Institutionen för systemteknik
Department of Electrical Engineering

Examensarbete

Two factor authentication in M2M

Fingerprinting of mobile devices for making a two factor authentication between the devices

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

Anna Karlsson

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

Linköping 2015

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

Linköpings tekniska högskola
Linköpings universitet
581 83 Linköping

Two factor authentication in M2M

Fingerprinting of mobile devices for making a two factor authentication between the devices

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, 12 juni 2015

Avdelning, Institution
Division, Department

Information Coding
Department of Electrical Engineering
SE-581 83 Linköping

Datum
Date

2015-06-12

Språk
Language

☐ Svenska/Swedish
☒ Engelska/English

☐ _____

Rapporttyp
Report category

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

☐ _____

ISBN
__

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

Serietitel och serienummer
Title of series, numbering

ISSN
__

URL för elektronisk version

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

Titel
Title

Tvåfaktorauthentisering mellan maskiner
Two factor authentication in M2M

Författare
Author

Anna Karlsson

Sammanfattning
Abstract

If your thesis is written in English, the primary abstract would go here while the Swedish abstract would be optional.

Nyckelord
Keywords

computer security, M2M, authentication

Sammanfattning

Sammanfattning är en sammanfattning på svenska...

Abstract

If your thesis is written in English, the primary abstract would go here while the Swedish abstract would be optional.

Acknowledgments

Vi tycker alla har varit så himla goa hela den här långa och tuffa tiden i våra liv.

Linköping, Januari 2020
Anna Karlsson

Contents

Notation

Notation

Notation	Meaning
G	G-force
η	
ϵ	
Θ	
ω	
$\boldsymbol{F}_C$	Coriolis force
T	Tesla (SI-unit of magnetic flux density)

Abbreviations

Abbreviation	Meaning
FAR	False Accept Rate
FRR	False Reject Rate
FTE	Fail To Enrollment
ICT	Information and Communication Technologies
IoT	Internet of Things
MEMS	Micro-electromechanical System
M2M	Machine-to-machine
NIC	Network Interface Card
PRNU	Photo-Response Non-Uniformity noise
RFF	Radio Frequency Fingerprinting
RFID	Radio-Frequency IDentification
FPN	Fixed Pattern Noise
CFA	Color-Filter Array
PCB	Printed Circuit Board
RMS	Root Mean Square

1

INTRODUCTION

This paper is the report for my master thesis in Computer Science and the last part of my education for become an engineer in information-technology in the field of secure systems. The thesis was performed on 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, refrigerator, coffee maker, watches, cat feeder, sewing machines..., the world of connected devices is growing rapidly. This world is known under the term 'Internet of Things'. For making this 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 using two factor authentication when accessing buildings, part of networks, our bank and so on. When talking about two factor authentication we usually use a combination of either three things; something you *know* like passwords, something you *have* like tag, passport, card or phone or something you *are* like iris or fingerprint. (More about those in chapter 2.) Something you know or have is things that can be copied, stolen or modified fairly easy and without know 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 on one person at a time. Machines or devices don't have those attributes as us human, they are build on hardware parts.
The background for this thesis is to explore the possibility for a machine to have a

fingerprint that can be used to more securely identify 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 an malicious user accessing your account.

1.2 Aims & Objectives

Today most of the solutions for M2M authentication involves a certificate, token, UUID etc., from my opinion is this something the machine know or have. The area of fingerprinting a machine 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 in to if the fingerprinting methods found today, can be used as something the machine *are* for two factor authentication between them. The problems I'll work to solve with this thesis is:

> Can you create a device fingerprint by using the unique hardware characteristics in a mobile device?
>
> Is this fingerprint suitable for using as a second factor for authentication between devices?

The problems above state a mobile device and not a general machine, which is one of my limitations in the thesis. When stating a mobile device leaves also leads to wireless network environment. The focus is also set to an authentication process where you are able to collect a set of data from the device in a database in a test environment. This means that new devices in the network has to go trough some kind of phase were collecting the unique hardware characteristics data, 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. As said in the background is a device building stone its hardware and something the devices *has* that is the point of view of the thesis.

The objectives of this work and can be summed up to:

> **Explore different unique hardware characteristics of a mobile device**
> Mobile devices today are equipped with a lot of sensors and since they like other hardware has some noise that may be unique enough to differ from a device of the same model. Measurements from the microphone-, speaker-, gyroscope-, accelerometer- and camera-sensor will be collected and valuated from the view as fingerprints. RFF (Radio Frequency Fingerprinting) is another perspective that also will be measured from the noise of the mobile devices in a wireless environment and compared together with the sensors.
>
> **Combining M2M, two factor and biometric authentication**
> Biometric authentication has ways of measure and compare fingerprints, this measurements and methods will be used to make the two factor authentication between the devices.

1.3 Thesis Outline

==== OBS!! ====
Denna är rätt fel nu, uppdatera det sista som händer... This introduction chapter including background, aims and objectives will give a quick view of what the thesis is about. The chapters that following is divided in different parts that maps to the different objectives listed above.

Ch.1: This will give an introduction to the work done in the thesis and motivation for doing it.

Ch.2: How authentication is made today between machines, two factor and in biometric.

Ch.3: The different unique hardware characteristics of a mobile device that has been found today and how they are collected.

Ch.4: Result of the collected unique hardware characteristics of a mobile device together with comparison and evaluation on if they can be used as mobile fingerprints or not. This chapter also presents the demo the made from the test result.

Ch.5: Conclusion will except conclusions also include ethical aspects and further work.

1.4 Related Work

TODO! or to be removed and covered in the next two chapters...

2

COMMUNICATION & AUTHENTICATION

Because just 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]
In this chapter will show ways of authentication (two factor, M2M and biometric) that is in the area of this thesis. The biometric part is in the area because it has good ways of measure strength in a biometric trait (especially fingerprint) that will be used when comparing strength of my tests 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 *have* like a key, card, passport and so on
- Something the authenticator *knows* for example password
- Something the authenticator *are*, 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 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 have and knows is the most common combination. The

third one, cost is the biggest reason form that biometrics isn't that common yet. [Anderson, 2008, p. 47]

2.2 Challenge-Response authentication

==== OBS!! ====
Om det blir resulatat av android applikationen så kommer den göra en challenge response och då skriver jag lite om vad det innebär här.

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, cause there is a human behind the mobile device calling. The reason for that using mobile devices in this thesis is that they have many hardware parts that can be used for no human communication with other devices (see chapter 3). These hardware parts can be found in other simpler devices such as accelerometer sensor probe that also can be applied on the result.
Often is M2M involving similar devices in the same M2M area network, interacting with an application. This makes it possible for devices to access public networks as well, via a gateway or router. An example is the heating system in smart homes. Devices are not a new thing, but when we have a growing world of IoT devices with very specific characteristics is growing. Thus makes the area of M2M more important to make these devises talk without a human behind. This affecting the requirements on the application and networks dealing with the devices. Characteristics of this devices is listed blow;

Multitude, they say that connected devise not directly interacting with humans, the big part of IoT is soon 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 of connected devices with 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 voltage. Therefor reliability is an important factor.

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

All this 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

Internet-of-Things, meaning to making everything connected to everything in the Internet. IoT is 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 doesn't. The common denominator is according to Polsonetti *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 much more 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]
In this thesis is M2M a subset of IoT, since it always one mobile device that wants to authenticate then it can communicate with other deceives.

2.3.2 M2M authentication

There are no standardized way of authenticate in M2M, but effort is done in the area. An example is [He, 2012] where he based authentication on a machines fingerprint. But this fingerprint isn't of the same character as the one this thesis is focusing on. In his article the fingerprint consist of hardware message of computers, such serial number of CPU, MAC address of network card, Machine ID etc. These things have through the years been proven to bee pretty easy to spoof. There are hundreds of guides of how to do that in many platforms like mobile devices (iPhone [**?**] and Android [**?**]) that is the thesis area.
Like [Ren et al., 2013] that states in their article that "...traditional methods such as "what you know and who you are" may not be applied". But the aim in section 1.2 is to do precisely that and with the advantage that using "regular" authentication that is more tried and tested. Thus the next section will be about biometric systems and how they authentication which is used for who you are.

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 recognized to the system. Buy, showing his or her biometric identifier (fingerprint, iris, DNA, etc.) to the biometric system, thus 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 biometric these characteristics is called *traits*, *indicators*, *identifiers*, *or modalities*, but for for the aim of this thesis will it still be called characteristics. For designing, implementation and evaluation when building a biometric system there are some steps that has to be done;

The first step is to collect biometric data and store it in a database with the users identity. The recognition is then done by again collect biometric data from the user and compared 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 a image, amplitude signal, online signature, odor or chemical-based.
2. *Feature extractor* - first has to make the raw biometric samples comparable, mostly done in three pre-process operations;
    - Quality assessment, is the sample good enough?
    - Segmentation, remove background noise from sample
    - Enhancement, by using an algorithm to improve the sample
3. *Database* - that has the data from the enrollment phase together with some identity data (like name orID). This database should having a 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's 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

Biometrics 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 aim 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 give the sample to the system, which then looks if the user is known to the system or not. The identification look-up takes longer time since you look for all samples in the database and compare them, in authentication you only look for the claimed identity. [Jain et al., 2011, ch. 1]

2.4.4 Measurements

Biometric measurements is a bit more tricky than in a password-based system where the answer just is 'match' or 'no match'. The accuracy of the biometric system must be consider when you choose characteristics. This is measured by two rates (False Reject Rate) that is the probability that two samples from the same user is not a match and (False Accept Rate) is the probability that two samples from different users is a match. A match is decided authentic between two samples from the same user is high enough and as a *impostor* is there is similarity between two samples from different users.
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. Which 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 Design a biometric system

When designing a biometric system it is done in a five activity cycle. Depending on the outcome of one activity, the next step could be forward or redoing earlier activity. The design cycle is represented as a flow-chart below (from page 27 in [Jain et al., 2011]), followed by a description of the five activities.

Understand nature of application - is about deciding functionality type and classified based on how well the system fits this six 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 are also the question 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.)

Choose biometric characteristics - is also classified, based on seven different factors. The thing with biometrics is that it will never be completely solid, thus all the factors can't be perfect. Counted to this is that the factors will have different value for different systems.

1. *Universality*, the fail-to-enrollment (FTE) rate should be low.
2. If the *uniqueness* of the characteristics is high will the rate of FAR 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 convenient.
5. The time of the authentication is measured in *performance*.
6. User should have a high *acceptability* in present their characteristics to the system.
7. *Circumvention*, in terms of how easy it is to malicious fake the characteristics.

Collect biometric data - is apart from the collecting also includes factors of time, cost and size of the equipment.

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

Evaluate the biometric system - by asking different questions. There are no framework for doing this and it has to account different perspective as require experts of different field such psychology, business, computer science and statistics. There exists no framework for these types of evaluation but [Jain et al., 2011] propose doing it in three evaluation-stages technology, scenario and operational.

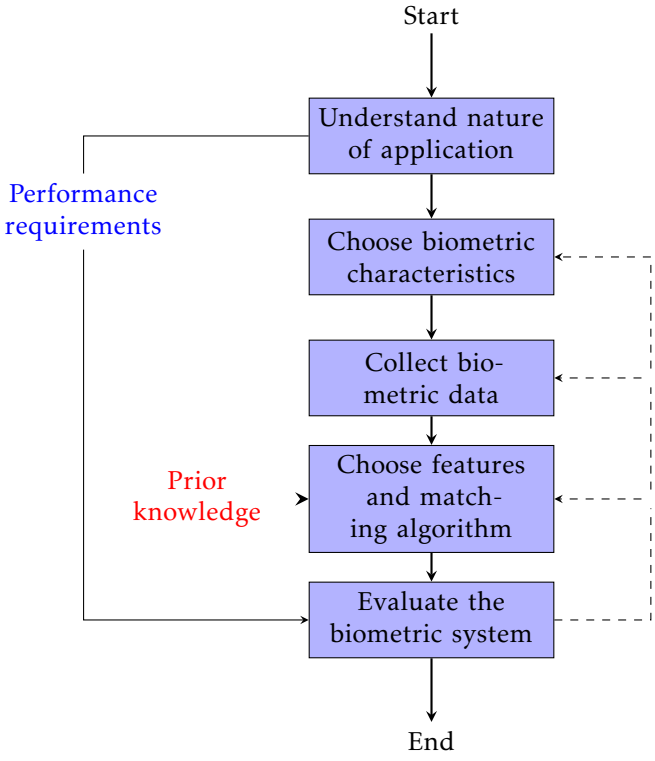

Figure 2.1: *The design cycle of a biometric system*

3

HARDWARE CHARACTERISTICS OF A MOBILE DEVICE

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. In the aim of this thesis it is feature characteristics that can be seen as an uniqueness of an mobile device. *Device fingerprint(ing)* is the term used for this feature characteristics and the pyramid seen in figure 3.1 from [Das et al., 2014] shows the different types of sources of device fingerprint. This thesis will focus on the top of quarter 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 from Android respective JavaScript depending on the preformed tests that is described in chapter 4.

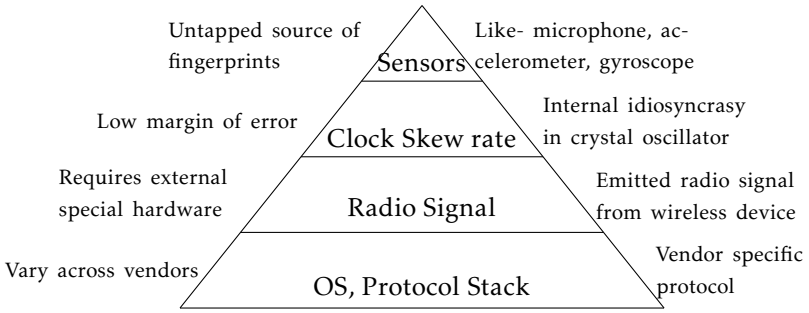

Figure 3.1: *The pyramid of features in a mobile device that can be used for fingerprinting.[Das et al., 2014]*

As seen above in figure 3.1 are sensors an untapped source of fingerprints in mobile devices and example of sensors are microphone, accelerometer, barome-

ter, speakers and gyroscope. The sensors investigated in this work is the accelerometer-, gyroscope-, magnetometer- 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 feature / 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 quadratically 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 feature / Bias

The gyroscope has some error characteristics like constant bias, white noise, bias instability, 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 Magnetometer

The magnetometer measures the magnetic field and was originally used for navigation and tracking. When it is used as a compass the Earth's magnetic field is measured. The type of sensor found in mobile devices is like accelerometer and gyroscope a MEMS sensor. They are known as e-compasses gaussmeters that is measuring of magnetic fields larger than 1 nT. [Y. Cai and Feneelly, 2012]

3.3.1 Fingerprinting feature / Bias

Note that normally bias in a magnetometer is called ***offset*** *but for uniformity reason of this report it will be referenced to* ***bias****.*
When try to measure the magnetic field of Earth with a magnetometer it also gets affected by other magnetic fields. The two main error sources from measurements of magnetometer are magnetic contamination in the sensor, called Soft and Hard-Iron distortion.
The hard iron distortion is caused by metals and magnets around the magnetometer. This field is constant even if the device moves, thus it is additive to the earths magnetic field. This distortion can be caused of e.g. the device speaker mounted near the magnetometer.
If there are magnetic fields in the surrounding environment of the device they will cause soft iron disorder. This error is smaller than the hard iron distortion. Any metal around the device can cause this error, such a car passing by. There are also a third bias that can effect the magnetometer and is as the hard iron distortion additive. This bias is the errors in the magnetometer itself such that all hardware has. But the result of this bias is similar to the soft iron distortion.
The good thing with magnetometer bias is that it is constant over time, thus once it is calculated it stays the same.

If the device is rotated one turn (360°)around its own z-axes (see figure 3.3) and plot x with respect to y there will be a circle. In a world without bias this circle will look like:

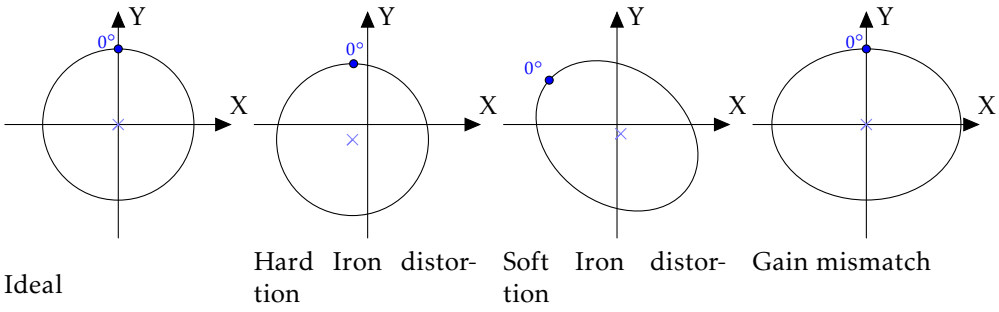

Figure 3.2: *Example of different magnetometer readings when affected by bias.*

[Merkel and Säll, 2011]

3.3.2 Magnetometer calibration

To do a magnetometer calibration the device is turned 360°around the z-axis (figure 3.3), with the device on a flat surface. This makes x and y-axis the surface and z-axis pointing (UP/DOWN?). To calculate the offset $O_{x,y,z}$ the average of min and max-value from each axis is calculated (the center of the circle) from the magnetic field $B_{x,y,z}$, like:

$$O_x = \frac{max(B_x) + min(B_x)}{2} \tag{3.3a}$$

$$O_y = \frac{max(B_y) + min(B_y)}{2} \tag{3.3b}$$

$$O_z = \frac{max(B_z) + min(B_z)}{2} \tag{3.3c}$$

And the hard iron compensated result $m^h_{x,y,z}$:

$$m^h_x = B_x - O_x \tag{3.3d}$$

$$m^h_y = B_y - O_y \tag{3.3e}$$

$$m^h_z = B_z - O_z \tag{3.3f}$$

The soft iron is sometimes compensated for in the equations above and sometimes additional compensation has to be done. Since soft iron offset is place specific it will not be considered in the thesis since a fingerprint of a device is not convenient if you have to be at the exact same spot every time the device tries to authenticate. [Merkel and Säll, 2011]

3.4 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 trough 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 image;

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 aren't 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]

3.5 Measurements of sensors on mobile devices

Measurements of sensors from mobile devices can be gather in different ways. In the work of this thesis two approaches is used, a browser application and an Android application. The camera is an exception where pictures provides the sensor characteristics.

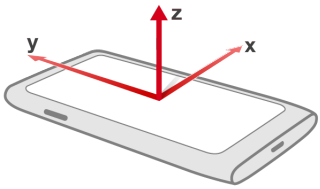

Figure 3.3: *The coordinate system used for both Android and JavaScript[Dixit, 2012]*

3.5.1 Android

Android sensor framework provides raw data with high precision from sensor that are built in the device such as different motions sensors (including accelerometer and gyroscope), environmental sensors and position sensors (e.g. magnetometer). [Android, 2015a] Android sensor framework classes that is used for gather sensor data is;

`SensorManager`, a manager used to access the sensor of the device.

`Sensor`, representing a sensor

`SensorEvent`, an event from a Sensor such data from the sensor, time-stamp or accuracy

`SensorEventListener`, gets a SensorEvent when a sensor or accuracy has changed

In Android you can't just read from the sensor whenever you need, instead the SensorEventListener has a function that is triggered every time the sensor is changed. You can however set how fast the delay from the sensor should be. The function provides the output of the sensor with a time-stamp of when (in nanoseconds) the change occured. The template code for this function:

```
SensorManager sensorManager = (SensorManager)
    getSystemService(Context.SENSOR_SERVICE);
Sensor sensor; //the sensor such accelerometer, gyroscope...

if (sensorManager.getDefaultSensor(Sensor.TYPE_OF_SENSOR) !=
    null) {
   // TYPE_OF_SENSOR such TYPE_ACCELEROMETER
    sensor =
        sensorManager.getDefaultSensor(Sensor.TYPE_OF_SENSOR);
    sensorManager.registerListener(this, sensor,
        SensorManager.SENSOR_DELAY_XXX);
}

public void onSensorChanged(SensorEvent event){
   // The time in nanoseconds of when the event occurred
```

```
    float timestamp = event.timestamp;

    // The measurements values from the sensor that changed
    float a = event.values[0];
    float b = event.values[1];
    float c = event.values[2];
    // and there may be more depending on which
    // sensor that caused the sensor change.
}
```

[Android, 2015b]

Accelerometer in Android

`TYPE_ACCELEROMETER` is the hardware measurements that measures the force of acceleration including the force of gravity with the SI unit m/s^2. Android also provide `TYPE_LINEAR_ACCELERATION` that is without gravity but that is a combined hardware and software sensor, thus this tests uses `TYPE_ACCELEROMETER` that provides almost raw data. There have been some bias removal from the sensor such bias from different temperature.
To get the acceleration applied to the device (a_d) the measurements of the force (F_s) applied the sensor is calculated from Newtons second law using the mass (m) of the device :

$$a_d = -\sum F_s/m$$

These measurements from the `SensorEvent` is in the x-,y- and z-axes like in figure 3.3 and collected from event like x=a, y=b and z=c in 3.5.1. [Android, 2015b]

Gyroscope in Android

The data from the gyroscope sensor is collected from the `TYPE_GYROSCOPE` that measures the rotation in rad/s around the x-, y- and z-axis of figure 3.3. The direction of the rotation is positive counter-clockwise if looking from a positive location of the axes. The values from the measurements when the gyroscope is changed is given like rotation in x=a, y=b and z=c in 3.5.1. Additional output is `values[3], values[4]` and `values[5]` that is the estimated drift arounf the axis in also in rad/s.
Android also provide a `TYPE_GYROSCOPE_UNCALIBRATED` that is the same as `TYPE_GYROSCOPE` except that no drift has been compensated for. There is still factory calibration and temperature compensation applied. [Android, 2015b] This makes it possible to calculate the linear sensor bias (equation 3.2] without Androids own bias compensation that is not known what it is.

Magnetometer in Android

Measures the magnetic field in x-, y- and z-axis in micro-Tesla (μT), as for the other sensors is that output for `values[0], values[1]` and `values[2]` (see

code 3.5.1). Android also provides a uncalibrated version that not has any calibration for the hard iron calibration. The uncalibrated type in Android is `TYPE_MAGNETIC_FIELD_` and as for the gyrocope it also comes with an bias estimation in x, y and z-axis for event-values `values[3], values[4]` and `values[5]`.

Fingerprinting: Sensor fusion in Android

`http://www.codeproject.com/Articles/729759/Android-Sensor-Fusion-Tutor`

3.5.2 JavaScript

JavaScript has since the use of smart-phones adapted a lot of new features, which makes it possible to access a lot of features in the devices. To access the gyroscope and accelerometer-data no permission from the user is needed, thus the user do not have to know that the sensors are measured.

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 3.3.

In JavaScript there are two types of acceleration, `accelerationIncludingGravity` and `acceleration`.`accelerationIncludingGravity` is acceleration made by the device. In context to `acceleration` not depending on influence of gravity only by the acceleration made on the device. Since the accelerometer in the device measures with gravity I assume that that is the most raw data you can get of the two. There is no documentation on this made, thus that makes this some kind of a guess.

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]

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 made in degrees of rotation angle. JavaScript has named this rotations as the figure 3.4 below.

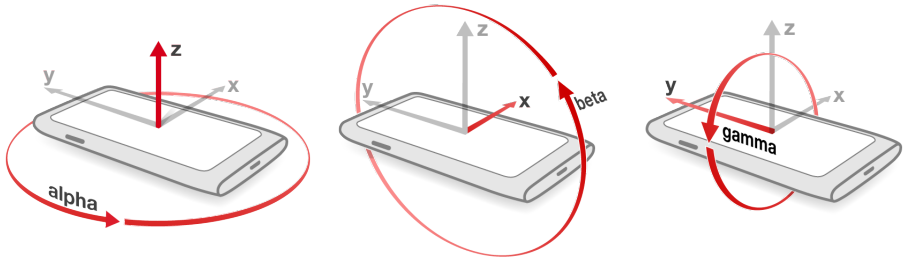

Figure 3.4: *The device rotation axes for the JavaScript* `DeviceOrientation`

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.

The JavaScript for measurements of the gyroscope:

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

[Dixit, 2012]

4

SENSOR MEASUREMENTS

In this chapter the methods used for testing the mobile devices for different characteristics is described in chapter 3.Three different test where performed to get sensor data to analyze for bias and characteristics.

Overview of the tests performed:

Motion sensor measurement I: Aimed to collect the data via a web-page since JavaScript can access *gyroscope* and *accelerometer*.

Motion sensor measurement II: Second test was implemented as an Android application to also measure sensor readings from *gyroscope* and *accelerometer* with the addition of the *magnetometer* together with some adjustments in the approach of the measurements.

Motion sensor measurement III:

- **Camera measurement I:** Collect one video from each device and extract pictures frames from the video. Calculate and compare the PRNU of the extracted pictures.
- **Camera measurement II:** Collected photographs instead of video from the device to get more sensor information.

4.1 Motion sensor measurement I: Web-page "Gyrotion"

Developed a web-page to collect the accelerometer and gyroscope data that JavaScript provide (section 3.5.2). The web-page is called "Gyrotion" due to it measures

gyro and motion. 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 and gyroscope data that then where saved as a CSV for further analyzing. The screen-shots below shows the web-page while measuring and the right one when finished and ready to send.

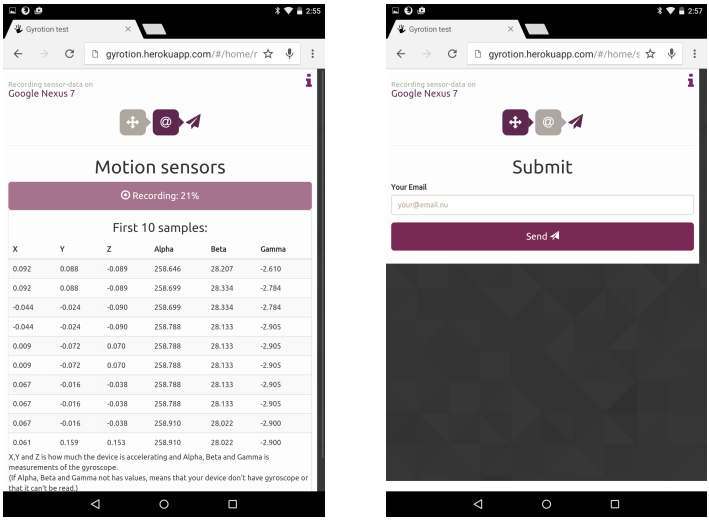

Figure 4.1: *Screen-shots of motion sensor measurements on web-page "Gyrotion"*

4.2 Motion sensor measurement II: Web-page "SensorRec"

Since added camera to the test the name "Gyrotion" no longer where applicable it became "SensorRec" (short for sensor recordings) From the last test some changes were made to improve the test result:

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 4.1).

3. It's 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. Also switch the accelerometer listener from `acceleration` to `accelerationInclu` since I suspect that that will give more raw accelerometer data, thus less bias removal by Android and JavaScript.

5. Added a step for video recording, more about that in section 4.4.

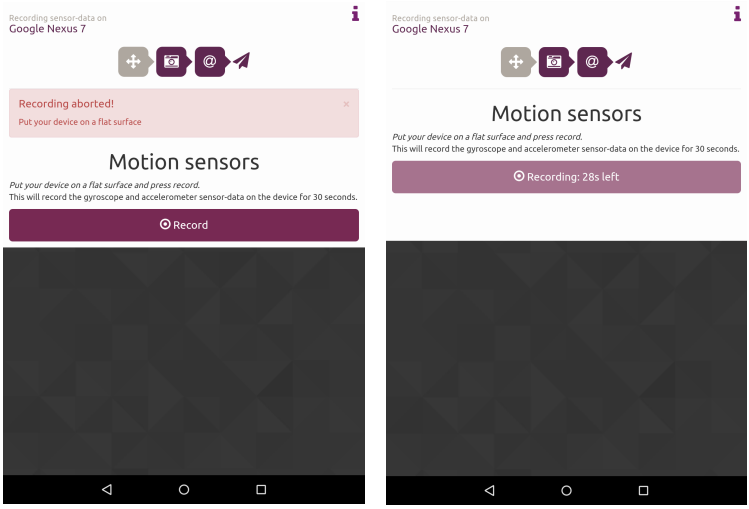

Figure 4.2: *Motion sensor measurements II on a Google Nexus 7*

The link for this page where spread and XXX measurements where made. The page is online on address: `http://sensorrec.herokuapp.com`

4.3 Motion sensor measurements III: Android application

==== OBS!! ====
Kanske flyttar denna till demo-delen beroende på hur det går... och då får jag inte gömma att flytta Android o JavaScript från kap3 till "rätt ställen"

1. An additional recording with the device placed in hand to make the measurements more diverse bias estimation.

2. Make 2 recordings with the difference of a 180 degree rotation alpha wise (see figure 3.3) for better bias estimation Kionix [2007]. REMOVE?

3. Added "Morse"-vibration in the recording to remove environmental bias which is one of the largest sources of bias [Kionix, 2007, p.8]. The "Morse"-vibration is a test pattern of Morse that the device vibrated like during the measurements. The reason for this is to have data for the demonstration

(REF TILL IMPLEMENTATION!!) of challenge response-authentication made by the device.

==== OBS!! ====
Lägga till lite scceenshots o kod o grejer...

4.4 Camera measurements

As most of the camera fingerprinting articles (REFERENSER!!) the aim has mostly been forensic and not focusing on the measurability or integrity of the pictures. That is why some limitations has been made in these measurements, e.g. black-motive (integrity) and not as large amount of picture required (measurability).
To measure the camera two measurements where gathered in both cases where the device on a flat surface which makes the camera result black. Both of this measurements is analyzed by the PRNU-method used by Jenkins [2009] described in section 5.2.

1. **Black video:** The recommended number of pictures for camera fingerprinting is 50 [Jenkins, 2009]. But that is not a convenient gathering purposes, thus to ask someone to take 50 black photos and send will not make many answers. Thats 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).

2. **10 black photos:** Simple as taking 10 photos, also with the camera pointing down on a flat surface. Since Jenkins [2009] where 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 “SensorRec”:

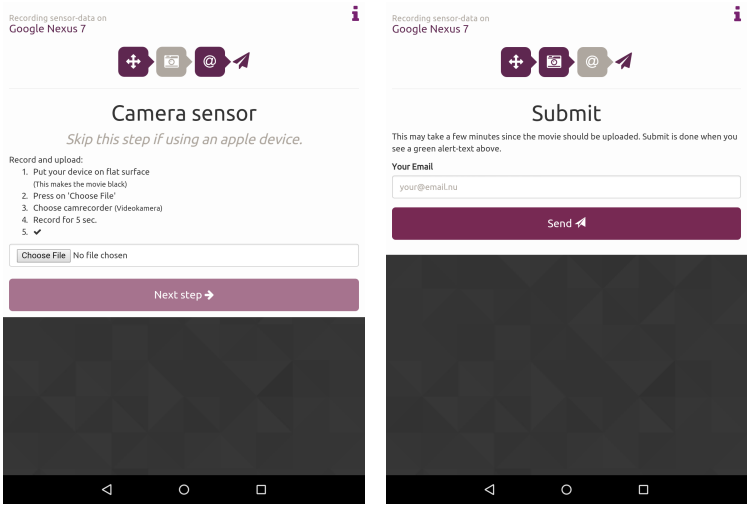

Figure 4.3: *Sensor measurements on a Google Nexus 7*

5

RESULT & ANALYZE OF TEST

TODO!

5.1 Result Accelerometer & Gyroscope-test

As described in chapter 4, two test have been preformed on the accelerometer and gyroscope data with the result presented here.

5.1.1 Test I

The data were gathered as described in section 4.1 from the web-page in figure 4.1 by sending the link of the page. When looking at devices with similar or same hardware you can see differences in measurements, for example here are the accelerometer recordings from 5 iPhone 6 and 1 iPhone 5S:

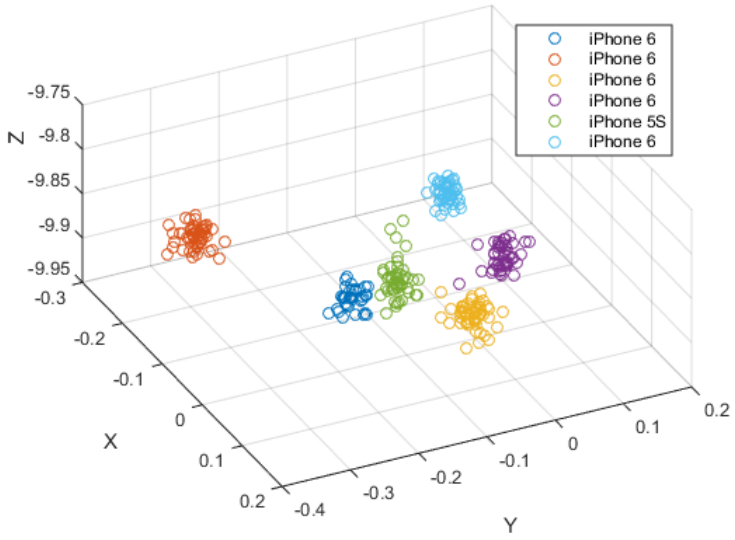

Figure 5.1: *Scatter graph on accelerometer recordings of 6 Apple devices*

The conclusion made from this made me ... SKRIV LITE OM RMS, SD, MEAN O LÄGG TILL PASSANDE GRAFER...

5.1.2 Test II

The page from Test I was developed by the test-result. The changes that were made is described in section 4.2 to improve that analyze data. The changes where improvements that as seen in figure 5.2 of six Android devices, that includes measurements for two of the devices with one month apart and you still see the measurements at the same spots.

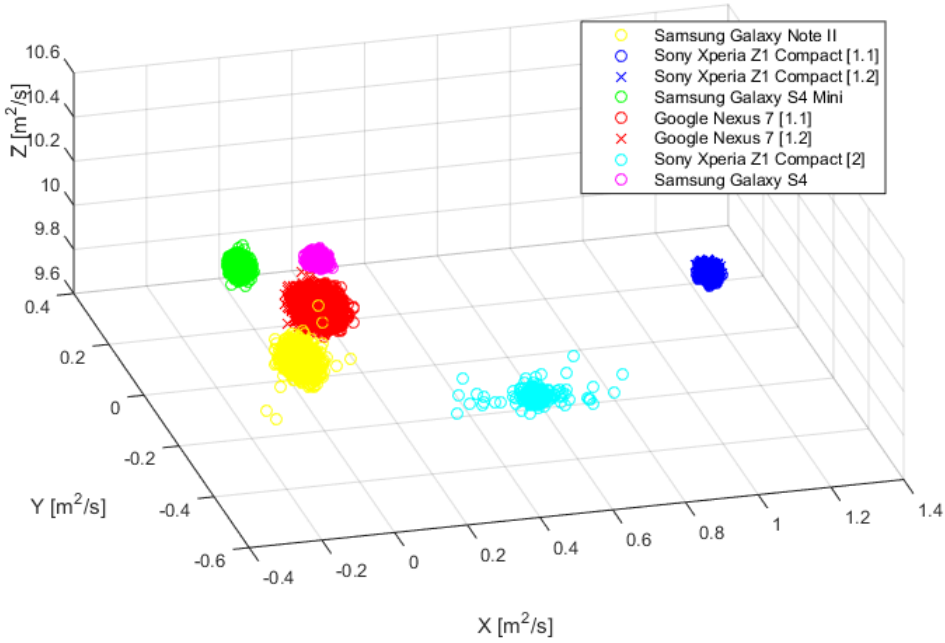

Figure 5.2: *Scatter graph on accelerometer recordings of 6 Android devices, where two has one month apart*

Another thing you can see from just these three measurements are that RMS (Root-mean-square) is still no good feature for fingerprinting eaven if the measurements are interpolated over time.

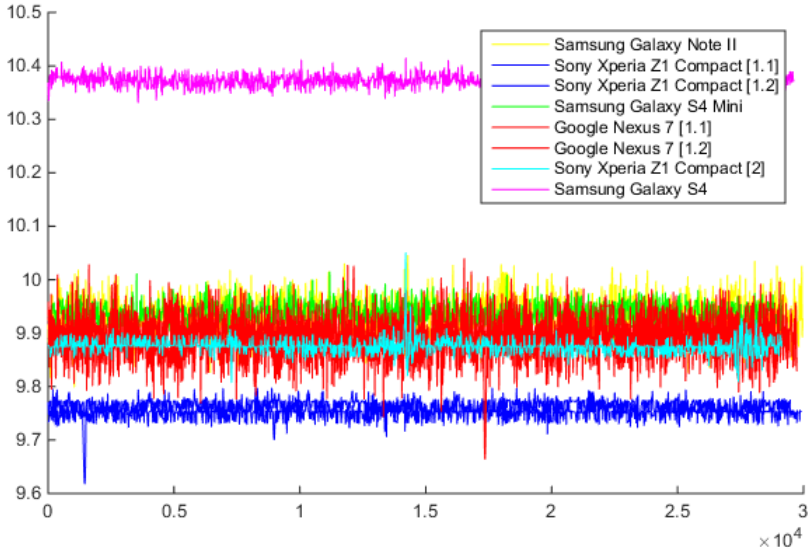

Figure 5.3: *Scatter graph on accelerometer recordings of 6 Android devices, where two has one month apart*

5.2 Result Camera-test

==== OBS!! ====
Flyttad text som måste skrivas om. For the test of the camera sensor the PRNU value is calculated as an approximation of the algorithm described in section 3.4 and also used by Jenkins [2009]. That 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. For the test the MATLAB `medfilt2` is used, which is an 2-D median filtering that outputs the median value of each pixel by its 3-by-3 neighbors.

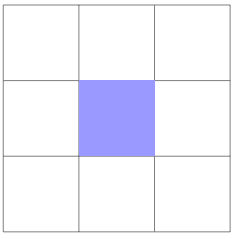

Figure 5.4: *the MATLAB `medfilt2` outputs the median of each pixel by it's 3-by-3 neighbors*

From the `medfilt2` we gain a picture without noise which is then subtracted

from the original to get the noise. 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 better noise is, thus the amount random noise is less and the fixed noise is more. Jenkins [2009] recommend a minimum of 50 images. This is then seen as the reference pattern used for correlating the noise from another pictures. This correlation is calculated like:

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

A threshold for acceptance on correlation is found by experimental on images taken with or without the camera. Then there is a balance between FAR and FRR. In section 4.4 i described two test preformed on the camera sensor of mobile devices.

5.2.1 Test I

Since the purpose of this thesis compared to earlier work REFERENSER!! 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 fist 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. This also makes it easier to get better noise since the same scene is used every 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');
```

To compare an 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' )
    );
```

5.2.2 Test 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.

Appendix

A

Trista saker

Långa beräkningar brukar bli rätt trista…

Detta är ett appendix-kapitel. Jämför med appendixet i chapter 5.

A.1 Bädda sängen

Den här beräkingen är så trista att vi kallar den *att bädda sängen*.

A.2 Diska

Den här beräkingen är så trista att vi kallar den *att diska*.

Bibliography

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

Open Source Project Android. Sensors Overview, April 2015a. URL `http://developer.android.com/guide/topics/sensors/sensors_overview.html`. [Online; 15-April-2015]. Cited on page 18.

Open Source Project Android. SensorEvent, April 2015b. URL `http://developer.android.com/reference/android/hardware/SensorEvent.html`. [Online; 15-April-2015]. Cited on page 19.

D. Boswarthick, O. Elloumi, and O. Hersent. *M2M Communications: A Systems Approach*. Wiley, 2012. ISBN 9781119994756. Cited on page 7.

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 page 13.

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 18, 20, and 21.

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 7.

Anil.K. Jain, Arun.A. Ross, and K. Nandakumar. *Introduction to Biometrics*. SpringerLink : Bücher. Springer, 2011. ISBN 9780387773261. Cited on pages 7, 8, 9, and 10.

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 17, 26, 32, 33, and 34.

Kionix. Accelerometer Errors. (AN 012), May 2007. URL http://www.kionix.com/sites/default/files/AN012%20Accelerometer%20Errors.pdf. Cited on page 25.

J. Merkel and J. Säll. Indoor Navigation Using Accelerometer and Magnetometer. Master's thesis, Linköping university, October 2011. Cited on page 16.

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 7.

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 7.

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.

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.

X. Ding Y. Cai, Y. Zhao and J. Feneelly. Magnetometer basics for mobile phone applications. February 2012. URL http://www.memsic.com/userfiles/files/publications/Articles/Electronic_Products_Feb_%202012_Magnetometer.pdf. Cited on page 15.

Index