>
<td>286,4273</td>
<td>0.9896</td>
<td>-7,2886</td>
<td>1,2444</td>
<td>0.1791</td>
<td>9,6552</td>
</tr>
<tr>
<td>1.4271E+12</td>
<td>286,4273</td>
<td>0.9896</td>
<td>-7,2886</td>
<td>1,2444</td>
<td>0.1791</td>
<td>9,6552</td>
</tr>
<tr>
<td>1.4271E+12</td>
<td>286,4273</td>
<td>0.9896</td>
<td>-7,2886</td>
<td>1,2444</td>
<td>0.1791</td>
<td>9,6552</td>
</tr>
<tr>
<td>1.4271E+12</td>
<td>286,4273</td>
<td>0.9896</td>
<td>-7,2886</td>
<td>1,2444</td>
<td>0.1791</td>
<td>9,6552</td>
</tr>
<tr>
<td>1.4271E+12</td>
<td>286,4273</td>
<td>0.9896</td>
<td>-7,2886</td>
<td>1,2444</td>
<td>0.1791</td>
<td>9,6552</td>
</tr>
<tr>
<td>1.4271E+12</td>
<td>286,4273</td>
<td>0.9896</td>
<td>-7,2886</td>
<td>1,2444</td>
<td>0.1791</td>
<td>9,6552</td>
</tr>
</tbody>
</table>


Upphovsrätt

Detta dokument hålls tillgängligt på Internet — eller dess framtida ersättare — under 25 år från publiceringsdatum under förutsättning att inga extraordinära omständigheter uppstår.

Tillgång till dokumentet innebär tillstånd för var och en att läsa, ladda ner, skriva ut enstaka kopior för enskilt bruk och att använda det oförändrat för icke-kommersiell forskning och för undervisning. Överföring av upphovsrätten vid en senare tidpunkt kan inte upphäva detta tillstånd. All annan användning av dokumentet kräver upphovsmannens medgivande. För att garantera äktheten, säkerheten och tillgängligheten finns det lösningar av teknisk och administrativ art.

Upphovsmannens ideella rätt innefattar rätt att bli nämnd som upphovsman i den omfattning som god sed kräver vid användning av dokumentet på ovan beskrivna sätt samt skydd mot att dokumentet ändras eller presenteras i sådan form eller i sådant sammanhang som är kränkande för upphovsmannens litterära eller konstnärliga anseende eller egenart.

För ytterligare information om Linköping University Electronic Press se förlagets hemsida http://www.ep.liu.se/

Copyright

The publishers will keep this document online on the Internet — or its possible replacement — for a period of 25 years from the date of publication barring exceptional circumstances.

The online availability of the document implies a permanent permission for anyone to read, to download, to print out single copies for his/her own use and to use it unchanged for any non-commercial research and educational purpose. Subsequent transfers of copyright cannot revoke this permission. All other uses of the document are conditional on the consent of the copyright owner. The publisher has taken technical and administrative measures to assure authenticity, security and accessibility.

According to intellectual property law the author has the right to be mentioned when his/her work is accessed as described above and to be protected against infringement.

For additional information about the Linköping University Electronic Press and its procedures for publication and for assurance of document integrity, please refer to its www home page: http://www.ep.liu.se/

© Anna Karlsson