Institutionen för systemteknik
Department of Electrical Engineering

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--YY/NNNN--SE

Linköping 2015

Linköpings universitet
TEKNISKA HÖGSKOLAN

Department of Electrical Engineering
Linköpings universitet
SE-581 83 Linköping, Sweden

Linköpings tekniska högskola
Linköpings universitet
581 83 Linköping

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--YY/NNNN--SE

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

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

Linköping, 5 juni 2015

Avdelning, Institution
Division, Department

Information Coding
Department of Electrical Engineering
SE-581 83 Linköping

Datum
Date

2015-06-05

Språk
Language

☐ Svenska/Swedish
☒ Engelska/English
☐ ________________

Rapporttyp
Report category

☐ Licentiatavhandling
☒ Examensarbete
☐ C-uppsats
☐ D-uppsats
☐ Övrig rapport
☐ ________________

URL för elektronisk version

http://urn.kb.se/resolve?urn=urn:nbn:se:liu:diva-XXXXX

ISBN
—

ISRN
LiTH-ISY-EX--YY/NNNN--SE

Serietitel och serienummer
Title of series, numbering

ISSN
—

Titel
Title

Fingeravtryck av Mobila Enheter

Device Sensor Fingerprinting

Författare
Author

Anna Karlsson

Sammanfattning
Abstract

The number of connected devices connected to the Internet is growing rapidly. When talking about devices it also covers the onces not having any contact with humans. This type of devices are the onces 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 use of the accelerometer, camera and gyroscope sensors of a mobile device as fingerprinting of a device. 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.

Nyckelord
Keywords

device fingerprinting, sensor identification, computer security, M2M, authentication

Sammanfattning

Antalet enheter som är anslutna till internet växer snabbt. När man talar om enheter så menar man också de som inte har någon kontakt med människor, ex. en uppkopplad temperaturgivare till en termostat. Dessa typer av enheter är de som förväntas växa mest, vilket är anledningen till att området för att unikt identifiera dessa enheter kräver mer undersökning. Det här examensarbetet inkluderar mätningar och utvärdering på användningen av sensorerna accelerometer, kamera och gyroskope på mobiltelefoner för att undersöka i vilken utsträckning de går att identifiera som unika enheter. Det kan liknas med ett fingeravtryck för mobiltelefonen. Den metod som används bygger på tidigare forskning inom sensoridentifiering tillsammans med metoder som används för att utforma ett biometriskt system. Kombinationen av långa beprövade metoder inom biometriområdet och ny forskning inom identifiering av sensorer är en nytt sätt för att titta på enheters fingeravtryck.

Abstract

The number of connected devices connected to the Internet is growing rapidly. When talking about devices it also covers the onces not having any contact with humans. This type of devices are the onces 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 use of the accelerometer, camera and gyroscope sensors of a mobile device as fingerprinting of a device. 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 thanks 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 and 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 classmate, 720 13-14 and LiU AIF IBK, love to all of you.

The biggest acknowledgment goes of course to my parents who have always been there for me, without them I would never have taken me through this education. Then of course my brother Johan who is always with me.

Linköping, June 2015
Anna Karlsson

Contents

List of Figures

List of Tables

Notation

Notation

Notation	Meaning
G	G-force
ϵ	Bias
$\boldsymbol{F}_C$	Coriolis force

Abbreviations

Abbreviation	Meaning
FAR	False Accept Rate
FRR	False Reject Rate
FTE	Fail To Enrollment
IoT	Internet of Things
M2M	Machine-to-machine
MEMS	Micro-electromechanical System
OS	Operating System
PRNU	Photo-Response Non-Uniformity noise
RMS	Root Mean Square
SVM	Support Vector Machine

1 INTRODUCTION

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. The thesis was performed at Cybercom AB in Linköping. This introduction chapter 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'. Making these things connect to each other we need secure authentication methods for knowing that they are connecting to the device they are suppose to and not anything or anyone else.

For us humans it has become an everyday thing to use two factor authentication when accessing buildings, part of networks, and our bank and so on. When talking about two factor authentication we usually use a combination of either three things; something you *know* like passwords, something you *have* like tag, passport, card, phone or something you *are* like iris or 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 as iris, fingerprint and DNA requires much more effort and time since you can only focus at one person a time. Machines or devices do not have those attributes as us human, they are build on hardware parts.

The background of this thesis is to explore the possibility for a machine to have a fingerprint that can be used to more securely authenticate them. This can be applied in several areas for example in the new smart homes where fridges, stoves, coffee makers and doors should communicate with each other. Another example could be when you only 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 has been more investigated in line with the world of connected devices that is called IoT (Internet of Things) has grown. 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:

- *Can you create a device fingerprint by using the sensor characteristics in a mobile device?*
- *Is this fingerprint suitable to be used as a second factor for identification of a devices?*

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 the title of the thesis implies, authentication is the focus not identification. Said in the background the devices building stone are hardware, thus something the devices *has* 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 and since they like other hardware has some bias that may be unique enough to differ from a 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 biometric authentication
Biometric authentication has methods of measure and compare 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 introduction chapter, includes background, aims and objectives, will give 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 previously 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 the 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 of the thesis.

2

COMMUNICATION & AUTHENTICATION

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 communicate. [Ren et al., 2013]
This chapter will present ways of authentication (two factor, M2M and biometric) that are in the area of this thesis. The section about biometric is included in the thesis because it has methods of measure strength of a biometric trait (especially fingerprint). These methods will be used when comparing strength of characteristic noise in the mobile device.

2.1 Two factor authentication

There are more ways to authenticate a user than password, however it is the most common. There are three different types of authentication;

- Something the authenticator *has* like a key, card, passport and so on
- Something the authenticator *knows* for example password
- Something the authenticator *is*, known as biometrics such as fingerprint or iris pattern

[Anderson, 2008, p. 31]
Authentication in two factor means a combination of two of the three types of authentication above. An example can be the use of a credit card (you have) in combination with a PIN-code (you know) to collect the money from an ATM. Something the authenticator has and knows is the most common combination.

The biggest reason for that biometrics is not that common yet is due to costs. [Anderson, 2008, 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 modern car keys 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 sends 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. This protocol is vulnerable to man-in-the-middle attacks.

Ross states that a much more common use of challenge response is in *two-factor authentication* (section 2.1). An example of use is if you have a bank card reader when accessing your bank on the Internet. When you want to log in there are 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 shard 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.

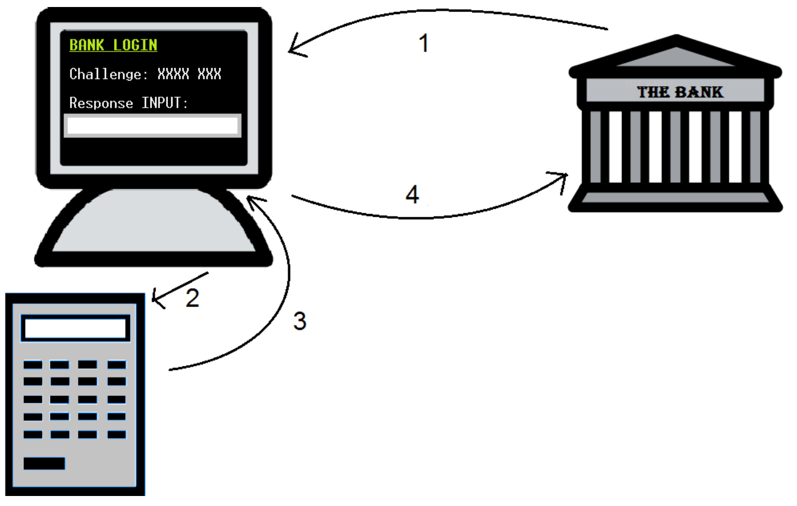

Figure 2.1: *Challenge-response authentication with bank card reader*

Describe figure 2.1:

1. Bank sending 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 o 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.

[Anderson, 2008, ch.3]

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 device 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 the same M2M area network that are 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. The area of M2M are important to make these devises talk without a human behind. This affects the requirements on the applications and networks dealing with the devices. Characteristics of these devices are listed blow:

- *Multitude* - The part of IoT that not directly interacts with humans is the part growing the most. It is soon expected to be significant more than the ones which interact directly with humans. This will put more pressure on application and networks dealing with all devices.
- *Variety* - The connected devices has requirements like data exchange rate, form factor, computing, or communication capabilities. M2M applications have to be built, in order to define and develop common enabling capabilities.
- *Invisibility* - Meaning that the device has virtually zero human control. The more invisibly the less likely for error caused by humans.
- *Criticality* - Devices that can harm humans like electrical errors. 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 is like told above very different, but many of them is similar in some ways, 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 between M2M communication and machine-to-human or human-to-human communication. [Boswarthick et al., 2012, p. 2-4]

2.3.1 Difference between M2M and IoT

The term Internet-of-Things meaning everything that is connected to the Internet. IoT are now in its starting pits and ready to start the race. Machine-to-machine communication is a part of that, but it also covers other areas and IoT some that M2M does not. 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 not only including same device communication but also passive and other low-power sensors that not can be motivated as a M2M hardware module. [Polsonetti, 2014]

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 a machines fingerprint. (This fingerprint is not of the same character as the one this thesis.) The fingerprint in the example consist of hardware message of computers, such serial number of CPU, MAC address of network card, Machine ID etc. [He, 2012]

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 in many platforms like mobile devices.

Quote about M2M authentication:

> *"... traditional methods such as "what you know and who you are" may not be applied".*
>
> [Ren et al., 2013]

This quote states pretty well the aim of this thesis (section 1.2). That is to use what the device is with biometric authentication that is more tried and tested.

2.4 The biometric process

> *"A biometric system measures one or more behavioral characteristics...information of an individual to determine or verify his identity."*
>
> [Jain et al., 2011, p. 3]

2.4.1 Recognition

As said before is biometric something you *are* and the person who wants to be recognised to the system. Buy, showing his or her biometric identifier (fingerprint, iris, DNA, etc.) to the biometric system, therefore seen as a *user* of the system. The strength in biometrics is also the fact that it knows if a user is known to the system even if the user denies it. [Jain et al., 2011, ch. 1]

2.4.2 Biometric systems

There are some blocks for building a biometric systems, which can measure characteristics of a user. In biometrics these characteristics are called *traits, indicators, identifiers, or modalities*, but 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 with the user's identity. The recognition is then done by again collecting biometric data from the user and compare to the database. This is the so called *enrollment and recognition phase*. The raw biometric data is often destroyed after enrollment and the recognition is all about pattern matching. This matching is done in four steps;

1. *Sensor* - to collect the raw biometric samples, that can be an image, amplitude signal, online signature, odour or chemical-based.
2. *Feature extractor* - Makes the raw biometric samples comparable, mostly done in three pre-process operations;

- Quality assessment - Checks if the sample is good enough.
- Segmentation - removes the background noise from sample.
- Enhancement - Uses an algorithm to improve characteristic features of the sample.

3. *Database* - that has the data from the enrollment phase together with some identity data (like name or ID). This database should having an access control mechanism for security reasons.

4. *Matcher* - where 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 counted in different way depending on the characteristics that is used in the system.

[Jain et al., 2011, ch. 1]

2.4.3 Biometric authentication

Biometric authentication, is sometimes also called verification that answers the question *Are you the one you say you are?*. There is also biometric identification that answers *Are you someone known to the system?* but that is not what this thesis aims to answer. The practical difference between authentication and identification is that the user has to give the system some kind of information (username, passport, email etc.) on who they claim to be. But in 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 with the claimed identity. [Jain et al., 2011, ch. 1]

2.4.4 Biometric measurements

Biometric measurements are a bit trickier 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 rates FRR (False Reject Rate) that is the probability that two samples from the same user is not a match and FAR (False Accept Rate) is the probability that two samples from different users is a match. There are a threshold η that is used to decide the FRR and FAR. The proportion of authentic scores (ω_1)) that are less than η is defined as FRR and the impostor score (ω_0)) that are greater than or equal to η is FAR. The rates can be described mathematical as;

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

$$FRR(\eta) = p(s \geq \eta|\omega_1) = \int_{-\infty}^{\eta} p(s|\omega_1)ds,$$

where $p(s \geq \eta|\omega_x)$ us the probability density function of the authentic respective impostor score. [Jain et al., 2011, 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 redoing earlier activity. These five steps are explained below followed by an flow-chart of the design cycle.

Understand nature of application
Deciding functionality upon type and classification based on how well the system fits this different behaviors; cooperative, overt, habituated users, attended, untenanted 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
This 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 fail-to-enrollment (FTE) rate should be low.

2. If 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 measured in *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
Apart from the collecting also includes factors of time, cost and size of the equipment.

Choose features and matching algorithm
A critical step since this is the heart of the system and has to bee done with a great deal of knowledge of the selected characteristics and the data extracted from it.

Evaluate the biometric system
The evaluation is done by asking different questions. There is no framework for doing this and it has to account different perspective that require experts of different field such psychology, business, computer science and statistics. The proposed method is in three evaluation-stages technology, scenario and operational. Jain et al. [2011]

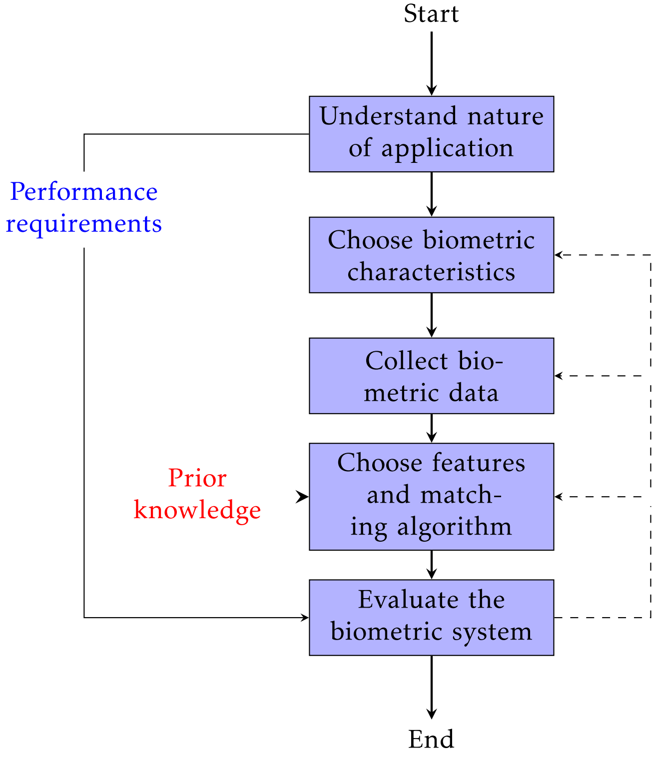

Figure 2.2: The design cycle of a biometric system

3

CHARACTERISTICS OF A MOBILE DEVICE

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

In the hardware of a device there are some features that can be used to distinguish devices from each other. In most cases it is not called features rather error sources, noise or bias. In the aim of this thesis it is feature characteristics that can be seen as a uniqueness of an mobile device. *Device fingerprint(ing)* is the term used for this feature characteristics and the pyramid seen in figure 3.1 shows the different types of sources of device fingerprint. This thesis will focus on the top of that pyramid that is the sensors. All error sources of sensors comes in form of bias and the bias from each sensor covered by the thesis is further explained in this chapter. There is also an explanation on how the sensors is measured in JavaScript that are used for measurements described in chapter 4.

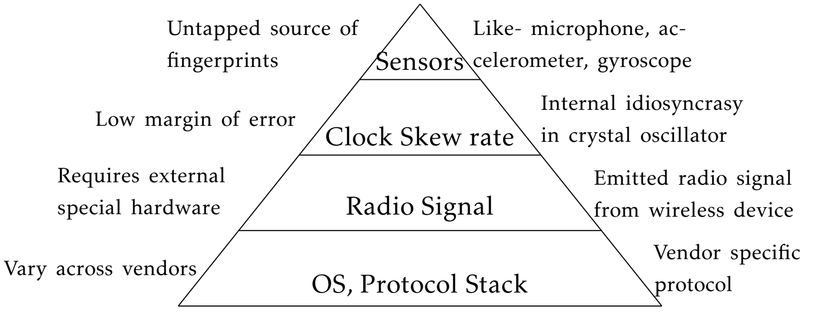

Figure 3.1: *The pyramid of features in a mobile device that can be used for fingerprinting.[Das et al., 2014]*

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 is the accelerometer, gyroscope, and camera sensors. All of them are common sensors in most of the mobile devices used today.

3.1 Accelerometer

The accelerometer is the sensor that detect movement on a mobile device, like when you changing orientation on your device. Acceleration is measured by sensing how much pressure the device has in terms of force. The type of accelerometer sensor found in a mobile device is a micro-electromechanical systems known as MEMS sensor. [Rodriguez and Shala, 2011]

3.1.1 Fingerprinting characteristics / Bias

Measure the characteristics from the accelerometer is done by taking the long term average of the output when the accelerometer is in rest. That is the biggest error source in the accelerometer and it grows quadratic over time, but when the accelerometer is in rest the error ϵ can be calculated as a function of time t;

$$s(t) = \epsilon * \frac{t^2}{2} \tag{3.1}$$

[Woodman, 2007][Rodriguez and Shala, 2011]

3.2 Gyroscope

The gyroscope is sensing how the device is moving in terms of angles, for maintaining or 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 a MEMS that is using silicon micro-mechanical techniques. Coriolis effect is measured with vibrating elements in the MEMS gyroscope. Coriolis effect is a change of moving objects direction when looking at it from a rotating reference system. The difference from the accelerometer is that the gyroscope measures relative to the device body rather than relative to earth. The equations of Coriolis force;

$$\boldsymbol{F}_C = -2\, m\, (\omega * v)$$

Where m is the mass of the particle, ω the angular velocity and v the velocity of the particle in the rotating system. [Woodman, 2007]

3.2.1 Fingerprinting characteristics / Bias

The gyroscope has some error characteristics like constant bias, white noise, bias instability, and calibration error and temperature effects. One of these error characteristics that can be tested by reading the output from a gyroscope in rest is the *constant bias*. That is bias of the gyroscope output when not having any rotation on it. This constant error ϵ of the bias over time t leads to an angular error that grows linear;

$$\theta(t) = \epsilon * t \tag{3.2}$$

If take the long term average output from the gyro in rest, the constant error of a rate gyro can be estimated.[Rodriguez and Shala, 2011]

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 basic is that light travels through a lens and hits a imaging sensor which contains pixels that has a filter array in front. The filter is for gives each pixel a detected color. The pixels is then put together again 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 kind 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* - there is a small electric current that leaks from photodiodes in each pixel, caused by dark current
- *Photo-response non-uniformity noise (PRNU)* - is a bias that is not affected by temperature or humidity. 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 really unlikely for two cameras to have the same pattern.

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, η_{ij} the shot noise, c_{ij} the dark current and ϵ_{ij} the additive random bias. The key for a unique fingerprint of the camera (in the mobile device) is to finding f. [Jenkins, 2009]

3.3.1 Fingerprinting characteristics / Bias

In this work the PRNU will be used as bias as in the research by Jenkins [2009]. 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 measure of stability known as Allan variance. This variance is a estimation of bias processes and not imperfections that temperature effects and frequency drift. [Allan]
This is also a common variance to use when calibrating gyroscope. [VectorNav] [Looney]

The mathematical term of Allan variance is $\sigma_y^2(\tau)$ and the square root of Allan variance is called *Allan deviation*, that mathematically becomes $\sigma_y(\tau)$.
Allan Variance:

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

Allan Deviation:

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

[Allan]

3.5 Previous work of device sensor fingerprinting

Accelerometer fingerprinting is a recent field of studies compared to the camera fingerprint that had been around for a longer time. The camera has for a long time been an object of identification in forensic purposes and therefor many research has been made and are applied today. Most of them uses advanced algorithm to extract the fingerprint and time of extracting has not been a concern. However in the aim of authentication the process ca not be to time-consuming. In the 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. [Michalevsky et al., 2014a]

Accelerometer

Year	Devices	Purpose	Fingerprint	Ref.
2014	107	Identification	Statistics	[Dey et al., 2014]
2014	3583	Tracking	Bias offset	[Bojinov et al., 2014]
This	100	Authentication	Statistics	This thesis

Table 3.1: *Comparing previous studies of accelerometer fingerprinting*

[Dey et al., 2014]: *AccelPrint: Imperfections of Accelerometers Make Smartphones Trackable*

This research shows that the accelerometer can be used in identification and tracking purposes of the device. It is performed on android devices with an android application and on standalone accelerometer chips. Their fingerprint consists of statistics values of the recordings such mean, standard deviation, skewness, min

and max-values in both time and frequency domain. The research makes recordings with and without vibrations and in different circumstances; in car, running, walking, standing still. Their test environment uses machine learning that uses the statistics to build a fingerprint.
The result is an accuracy on 98% when having alien devices among the already known devices which. Alien devices means that they are not previous known for the system, e.g. separate new users from already register users.
The research also states that the time needed for identifying a device is 30 seconds and that CPU-load less than 40% is not affecting the result. Another important thing to notice is that since the also used standalone accelerometer in different OS it rule out the possibility of that an OS can affect the output from the accelerometer. [Dey et al., 2014]

[Bojinov et al., 2014]: *Mobile Device Identification via Sensor Fingerprinting*

This research shows a much larger scale experiments of 3583 devices. Experiments are performed using JavaScript in a web-page. The fingerprint consists of calculating the bias offset on the accelerometer data. The result however is not as good as the previous with successful identification on 15.1%. To improve the result UserAgent-data were added and success rate rises to 58.7% but that is software-based identification that more easily can be modified at the client side. [Bojinov et al., 2014]

Since the research is in such different sizes they are difficult to compare it may be the case that *AccelPrint* gets similar success rate if scaling it up and vice versa.

Camera

Year	Devices	Purpose	Fingerprint	Ref.
2008	16	Identification	Probabilistic SVM classifier	[Celiktutan et al., 2008]
2009	150	Identification	PRNU correlation	[Jenkins, 2009]
2014	20	Authentication	PRNU correlation	This thesis

Table 3.2: *Comparing previous studies of camera fingerprinting*

[Celiktutan et al., 2008]: *Blind Identification of Source Cell-Phone Model*

Using a probabilistic SVM classifier based on different features they manage to get good result (success rate on 95.1%) even on images that are manipulated such cropped, resized or rotated. This however is a small scale experiment with more advanced techniques that not can 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 less quality than today's smart-phones. [Celiktutan et al., 2008]

[Jenkins, 2009]: ***Digital Camera Identification***

One of the experiments performed in this research included 150 devices with images that had random motives, zoom and other post-processing. The fingerprint consisted of the PRNU correlation and resulted in a false reject rate on 2.4% and a false acceptance rate on 0.043%. The difference to this work is the use of camera of a mobile device instead of a digital camera. [Jenkins, 2009].

4

METHOD OF COLLECTING DATA

As the title of the chapter implies this is the part of *collect biometric data* compared to the biometric design cycle. Can also be seen as a part of *choose biometric characteristics*.
This chapter describes the different measurements methods used for collecting the sensor data.

Overview of the tests performed:
Measurement I - Motion: Collected accelerometer and gyroscope data by using a JavaScript web-page. The purpose to find out which of `accelerationIncludingGravity` and `acceleration` is better in purpose of extract unique device characteristics.

Measurement II - Motion: Collected accelerometer and gyroscope data by using a JavaScript web-page. The 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 where collected by the same process as test 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 has 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.

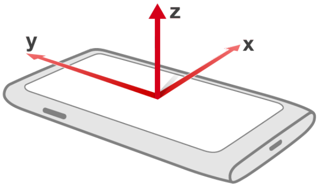

Figure 4.1: *The coordinate system used in JavaScript[Dixit, 2012]*

4.1.1 Accelerometer in JavaScript

To get measurements from the accelerometer an event listener called `devicemotion` is added. The output from measurements is the acceleration force in m/s^2 according to x-, y- and z-axes as in 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 with `accelerationIncludingGravity` and x=0,y=0 and z=-G for the `acceleration`. [Block and Popescu, 2011]
The rotation rate of the device is also available from the `devicemotion`, that is the acceleration (m/s^2) around the axes as seen in figure 4.2.

The JavaScript for measurements of the accelerometer:

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

[Dixit, 2012]

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.

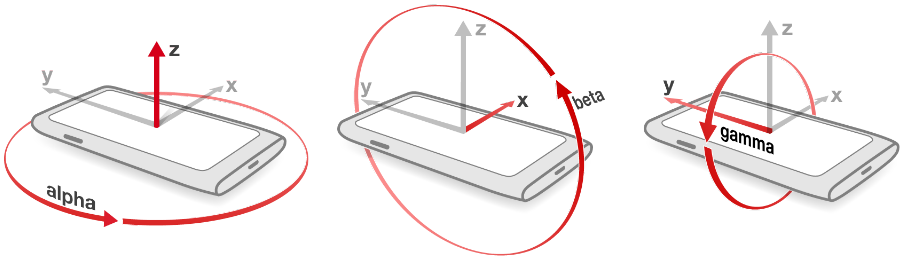

Figure 4.2: *The device rotation axes for the JavaScript* `DeviceOrientation`

Alpha is measured in the range of 0°to 360°around the z-axis, beta in in the range of -180°to 180°around x-axis and gamma in the range of -90°to 90°around y-axis.[Dixit, 2012]

```
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 first measurement had the purpose to test the accelerometer with and without the impact of gravity. The purpose was to see 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 where completely diverse in sense of device platform and only required a browser installed and Internet connection. This only require that the measured device has Internet connection and a browser installed, no additional installations and completely cross-platform.

The measurements required that the device where 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 below shows the web-page while measuring and the right one when finished and ready to send.

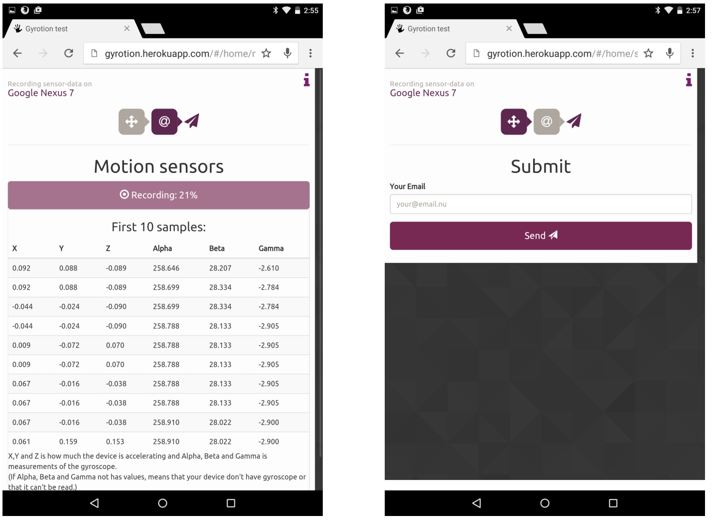

__Figure 4.3:__ Screen-shots of web-page during accelerometer measurements in test I

4.3 Measurement II - Motion

The second measurements were also performed from a web-page using JavaScript to collect gyroscope and accelerometer data and a file-upload to collect measurements from the camera of the device. As of the result in last test there where 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 test (section 4.2).
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. Accelerometer listener used is only `accelerationIncludingGravity`, due to results described in section 5.2.
5. Added a listener for the gyroscope, section 4.1.2.
6. Collecting camera sensor by a five seconds black video, section 4.4.

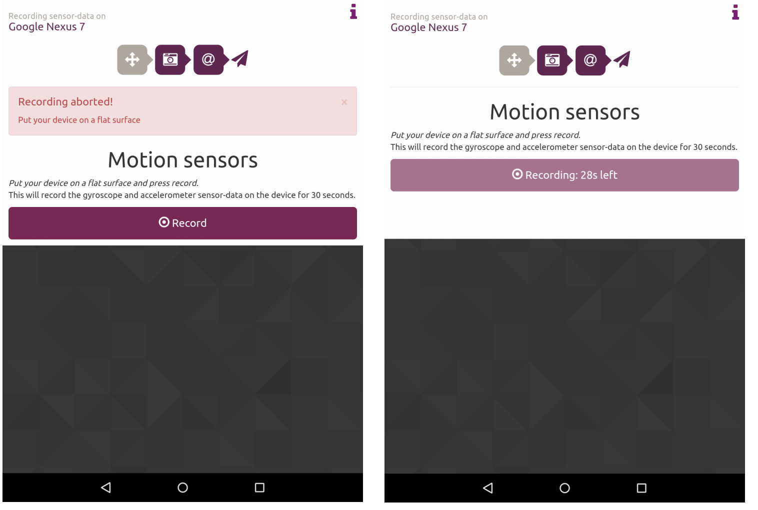

Figure 4.4: *Motion sensor measurements II on a Google Nexus 7*

4.4 Camera measurements

The research found on identifying a camera based on pictures has been in forensic purposes. The difference with forensics and to be used in an authentication to a system or application is that there are tighter limits to the time of an identification. Integrity is also a factor that comes into play for the system should 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 where gathered. In both cases was the device put on a flat surface which makes the camera result black. Both of this measurements are analysed by the PRNU-method used in [Jenkins, 2009] described in section 5.4.

Collecting I - Black video:
The recommended number of pictures for camera fingerprinting is 50 [Jenkins, 2009]. That is not a convenient 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 for 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:

Simple as taking 10 pictures, also with the camera pointing down on a flat surface. Since [Jenkins, 2009] were using pictures of diverse motive this aims to investigate if there may be enough with 10 pictures when the motive is the same.

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

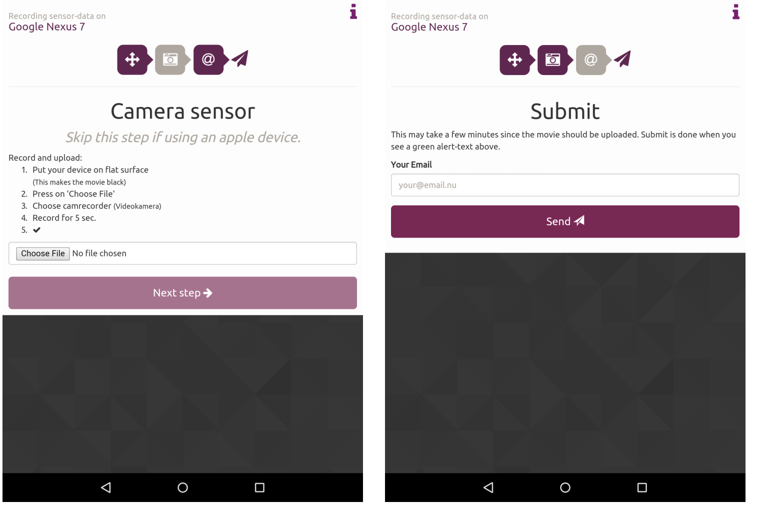

Figure 4.5: *Sensor measurements on a Google Nexus 7*

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

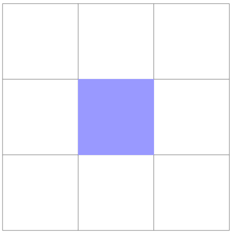

Figure 4.6: *the MATLAB* `medfilt2` *outputs the median of each pixel by its 3-by-3 neighbors*

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

$$corr(\boldsymbol{n}, \boldsymbol{r}) = \frac{(\boldsymbol{n} - \bar{\boldsymbol{n}})(\boldsymbol{r} - \bar{\boldsymbol{r}})}{||\boldsymbol{n} - \bar{\boldsymbol{n}}||||\boldsymbol{r} - \bar{\boldsymbol{r}}||}$$

where $\boldsymbol{n}$ is the reference pattern and $\boldsymbol{r}$ the noise from another picture. [Jenkins, 2009]

5

RESULT OF MEASUREMENTS

This chapter covers the results of the measurements described in chapter 4. The first two sections cover the measurements made on the accelerometer and gyroscope sensor. Third section cover the result of the two camera measurements. This can be seen as one part of *choose biometric characteristics* and also a part of *choose features and matching algorithm*.

5.1 Pre-measurements

To get a hint if accelerometer were a possible fingerprinting candidate some pre-measurements were performed. This were in the early state of the development of the web-page used in measurements I and II. The measurement preformed on six different iPhones showed in figure 5.1 indicates that the accelerometer is a sensor that may be good in fingerprinting purposes.

5.2 Result of measurements I - Motion

The data were gathered as described in section 4.2 from the web-page in figure 4.3 by spreading the the page. This resulted in over a hundred recordings with an FTE on 5%, which had diversity in platforms, brands and models. Since the web-page where spread mostly to company employees the amount of devices with the same model is high as seen in figure 5.3. The purpose of this measurement where to identify if there were differences in terms of bias characteristics between the JavaScripts `accelerationIncludingGravity` and `acceleration`. The result of the measurements can be showed by making scatter-plots of the output acceleration of the devices. Shown in the figure 5.3 the *Sony Xperia* devices represents more than a fifth of the total devices in the measurement. The result of

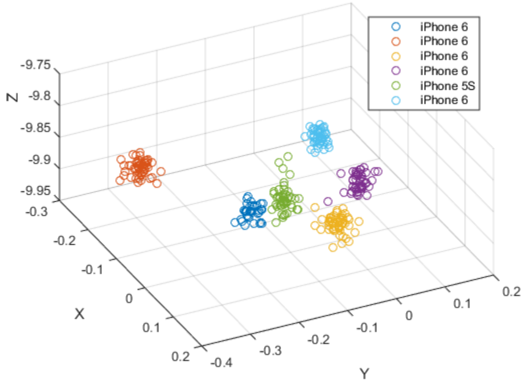

Figure 5.1: *Scatter-plot on accelerometer recordings of 6 Apple devices*

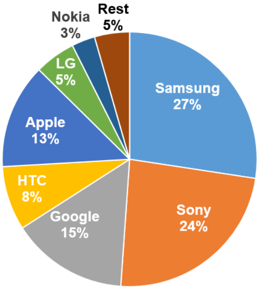

Figure 5.2: *Diversity of device brand sampled in measurements I*

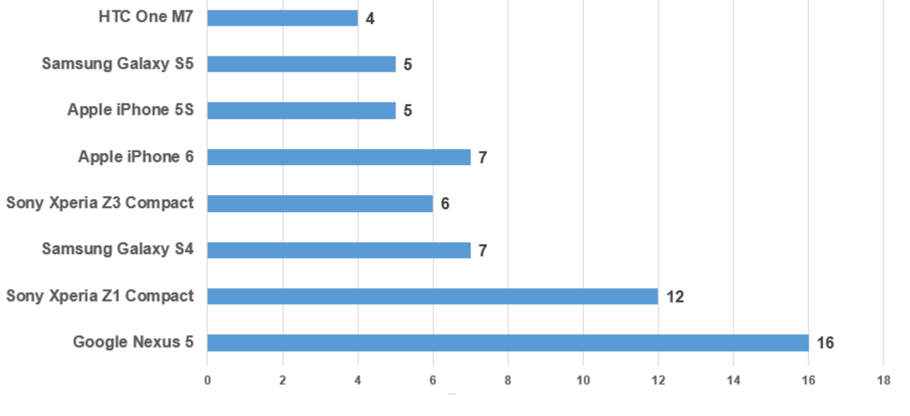

Figure 5.3: *Most common devices models in measurements I*

scatter-plots of measurements of 12 *Sony Xperia* devices with and without gravity in accelerometer readings is shown in figure 5.4 and figure 5.5.

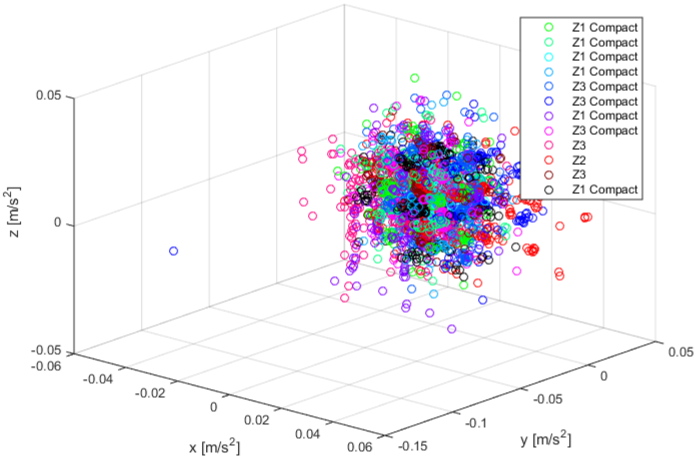

Figure 5.4: *Bias from twelve Sony Xperia devices measured with JavaScripts* `acceleration`

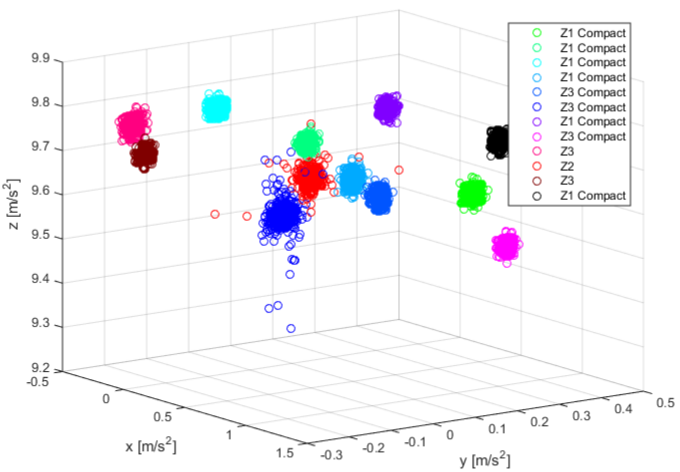

Figure 5.5: *Bias from twelve Sony Xperia devices measured with JavaScripts* `accelerationIncludingGravity`

5.3 Result of measurements II - Motion

The result here is an analyses of the gyroscope and accelerometer data collected from 60 devices with an FTE of 2% by an improved version of the JavaScript webpage used in measurements I. The changes that were made is described in section 4.3 to improve that analyze 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 is permanence described in section 2.4.5, that is the trait not changing significantly over time. To test this 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 that data is collected from. 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 on this measurements among more devices the

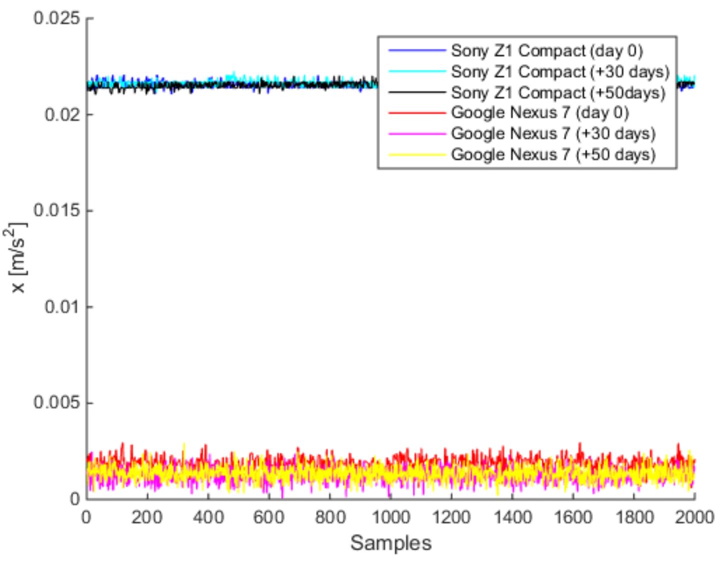

Figure 5.6: *Accelerometer readings of x-axes on a Sony Xperia Z1 Compact and a Google Nexus 7 over 50 days*

scatter-plot 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.

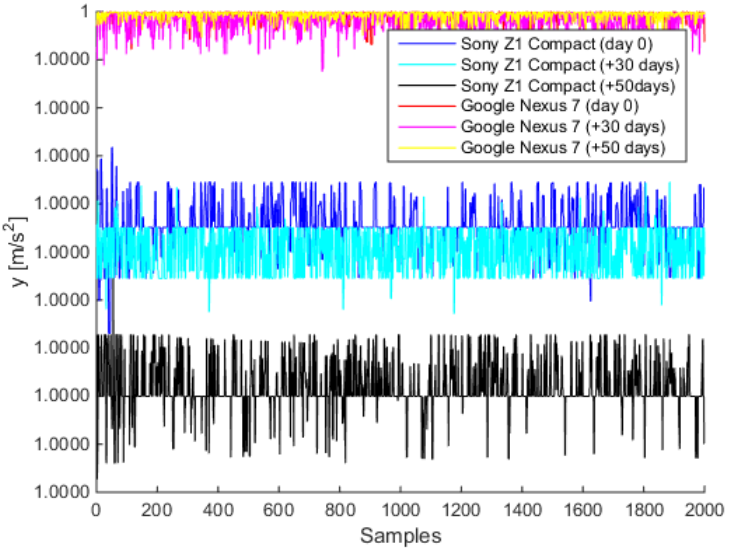

Figure 5.7: *Accelerometer readings of y-axes on a Sony Xperia Z1 Compact and a Google Nexus 7 over 50 days*

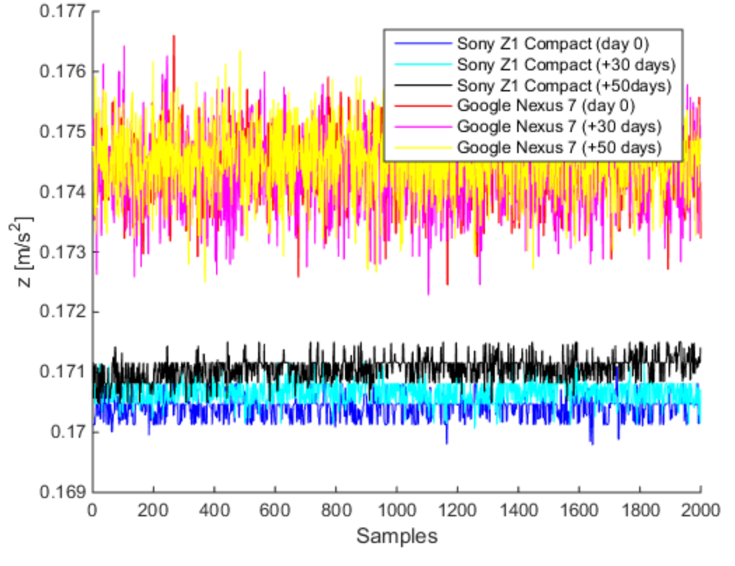

Figure 5.8: *Accelerometer readings of z-axes on a Sony Xperia Z1 Compact and a Google Nexus 7 over 50 days*

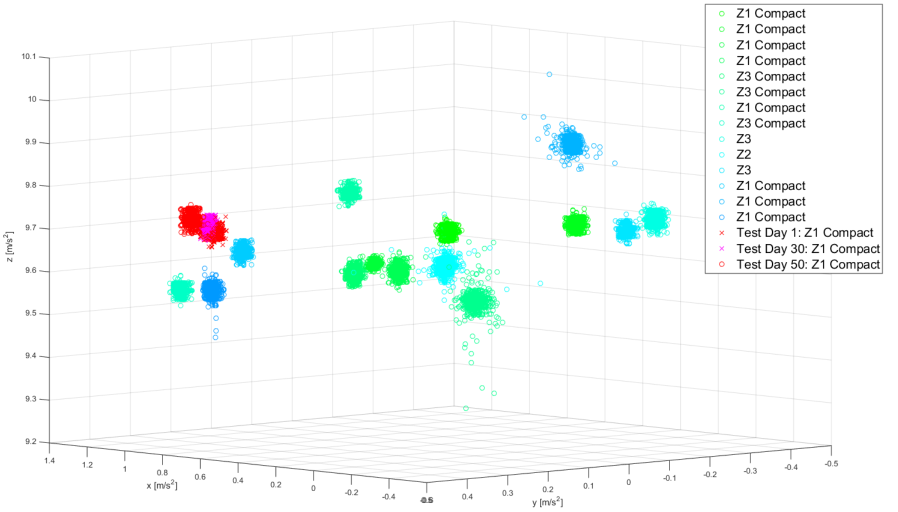

Figure 5.9: *Scatter-plot of accelerometer readings Sony Xperia-device, one of them with measurements performed on the same device with 50 days apart.*

5.3.2 Features of accelerometer data

As in Dey et al. [2014] I used statistical features calculated by the time domain. The features used and calculated as followed: To compare these features and

Feature Name	Description
Mean	$\bar{x} = \frac{1}{N} \sum_{i=1}^{N} x(i)$
Std-Dev	$\sigma = \sqrt{\frac{1}{N-1} \sum_{i=1}^{N} (x(i) - \bar{x})^2}$
Average Deviation	$D_{\bar{x}} = \frac{1}{N} \sum_{i=1}^{N} \lvert x(i) - \bar{x} \rvert$
Skewness	$\gamma = \frac{1}{N} \sum_{i=1}^{N} \left(\frac{(x(i) - \bar{x})}{\sigma} \right)^3$
Kurtosis	$\beta = \frac{1}{N} \sum_{i=1}^{N} \left(\frac{(x(i) - \bar{x})}{\sigma} \right)^4 - 3$
RMS Amplitude	$A = \sqrt{\frac{1}{N} \sum_{i=1}^{N} (x(i))^2}$
Lowest Value	$L = (Min(x(i))\rvert_{i=1 \ to \ N})$
Highest Value	$H = (Max(x(i))\rvert_{i=1 \ to \ N})$

Figure 5.10: *Calculations of statistical accelerometer features. From [Dey et al., 2014, p.6]*

get a picture of if any of them are good for fingerprinting plots of devices were made. These can be found in appendix A. The chosen devices for the plots is the twelve *Sony Xperia Z*-devices including the *Sony Xperia Z1 Compact* that have measurements over 50 days. In these graphs it is possible to see that medium, min, max and the RMS paragraphs *Sony Xperia Z1 Compact* measurements still are right gathered from the other device. Standard deviation looks to separate a bit more and kurtosis, and skewness means deviation looks nothing like being concerned.
In order to compare the properties that are 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. From the distances are used so minimum and the median value 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. Also adding the median as a feature since mean-value can be unreliable if there are outliers. As seen in table 5.1 the values strengthens the result read from appendix A.

Minimum distance						
	Mean	RMS	Std.dev.	Min	Max	Median
All	0,018	0,0193	0,0001	0,0287	0,0365	0
Z1Comp	0,0171	0,0171	0,0002	0,0224	0,0144	0,0175
	95%	89%	200%	78%	39%	
Nexus7	0,0237	0,0182	0,0008	0,0267	0,0119	0,0225
	132%	94%	4%	93%	33%	
Median distance						
	Mean	RMS	Std.dev.	Min	Max	Median
All	0,7934	0,3925	0,0202	0,89	0,9199	0,7953
Z1Comp	0,0519	0,0519	0,0009	0,0447	0,054	0,0575
	7%	13%	690%	5%	6%	7%
Nexus7	0,0285	0,0275	0,0019	0,0361	0,0302	0,0283
	4%	7%	10%	4%	3%	4%

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 analysis of the measurement values as for the accelerometer has been done with the gyroscope. Since the output of the measurements is in degrees and as described 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. The output first is calculated trough sinus, cosine and tangent, ($\alpha = sin(alpha), \beta = cos(beta), \gamma = tan(gamma)$). 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 non-bias sensor. That however did not result in the same stability in permanence as seen in table 5.2.

	Mean	Std.dev.	RMS	Min	Max
Minimum distance					
All	0,000188	1,31E-05	0,000112	2,63E-05	0
Z1Comp	0,00924	0,001157	0,00896	0,009478	0,001348
Z1Comp/all	«100%	«100%	«100%	«100%	
Nexus7	0,006013	0,003204	0,006512	0,000738	0,000126
Nexus7/all	«100%	«100%	«100%	«100%	
Median distance					
All	0,019079	0,005938	0,016074	0,012646	0,007945
Z1Comp	0,00924	0,001157	0,00896	0,009478	0,001348
Z1Comp/all	48%	19%	56%	75%	17%
Nexus7	0,006013	0,003204	0,006512	0,000738	0,000126
Nexus7/all	32%	54%	41%	6%	2%

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 a 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 and compared in table 5.3 as the time features of the gyroscope and accelerometer. If the variance stays the same between measurements for each device it would be a good fingerprinting feature. As read from the table 5.3 is not the Allan variance 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 does not make the Allan variance to a candidate of a fingerprinting feature for the motion sensors.

Minimum distance	All	Z1Comp	Nexus7	Z1C./All	Nex./All
Accelerometer	2,28E-14	9,06E-14	1,02E-12	«100%	«100%
Gyroscope	1,91E-19	2,85E-17	2,57E-17	«100%	«100%
Median distance	All	Z1Comp	Nexus7	Z1C./All	Nex./All
Accelerometer	3,64E-12	3,57E-13	4,96E-12	10%	< 100%
Gyroscope	1,68E-16	4,17E-17	1,44E-16	25%	86%

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 that the features above is way of fingerprinting devices a simulation were performed in MATLAB. The concept is that a fingerprint of all devices is calculated. It contains the features described in section 5.3.2 that resulted in the most stable values over time, thus min, max, mean and RMS. The code for making a fingerprint can be found in B (listing B.1).

When a new measurement is sent in to the simulation, features are calculated and compared to the once 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 smallest 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 chosen devices is the input device there is a match. The code for the simulation can be found in B (listing B.2).
As in biometric system the threshold decides how far from the values an input can be and sill be a match. This threshold creates a rate of match error in the system called FRR and FAR (see section 2.4.4). There are two numbers that can be changed in the simulation that affects the error rates that is `th1` and `th2`. The result of these changed values is presented in table 5.4.

5.4 Result Camera-measurements

For the test of the camera sensor the PRNU value is calculated as an approximation of the algorithm described in section 4.4 and also used by Jenkins [2009].

FRR	th1/th2	1	2	3
	1	2,27%	8,62%	29,55%
	2	20,45%	20,45%	29,55%
	3	34,09%	34,09%	38,64%
FAR	th/F<	1	2	3
	1	0,00%	0,00%	0,00%
	2	0,89%	0,45%	0,43%
	3	1,77%	0,86%	0,44%

Table 5.4: *The FAR and FRR of the MATLAB simulation when changing threshold values* `th1` *and* `th2` *see appendix B*

5.4.1 Result of camera measurement I

Since the purpose of this thesis compared to earlier work (section 3.5) has the purpose of authentication and not forensics, is convenience for the collecting and measurability a factor to take in account. That is why the first experiment is asked the users to record a 5 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 takes a lot of 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. The MATLAB code for this is:

```
% 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 heigt 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 to get the noise is done. Then the noise from the reference pictures is compared to all collected PRNU and correlation is calculated like the formula above in 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*

The result of identify an input PRNU with the PRNU from already known devices head a limit on only six devices there only two of them were correctly identified. Since Jenkins [2009] made better result than this, the conclusion that the bad result were 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 Result of camera measurement II

Since the earlier test leaved out some of the PRNU noise when recorded a video instead of taking a picture the new test consist of 10 images from every device. The recommendation from Jenkins [2009] to use at least 50 images is here compensated by again using black images (picture taking with device camera facing down). Since the scene is always the same the noise removal will be better in fewer images. The same code is used as above with the different that the video

to image step is removed. The sizes of the images in this case is better since the camera on the mobile devices by default uses higher resolution when taking a picture then when recording.
The result of this measurements started out good with none false match at five devices, but that number increased rapidly as you can see in table 5.5. As the value grew that quickly no more samples from more devices were gathered.

Devices	FRR	Time [s]
5	0%	15-20s
7	50%	17-26s
10	67%	25-46s

Table 5.5: *False rate and time taken to compare PRNU of camera images.*

6 DISCUSSION

This chapter interweaves the theory and method with the result. What the difference between the theory and result is and why. The limitations of the method used is also discussed.
Discussion followed by conclusion in the view of a biometric system can be seen as *choose features and matching algorithm* and also *choosing feature and matching algorithm*.

6.1 Accelerometer

6.1.1 Result

The result of the first measurements of the accelerometer resulted in some unexpected result, the fact that JavaScripts listener without gravity does not seem to have any static noise at all. The reason could be some software modification of the sensor data before it reaches the `event`. The recommendation from MEMS accelerometer manufacturers is to calibrate the sensors. [Kionix, 2007]
Doing some research on Android sensors revield that their `SensorEvent` also have 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 m/s^2.
`TYPE_LINEAR_ACCELERATION` is without gravity but a combination of hardware and software sensor. Where as `TYPE_ACCELEROMETER` comes the measurements only from hardware. But there have been some bias removal from the sensor such bias from different temperature. [Android, 2015]
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. [Block and Popescu, 2011] [Mozilla, 2015]
As a result of measurements I the used measurement of the accelerometer in 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 its OS 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 noticeable 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 more 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. For 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 doing some calibration with the sensor that effects the result if not dealing with raw data. But as also 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 been hard since manual calibration of noise caused be temperature had to be done.

6.2 Gyroscope

Discussion about the result of the gyroscope measurements and the method used to get gyroscope data from the mobile device.

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 result.
The other method where calculating the Allan variance that is used for calibration of gyroscope, did not give the expected result. 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 that has been done before the developer get the measurement data, which makes that part harder. That the gyroscope is much more sensitive than the accelerometer can be seen when 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 in account before the constant noise 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.
The Android developer page about sensor events state that they make factory calibration and temperature compensation even on their uncalibrated sensor events (magnetometer and gyroscope). That is relativity new feature added in Android 4.3 Jelly Bean (API level 18 McEntegart [2013]) from 2013 but the original once used since Android 1.5 Cupcake (API level 3 Ducrohet [2003]) from 2009 makes some more noise compensation and calibration. What kind of compensation and calibration done is not public. [Android, 2015]
Since the output of both the calibrated and uncalibrated sensor is in rad/s implies that it could be some software calibration in the date, not knowing where it is done.

6.2.2 Method

The method 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 McEntegart [2013]. This rawer data may result in better feature values in time domain or Allan variance.

6.3 Camera

This section discusses the result and method used for evaluating the camera sensor as a fingerprinting characteristic.

6.3.1 Result

The result of the camera sensor were not as good as expected or as good as in the research by [Jenkins, 2009] 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. [Boström, 2009] This is about the average smart-phones camera today, but some other specifications may have other 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 just send. It generated 100-200 which also made an 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 sensor of the device. To begin with 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 have an advantage in terms of user-ability, but as valuable your personal information is today for the commercial and advertising it's hard to set a value for something that could identify you everywhere on the Internet. The tracking possibilities are enormous and have to be concerned if this type of identifying can be done.

There are of course some good things in the view of ethical and societal aspects. If the 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 send signals to your thermostat, or that only your mobile device that can unlock the front door.

The point here is that fingerprinting features of a device should be treated in the same way as your biometric trait. This means that you want to have control over were your biometric trait is used. Most of us think that it is legit that 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. The concept should be considered when fingerprinting a device as well. The accelerometer data may be applicable to use by banks, to your door or car. But you may not want as a login feature to a commercial site that may sell that information.

7 CONCLUSIONS

This chapter will reconnect to the aim and objectives of the thesis. In comparison to designing a biometric system this would be the part final of *choosing feature and matching algorithm*.

7.1 Choose of characteristics

In the selection of characteristics, there are seven different factors that must be considered (described in section 2.4.5). The sensors of the thesis are compared to decide which sensor*(/s)* that 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 (code on GitHub [Karlsson]). This 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 was the best choice of uniqueness since the FAR is zero when both threshold values is set to one. The FRR were that 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 high uniqueness. [Celiktutan et al., 2008]

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 less than once a month further testing is recommended. Also some kind of machine learning of the drift of bias would be preferable as used by [Dey et al., 2014].
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. [Jenkins, 2009]

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 do 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''

That is a Stanford security research proving 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. [Michalevsky et al., 2014b]
The conclusion is that it is acceptable of the majority of people today but may be

would 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 a authentication system would not be acceptable.
To conclude this all the sensors is probably social accepted to use for authentication today. The question is what could happen 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 of that. 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).

Characteristics\Sensor	Accelerometer	Gyroscope	Camera
Universality	Good	Good	Good
Uniqueness	Good	Bad	*
Permanence	Good	Bad	Good
Measurability	Good	Good	Bad
Performance	Good	Good	Bad
Acceptability	*	*	Good
Circumvention	Good	Good	Good

Table 7.1: *Conclusions about the factors of choosing fingerprint sensor. (Factors from biometric characteristics see section 2.4.5)*
**See explanation respective title above.*

7.2 Further work

When 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. This work would contain more devices, and therefore check the scalability of using an accelerometer. 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. Not knowing of any malicious devices today, and therefore 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 noise would be a area to explore and evaluate if they have lower rates of FAR and FRR or is 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

A

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

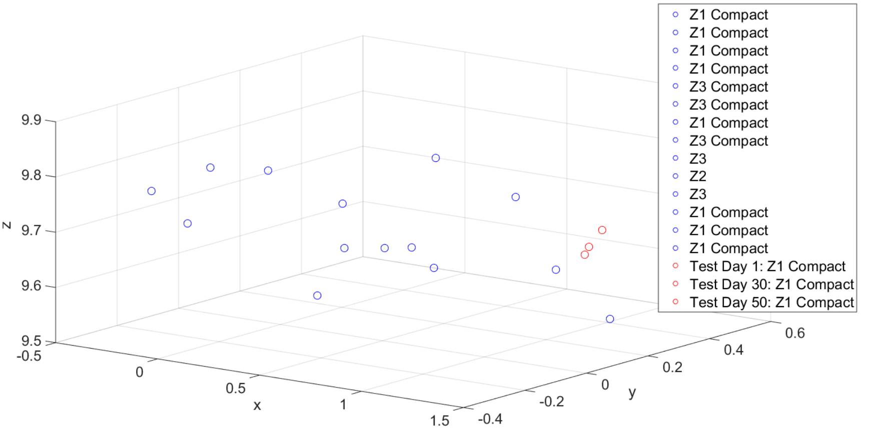

Figure A.1: *Scatter-plot of mean values of 12 Sony Xperia Z-devices including one device with readings over a period of 50 days*

Scatter-plot of standard deviation values

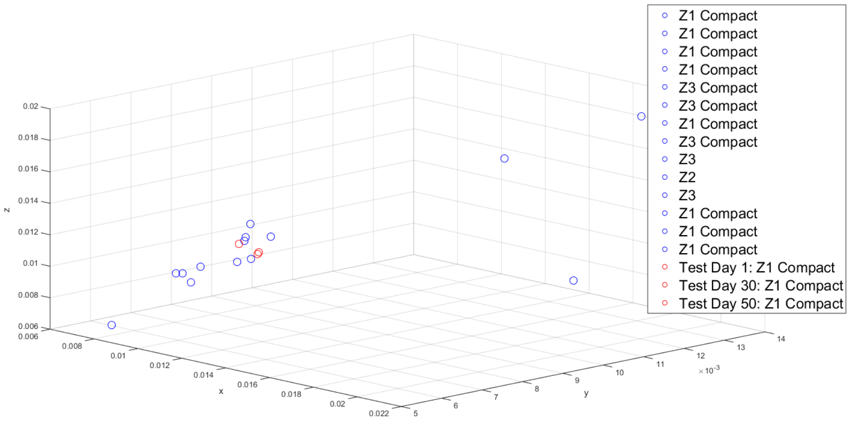

Figure A.2: *Scatter-plot of standard deviation values of 12 Sony Xperia Z-devices including one device with readings over a period of 50 days*

Scatter-plot of average deviation values

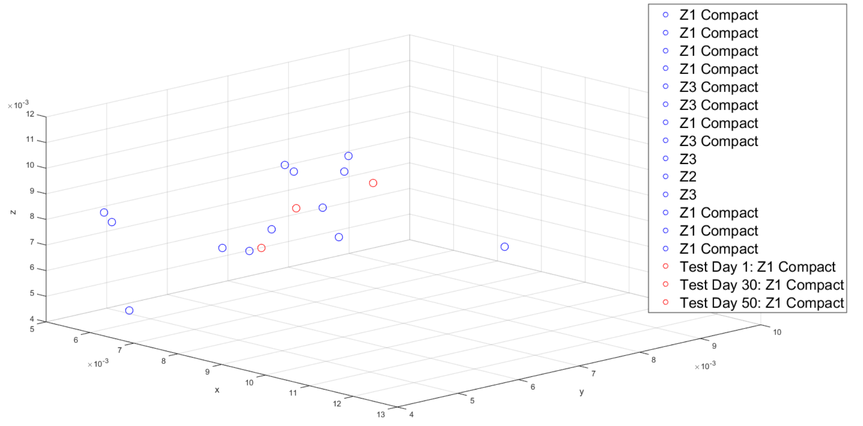

Figure A.3: *Scatter-plot of average deviation values of 12 Sony Xperia Z-devices including one device with readings over a period of 50 days*

Scatter-plot of skewness values

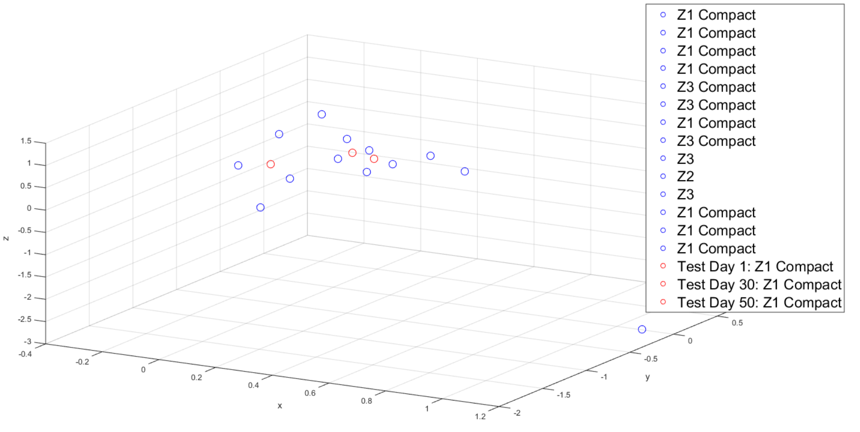

Figure A.4: *Scatter-plot of skewness value of 12 Sony Xperia Z-devices including one device with readings over a period of 50 days*

Scatter-plot of kurtosis values

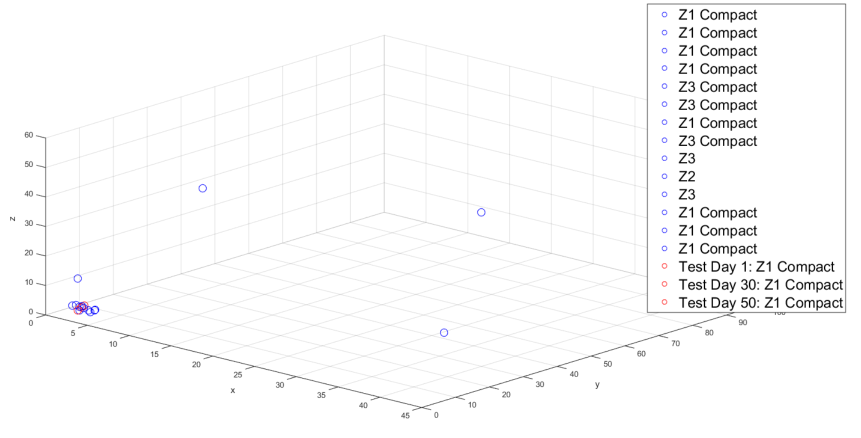

Figure A.5: *Scatter-plot of kurtosis values of 12 Sony Xperia Z-devices including one device with readings over a period of 50 days*

Scatter-plot of RMS values

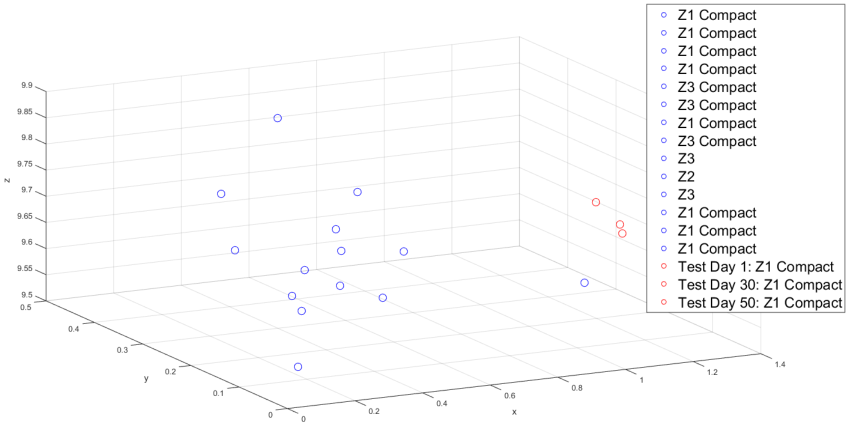

Figure A.6: *Scatter-plot of RMS values of 12 Sony Xperia Z-devices including one device with readings over a period of 50 days*

Scatter-plot of min values

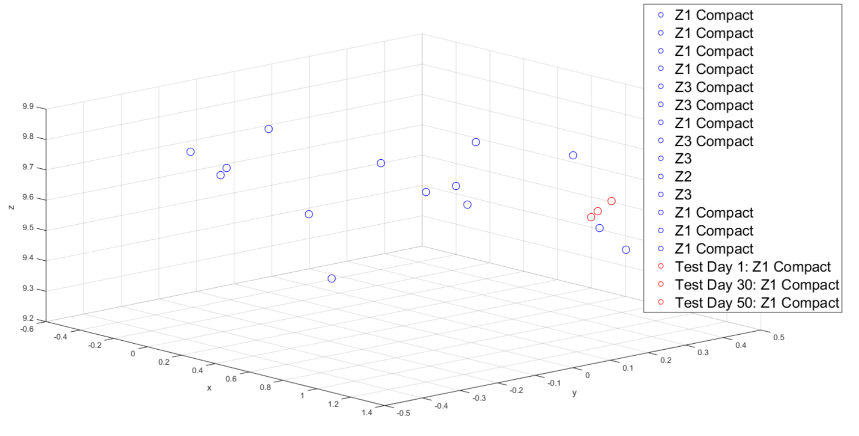

Figure A.7: *Scatter-plot of min values of 12 Sony Xperia Z-devices including one device with readings over a period of 50 days*

Scatter-plot of max values

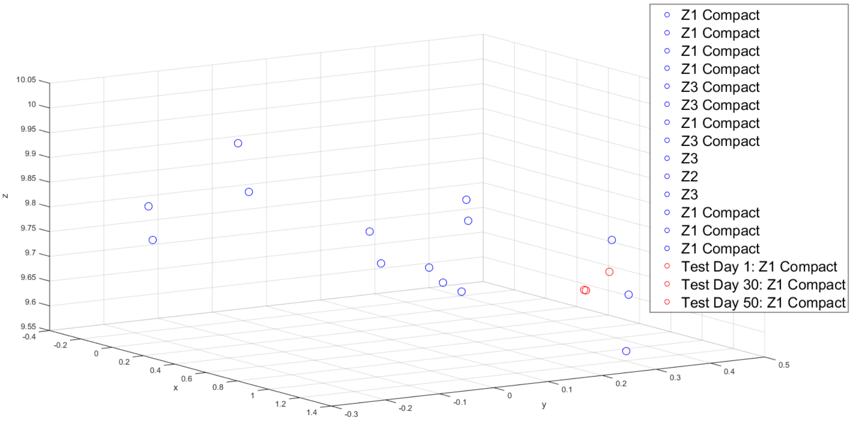

Figure A.8: *Scatter-plot of max value of 12 Sony Xperia Z-devices including one device with readings over a period of 50 days*

B

MATLAB accelerometer fingerprinting simulation

```
function fingerprint_calc(device_id)
%FINGERPRINT_CALC recivce the device id and save finerprint in a
    mat-file
%  The CSV-file is recived 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 acclerometer data input
```

```
%  The input file is a CSV-file with acclereometer 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*

C

Example of CSV-file of measureing 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.

time	alpha	beta	gamma	x	y	z
1,4271E+12	286,4273	0,9896	-7,2886	1,2253	0,1466	9,6552
1,4271E+12	286,4273	0,9896	-7,2886	1,2322	0,1473	9,6552
1,4271E+12	286,4273	0,9896	-7,2886	1,2322	0,1473	9,6552
1,4271E+12	286,4273	0,9896	-7,2886	1,2322	0,1473	9,6552
1,4271E+12	286,4273	0,9896	-7,2886	1,2444	0,1791	9,6552
1,4271E+12	286,4273	0,9896	-7,2886	1,2444	0,1791	9,6552
1,4271E+12	286,4273	0,9896	-7,2886	1,2444	0,1791	9,6552
1,4271E+12	286,4273	0,9896	-7,2886	1,2444	0,1987	9,6631
1,4271E+12	286,4273	0,9896	-7,2886	1,2444	0,1987	9,6631
1,4271E+12	286,4273	0,9896	-7,2886	1,2444	0,1987	9,6631
1,4271E+12	286,4273	0,9896	-7,2886	1,2353	0,1918	9,6744
1,4271E+12	286,4273	0,9896	-7,2886	1,2353	0,1918	9,6744
1,4271E+12	286,4273	0,9896	-7,2886	1,2353	0,1918	9,6744
1,4271E+12	286,4273	0,9896	-7,2886	1,2444	0,1418	9,6744
1,4271E+12	286,4273	0,9896	-7,2886	1,2444	0,1418	9,6744
1,4271E+12	286,4273	0,9896	-7,2886	1,2444	0,1418	9,6744
1,4271E+12	286,4273	0,9896	-7,2886	1,2556	0,1556	9,6633
1,4271E+12	286,4273	0,9896	-7,2886	1,2556	0,1556	9,6633
1,4271E+12	286,4273	0,9896	-7,2886	1,2556	0,1556	9,6633
1,4271E+12	286,4273	0,9896	-7,2886	1,2273	0,1717	9,6673
1,4271E+12	286,4273	0,9896	-7,2886	1,2273	0,1717	9,6673
1,4271E+12	286,4273	0,9896	-7,2886	1,2273	0,1717	9,6673
1,4271E+12	286,4273	0,9896	-7,2886	1,2193	0,1849	9,6744
1,4271E+12	286,4273	0,9896	-7,2886	1,2193	0,1849	9,6744
1,4271E+12	286,4273	0,9896	-7,2886	1,2253	0,1909	9,6604
1,4271E+12	286,4273	0,9896	-7,2886	1,2253	0,1909	9,6604
1,4271E+12	286,4273	0,9896	-7,2886	1,2253	0,1909	9,6604
1,4271E+12	286,4273	0,9896	-7,2886	1,2253	0,1909	9,6847
1,4271E+12	286,4273	0,9896	-7,2886	1,2253	0,1909	9,6847
1,4271E+12	286,4273	0,9896	-7,2886	1,2253	0,1909	9,6847
1,4271E+12	286,4273	0,9896	-7,2886	1,2558	0,1756	9,7088
1,4271E+12	286,4273	0,9896	-7,2886	1,2558	0,1756	9,7088
1,4271E+12	286,4273	0,9896	-7,2886	1,2636	0,1554	9,6801
1,4271E+12	286,4273	0,9896	-7,2886	1,2636	0,1554	9,6801
1,4271E+12	286,4273	0,9896	-7,2886	1,2636	0,1554	9,6801
1,4271E+12	286,4273	0,9896	-7,2886	1,2464	0,1697	9,6572
1,4271E+12	286,4273	0,9896	-7,2886	1,2464	0,1697	9,6572
1,4271E+12	286,4273	0,9896	-7,2886	1,2444	0,1897	9,6911
1,4271E+12	286,4273	0,9896	-7,2886	1,2444	0,1897	9,6911
1,4271E+12	286,4273	0,9896	-7,2886	1,2256	0,2097	9,6747
1,4271E+12	286,4273	0,9896	-7,2886	1,2256	0,2097	9,6747
1,4271E+12	286,4273	0,9896	-7,2886	1,2256	0,2097	9,6747
1,4271E+12	286,4273	0,9896	-7,2886	1,2256	0,2097	9,6747
1,4271E+12	286,4273	0,9896	-7,2886	1,2256	0,2097	9,6747
1,4271E+12	286,4273	0,9896	-7,2886	1,2256	0,2097	9,6747
1,4271E+12	286,4273	0,9896	-7,2886	1,2253	0,1906	9,6555
1,4271E+12	286,4273	0,9896	-7,2886	1,2253	0,1906	9,6555
1,4271E+12	286,4273	0,9896	-7,2886	1,2253	0,1693	9,6756
1,4271E+12	286,4273	0,9896	-7,2886	1,2253	0,1693	9,6756
1,4271E+12	286,4273	0,9896	-7,2886	1,2471	0,1361	9,7046

Bibliography

D. Allan. Statistics of atomic frequency standards. In *Proceedings of IEEE*. Cited on page 16.

R. Anderson. *Security Engineering: A Guide to Building Dependable Distributed Systems*. Wiley, 2008. ISBN 9780470068526. Cited on pages 5, 6, and 7.

Open Source Project Android. SensorEvent, April 2015. URL `http://developer.android.com/reference/android/hardware/SensorEvent.html`. [Online; 15-April-2015]. Cited on pages 39 and 41.

Steve Block and Andrei Popescu. Deviceorientation event specification. *W3C Working Draft*, December 2011. URL `http://www.w3.org/TR/orientation-event/`. Cited on pages 20 and 40.

Hristo Bojinov, Yan Michalevsky, Gabi Nakibly, and Dan Boneh. Mobile device identification via sensor fingerprinting. *CoRR*, abs/1408.1416, 2014. URL `http://arxiv.org/abs/1408.1416`. Cited on pages 16 and 17.

Martin Boström. Digitalkamera - test. October 2009. URL `http://www.gp.se/konsument/tester/1.230520-digitalkamera-test`. [Online; 13-May-2015]. Cited on page 42.

D. Boswarthick, O. Elloumi, and O. Hersent. *M2M Communications: A Systems Approach*. Wiley, 2012. ISBN 9781119994756. Cited on page 8.

O Celiktutan, B. Sankur, and I. Avcibas. Blind identification of source cell-phone model. *IEEE Transactions on: Information Forensics and Security*, 3:553—-566, August 2008. ISSN 1556-6013. Cited on pages 17 and 46.

Anupam Das, Nikita Borisov, and Matthew Caesar. Fingerprinting smart devices through embedded acoustic components. *CoRR*, abs/1403.3366, 2014. URL `http://arxiv.org/abs/1403.3366`. Cited on pages xi and 13.

S. Dey, N. Roy, W. Xu, R. Choudhury, and S. Nelakuditi. Accelprint: Imperfections of accelerometers make smartphones trackable. Briefing Papers UCAM-CL-TR-696, University of Illinois and University of South Carolina, February 2014. Cited on pages xi, 16, 17, 32, and 46.

Shwetank Dixit. The w3c device orientation api: Detecting orientation and acceleration. *dev.opera.com/articles/*, July 2012. URL `https://dev.opera.com/articles/w3c-device-orientation-api/`. Cited on pages xi, 20, and 21.

Xavier Ducrohet. Android 1.5 is here!, April 2003. URL `http://android-developers.blogspot.se/2009/04/android-15-is-here.html`. [Online; 12-May-2015]. Cited on page 41.

Dinghua He. Remote authentication of software based on machine's fingerprint. In *Software Engineering and Service Science (ICSESS), 2012 IEEE 3rd International Conference on*, pages 543–546, June 2012. doi: 10.1109/ICSESS.2012.6269524. Cited on page 8.

Anil.K. Jain, Arun.A. Ross, and K. Nandakumar. *Introduction to Biometrics*. SpringerLink : Bücher. Springer, 2011. ISBN 9780387773261. Cited on pages 9, 10, 11, and 12.

Neil Jenkins. Digital camera identification. Technical report, Forensic Signal Analysis, University of Cambridge, November 2009. URL `https://www.cl.cam.ac.uk/teaching/0910/R08/work/essay-nmj27-cameraid.pdf`. Cited on pages 15, 17, 18, 23, 24, 25, 35, 37, 42, and 46.

Anna Karlsson. Source code of motion measurements. Cited on page 45.

Kionix. Accelerometer Errors. (AN 012), May 2007. URL `http://www.kionix.com/sites/default/files/AN012%20Accelerometer%20Errors.pdf`. Cited on page 39.

Mark Looney. A simple calibration for mems gyroscopes. Technical article, Analog Devices. Cited on page 16.

Jane McEntegart. Google Announces Android 4.3, Update Rolling Out Today. 2013. URL `http://www.tomshardware.com/news/Android-4.3-Update-Roll-out-Release-Nexus,23671.html`. Cited on pages 41 and 42.

Yan Michalevsky, Dan Boneh, and Gabi Nakibly. Gyrophone: Recognizing speech from gyroscope signals. In *23rd USENIX Security Symposium (USENIX Security 14)*, pages 1053–1067, San Diego, CA, August 2014a. USENIX Association. ISBN 978-1-931971-15-7. URL `https://www.usenix.org/conference/usenixsecurity14/technical-sessions/presentation/michalevsky`. Cited on page 16.

Yan Michalevsky, Dan Boneh, and Gabi Nakibly. Gyrophone: Recognizing speech from gyroscope signals. In *23rd USENIX Security Symposium (USENIX Security 14)*, pages 1053–1067, San Diego, CA, August 2014b. USENIX Association. ISBN 978-1-931971-15-7. URL https://www.usenix.org/conference/usenixsecurity14/technical-sessions/presentation/michalevsky. Cited on page 46.

Contributors Mozilla. DeviceMotionEvent.accelerationIncludingGravity, February 2015. URL https://developer.mozilla.org/en-US/docs/Web/API/DeviceMotionEvent/accelerationIncludingGravity. [Online; 24-Feb-2015]. Cited on page 40.

C. Polsonetti. Understand the difference between iot and m2m. *Chemical Processing*, April 2014. URL www.chemicalprocessing.com/articles/2014/understand-the-difference-between-iot-and-m2m/. [Online; posted 24-April-2014]. Cited on page 8.

Wei Ren, Linchen Yu, Liangli Ma, and Yi Ren. How to authenticate a device? formal authentication models for m2m communications defending against ghost compromising attack. *International Journal of Distributed Sensor Networks*, 2013(Article ID 679450):9, 2013. Cited on pages 5 and 9.

Angel Rodriguez and Ubejd Shala. Indoor positioning using sensor-fusion in android devices. Master's thesis, Kristianstad University, September 2011. Cited on pages 14 and 15.

Technologies VectorNav. Gyroscope. Cited on page 16.

Oliver J. Woodman. An introduction to inertial navigation. Technical report UCAM-CL-TR-696, University of Cambridge, Computer Laboratory, August 2007. Cited on page 14.

Linköpings universitet