Radboud Universiteit

Master Thesis Computer Sciences

Creating secure IoT systems

Author:
Max Tijssen

Supervisors:
Erik Poll
Martin Sandren

March 3, 2017

Contents

Acknowledgments

I would like to warmly thank everyone who played a part in helping me create this thesis.

Firstly I would like to thank Accenture and their security team for allowing me to write this thesis during an internship at their company, as well as providing me with all the help and tools I needed. Special thanks go out to Martin Sandren, my supervisor at Accenture, our regular meetings helped make both my internship as well as this thesis a success.

I would also like to thank Guido Pommer from Schiphol as well as Peter Huisman from KLM. They were able to provide a unique and very relevant perspective on this research.

Lastly I would like to give a special thanks to my academic supervisor Erik Poll, of the Radboud University. Without his feedback and input this thesis would not have been possible.

Abstract

This thesis explores the security issues which arise when implementing an IoT system. To this end we will implement an RFID system as a simplified IoT example, and explore these issues this way. The research will also show a number of different technologies available when implementing such a system, showing their differences and why certain ones can be chosen over others for certain functional or security requirements.

As stated the prototype IoT system will serves as a running example. The system implemented serves as a way for passengers at an airport to track their baggage after checking it in.

The findings of implementing this system, combined with a literature study led us to find five main differences between IoT and traditional systems. Briefly summarized these differences are the following:

1. Technical limitations of IoT devices.
2. Physical environment plays a larger role in IoT systems. Many components of an IoT system will not be in a controlled environment.
3. Lack of security-focus during design and implementation process.
4. IoT devices are an interesting target for attackers as tools for DDoS attacks.
5. The use cases of IoT systems are more often privacy sensitive.

We also produced an overview of seven different communication technologies that can be used for IoT systems. The technologies are compared on a functional level (range, throughput and cost) as well as some specific security features (one/two-way authentication, confidentiality, unauthorized tracking countermeasures and identification). We feel this overview would be of value to anyone implementing such a system and is still searching for an appropriate technology. This overview is available as a overview table, complemented by more in-depth background information per technology.

Chapter 1

Introduction

The Internet of Things(IoT) is a relatively old concept [17] that is just now starting to gain a lot of traction. By connecting all imaginable devices to the Internet, from cameras and phones to fridges and wind turbines, people are quickly coming up with new ways to use IoT to improve our quality of life.

A problem with this fast expanding sector is that security unfortunately takes a backseat to functionality and cost of production. Most of the devices in IoT systems are quite simple, but this does not at all mean they are harmless. This is shown for instance by the recent large scale DDoS attack, which left many major websites (Twitter, Amazon, PayPal, etc.), unreachable by flooding a key DNS server [18]. For this attack a botnet was used which contained (among other devices) improperly secured IoT devices, often with default passwords. This highlights the importance of adequately securing IoT systems.

In this thesis we will be exploring how to design an IoT system in such a way that it can be said to be reasonably secure. The primary research question of this thesis is the following: **What security concerns arise when implementing an IoT system?** This can be reduced to the following sub-questions:

1. What aspects of IoT set it apart (security-wise) from traditional IT systems?
2. What concerns stem from these differences?
3. Which different technologies are available to develop an IoT system, and what are their differences?

These questions will be explored by implementing an example IoT use case up to the level of a working prototype, in addition to literature research.

These questions could also be explored exclusively through literature research, so why go through the effort of creating this prototype? We feel that many issues which might be otherwise missed or considered trivial will be revealed this way. By going through the entire process from the ground up we will by necessity experience the design decisions that have to be taken at each step of the way, and think about their security impact. Furthermore a running example serves to clearly illustrate many of the security issues to tackle.

The use case that we will be exploring is the tracking of a passenger's baggage after checking it in at an airport. In the system a passenger can use his

smartphone to track when his bag has passed certain scanners in the system, as a toy implementation. A cheap passive RFID-tag will be attached to every bag entering the system, which can then be picked up by scanners set further along the track. The reasoning for this technology can be found further on in this thesis in Chapter 6.

It can be said that this system is not "true IoT", or at least a quite simplified version, since it is a closed system with a limited amount of readers, which only communicate with one central application server. In a well set-up system this server is the only device actually connected to the wider Internet. We feel this does not hinder our goals, as the technological difficulties which would arise in a more complex or widespread system would offer little new insight. The system being rather straightforward helps both by reducing implementation time, as well as making the research more compact and easy to follow.

As noted in sub-question 3. we will also be looking at some different technologies which can be used for an IoT system. The technologies we will examine are: RFID, Visual Identification, NFC, Bluetooth Low Energy, Cellular, Lora and SigFox. We will examine both constraints with regard to usability, as well as looking at their differences regarding security aspects. For the security aspects we will mainly focus on their ability and ways to perform identification, authentication, prevent unwanted tracking and confidentiality.This comparison can be found in Chapter 4, and an overview table be found in Section 4.7.

Chapter 2

Background on IoT (security) and RFID

2.1 The Internet of Things

The Internet of Things(IoT) is the concept of not only our home PCs and large servers, but everything that can fit a sensor, from our fridges to our bikes, being connected together through the Internet. Through this interconnectivity these devices work together to reach some common goal [14]. An example is using your phone to preheat your oven and turn on your lights when you are on your way home; see [14] for more examples.

The concept IoT was actually first proposed as early as 1997 by Paul Saffo [17] where, discussing great technological catalysts, he says the following:

> And that is what the coming decade is going to be shaped by - cheap, ubiquitous, high-performance sensors. We are going to begin adding sensory organs on our devices and our networks.

Although he might have been a decade off we are now firmly in the age which Saffo predicted, where sensors are being added to all objects imaginable. These sensors are being used for a plethora of goals, from predicting the wind speed [3] and energy needs [1] to E-Health possibilities [5].

IoT is more than simply adding sensors though. It is about creating 'smart' devices and systems. There are three main visions when talking about IoT [14]:

- **"Things"-oriented vision**: Here the focus lies on the physical devices being attached to the network, and their sensors.
- **"Internet"-oriented vision**: The focus lies on the protocol used to connect all these devices, for instance the Internet Protocol.
- **"Semantic"-oriented vision**: This is the most abstract of the three visions. It focuses on the sheer amount of devices (and device types) being connected. It looks at how to search, store, address, etc. these devices.

  Figure 2.1, taken from [14] gives a useful overview of these different visions, and where they meet.

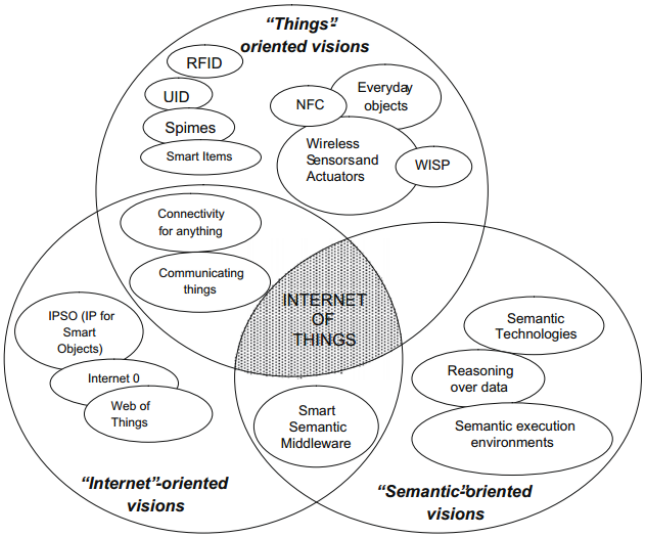

Figure 2.1: Venn-Diagram showing the different visions of IoT, from [14].

2.1.1 IoT and Security

Due to its very nature IoT brings with it a whole slew of security issues. These issues in our opinion stem from the following major factors of IoT.

The first set of factors is of a technical nature. Most IoT systems consist of a very high amount of (unattended) actors with low computational capabilities. In theory these are all avenues of attack, both as a way into the system or as a way to abuse the system to attack another (DDoS). Existing problems of securing any information system will be further aggravated due to this aspect. If even only one of these actors is improperly secured the entire system can become vulnerable. There are also limitations with regards to the low amount of power these devices are allowed to use, either because they can not receive more power (passive devices) or because their batteries are designed to last for a very long time. These limitations don't prohibit forms of cryptology [28], but the solutions do have to be created with these restrictions in mind, which will lead to higher development costs.

An often heard idiom when talking about security is that all bets are off when an attacker has physical access to a device, meaning that it is close to impossible to give security guarantees about these devices. Attackers with physical access gain many new ways of attacking the devices, from directly reading out memory to some side-channel attacks. A characteristic of IoT systems is that they, or at least some of their components, will often be in publicly accessible places. This leads to the need to secure the entire system, knowing that many of its participants could be compromised. In this way context and surroundings of an IoT system is a very important factor to consider, since the use case usually restricts your choices. Some examples of questions have to be asked with regards to this: Where does certain confidential data get stored/processed? What network can be used? What devices in the system can be physically accessed by outsiders and which can not?

The next issue regards security not being a great concern in many of the

IoT design processes. The devices involved will often have to be mass produced cheaply, and there seems to be a general apathy towards security by consumers when buying these devices. These two factors put together make it quite obvious why producers would not be to interested in investing heavily in security. This problem is further compounded by the fact that when the inevitable security issues are found there is often no way to patch these devices.

IoT devices have also proven themselves as being an interesting target, not necessarily for any data stored on them, but in order to use them as a further vector in a DDoS attack. This was proven by the recent large scale DDoS attack on a DNS server [18] which took down several large American based websites. The overall lack of security (thus ease of hacking) and large number of IoT devices naturally make them ideal targets to use in a botnet for these kinds of attacks. A Gartner forecast strengthens this claim [24]:

> By 2020, over 25% of identified attacks in enterprises will involve IoT, although IoT will account for less than 10% of IT security budgets.

This also signals the need for increased spending on IoT security. This relates to our previous point where we discussed cost often being seen as a more important factor than security in IoT systems.

The last issue concerns the usage of many IoT systems. They will often be systems which somehow compromise the privacy of their user by making them trackable, either directly (tagged clothing) or indirectly (transportation). Due to these privacy concerns the security of these systems should be paramount, although as already mentioned, it unfortunately is not. A current major hurdle to broad acceptance of IoT systems is trust from the users [14]. These privacy concerns of course play a major role in this trust, so if IoT systems are to become more widely accepted these concerns have to be addressed.

These factors combined indicate that security has to be a focus point in the development of any IoT system, more so even than for other information systems. This opinion is shared by the American Federal Trade Commission which published a FTC staff report: "Internet of things: Privacy & security in a connected world" [20] where they seek to inform producers of the possible benefits, but also the risks of IoT systems.

2.2 RFID

As noted in the previous section, in an IoT system there are many (possible different kinds of) devices. These devices have to somehow be identified in order for the system to function. Radio Frequency IDentifiaction (RFID) technology is one of the more common ways of doing this. Note that there are many other technologies that are used in IoT systems for identification (as will be discussed later on), as well as many other uses for RFID other than IoT systems. As the name implies it uses radio wave communication to send data between two devices. In the most simple use case a simple "tag" (often little more than an antenna and a chip) sends an ID number stored in its memory to a more complex reader in order for the tag to identify itself to the reader. Although a fairly old concept (with earliest publication stemming from 1948 [10]) it has

only the last couple of decades really started to take off, due to advances in overall computer capabilities and the ever decreasing size of transistors.

There are two variants of RFID devices: active and passive. Active devices have their own power source, while passive ones rely on the energy they capture through their antenna to power the chip. The passive chips either use magnetic induction (near field) or electromagnetic waves (far field) to capture the energy sent by the reader [11]. There are also three different frequency bands which passive RFID uses. This frequency influences both its range as well as how easily it penetrates materials. A general rule of thumb is that lower frequencies lead to a better penetration of non-air mediums, but smaller read ranges. The different types of passive devices and their ranges are the following:

- **125 - 134 KHz Low Frequency (LF)**: Range less than 10 cm, typically around 1cm.
- **13.56 MHz High Frequency (HF)**: A range of about 10cm up to 1 meter. Note that this is the same frequency that NFC uses, and NFC readers can also read these tags.
- **865- 960 MHz Ultra-High Frequency (UHF)**: A range of up to about 12 meters.

Powered devices operate on either 433 or 915 MHz, both of which have a range of up to about 100 meters [15]. A functioning system will always exist of either an active and a passive device (with the active one powering the passive one), or two active ones. In either case there will always be a "Tag", which sends information, as well as a "Reader", which reads it. Two-way communication is possible (for instance for some authentication protocols), but for the most part information travels only from the tag to the reader.

The active as well as the passive variants have their own use cases of course. Active ones are often more expensive (and bulkier for protection), but also able to communicate over larger distances and have a higher possible power usage. They are perfect for identifying large expensive devices such as planes or cars. Passives tags meanwhile are much cheaper and can easily be mass produced, to be used for instance in retail industries to track stock and sales, as companies like Wal-Mart are doing [11]. There are of course many different ways for RFID technology to benefit a corporation(see [13] for a good overview) but they all come down to somehow identifying (and possibly authenticating) some mobile thing to a mobile or stationary reader.

The costs of an RFID system depend largely on the system being implemented. Individual tags are extremely cheap (a few cents, up to a few dollars for very heavy duty ones), while the readers are more expensive and cover a wider price range, from ten dollars for a cheap featureless system to a few thousand for a fully equipped industrial reader. Even with tags being as cheap as they are they can still be the bottle-neck if you are identifying millions of items (such as at an airport or large warehouse). In these scenarios the cost of the readers is relatively minor, while the tags will make up the majority of the costs. Obviously the inverse can also be the case, such as for the access-management of a building. In this last example you will only need relatively few tags (as many as their are people to identify) while, depending on the amount of entrances and different areas within the building itself, you might need significantly more readers.

RFID contains no inherent confidentiality or authentication, although individual tags or readers may offer this. There is no direct link between type (unpowered or powered) or frequency of the device and security features, although there is some legacy of low frequency devices only being used for identification and thus not having received much security attention. Each type of device in theory has enough computational power to implement a algorithm like AES [28], but it is not standard, so care has to be taken when a system requires this.

Unauthorized tracking is a major concern when dealing with RFID technology. For the most part, RFID only serves to identify or authenticate a certain device. It makes sense that if there is no confidentiality or reader authentication involved that this same functionality can be used by unauthorized attackers to track these devices.

The frequency chosen has a large impact on security because it determines the range at which the tags can be read. Although from a usability perspective a longer range might sound like an advantage, do keep in mind that this also increases the range for eavesdropping or using illegitimate readers.

For a full technical specification of RFID see ISO 18000 [44], 14443 [45] and 15693 [46]. ISO 18000 is a set of a documents which describe RFID technologies for item identification at five different frequency ranges. ISO 14443 and 15693 describe Proximity and Vicinity cards respectively. Proximity cards function within 10cm and Vicinity cards 50cm. Furthermore security is more of a focus in Proximity cards (ISO 14443), due to their uses.

A note about the ranges mentioned: using antennas specifically designed for the purpose of capturing the radio waves used by RFID cards the range can be further extended as shown in [48]. Here the authors managed to activate and read the response of a ISO 14443 Proximity Card at a distance of 75cm. The authors expect further distances can be reached by using Digital Signal Processing. This means that when technologically advanced attackers are a threat you can not trust upon the normal maximum ranges of RFID, or at least this subset of the technology.

A good way to make sure a chosen system has certain security features in place is by making sure it is compliant with ISO/IEC 29167 [43]. This ISO standard defines for ISO 18000 the architecture for implementing security, as well as providing a more general architecture for other RFID standards. Note that ISO/IEC 29167 is split into multiple documents, each describing a different possible security suite that can be used. The following suites are described:

- AES128
- ECC-DH
- CryptoGPS
- Grain-128A
- AES128-OFB
- ECDSA-ECDH
- Present-80
- Ramon

These suites are named after the cryptographic primitives which are used in their protocols.

Be careful when selecting a suite, since they don't necessarily offer the same features. Some suites for instance only support tag authentication or don't have any encryption in place to facilitate confidentiality. For a full overview see Table 4.1 on Page 24. Be warned that although the ISO gets cited as a way of checking compliance it is quite difficult to find tags/readers which actually implement this standard, or at least ones that advertise this fact. We hope that in the future these standards would see more widespread use.

Chapter 3

Related work

This chapter serves as an overview of the work that has already been done on our subject. We do this in two parts. First by giving an overview of existing literature on IoT security. After this we showcase some other solutions already developed and possibly on the market when it comes to tracking baggage at an airport.

3.1 Literature on IoT (security)

As noted in Section 2.1.1 there are plenty of reasons for more focus on IoT security. Fortunately there are quite a few good articles and sources already available on this topic, as well as on IoT in general.

IoT is currently quite a hot topic in both academia and business. An example of IoT being explored is a very useful survey by Atzori et. al, [14], which contains some history, definitions of IoT, different usable technologies and examples use cases. In short this is a very good starting point for anyone doing research related to IoT without having much prior knowledge. A similar paper which also focuses on current and future trends is [31], which makes a nice complement to the previously mentioned paper.

Applications for IoT have also received a lot of attention and many different publications with proposals have been made. See for instance the papers on smart meters [1] [4] or wind speed prediction [3]. Smart-healthcare is also a often heard topic, see for example [5] for a proposed architecture which incorporates IoT. Two particularly interesting papers about IoT applications are one that gives a Chinese perspective on the future of IoT and where it can be used [2] and a paper on creating a framework for a smart city through IoT [6]. The value in these works comes from the fact that they show a variety of possible use cases combined.

The most directly relevant literature for this research is of course literature pertaining to IoT security. There is fortunately already some work done on this. One useful source for this research was [8], which is a document by the Cloud Security Alliance. The Cloud Security Alliance is a not-for-profit organization compromised of many members from large industry players, seeking to promote the use of security best practices in cloud environments. This document gives a good overview of what is necessary to create a secure IoT system, in a step

by step fashion. A more general overview of security in IoT is given by [19]. Gartner also has a forecast on how they expect IoT security budgeting and threats to evolve [24]. A more academic view on IoT security is given in [33], where the authors propose a systematic approach for looking at IoT security. They do this by highlighting the tensions between different aspects of an IoT system and capturing this in a model. For a much more complete overview of the different guidelines about handling or implementing IoT systems see [55], a recent blog post by Bruce Schneier.

Our own research contributes to the topic of IoT security by taking a different approach than these other sources. Where these sources follow a more theoretical line of thinking, our own combines this approach with the practical work of implementing a system. The findings about the difficulties of creating secure IoT systems in this research stem from the design decisions taking during implementation of our prototype system. We have also created an overview of the different attributes of communication technologies, both security specific and more general, which can aid academic and business personnel in choosing a technology that fits their system requirements.

3.2 Different solutions for luggage tracking

Our IoT system is of course not the only one available on the market. In this section we will describe some of the commercial products available. Although all of the solutions below are similar they all have a different end usage, which is why they were selected. When further analyzing these solutions we found each to rely on a different kind of technology, which indicates that each of the technologies available for IoT fill their own niche. We found two more solutions, but will not go further in depth for these, since they are both similar to Track & Go. These are Bag2Go[1] from Airbus and RIMOWA[2] from Lufthansa. To see how our solution compares to these please refer to Chapter 7.

The standard baggage handling system (without tracking) used by airports without these more advanced solutions is also important to keep in mind, to see just what the added value of these systems is. For this reason we have also included it in this section, as Subsection 3.2.2.

3.2.1 Overview

An overview of all of the solutions we will discuss can be found in Table 3.1. As can be seen from this table, although the use of all these solutions is similar, there are still significant differences. We argue that these differences are largely caused by the technologies chosen. In the next chapter we will discuss these (and other) technologies, and show these differences. See Sections 3.2.2 up to 3.2.6 for a more detailed description of these technologies.

[1] `http://www.airbus.com/newsevents/news-events-single/detail/applying-innovation-to-improve-the-airline-luggage-experience/`

[2] `https://www.rimowa.com/`

Table 3.1: An overview of baggage tracking solutions

	Technology	**System Owner**	**Usage range**
Base (no tracking)	Visual identification (barcode/id)	Airport	Global
TRACE ME	Visual identification (barcode/id)	Passenger (cooperation from airport)	Airport
BagTrack	RFID	Airport	Airport
EVIATE	GSM, GPS, Bluetooth & RFID	Passenger	Global
Track & Go	Bluetooth	Passenger	70m range from smartphone

3.2.2 Current Baggage Handling

As an example of how baggage handling systems currently works we will explain the one in use at Amsterdam Airport Schiphol. We base our explanation on a blog post by an employee, where he describes the process [60].

After baggage is checked-in, either at an employee or at the self-service desk, a barcode tag is printed and attached to the piece of baggage. The identifier contained in this tag is an international standard and recognized by airports worldwide. For transfers the process is similar, only the tagging was done at another location and the identity is known due to another airport initially entering this into the system.

The baggage then enters the baggage handling system, where it is scanned and identified. From this point onward the location of the luggage is tracked by keeping the starting point and speed of the conveyor belt in mind, and using them to calculate the current position. Since most baggage is checked in significantly before the airplane is ready to be loaded, it is first stored in a temporary storage, called a "buffer". When the airplane is ready to be loaded the baggage is put back into the conveyor system and is transported to a baggage cart, which transports it the last part of its journey to the airplane. The loading between different conveyor bands in or out of the buffer is done by robots.

Schiphol owns the system, and allows the airlines to use it. Schiphol's responsibility with regard to the baggage runs from when the baggage enter the system, until the moment it gets loaded into the baggage carts. Before and after this the airline is responsible.

3.2.3 TRACE ME

TRACE ME luggage tracker (`https://www.tmlt.co.uk/`) is a solution developed by a British company by the same name. It consists of a barcode tag that can be attached to your luggage, with a unique ID number. This system relies on existing airport luggage tracking, which it communicates with through SITA, the baggage tracking system used by the majority of major airports. It then forwards this information onto the passenger, who can this way stay up to date on where his bag is, depending on airport implementation. As such we feel this is more a service, than truly a technical solution.

3.2.4 BagTrack

A very similar system to the one we plan to implement is BagTrack by Lyngsoe Systems (`http://www.lyngsoesystems.com/en/solutions/airports-airlines/`). Implemented already at Hong Kong international airport, Milan, and Lisbon, it uses RFID tags attached to the device and readers to track the passenger's luggage throughout the system. Although it is not claimed on the website itself we believe the tags used are passive UHF ones, since these are the ones recommended by the International Air Transport Association, which Lynggsoe claims to abide by.

This system is designed to replace existing baggage handling systems. This system is thus not only concerned with tracking baggage for the passenger, but also for the airport itself, to use in for instance sorting.

3.2.5 EVIATE

The EVIATE system (`http://www.fasttrackcompany.com/`) was developed by the FastTrack company, in collaboration with (among others) Airfrance KLM. It is a separate high tech device that has to be inserted into your bag during travel. It allows for real time tracking of your luggage through GPS. It contains its own power source (rechargeable batteries). Fastrack's website claims EVIATE uses GSM, GPS, Bluetooth & RFID, although it fails to mention which technology is used for what purpose.

Besides tracking your luggage it also allows for some other features, such as automatically shutting off when it arrives on the plane (to save power and presumably complying with airline regulations) and alerting the passenger if his luggage is opened.

The system is scheduled to be released somewhere in 2017, so some of the details might be subject to change.

3.2.6 Track & Go

Samsonite has developed the Track & Go system[3], which are beacons built into suitcases by Samsonite, serving a similar purpose to the EVIATE device. The system uses Bluetooth beacons, which allows users to track their bags up to 70 meters away [16]. It uses Google's Eddystone Ephemeral Identifier[4], which broadcasts an identifier that changes every few minutes. The goal of this EID is to allow "things" to identify themselves, in such a way that only someone who has the linked ephemeral key can understand it. This feature serves to protect privacy.

If a user flags a bag as lost, other users of the system who happen to come within 70 meters of the bag will automatically signal this to server, which notifies the original owner. This is done the same way that tracking your own baggage happens, by an application on your smartphone. This system seems to be less about tracking your baggage while it is being handled by the airport, and more about keeping track of it while it is still in your possession.

[3] `http://www4.samsonite.com/_investordocs/20160414153416_ENG%20Samsonite%20TrackGo%20Press%20Release%20160406.pdf`

[4] `https://developers.google.com/beacons/eddystone-eid`

Chapter 4

Communication Technology comparison

This chapter examines some of the technologies that could be used for our use case, or similar use cases. These technologies all serve to facilitate the communication between different devices in IoT systems. The technologies we examine are: RFID, Visual Identification, NFC, Bluetooth Low Energy, Cellular, Lora and SigFox. There are of course many more technologies that could have been used (for instance Wi-Fi, Zigbee and Satellite), but we chose the ones mentioned since we felt they were the most relevant. Note that we evaluate the technologies not only based on the specifications of our use case, but also their more general aspects.

Gathering the information for this chapter proved to be a more difficult task than expected. Many of the technical specifications such as the range vary widely depending on which vendor is making the claims, while clear descriptions of security features were often even harder to find. Often it is easy to find some promise of a system being "secure", but it becomes necessary to dive into white papers and technical manuals to find just what is meant by the system being "secure", and how these features are achieved.

4.1 Security considerations

The technologies we discuss in this chapter will be evaluated on the following security features.

- ***Identification***: This involves one party claiming an identity to another party. Most often this is done by communicating some ID-number.
- **One-way authentication**: Here one party of the communication authenticates itself to the other party, so it proves itself to be a certain identity, rather than simply claiming it. In our case the party which identifies itself to the other will always be the "tag" (that which identifies devices).
- **Mutual authentication**: An extension to the previous feature. Here both parties authenticate themselves to each other. Concretely in our

case this would mean the reader also authenticates itself to the tag, so the tag will only reveal its information or identity to a legitimate reader.

- **Confidentiality**: This means that the communication between the two parties can not be overheard. In practice this will often mean that communication gets encrypted, so that even though a attacker can capture what is being sent it is meaningless to him.

- **Unauthorized tracking countermeasures**: What the technology offers in order to prevent unauthorized tracking.

All technologies have the minimum security feature of "Identification", which means they are able to claim a certain identity (but not prove this). This is both necessary for our use case, as well as true of all communication technologies.

There are of course more factors that could be considered, for instance availability or integrity of data, but we decided on the the ones written above, since they fit best with the security requirements for our use case, which we describe in Section 5.5.

Please note that what we describe in the following sections concerns claims made by the specifications or producers of these technologies. Often it is only described what cryptographic primitive (such as AES) their cryptographic protocol is based on. Even if the primitive is secure, the protocol might be fundamentally flawed and insecure. For this reason this chapter should not be used as hard evidence that a certain technology guarantees some security feature, only that it has the potential to do so. Truly evaluating and proving these kinds of claims is difficult and labor-intensive enough that we feel it makes more sense to do this per technology, or even per implementation, basis.

4.1.1 Layers

In our consideration of the technologies we focus on two different layers of the software & communication stack, the "Application layer" and the "Transport layer". For a graphic representation of these layers see Figure 4.1. This model is a very simplified version of the widely used OSI model [50]. The Application Layer contains the application which you control, the Transport Layer is dictated by the technology chosen, so mostly immutable.

Our comparison will focus on whether the Transport Layer has the security features described in the previous section. There are a few reasons for this:

1. It is not always feasible to implement your own security protocols in the application layer, due to shortcomings of the device. It might have enough computing power in theory but have a high development cost due to having to work with these limitations, or the limitations might be more strict such as only being able to send packets of 12-bytes, and being limited in the amount of packets you can send. The device might not even be programmable at all, such as QR-codes or very basic RFID tags.

2. The ease of making mistakes when working with cryptology. Developing an application without sufficient specific security knowledge, even when using secure primitives, is likely to result in the end product containing vulnerabilities.

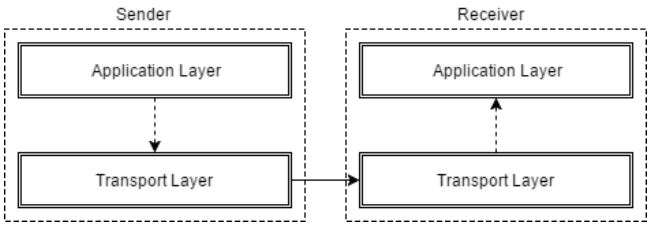

Figure 4.1: Layers for our technology comparison.

3. There are some things in the Transport Layer that can not be fixed in the Application Layer. Often a packet being send by the Transport Layer consists of a few parts, with the payload (which the application chooses) only being one of these. If you are for instance interested in privacy and want to make sure your messages cannot be linked to one another you can not use a technology which packets contain some unique identifier. Nothing you do on the Application Layer would fix this problem, since you only control the payload.

4.2 Visual Identification (barcode/QR-code)

Visual identification is, among the technologies shown here, both the least technologically advanced, but also by far the cheapest option. Whereas the other options all trust upon both of the parties involved in the communication being capable to send and/or receive information, here the reader does all the heavy lifting. The tag will contain some machine readable code contained in a visual representation, which is captured by the reader (for instance the camera of a smartphone) and translated. This does of course mean that the reader has to have a direct and clear view of the tag in order to read it. The range of this technology is fully dependent on the power of the camera which reads the tag and the size of the tag itself.

As you would expect, security features in this kind of technology are rather limited. There is no way for the tag to decide whether it wants to reveal its contents, so reader authentication is no longer useful. Copying tags is also trivial, a picture of a tag is just as valuable as the tag itself. The only way to protect these kinds of tags is to make sure line of sight is only possible for legitimate readers, but there is no feasible way to guarantee this. The only countermeasure against unauthorized tracking is once again this line of sight. When there is line of sight, it becomes trivial to track some device which is identified by visual identification.

Visual identification is always useful as a baseline solution, due to its low cost. The low complexity of the technology would also indicate an easier to protect system (keeping the restrictions of the openness of the tag in mind), although even this system has its own threats [29].

4.3 NFC

NFC is a closely related to RFID technology [27], with NFC readers even being compatible with some RFID tags. NFC, like the other technologies, serves to allow wireless communication between two devices. Just like RFID the reader can supply power to the card, or both can have their own power source [26]. NFC functions at the same frequency as High-Frequency RFID (13.56 MHz), which means it only has a range of up to about 10cm. Because of the similarities with RFID we won't dive much further into technological descriptions.

What makes NFC unique is the number of devices which can function as a reader, most brands of smart phone for instance. This leads to a technology that is very consumer friendly, and less industry focused than RFID, with its dedicated readers, of which the higher end ones can cost thousands of dollars.

Security in NFC is quite a hot topic with much academic research published on it [26,40,41,48]. A full technical specification can be found in [47]. In this ISO specification there is no mention of any encryption or authentication standard, so this would have to be implemented in the Application Layer. Eavesdropping is for this reason considered a significant threat in all of the research cited, since it is quite cheap and easy to capture radio waves. An advantage NFC has in this regard is that the range that the signal can be captured is quite small, according to [40] 1m for passive and 10m for active devices.

Unauthorized tracking is in our opinion a major concern as well. Due to there only being one-way authentication, it is trivial to use unauthorized readers in key locations to track who goes where. It could be argued that with NFC this is even easier than with visual identification, because the tags can be scanned even when concealed. The range helps lessen this issue, but this could be overcome by placing your readers in locations that you know people will be forced to be close to, such as door frames.

4.4 Bluetooth Low Energy

Bluetooth Low Energy (BLE) is the low energy alternative to the widespread and well-known Bluetooth protocol [21]. BLE serves the same purpose as its more energy consuming cousin, namely facilitating communication between two devices over short ranges.

Original Bluetooth could also be used to implement an IoT system, since it facilitates this same communication. Original Bluetooth can handle a much higher throughput than BLE, but pays the price by requiring much more power. BLE was designed specifically to alleviate this power concern, so that it can be used for IoT systems, which are often restricted in this regard. For this reason we do not examine original Bluetooth further, and stick with BLE. If power-draw is not a concern in the design of a system, but throughput is, then Bluetooth should also be considered.

BLE functions in a slave-master fashion, where the master sends a poll to the slave, which the slaves responds to with certain values or measurements, with the option of a two way acknowledgment after this. In this way the BLE system is quite similar to RFID, with slaves and masters representing tags and readers.

BLE slaves need a dedicated power supply, and can have a lifetime of 2 days

up till 14 years, depending on the frequency of polling and the number of times the slave can ignore a polling attempt, and thus not turn on its radio. These factors combined lead to a choice about how often the slave has to respond, from once every 7.5ms to once every 32 seconds. The power supply delivers 3V and 220 mAh, which amounts to 660mW. Furthermore BLE can communicate up to 58.48 kbps. These figures were taken from [21].

An interesting example of the BLE technology is called "BLE beacons". They are similar to masters in that they initiate communication but all they do is simply periodically send an "advertisement" of where they are. These beacons have a significantly higher range than conventional BLE slaves. Their signal can then for instance be picked up by a mobile phone, which can use it to determine its own position (with much better accuracy and coverage than something like GPS could). An example of this being used in practice is Schiphol airport, where the beacons can be used to navigate [22].

Security-wise BLE offers mostly everything that could be necessary. The following facts were taken from the technical specifications of the Bluetooth Protocol version 5.0 [34], as well as from a white paper published by the Bluetooth Special Interest Group describing the security that BLE offers [54]. Note that BLE is described in the technical specification alongside the larger Bluetooth protocol. Authentication and Confidentiality is achieved through AES in CCM mode. BLE uses three different cryptographic keys (all three unique per device, and generated on the device). They are used for the following:

- Confidentiality of data and device authentication
- Authentication of unencrypted data
- Device Identity

The pairing of devices is done by entering a code generated by the master into the slave. This is necessary for the other security features, since it is used to create the shared symmetric key between the two devices. The goal of this pairing is to prevent passive eavesdropping, as well as man-in-the-middle attacks.

There are even some precautions to prevent unauthorized tracking of BLE devices, thus increasing privacy. This is done by frequently changing the address (identity) of the BLE device, which means that the device has to be re-identified by the tracker each time it changes, complicating the process of unauthorized tracking.

Of the technologies discussed BLE is the only one where pairing is a part of the communication process. The other technologies are either too simplistic (visual ID), designed to work with a wide variety of scanners (RFID, NFC) or simply designed not to require pairing (Lora, SigFox, Cellular).

4.5 Cellular Network

The cellular network is the technology that is probably the most well-known among consumers. It is through this network that our mobile phones call each other or connect to the Internet. There is of course nothing stopping other devices from connecting and communicating through this network. An example of this is cars being fitted with sim-cards, in order for them to have mobile

Internet access. This way virtually any device can obtain Internet access, and thus communication abilities, anywhere there is cellular coverage.

A large advantage of the cellular network is coverage. This network can be connected to in nearly any civilized part of the world, the exception being very rural areas. A possible difficulty with the coverage lies in the fact that although many areas are potentially covered, some interference (tunnels, thick walls, etc.) can block the signal, with no easy way to boost it yourself, apart from convincing the telecommunication companies from building another tower. Throughput on the other hand is another great advantage of cellular networks, with 4G networks being able to reach up to 1 Gbps.

The major drawback in using cellular services lies in the price. Although the hardware needed to connect is not that expensive (antenna), you need a subscription in order to connect to the local telecommunication network, where prices vary greatly. This price is aimed at having maybe one or two devices per person which connect to the network and use it extensively, making it a rather poor choice for many IoT use-cases, since you will often have many devices using only very limited bandwidth.

A possible way of reducing these costs is using GPRS [56] instead of a more expensive protocol like 4G. GPRS is a service which allows 2G and 3G protocol using devices to send IP-packets through the standard GSM network to the wider Internet. A downside of GPRS is that it functions as as best-effort service, so there are no guarantees on latency or throughput, as well less error correction. The throughput when using GPRS is also rather low compared to the 4G network, with speeds only reaching up to 100 kb/s The major advantage in pricing comes from the fact that you are charged per MB used, instead of a flat subscription service with a cap. The price per MB is significantly more expensive (roughly a euro per MB) than a subscription where you can use a capped amount of GB per month, but the total costs will still be much lower when only using a few MB. If the IoT device in question does not need to send a lot of data you could this way end up paying significantly less. An example of GPRS being used for cost saving reasons can be found in [57], where GPRS was found as a cost-effective way to communicate between a GPS system and a web server.

Security wise is it quite hard to make assumptions about the cellular network. Eavesdropping on standard GSM (voice) data is quite possible using open-source software and relatively cheap hardware [42]. The security of mobile data is more up for debate, due to the many different protocols available (2G,3G,4G etc.), as well as differences between carriers in what kinds of encryption or authentication are used. Fortunately there are standard protocols which can be used on top of the communication layer (TLS, IPSEC, etc.) which can be used to create a secure tunnel, even over a insecure network. Although it could be argued that this is true for all technologies mentioned, cellular is one of the only ones with the throughput necessary to guarantee these function properly.

Trackability is a major concern when using cellular networks. When communicating over GSM, the authentication process includes sending the identity of mobile system [58]. GSM attempts to fix this by introducing Temporary Mobile Subscriber Identifiers, which is a good first step, but the problem is that when switching the phone on the base-identity is send out, so someone tracking a device from power up can still easily track this. These concerns are of course only made more severe by the great range of cellular devices.

Anyone seeking to use cellular networks for an IoT system would do well to carefully examine just over what carrier and using what protocol the data will travel. Also keep in mind that using mobile devices it might be possible to switch carriers when changing locations, possibly exposing your system to for instance weakened encryption. Many of these concerns can be alleviated by using the protocols described earlier to create a secure tunnel.

4.6 Lora and SigFox

Lora and SigFox are two competing vendors that offer a quite similar solution, namely a Lower-Power Wide-Area (LPWA) network. LPWA networks are networks which are designed to offer a range similar to that of cellular, but with decreased costs in both power consumption as well as direct monetary usage costs. For IoT such a service is very valuable, since it takes care of a large part of the trouble with setting up an IoT system, namely connecting the devices together, in a affordable way. With other solutions you control both the sender/receiver (tag and reader for instance), while with these solutions you simply connect your device to a Lora or SigFox network. The effect of this is quite similar to for instance the cellular network, where as long as the IoT device is in an area with coverage it can communicate with the larger IoT system. There are many more LPWANs, such as HayStack[1], NWave[2] and Narrowband IoT[3]. We chose Lora and SigFox since they are, to our knowledge, the most popular and widespread ones.

Both technologies offer impressive ranges for their receivers, up to 30km on open water and a more conservative 10km in urban environments. These figures were independently confirmed for Lora [52], while for SigFox we only have a claim by the company itself [53]. Due to the similarities in the technologies however, we feel this claim is likely realistic.

These vendors are largely similar, although there are some differences as noted in [32] and on the websites of the technologies[4][5]. There are some technical differences, such as throughput, see [32] for these. There are also some important differences in business models between the two. For the Lora technology, anyone can create their own network and connect this to the wider Lora network using GateWays. The largest organized group doing this is "The Things Network"[6]. This group seeks to create a network with global coverage, upon which you can (free of charge) connect any devices you wish. Sigfox on the other hand is not only a technology producer but also a service provider. It works with select partners to roll out their network, which you can connect to for a monthly fee. Effectively this means that if there is no coverage in a certain area, that with Lora you can provide your own, while for SigFox you would have to wait for SigFox partners to roll it out there. With Lora you thus have the option of connecting your private network, through a Gateway, while SigFox only supports you using 3rd party networks.

[1] http://haystacktechnologies.com/lpwans/
[2] http://www.nwave.io/nwave-network/
[3] https://www.u-blox.com/en/narrowband-iot-nb-iot
[4] https://www.lora-alliance.org/
[5] https://www.sigfox.com/
[6] https://www.thethingsnetwork.org

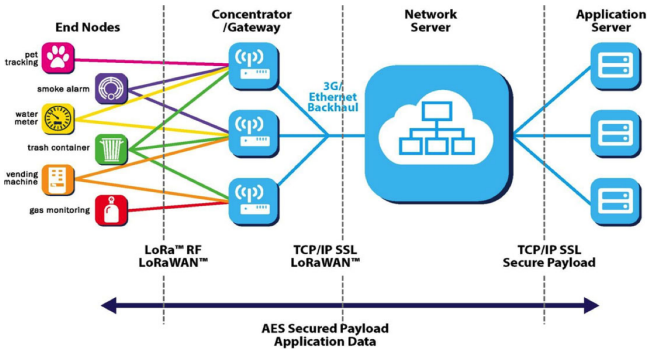

Figure 4.2: Lora Architecture. From [7].

An overview of the architecture of a Lora network can be found in Figure 4.2. A new member joining a Lora network will be able to add his own Gateways and End Nodes, or just End Nodes if coverage is already present. The network server handles responsibilities such as provisioning. Someone adding his own Gateways to an existing network has no control over these network wide configurations, other than disconnecting his Gateways.

The coverage of the two networks differs greatly. See Figures 4.3 and 4.4 for an overview of the current coverage of Europe of "The Things Network" and SigFox respectively. In the Lora network map each number indicates one Gateway. As these figures show, the Netherlands with its 583 Gateways is well covered, while a country such as Spain only has about twenty, which of course leaves most of the country without coverage. SigFox on the other hand seems to cover the vast majority of countries which it does cover. Do note that both of these technologies are relatively new and that the coverage is increasing rapidly, so future readers would do well to look up current coverage before making decisions based on these figures. From these maps you can conclude that if you are in a country which SigFox covers, you can likely get better coverage from their service. In countries SigFox does not cover there is a possibility of an already existing Lora network, or otherwise you can extend this yourself by purchasing a Gateway and coupling this to the network.

The extendibility of Lora's network might not seem important when the system to be implemented gets coverage in both systems, but even then there are some cases where this can be beneficial. When dealing with systems that are (partially) indoors, there is a chance that reception might be poor even if there is coverage in theory. If the system is in some industrial area with many thick concrete walls, the signal can get blocked very easily. With Lora you can in such a case place your own Gateways inside of these areas, and extend the network as is necessary. With SigFox you would have to try and do this through one of their partners.

A major benefit of using Lora is that security gets taken care of. Lora offers confidentiality, authentication and integrity on the network level, as well as end-to-end encryption on an application level[7]. Lora does its authentication and encryption with a protocol based on AES 128 bit [35,36], using separate keys for both. With a properly implemented protocol, it is fair to assume data sent

[7] `https://www.lora-alliance.org/What-Is-LoRa/Technology`

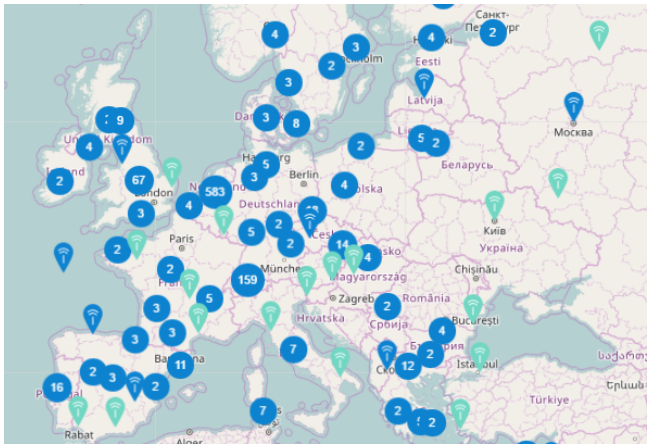

Figure 4.3: Map of The Things Network coverage in Europe.
Numbers indicate Gateways. From: `https://www.thethingsnetwork.org/`

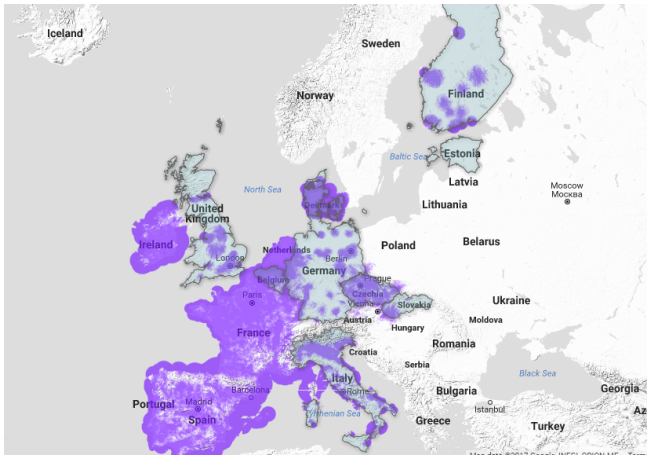

Figure 4.4: Map of SigFox coverage in Europe.
Purple is current coverage.
Light blue are countries currently being deployed.
From: `http://www.sigfox.com/en/coverage`

through the Lora network to be confidential, as well as authenticated. Integrity is assured by the use of MAC values.

There does seem to be some possibility of tracking Lora devices. As part of the header of any message send by the end-device to the Gateway is the DevAddr, which is the address of the device in the current network [59]. From our understanding this part of the message is not encrypted, and can thus be intercepted and read. The fact that this key is unique per network does increase the difficulty of tracking, since devices which switch networks will also switch key, so continuous tracking would be necessary.

Finding information about security features of Sigfox was the most difficult out of all technologies. From what we can gather Sigfox does do message authentication using Message Authentication Codes (MACs) made with a unique symmetric key on the device [37]. They call this "AES encrypted signatures" [49], but we feel this term is a little misleading due to the fact that signature seems to imply asymmetric cryptography, which is not the case. The receiver decrypts the MAC and authenticates the sender to itself, as well as checking the message integrity. The message itself does not seem to be encrypted. Instead Sigfox relies on two tricks to increase the difficulty of eavesdropping: channel-hopping and communicating over the Ultra Narrow Band(UNB) [38, 39]. We feel these techniques are not nearly sufficient to protect against eavesdropping. Channel-hopping can be trivially circumvented by listening to all channels, and the equipment used simply has to able to receive UNB. For these reasons we expect tracking to be possible, as long as the messages contain some device identification. Note that our sources for this information are a forum post, a sales pitch and a blog post, so they are of questionable quality, but they were the best available to our knowledge. A possible explanation of the difficulty of obtaining this information is SigFox's business model, where they feel no need to reveal these details since nobody would implement the systems themselves, although this is purely speculative.

Both technologies thus use symmetric cryptography (with the same key for both encrypting and decrypting), since they both use AES as a their security-primitive. For this reason it is important that each device has a unique key, otherwise the leaking of one key would bring down the entire system instead of one device. Fortunately for both of the technologies this is true.

4.7 Technology overview

Table 4.1 gives an overview of the specifications of all technologies discussed. Please refer to their respective sections for a more complete description. Figure 4.7 gives a quick overview of the throughput and range of the technologies discussed. Many use cases mainly depend on the range or throughput of the technologies chosen, so we felt this warranted its own figure.

Cost

Table 4.1 has two different columns for cost, "Device cost" and "Data cost". Device cost capture the cost of a single device in the network, for example an RFID tag, BLE beacon or QR-code. Similarly the Data cost capture the cost per month of one device communicating in a network, so for a single device

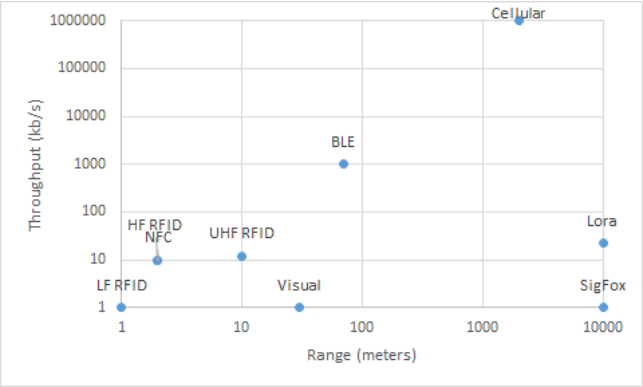

Figure 4.5: Overview of range and throughput of the different technologies.

using the SigFox or cellular network. There is also a third kind of cost, namely "Infrastructure cost", which we did not add to the table. Within Infrastructure cost we capture the cost of for instance readers or connecting the system with itself, or other systems. The reason for not including this cost is that it is too dependent on the specifics and environment of the system being implemented. The infrastructure of an RFID system can be extremely cheap if it only needs 2 cheap readers which do not need to connect to further systems, or it can be massive it it requires many industrial-strength readers. Similarly the infrastructure cost of setting up a global RFID system versus a global cellular system are very different. Since there is no real ground to objectively compare the technologies this way we simply recommend that anyone choosing a technology carefully consider the infrastructure costs they would incur.

The costs have values ranging from "Low" to "High". We have made a deliberate choice not to attach concrete values to these labels, since the actual costs differ enormously depending on all kinds of factors such as vendors, bulk orders, implementation, etc. These values roughly translate as follows: Low →a few cents, Medium →a few euros, high →more than ten euros.

Table 4.1: Overview communication Technologies

	Range	**Throughput**	**Device cost**	**Data cost**	**Authentication**	**Confidentiality**
Visual ID (Barcode/QR)	Line of sight	Limited by tag	Low	None	None	No
NFC	<10cm	Up to 2.5kb/s	Low	None	Implementation dependent	Implementation dependent
Bluetooth Low Energy	<100m	1 mb/s	Medium to low	None	One-way	Yes
Cellular	2-3km	>1Gb/s	Medium	(4G) High (GPRS) Medium	Carrier-dependent (but can create secure channel)	Carrier-dependent (but can create secure channel)
Lora	10km	Up to 22 kb/s, limited by day	Medium	Possibly none	Mutual	Yes
Sigfox	10km	12 byte packets, limited per day	Medium	Low	One-way	No, but some counter-measures in place (channel-hopping, UNB)
RFID Base	(LF) <10cm (HF) Between 10cm and 1 meter (UHF) <12m (Powered) <100m	From a few to multiple hundred tags a second	Low	None	Implementation dependent	Implementation dependent
RFID ISO 29167 Compliant	(LF) <10cm (HF) Between 10cm and 1 meter (UHF) <12m (Powered) <100m	From a few to mulitple hundred tags a second	Medium to low	None	AES-128 (One-way) ECC-DH (One-way) CryptoGPS (One-way) Grain-128A (Mutual) AES-OFB(Mutual) ECDSA-ECDH (Mutual) Present-80 (One-way) Ramon (Mutual)	Yes, apart from the CryptoGPS suite, which does not offer confidentiality

Chapter 5

Requirement analysis for baggage tracking

This chapter showcases the requirements that our system should fulfill, as they were set before the implementation of the system. The system in question is a way for passengers to track their luggage after they check it in at an airport. Note that in the development of a 'real' system more care should be given to the functional and non-functional requirements, and things like user-stories etc. should be taken in account. For simplicity's sake we have decided to keep these rather limited since they are not the focus of this research.

The security requirements on the other hand are of course vital. Here we will first create an attacker model of the kinds of adversaries we want to protect against, and which kinds of attacks these might be capable of (threat model). We will then use these to create concrete security requirements. Lastly we will create some Security Scenarios, which serve to evaluate how our system would perform under certain kinds of attacks or threats. Doing this kind of analysis and creating these requirements prior to implementation is vital in the process of Security by Design [8], in contrast to only looking at security as an afterthought.

There are many ways of conducting requirement analyses, we have chosen a rather informal one since it best serves our purposes, but a more advanced or extensive system could very well benefit from a more formal analysis.

5.1 Prerequisites

In order for our system to function there are certain prerequisites which have to be met. In this section we will state which these are.

P1. A tag is inserted into the baggage label, which can be scanned and identified.

P2. The passenger has a smartphone with a connection to the Internet.

P3. There is an existing baggage handling system which our tracking system can piggyback onto.

P4. The baggage handling system and path to the airplane are physically protected from unauthorized persons from entering.

P5. The system is used in both the airport of departure as well as the destination.

P6. Participating airports share information on enrolled tags.

5.2 Stakeholder Analysis

For each of the different kinds of requirements it will be important to think about the different parties which will be involved with the system. These parties will all have different motivations and goals, and thus produce different requirements.

The parties which we identified as having a stake in this system are the following:

- **The airport**: The airport will be the system owner. Their main concerns lie in increasing passenger satisfaction and lowering the cost associated with lost baggage. They would furthermore not want the system to impact their existing baggage transport/sorting systems, and minimize the cost per item as well as the costs associated with getting passengers familiar with using the system.
- **Passengers**: Passengers will be the ones whose baggage will be traversing the system. Their main concerns will be being able to easily track their baggage and ensuring its safety.
- **Airlines**: Airlines will indirectly be impacted by the system, by the baggage being tracked being put on their airplanes. They are also interested in passenger satisfaction, since a passenger who lost his baggage is just as likely to complain to the airline as to the airport, and otherwise will want the new system to impact their processes as little as possible.
- *Service provider*: If a solution is chosen where some third party offers their network (e.g Sigfox), to be used by the system, then they too are a stakeholder. Their concerns will be twofold. On the one hand they will want to make sure their network meets the demands of the customer (so the system owner in this case), while on the other hand making sure that this new service does not somehow disrupt other parts of their network.

5.3 Functional Requirements

In this section we will describe the functional requirements the system should fulfill. These requirements show what the system should be able to do, what functions it should have.

FR1. The passenger should be able to use their smartphone to track their luggage passing the following points:

(a) Check-in

(b) Entering baggage handling system

(c) In temporary storage

(d) On baggage cart

(e) At airplane loading

(f) On baggage carousel number N

FR2. Passengers should be able to opt-out of participating in the system.

FR3. Tags should be scan-able as long as they are attached somewhere to the outside of the bag.

FR4. Passengers checking in multiple bags can track each bag individually.

FR5. A piece of baggage has to be able to be removed from the tracking system without the passenger's knowledge. See Section 5.5.2 **Tracking a bag** for the reason for this Requirement.

5.4 Non-functional Requirements

This section will describe the non-functional requirements, which describe certain quality aspects that the system should fulfill.

NFR1. The technology chosen has to comply with airline regulations, and thus not sent any radio signals while on a plane.

NFR2. The existing baggage handling system should not be slowed by this new tracking system.

NFR3. The tracking system should be able to process the same quantity of bags as existing baggage handling systems.

NFR4. The tracking system should be usable by passengers with only guidance from the system itself, no intervention from personnel should be necessary.

NFR5. The system should accept any bag that conventional airport baggage handling systems would accept.

NFR6. Problems with the tracking system should not cause problems with the regular baggage handling system.

NFR7. The cost per item of luggage has to be low.

5.5 Security Requirement Analysis

5.5.1 Attacker model

From our analysis we discern the following possible attackers of our system.

- **Paparazzi**: This attacker is someone with limited means whose main goal is to be able to following someone's baggage without proper authorization for this. Although we have given this attacker the name "Paparazzi" this could of course be anything from paparazzi following a celebrity to a jealous spouse tracking their partner.

- **Thieves**: This attacker describes a technically limited attacker whose primary goals it is to steal the baggage. The system described can be both detrimental and beneficial to this category of attackers. On the one hand it can help them identify the specific piece of luggage they want, if they can somehow breach the system. If the thief identifies and scans a target bag he wishes to steal in the time between the moment it is tagged and before it enters the sorting area they can later use this same identifier to re-identify the bag. It is even possible this could be done on the baggage carousel, even if the tag is no longer recognized by the system it can still be scanned. The tracking system can also be a hinder, by alerting the owner of the luggage when the bag has disappeared after a certain point in the system. Note that there are two variants of this attacker, both an insider (baggage handler) and an outsider.

- **Organized crime**: Criminals with a lot more technical and financial means to deploy their attacks. As has been shown in the past these criminals will often employ people working at the baggage handling areas itself so they can be assumed to have at least limited access to the physical system. The end goal of "Organized crime" is to somehow make money by exploiting some part of the system. The possibilities for this are discussed in the next Section. Note that we consider "professional" hackers as being a part of this category, even if they work solo, as well as the more traditional large criminal organizations.

- **Government actors**: With any security related system government actors have to be at least considered. These actors will have near unlimited means to break through a system, and can easily force physical access, in this case probably with the help of the airport. For our system we will not specifically defend against these attackers, since this would be both extremely difficult, as well as ultimately futile since the system owner will likely cooperate.

- **Terrorists**: As with any system that operates within or around an airport terrorists are always a concern that has to be addressed. Terrorists would most likely use the system as intended, just for a more nefarious purpose. By tracking their bag, they can be assured of it having reached a certain destination (such as the plane). If such an attacker has managed to for instance smuggle explosives into their bag, it would be a great advantage to be sure that the bag has reached its destination before detonating these.

5.5.2 Threat model

We consider the following possible attacks that the attackers as described in 5.5.1 could attempt on the system.

- **Tracking a person**: Using the tracking system to track someone else's bag, with the intention of learning what plane they boarded or where the bag came from, and thus learning about their travels. This attack is the one that attacker *Paparazzi* is interested in. The PR blow-back that a attack like this could cause is a lot more serious than the attack itself, and should not be underestimated.

- **Tracking a bag**: Similar to the last threat but subtly different. Here the attacker only wishes to track a certain bag, irregardless of who the owner is. This attack is the primary interest of *Organized crime*. The reason for this is that smugglers are known to employ the bags of random travelers in order to smuggle illegal substances (often drugs) from one country to another[1]. A baggage handler in country A will plant the goods and identify the bag using the RFID tags, after which a baggage handler in country of destination B will scan and find the same bag, and remove these substances.

  There is also the variant of this threat where the owner of the bag is actually the one doing the tracking. This would be the attacker described as *Terrorist* in the previous section. Although this attacker is using the system precisely as it is meant to be used, we still consider this a threat to the system. If a terrorist is able to use this system to assist in his terror attack, then beside the obvious other horrible consequences, this system would also lose all its credibility and most likely be discontinued.

- **Stealing a bag**: This attack is not made possible due to the system, but actually made more difficult. A bag that has been stolen will no longer be traveling its standard route, so the tracking system will (unless fooled) detect this, and the owner of the bag will be alerted. Someone wanting to commit this attack would have to take this into account.

- **DoS-ing the system**: Somehow disrupting the system enough that it (temporarily) seizes functioning. An *Organized crime* attacker could use the threat of DoSing the system as a way to extort money from the airport, or simply do this out of spite. It is not even necessary for someone to consciously attack the system, a system that is not adequately set up could be disrupted by something as simple as old tags not being removed.

- **DDoS-ing *with* the system**: Instead of DoS-ing the system, the readers in the system can also be used as a way of DDoSing other online services, if they are able to communicate through the Internet. An example of this is the recent attack already mentioned earlier in the paper [18]. This kind of attack is of course an inherent threat nearly every IoT system faces, we won't go to much in depth on this attack other than saying that Readers (or any connected device) should not be directly connected to the Internet, unless necessary.

5.5.3 Security Requirements

Analyzing the threat model in Section 5.5.2 we came to the following additional Security Requirements that the system should posses.

SR1. Tags of previous journeys should not be recognized as valid.

SR2. Tags contain a minimal amount of information, just an id.

SR3. An ID number can not be coupled to a passenger's identity, without information on either the application server or the passenger's phone.

[1] `http://www.thejournal.ie/baggage-handler-drugs-2494944-Dec2015/ https://www.theguardian.com/world/2005/may/28/indonesia.australia`

SR4. Readers transmit only a read ID number in addition to their own id.

SR5. All channels of communication need to be secured in such a way that confidentiality and integrity of data can be reasonably assumed.

5.5.4 Security Scenarios

In Subsection 5.5.2 we focused on the motivations of possible attackers. In order to assess whether our system fulfills the requirements set out in Subsection 5.5.3, which are designed to counter the attacks related to these motivations, we will evaluate the following concrete scenarios:

SS1. What is the impact of an enrolled tag getting lost or stolen?

SS2. What is the impact of a reader getting lost or stolen?

SS3. What damage can a compromised enrolled tag do?

SS4. What damage can a compromised reader do?

SS5. How easily can the system be (D)DoS'ed?

SS6. How easily can the system be used to DDoS other systems?

SS7. What is the impact of an enrolled tag failing completely?

SS8. What is the impact of a reader failing completely?

SS9. Can a bag be removed from the system without its owners knowledge?

Chapter 6

System Design and Implementation

The design of a system is of course of great importance, especially in a system such as the one described where we place an emphasis on the security of the system. In this chapter we will walk through the design process for this system, starting with an overview of its workflow, followed by showing the components used to implement this.

6.1 System Workflow

See Figures 6.1 and 6.2 to see the workflow of our system. These figures show the different key-points at which the RFID readers update the application with the location of luggage. Due to these updates the passenger is able to track their luggage.

See Figure 6.3 for a more detailed view of what happens at one of the "Scan RFID" blocks, and what is communicated between the different parts of the system.

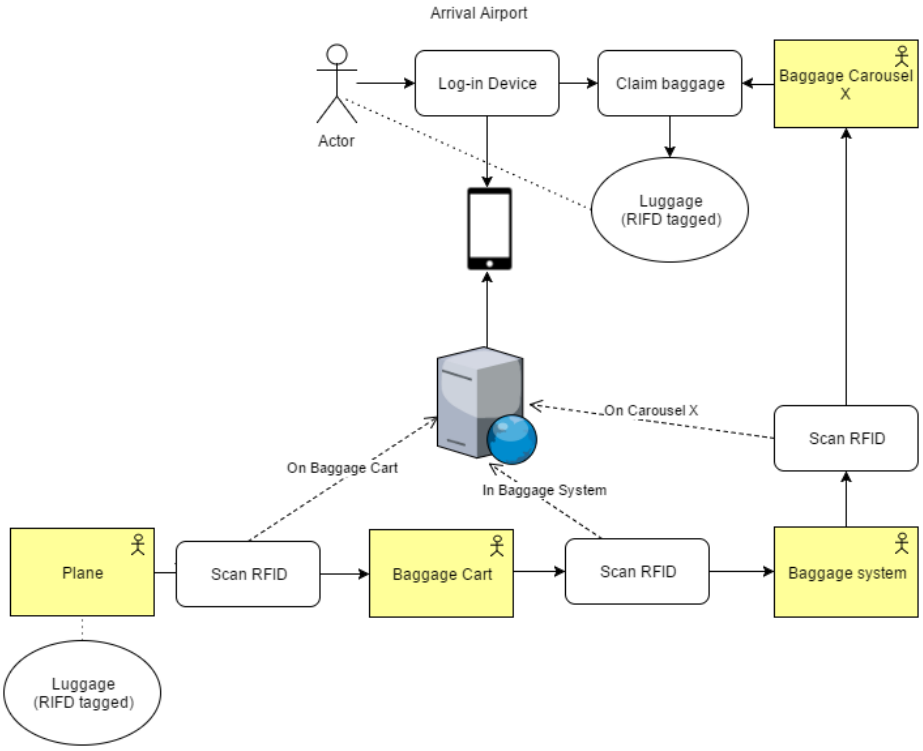

Figure 6.1: Use case upon arrival.

6.2 Components

6.2.1 Communication/Identification Technology

In our system the identification of luggage is of course vital, so a technology needs to be chosen in order to allow the luggage to send this identification. As noted in Section 5.5.3 SR5 it is necessary for communication to be secured (to safeguard integrity and confidentiality of data). This is another factor to keep in mind when selecting a technology.

See Chapter 4 for an overview of the different technologies we considered, in addition to Section 2.2 for RFID. As is already hinted at by the fact that it has its own section RFID is the technology we have chosen. In this section we will expand on how we came to this decision.

First let us consider condensed the requirements that this technology has to abide by. These are the following:

1. The price per item being identified has to be very low (a few cents), since each piece of baggage has to be separately identified, and the system owner (airport) has to pay for each one.

2. Throughput is not a concern, only an ID has to be sent.

3. The system needs to somehow secure the ID being sent, in order to prevent unwanted tracking. This can be done in two ways:

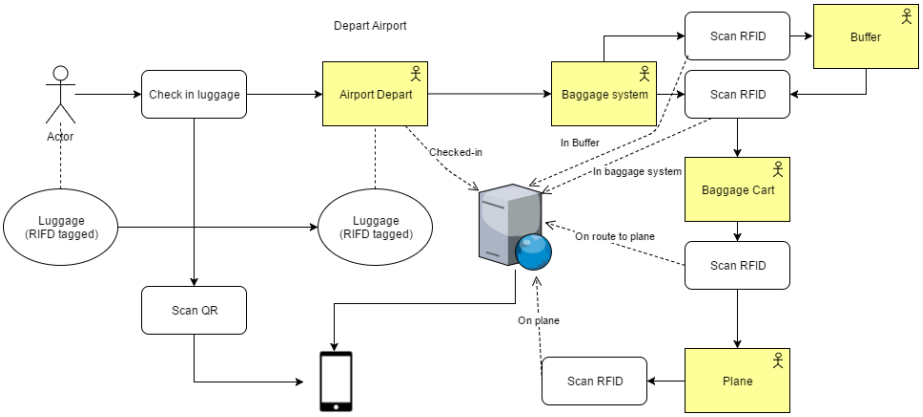

Figure 6.2: Use case upon depart.

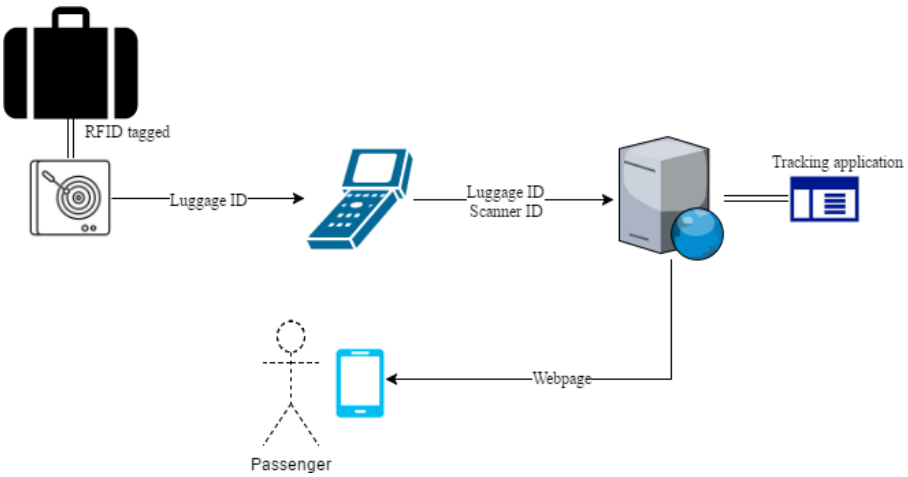

Figure 6.3: Overview standard scanning process.

(a) Enforcing authentication (so only legitimate readers can read) and being non-eavesdroppable (so legitimate readers cannot be overheard).

(b) The range of the technology is short enough that it is not-feasible for an attacker to track the baggage, since the area that the baggage enters is secured.

Requirement 1 and 3 have an interesting interaction. Cheap technologies have their limited ranges to prevent tracking, so fulfill 3(b) . The more expensive technologies often use cryptography, so they fulfill 3(a). Combining these two requirements thus only leaves the technologies relying on short ranges, so NFC, Low-Frequency RFID (LF RFID), High-Frequency RFID (HF RFID) and visual identification. An important aside here is that we thus consider the low range to be a feature of these technologies, since it complicates unwanted tracking. Normally limited range would be seen as a disadvantage, but as we show here this does not always need to the case.

Out of these options we have chosen for LF RFID. NFC is more consumer focused instead of industrial (so higher costs per product) and visual identification requires precise line of sight, which means that luggage all has to be carefully placed, and even then it might require the conveyor belts to slow down in order to adequately read them. HF RFID is also a viable option (and we would recommend it), but for practical concerns we chose LF for implementing our prototype, since the readers were easier and cheaper to obtain. As stated in the non-functional requirements in Section 5.4 the tracking system should not (or at least minimally) slow down the existing system. Since high end RFID readers can read many tags per second this should be no issue, as well as these readers not requiring line of sight.

A further advantage of RFID is that there is an easy way to identify implementations which do have some security features, depending on how much can be invested, as can be seen in Table 4.1 on page 24. There are quite a few different security suites available which can provide both mutual authentication as well as protection against eavesdropping. Some of these implementations might be high or even ultra-high frequency, but the range will no longer be an issue, due to the fact that eavesdropping already gets protected against. In this way they now fulfill 3(a) instead of 3(b). Powered tags will in all likelihood prove to be too expensive, but even if this were not true their large ranges might also cause issues with tags being read from to far away or by multiple readers, confusing the system.

These RFID tags can be added into the existing baggage-labels, this way this will cause minimal disruption to the existing sorting system.

For our prototype the RFID readers will be connected to laptops using a USB-cable. The laptops will run a Java-program which takes the output of the readers and sends this through SSL to the Application Server (in our case another laptop). We use this setup because our reader can not communicate independently. See Figure 6.2.1 for a visual representation. In a real system the readers should securely communicate independently. With readers communicating separately you have to keep things such as certificate distribution in mind.

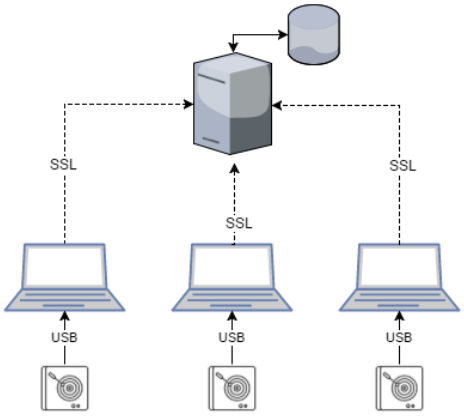

Figure 6.4: Structure of communication.

6.2.2 Application

As described in the previous section the reader and laptop configurations send their output through SSL to a Java program. This Java program (which we will refer to as "the tracking application") has a few key responsibilities.

Firstly it manages the enrolled tags. It handles both enrollment of the tags, as well as their end-of-life. At the startup the tracking application will first request you to scan the tags that will be active during this run. It will store these IDs in an internal data structure (Array, Collection, etc.). In a real world example this would be a database, in order to efficiently store many different tags. The tracking application also removes tags from the enrolled store upon them reaching their end-destination, so that left-over tags do not accidentally (or purposefully) confuse the system on subsequent journeys. At this point, although the tags are still functional (in that they can be read), they will no longer be recognized by the tracking application.

Next it stores the currently known whereabouts of enrolled tags (and thus baggage). When the tracking application gets input from the readers it does several things. First, it checks whether the ID it received is of an enrolled tag, if not it disregards the message. Then it checks whether the reader it received the ID from has a recognized location and whether it is the next location along the path. Here a choice can be made whether the path should be followed strictly or not. Being lenient and accepting out-of-order reads makes the system more fault tolerant.

6.2.3 Passenger Interface

The Java application described in the previous subsection produces a store of the current location of tags in the system. When a passenger checks in his luggage, he receives two things: a QR-code (which translates to a URL) and a set of login credentials. When the passenger scans this QR-code it leads him to website which searches the store for the ID which matches the login credentials. The found information is then displayed to the passenger, on an automatically updating screen.

It would be possible to tie these login credentials to systems such as existing

loyalty programs. We would in principle advice against this, or at least advise carefully considering the consequences, which we will now discuss. In the current system the ID in a tag can not be tied to a specific passenger, other than through already having knowledge of the ownership of the bag. This alleviates a lot of privacy concerns, and makes the system much less sensitive. When coupling this system with existing loyalty systems you lose this advantage. There is also the concern that coupling the tracking system to the larger passenger loyalty systems means that these loyalty systems get exposed to attack from this potentially less secured tracking system.

Chapter 7

Results and Analysis

In this chapter we will evaluate to which extend the system meets the requirements stated earlier in Chapter 5. We will also compare it to the other baggage tracking applications available, to see how it holds up.

7.1 Requirement Evaluation

In this section we will go through each of the requirements set forth in Chapter 5, on a per type basis, and argue to what extend system meets them. Please refer to Chapter 5 for the individual requirements.

7.1.1 Functional Requirement Evaluation

FR1. Our prototype system only makes use of two reader-laptop combinations and thus only models (b) "Entering baggage handling system" and (e) "At airplane loading", but it is trivial to extend the system with more readers-laptops and thus locations. For this reason we would argue that our system meets this requirement.

FR2. This is met, a passenger can ask not to have his luggage tracked. The best way to implement this would be to not enroll the tag at this point. This way there is no need to produce both tagged as well as non-tagged luggage labels, since a tag that is not enrolled will be ignored by the system. There is some lingering threat here that the not enrolled tag can still be tracked by illegitimate readers. If you want to counter this there would need to be untagged labels, or simply pierce the label at the spot of the RFID tag to disable it. There is a device capable of destroying these tags remotely called a "RFID Zapper" [61]. To the best of our knowledge it was never commercially produced, but the specifications for creating it are out there. Such a device could be used to more easily permanently disable the tags.

FR3. Our prototype system uses tags/readers that are cheap enough so that this requirement is not met. The tag can be read from approximately 1cm, which means that the reader and tag would have to positioned very carefully in order for them to always connect. In a real system we would recommend using HF tags, with a range of between 10cm and 1m. This

would leave plenty of wiggle room for the tag to be positioned anywhere on the bag.

FR4. Each bag has a unique tag and its location is tracked separately, so this requirement is met.

FR5. Removing a bag from the system without alerting the bag's owner can be done easily, simply physically remove the bag from the baggage handling system but keep the baggage-label. The baggage-label can then proceed along the normal route, and be scanned along they way. The passenger will thus be updated of the progress of the tag, instead of the bag, and there is no way for him to be determine the difference. If removing the label is undesirable (for instance because it could later be discovered by the passenger or other parties) it is even possible to simply directly manipulate the database, since this is under the direct control of the airport.

7.1.2 Non-Functional Requirement Evaluation

NFR1. The tags we have chosen are passive, and will be physically unable to send any radio waves, without being activated. This way they will be unable to break airline regulations of sending radio waves while on board.

NFR2. Most RFID readers can read from ten to a few hundred tags a second, so there is no need for this to slow down the existing baggage handling systems. The only slow down will be at check-in desk, where passengers will be handed their QR-code and login credentials. We expect that with wider adoption and familiarity with the system this slow-down will also become minimal.

NFR3. As noted before the readers can handle many tags a second so they will not become a bottle-neck, neither will the program nor its storage. The system can thus easily handle the same amount of bags as the existing baggage handling systems.

NFR4. For the most part there is no need for personnel to interact with the system, with the exception of handing over the qr-code and login credentials upon check-in. This could also easily be handled by an automatic terminal, but we feel that for early adoption it might be better to leave this to the attendant. This way any questions, for instance about privacy, can immediately be answered. After logging into the application there is only a screen with information on it, which automatically updates, so passengers should have little difficulty using this.

NFR5. Baggage already has to comply with size regulations and has to able to have a label attached to it. Our system adds no new requirements in this regard and can thus accept any bag that the existing baggage handling system can.

NFR6. The regular baggage handling system is not influenced by this tracking system and can proceed regardless of any issues with the tracking system. The only possibility is that repairs have to be made to the tracking system for which the baggage handling system would have to be temporarily

suspended. This should be a rare enough occurrence not to cause any real issues, and even when it does happen the repairs could be done during the downtime of the baggage handling system.

NFR7. Each piece of baggage only needs a tag which costs a few cents at most, we consider this to be sufficiently low.

7.1.3 Security Scenario Evaluation

SS1. The impact of an enrolled tag getting lost in our system is small. The tag contains no information other than its ID, which in itself is harmless. So no sensitive information get leaked.

An attacker which obtains this lost or stolen tag could in theory duplicate it, but we see no motivation or attack scenario for this. Other systems might be DoS'ed this way, but our system will simply scan the tags and update locations accordingly.

Furthermore tags can simply be unenrolled, and this way made harmless. This way the only real impact comes from the passenger which originally would have used the tag being assigned a new one, or simply missing the functionality of tracking their luggage for one flight.

SS2. We would first like to mention that these readers are in well guarded areas, namely the baggage handling areas in airports and the runways. This is also noted in Prerequisite P4 in Section 5.1. If these areas are breached, attacks much more severe than the ones we describe and unrelated to our system, could be committed. This means that the chance for readers to be stolen is quite low.

Due to the simple nature of our tags and readers there is no real impact of a reader getting lost or stolen, other than it needing to be replaced. Our readers contain no key material, and can only be stolen by being physically disconnected from the system, since they can not communicate independently.

In order to do some more interesting analysis we will also take a look at the case where the readers contain key information, as well as them staying connected to the system after being stolen. The key material would concern both the keys necessary for authentication with the tag as well as the back-end application. This would be true for the majority of readers you would use for a fully implemented system.

If there is some key material (or even a master key) on these readers then the picture changes. An attacker which manages to recover this key might be able to decrypt all information sent between tags and readers, or create his own authenticated readers. As has been shown in research like [51] if there are vulnerabilities in the implementation of for instance authentication, it is possible to in trivial time recover a master key protocols. Even if the protocol has absolutely no weaknesses there are still ways of recovering this key if you have unlimited access to the physical device, such as side-channel analysis. The best choice would be to make sure that the information on one reader being lost does not break the entire system. This can be done by making sure that these keys are unique per reader

and that there is some way for new tags to no longer trust the key of a lost reader. At that point old tags can be disenrolled, and the lost reader in theory be made harmless.

More problems arise when a reader can be stolen without disconnecting it from the system. An attack would be possible where you have the reader continuously send updates on as many id's as he can, and effectively DOS a part of the system this way. This can be mitigated by making sure that the connection to readers can be disconnected from the application server side, which our tracking application can, so that these readers can be ejected from the system.

SS3. This is similar to the scenario in SS1, but here a tag which remains in the system becomes malicious. As noted before the tag contains no sensitive information and can not do a lot of damage. The tags used in our system are not programmable, and thus they can not simply "become malicious". What could happen though is for an attacker to supply his own pre-programmed tag.

A tag could lie about its ID, but due to the small chance of the chosen ID colliding with that of a currently enrolled tag this would in effect only impede the tracking of the malicious tag itself. This assumes that IDs are chosen randomly and not ordered, which would be a good practice regardless. Note that this attack is not possible if the tag has to authenticate itself, instead of only identification.

SS4. A malicious reader could quite easily temporarily disrupt the service. We define a malicious reader as a legitimate reader in the system, which has somehow been corrupted by an attacker to attack the system. As already noted in SS2, this is difficult to manage in reality, due to the readers being in a controlled and protected environment. For the rest of this evaluation we will assume this has somehow been done and that the primary goal of the attacker is to disrupt the system.

The reader can do all kinds of things to make the data being shown to the passenger unreliable. It could for instance put random delays on sending the IDs to the application server, omit sending some, or even simply send stored IDs. This all serves to obfuscate which data is real, and which isn't. If done correctly this is attack would require some time to detect, since the malicious reader would seem to be functioning as a normal reader.

The reason the attack described above is a threat in this scenario, and not in SS2 is that in SS2 you know which reader is maliciously sending wrong information, while here you do not. For a considerable amount of time you might not even know that you are being attacked in the malicious reader scenario, since the system would function but simply produce unreliable data. In SS2 you know that a reader has to be removed from the system.

This scenario becomes a lot more dangerous if there is some kind of master material on the reader, as explained in scenario SS2.

SS5. The readers are a in a secured area per Prerequisite P4. Thus the only thing that a attacker can control is what is being read by the readers.

The most obvious way of conducting a DoS attack on the system is simple putting hundreds or thousands of tags in your bag, and hoping to overload the system this way. Here there are three factors that could become overloaded: the reader, the application or the connection between these two.

Industrial RFID readers can read hundreds of tags a second, so most of the tags within the bag will be read. Even if there are too many tags for the reader to keep up with, or a cheaper reader which reads fewer tags is used, this will cause no long term problems. It will simply miss some of them. Due to these tags not being part of the system nothing of value is lost. The only possible problem here is that the tags of the luggage immediately next to the attacking one might be missed as well.

The connection between the reader and the application could in theory be overloaded by forcing the reader to send too much data, but due to the small amount of data contained in a RFID tag we do not find this to be a realistic attack.

Lastly the tracking application itself. The steps undertaken for each message received by the application are simple enough that we feel that it could handle every reader sending its maximum amount of read tags at any one time. We verified this by having our reader laptops send as many messages to the server as they were able, which were properly handled. Do note that there is always the threat of the server upon which this application is running getting DDoS'ed, but this is a more general problem which we consider out of scope.

Due to the analysis above we feel our system is sufficiently robust against (D)DoS attacks.

There is one more attack that we do feel still warrants some attention and that is disabling the system by using a radio wave jammer. These jammers are cheap to produce and there are plenty of sources online on how to create your own. When one of these jammers with sufficient range is used the system will no longer function, since the tags cannot be read.

There are no real technical defenses against this kind of attack, since you are effectively making the communication medium (radio waves) unusable. This is an area where we have to trust upon the physical security already in place at airport, luckily these should be more than well enough equipped to locate and deal with these kinds of jammers.

SS6. This scenario looks at how easily the system can be used to DoS other systems. If the readers get compromised, and if they are able to communicate directly or indirectly through the Internet then there is a risk of them being turned against other systems. Due to the relatively low number of readers (a few dozen at most), we feel this is not a major concern. Regardless since there is no reason for the readers to communicate directly with the Internet this should be restricted as much as possible, so they can only directly communicate with the application server.

SS7. The impact of an enrolled tag failing is minimal, it will simply stop updating its location. This might cause some confusion for passengers, but

this should be a rare enough occurrence that it doesn't warrant to much attention.

One detail here is that if a tag fails midway through the process it won't reach its destination and thus never get disenrolled. The easiest way to fix this is simply to have enrolled tags wiped after the last flight of the day, or after a set amount of time.

SS8. A reader failing would mean that one of the key-points being tracked in the system will no longer update. This should be easy to detect, since one of the reader will cease to send messages. Until this reader is repaired or replaced the tracking system should be considered non-functional, but there will be no lasting effects.

SS9. As described in Subsection 7.1.1 Requirement 5., the answer to this scenario is yes, a bag can be silently removed. This takes care of the threats attributed by the *Terrorist* attacker model (see Section 5.5.1), where they would use the system to track when their baggage reaches a certain destination to aid in their terror attack. Of course this does not mean that the system can in any way detect which bag would have to be removed, but if this is detected by the regular means at an airport, then the system will not hinder the process by alerting the terrorist of his bag being seized.

7.1.4 Security Requirement Evaluation

In the previous subsection we evaluated how the system performs under certain specific attack scenario's. This section shows for the more general security requirements of Subsection 5.5.3 to what extend they are met by our system.

SR1. This requirement is met, tags are disenrolled upon reaching their destination, as well as at every startup of the system.

SR2. This requirement is met, tags contain no information other than their ID.

SR3. This requirement is met, in the current application there is no way to link an ID to a passenger, other than by physically observing who receives what tag.

SR4. This requirement is met, the messages send by the application have the following form: *Read ID + ReaderID*.

SR5. In our current system this is met by having a very low read range, so that an attacker would need to be almost touching the bag in order to identify/trace it. If in a system RFID technology with a larger read range is used then there would need to be some safeguards such as mutual authentication and confidentiality in order to meet this requirement.

7.2 Compared with other solutions

Here we will compare our system/solution with the ones described in Chapter 3.

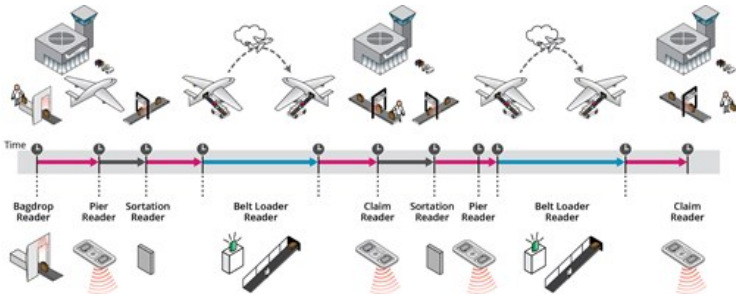

Figure 7.1: BagTrack RFID readers, from `http://www.lyngsoesystems.com/en/solutions/airports-airlines/`.

7.2.1 TRACE ME

As noted in its description in Section 5.5.2 on page 28 we feel TRACE ME is more of a service offered to passengers than a technological system. It satisfies roughly the same goal of tracking your luggage, but it is very much dependent on implementations at airports for how precise this can be done.

Furthermore we feel that the privacy issues with a system like TRACE ME are a major concern. The barcode tag which you attach to your luggage is unique and links directly to your person, meaning that outside parties can trivially use it to track you. An advantage for TRACE ME here is that they use traditional barcodes, so that line of sight is necessary in order to read them.

7.2.2 BagTrack

BagTrack is basically our system implemented in the real world. From the passenger's perspective there are a few small differences such as the defined key-points or how a passenger is alerted to the location of their bag but these are quite minor. BagTrack is a more complete overhaul of existing baggage handling systems then our own. Whereas our system seeks to work alongside existing sorting systems, BagTrack seeks to replace (and improve) these systems. In Figure 7.2.2 you can see at what points during the journey the BagTrack system has set up RFID readers, this can be compared to our own in Figures 6.1 and 6.2.

There is little indication of whether or not security was major concern during the design of this system, so this might be an area where our solution holds an advantage. More thorough information about the BagTrack system would be necessary to guarantee this. On that note there is also the choice of technology. As stated in Subsection 3.2.4, we believe BagTrack uses UHF RFID tags. We recommend using either LF or HF tags, because of their reduced ranges, which complicates unwanted tracking. In a real world scenario there probably would be no choice but to use UHF, since this is the standard set by the International Air Transport Association, but we feel this choice should be reversed. The range with UHF is sufficient to easily allow unwanted tracking. We also expect it complicates the sorting process in an unnecessary way, since any reader will be reading multiple tags at the same time, and be unable to pinpoint which tag belongs to what bag.

This solution is a good indication that systems such as the one we proposed and implemented can function in the real world.

7.2.3 EVIATE

The most notable difference between EVIATE and our own system lies in the way that this solution allows you to track your bags. Our system only alerts you when your bag reaches certain key-points. The EVIATE system on the other hand will through using GPS allow you to continuously pinpoint the location of your bags. A side note we would like to make here is that we are not aware of just how the producers overcame the shortcomings of GPS in hard to penetrate buildings (such as airports), if they did indeed overcome this.

Furthermore EVIATE has some additional features which our solution does not, such as alerting you when your bag is opened.

These advantages come with a price though. The cost of a device such as the EVIATE system is high enough that this is truly a consumer product, instead of our solution where the airport/airline offers this to all passengers. Since the system is not released yet some of the facts mentioned might change.

7.2.4 Track & Go

Track & Go fills a different niche than our system. Whereas our system is about knowing where your bag is after handing it off at the airport, Track & Go seeks to make sure you do not lose it before this time.

We originally had some concerns about unauthorized tracking with this system. The BLE beacons used in this system have a range of 70 meters and send their identity to other users of the system within this range. This is more than enough to track someone in an inconspicuous way, possibly even by just setting up readers at some strategic points. Discovering that this technology uses Google Eddystone Ephemeral Identifier alleviated most of these concerns. As long as this implementation is secure only the back-end server will be able to translate this ephemeral identifier to the actual identity of the system owner.

Chapter 8

Conclusion and future work

In this chapter we will give a conclusion to our thesis, by summing up what our contributions and important findings were. We will also look at what work is still to be done in Future Work, and what other avenues of research related to this subject could be pursued.

8.1 Conclusion

One of the major contributions of this research is our technology comparison in Chapter 4. It gives an easy to understand overview table, as well as more in depth information per technology in their relevant sections. The aspects we examined are both general technical aspects such as range, cost and throughput as well as the security aspects authentication, unauthorized tracking countermeasures and confidentiality. We found that the differences per technology were large, but that each technology fulfilled a special niche, especially when considering cost. This chapter will be useful for people seeking to implement their own (IoT) systems as well as academics wanting to expand their knowledge of relevant technologies in the field of IoT.

In our comparison we found some interesting differences between Lora and SigFox, which mostly stem from their different business models. Lora provides technology and a protocol which can setup a world-wide network, but leaves the implementation of this network to its users. SigFox on the other hand sells both the device to connect to the network, as well as setting up the network itself (through partners), and providing this network to its customers. This leads to differences like paying subscription costs to SigFox, as well as differences in coverage between the two. Lora's more open design also means that there is a lot more information available about the workings of Lora, while SigFox keeps most of its proprietary protocol a mystery.

Here we would again like to remind the reader to be careful when making a choice for a certain technology. As we also experienced while gathering this information, the promises made by vendors of the technologies we compared differ greatly. Certain implementations might miss certain security features for instance, or even offer more than the base technology. Range is another attribute which differs greatly per implementation, with many implementations not even approaching the theoretical maximum of the technology. For this reason it is

always wise to do some independent research when selecting a vendor after making a technology choice.

The other part of our research questions was examining the differences between traditional IT systems and IoT systems. We found five major differences:

1. Technical limitations of IoT devices.
2. Physical environment playing a larger role. Many components of an IoT systems will not be in a controlled environment.
3. Lack of security-focus during design and implementation process.
4. IoT devices being an interesting target for attackers as tools for DDoS attacks.
5. The use cases of IoT systems are more often privacy sensitive.

For the discussion of these see Section 2.1.1.

During the implementation of our own system we encountered these same differences. We encountered these differences as follows (numbers correspond to the ones above):

1. It would have been possible to use RFID tags and readers with security features such as authentication and confidentiality. The cost of these tags and readers would have been much higher though, due to the cost implementing these on such limited hardware. Due to this cost we decided on using cheaper LF RFID tags and readers and relying on limited range instead.
2. In the evaluation of our requirements we relied on the fact that our readers are in a controlled environment, and not easily accessible. If this were not the case, additional measures would be necessary to safeguard the system.
3. We explicitly considered security in our design process, in order to counter this. An example of this is how we consider short ranges to be an advantage, due to them increasing the difficulty of eavesdropping.
4. Less relevant for our system. The "Things" of our system of which we have many, tags, are not able to DDoS another system.
5. Our system is a good example of something seemingly innocent which can have many privacy implications as described in Section 5.5.

We found the implementation of our use case to be valuable, but the supporting literature research to also be crucial. The value of the implementation lay mostly in guiding what questions to pursue in our literature research. The choice of technology is the best example of this. The technology comparison chapter (Chapter 4) came about because we needed a technology for our own use case. We found no comparison in existing literature which would meet our demands so we created our own. So although the implementation process did not answer any questions in of itself, it did help us create the questions which we end up answering and confirmed some of our findings.

We do feel that despite the positive results the implementation brought, that the amount of time that was spent on this was not in proportion to these results.

The chapters of this thesis also reflects this: Chapters 5,6 and 7 are all specific to our system and design/implementation process. While these chapters do contain interesting information, it is less relevant to a more general case. We feel the earlier chapters are more useful in this regard. If in our research we placed less focus on implementing our use case and more on the literature research, that these general chapters could have been improved or extended.

An interesting observation is that our main use case, tracking a bag, is also our main security concern: the unauthorized tracking of a person or bag. We feel that for most IoT systems unauthorized tracking will be one of the major threats, because of obvious privacy concerns as well as the technology's tendency to allow precisely this. Use cases which have nothing to do with tracking something or someone might still use a technology which allows unauthorized tracking. An example of this would be tracking drivers by their vehicles sporadically using SigFox to send their fuel-levels to a central server. Bad press related to these kinds of incidents have to ability to greatly undermine the acceptance of these technologies. In our case we tackled this by making sessions short-lived and the ids unlinkable to real identities, but each system will require its own solution and attention spent on this problem.

During our security requirement analysis we also discovered some threats less obvious than tracking. For instance the need to remove bags without the owner knowing to counter terrorist threats, or the system being used to smuggle drugs in other passengers' bags. This strengthens the case for systems requiring specific security attention during their design phases, because otherwise these might have been missed.

8.2 Future Work

IoT security is a large subject, with many different facets that can be examined and improved. This section gives some suggestions as to how our own research could be extended or improved.

In our technology overview, we reviewed what security primitives (AES for instance) were used, and whether or not they were secure. There is of course no guarantee that even if a secure primitive is used that the protocol implemented by the technologies is also secure. This research could be extended by taking one or several of these technologies and trying to piece together their protocol and evaluate this. From our own preliminary findings in this regard it seems far from straightforward to discover these protocols, since they are often proprietary and not described in any real detail. This could be tackled by working directly with the producers, or doing some reverse engineering/side-channel analysis to piece it together yourself.

Our technology overview could also be extended with more technologies, or other aspects of these technologies. Some example technologies that could be examined are: Wi-Fi, Zigbee, Satellite, HayStack or traditional Bluetooth. Other aspects that might be examined could for instance be wall penetration grade, battery life, ease of disruption or non-repudiation.

Some of the information could also be refined, mainly some of the security features of SigFox. As stated in Section 4.6 our current sources of information on SigFox are questionable, and contain little information about just what cryptographic protocol is used. The problem being that to our knowledge there

is no real technical specification, or something resembling one, available for SigFox. These technical details could be made more clear by means of either reverse-engineering, or by trying to work together directly with SigFox, if they are willing to reveal this information.

In our Chapter 3 (Related Work) we already touched upon a blog post by Bruce Schneier [55], where he outlines nineteen different security guidelines for handling or implementing IoT systems. This overview is of course already very useful as a general view on what guidelines are available but still leaves the reader with nineteen different documents and hundreds of pages of technical text. It would be interesting to attempt to combine these guidelines together, see where they differ or on what they agree. This way a much more manageable general guideline could be created. There is also the possibility of finding interesting conflicting views, finding out what causes these could lead to more interesting insights.

Repeating the research, but taking a very different use case would lead to some interesting new questions being asked. As we noted, many of the things we ended up diving into were a result of the system we were implementing. Choosing a use-case sufficiently different that you would, for instance, use Lora instead of RFID could lead to new interesting questions. We also feel that choosing a larger scale project, instead of the prototype we developed, could lead to new insights in the same way.

Another interesting angle would be to take an IoT use case and find different implementations of this system. See Section 3.2 for an example of the ones we found for our use-case. After gathering these analysis could be done on how these systems would be vulnerable to attack, this could even be extended into trying to turn this into workable attacks. We feel that this could give a good overview of flaws often present in IoT systems.

Bibliography

[1] Lanzisera, S., Weber, A. R., Liao, A., Pajak, D., & Meier, A. K. (2014). Communicating power supplies: Bringing the internet to the ubiquitous energy gateways of electronic devices. *IEEE Internet of Things Journal*, 1(2), 153-160.

[2] Chen, S., Xu, H., Liu, D., Hu, B., & Wang, H. (2014). A vision of IoT: Applications, challenges, and opportunities with China's perspective. *IEEE Internet of Things journal*, 1(4), 349-359.

[3] Wang, D., Ni, Y., Chen, B., Cao, Z., Tian, Y., & Zhao, Y. (2015, August). Wind Speed and Direction Predictions Based on Multidimensional Support Vector Regression with Data-Dependent Kernel. *In International Conference on Cloud Computing and Security* (pp. 427-436). Springer International Publishing.

[4] Sun, Q., Li, H., Ma, Z., Wang, C., Campillo, J., Zhang, Q., Wallin, F., & Guo, J. (2016). A Comprehensive Review of Smart Energy Meters in Intelligent Energy Networks. *IEEE Internet of Things Journal*, 3(4), 464-479.

[5] Catarinucci, L., De Donno, D., Mainetti, L., Palano, L., Patrono, L., Stefanizzi, M. L., & Tarricone, L. (2015). An IoT-aware architecture for smart healthcare systems. *IEEE Internet of Things Journal*, 2(6), 515-526.

[6] Jin, J., Gubbi, J., Marusic, S., & Palaniswami, M. (2014). An information framework for creating a smart city through internet of things. *IEEE Internet of Things Journal*, 1(2), 112-121.

[7] Lora Alliance. (2015). A technical overview of LoRa and LoRaWAN. *White Paper*. Retrieved from: `https://www.lora-alliance.org/portals/0/documents/whitepapers/LoRaWAN101.pdf`

[8] Future Proofing the Connected World. (2016). *Online Article* Retrieved October 17, 2016, from https://cloudsecurityalliance.org/download/future-proofing-the-connected-world/

[9] Juels, A. (2006). RFID security and privacy: A research survey. *IEEE journal on selected areas in communications*, 24(2), 381-394.

[10] Landt, J. (2005). The history of RFID. *IEEE potentials*, 24(4), 8-11.

[11] Want, R. (2006). An introduction to RFID technology. *IEEE Pervasive Computing*, 5(1), 25-33.

[12] Vogt, H. (2002, August). Efficient object identification with passive RFID tags. In *International Conference on Pervasive Computing* (pp. 98-113). Springer Berlin Heidelberg.

[13] Weinstein, R. (2005). RFID: a technical overview and its application to the enterprise. *IT professional*, 7(3), 27-33. ISO 690

[14] Atzori, L., Iera, A., & Morabito, G. (2010). The internet of things: A survey. *Computer networks*, 54(15), 2787-2805.

[15] Smiley, S. (2016, March 4). Active RFID vs. Passive RFID: What's the Difference? *Blog Post*. Retrieved October 26, 2016, from`http://blog.atlasrfidstore.com/active-rfid-vs-passive-rfid`

[16] Mallik, N. (2014, August 5). Beacon FAQs: Everything you need to know. *Blog post*. Retrieved February 13, 2017, from`https://blog.beaconstac.com/2014/08/beacon-faqs-everything-you-need-to-know`

[17] Saffo, P. (1997). Sensors: the next wave of innovation. *Communications of the ACM*, 40(2), 92-98.

[18] Menn, J., Finkle, J. & Volz, D. (2016, October 21). Cyber attacks disrupt PayPal, Twitter, other sites. *Reuters*. Retrieved from `http://www.reuters.com/`

[19] River, W. (2015). Security in the Internet of Things. *White Paper. Wind River Systems, Tech. Rep.*. Retrieved from `http://www.windriver.com/whitepapers/security-in-the-internet-of-things/wr_security-in-the-internet-of-things.pdf`

[20] Federal Trade Commission. (2015). Internet of things: Privacy & security in a connected world. *Washington, DC: Federal Trade Commission.*. Retrieved from `https://www.ftc.gov/system/files/documents/reports/federal-trade-commission-staff-report-november-2013-workshop-entitled-internet-things-privacy/150127iotrpt.pdf`

[21] Gomez, C., Oller, J., & Paradells, J. (2012). Overview and evaluation of bluetooth low energy: An emerging low-power wireless technology. *Sensors*, 12(9), 11734-11753.

[22] Amsterdam Airport Schiphol first Airport in Europe with full beacon coverage. (2016, July 02). *Online Article*. Retrieved November 15, 2016, from `http://www.schiphol.nl/SchipholGroup1/NieuwsPers/Persbericht/AmsterdamAirportSchipholFirstAirport/InEuropeWithFullBeaconCoverage.htm`

[23] Zimmerman, T. (2016, March 4). Essential Best Practices for Tracking Critical Assets Using RFID. *Gartner March 2016*. Retrieved from `http://www.gartner.com/document/3239618?ref=solrAll&refval=176884688&qid=5a9091b8166a429e92fe1425ff98ba1d`

[24] Contu, R., Middleton, P., Perkins, E. & Akshay, L. (2016, April 7). Forecast: IoT Security, Worldwide, 2016. *Gartner April 2016*. Retrieved from `http://www.gartner.com/document/3277832?ref=solrAll&refval=176884805&qid=70e0ea20467191106d883d1fac0a98ad`

[25] Comparison of NFC & Barcode to RFID Inventory Tracking — RFID4U. (2016, September 7). *Online Article* .Retrieved November 16, 2016, from http://rfid4u.com/comparison-of-rfid-nfc-and-barcode-for-inventory-tracking-part-2-nfc-barcode/

[26] Haselsteiner, E., & Breitfuß, K. (2006, July). Security in near field communication (NFC). In *Workshop on RFID security* (pp. 12-14).

[27] Trasher, J. (2013, October). RFID versus NFC: What's the difference between NFC and RFID? *Blog Post*. Retrieved November 16, 2016, from http://blog.atlasrfidstore.com/rfid-vs-nfc

[28] Feldhofer, M., Dominikus, S., & Wolkerstorfer, J. (2004, August). Strong authentication for RFID systems using the AES algorithm. In *International Workshop on Cryptographic Hardware and Embedded Systems* (pp. 357-370). Springer Berlin Heidelberg.

[29] Kieseberg, P., Leithner, M., Mulazzani, M., Munroe, L., Schrittwieser, S., Sinha, M., & Weippl, E. (2010, November). QR code security. In *Proceedings of the 8th International Conference on Advances in Mobile Computing and Multimedia* (pp. 430-435). ACM.

[30] Takiishi, K. & Foong, K. (2016, January). Market Trends: Low-Power Wide-Area Access Technologies for the IoT. *Gartner January 2016*. Retrieved from `http://www.gartner.com/resources/293500/293516/market_trends_lowpower_widea_293516.pdf`

[31] Gubbi, J., Buyya, R., Marusic, S., & Palaniswami, M. (2013). Internet of Things (IoT): A vision, architectural elements, and future directions. *Future Generation Computer Systems*, 29(7), 1645-1660.

[32] November Five - How to connect the internet of things: Lora vs Sigfox. (n.d.). *Online Article*. Retrieved December 14, 2016, from `https://novemberfive.co/blog/internet-of-things-lora-vs-sigfox/`

[33] Riahi, A., Natalizio, E., Challal, Y., Mitton, N., & Iera, A. (2014, February). A systemic and cognitive approach for IoT security. In *Computing, Networking and Communications (ICNC), 2014 International Conference on* (pp. 183-188). IEEE.

[34] *Specification of the Bluetooth System Core Version:5.0* [Vol 0, Part A] (2014, December 2). Technical Specification. Retrieved from: `https://www.bluetooth.com/specifications/adopted-specifications`

[35] Miller, R. (2016, March). LoRa Security: Building a secure LoRa solution. *White Paper. MWR Labs*. Retrieved from: `https://labs.mwrinfosecurity.com/assets/BlogFiles/mwri-LoRa-security-guide-1.2-2016-03-22.pdf`

[36] Ducrot, N. et al., (2016, April). LoRa Device Developer Guide. *Orange*. Retrieved from: `https://partner.orange.com/wp-content/uploads/2016/04/LoRa-Device-Developer-Guide-Orange.pdf`

[37] Oudot, F. (2015, December). Sigfox Data Security. *Online forum comment*. Message posted to: http://ask.sigfox.com/questions/155/data-security.html

[38] Rakon Thinxstra, (2016, March). Rakon Thinxtra Sigfox Your Questions Answered. *Company sale pitch* Retrieved from: http://www.rakon.com/component/docman/doc_download/499-rakon-thinxtra-sigfox-your-questions-answered?Itemid=

[39] Author anonymous, (2016, April 13). Make your IoT design on Sigfox or LoRa ? *Blog post*. Retrieved from https://www.disk91.com/2016/technology/internet-of-things-technology/make-your-iot-design-on-sigfox-or-lora/

[40] Mulliner, C. (2009, March). Vulnerability Analysis and Attacks on NFC-Enabled Mobile Phones. In *ARES* (pp. 695-700).

[41] Van Damme, G., Wouters, K., & Preneel, B. (2009). Practical experiences with NFC security on mobile phones. *Proceedings of the RFIDSec*, 9, 27.

[42] Broek, F. V. D. (2011, January 3). Eavesdropping on GSM: state-of-affairs. *arXiv preprint arXiv:1101.0552*.

[43] ISO/IEC, (2012). ISO/IEC 29167-1:2012 Information technology – Automatic identification and data capture techniques – Part 1: Air interface for security services and file management for RFID architecture. Geneva, Switzerland: ISO/IEC.

[44] ISO/IEC, (2008). ISO/IEC 18000-1:2008 Information technology – Radio frequency identification for item management – Part 1: Reference architecture and definition of parameters to be standardized. Geneva, Switzerland: ISO/IEC.

[45] ISO/IEC, (2016). ISO/IEC 14443-1:2016 Identification cards – Contactless integrated circuit cards – Proximity cards – Part 1: Physical characteristics. Geneva, Switzerland: ISO/IEC.

[46] ISO/IEC, (2010). ISO/IEC 15693-1:2010 Identification cards – Contactless integrated circuit cards – Vicinity cards – Part 1: Physical characteristics. Geneva, Switzerland: ISO/IEC.

[47] ISO/IEC, (2010). ISO/IEC 18092:2013 Information technology – Telecommunications and information exchange between systems – Near Field Communication – Interface and Protocol (NFCIP-1). Geneva, Switzerland: ISO/IEC.

[48] Habraken, R., Dolron, P., Poll, E., & De Ruiter, J. (2015, June). An RFID skimming gate using Higher Harmonics. In *International Workshop on Radio Frequency Identification: Security and Privacy Issues* (pp. 122-137). Springer International Publishing.

[49] Derouin, R. (2016). Getting started on SIGFOX. *Powerpoint slides*. Retrieved from: http://www.slideshare.net/RyanDerouin/get-started-on-sigfox.

[50] Stallings, W. (1987). *Handbook of computer-communications standards; Vol. 1: the open systems interconnection (OSI) model and OSI-related standards*. Macmillan Publishing Co., Inc..

[51] Garcia, F. D., de Koning Gans, G., Muijrers, R., Van Rossum, P., Verdult, R., Schreur, R. W., & Jacobs, B. (2008, October). Dismantling MIFARE classic. In *European Symposium on Research in Computer Security* (pp. 97-114). Springer Berlin Heidelberg.

[52] Petajajarvi, J., Mikhaylov, K., Roivainen, A., Hanninen, T., & Pettissalo, M. (2015, December). On the coverage of LPWANs: range evaluation and channel attenuation model for LoRa technology. In *ITS Telecommunications (ITST), 2015 14th International Conference on* (pp. 55-59). IEEE.

[53] Centenaro, M., Vangelista, L., Zanella, A., & Zorzi, M. (2016). Long-range communications in unlicensed bands: The rising stars in the IoT and smart city scenarios. *IEEE Wireless Communications*, 23(5), 60-67.

[54] Taplett, N. (January, 2016). *White paper*. Bluetooth low energy security. Retrieved from: `https://www.bluetooth.com/~/media/files/specification/bluetooth-low-energy-security.ashx?la=en`.

[55] Scheiner, B. (February, 2017). *Blog post*. Security and Privacy Guidelines for the Internet of Things. Retrieved from:`https://www.schneier.com/blog/archives/2017/02/security_and_pr.html`

[56] Kalden, R., Meirick, I., & Meyer, M. (2000). Wireless Internet access based on GPRS. *IEEE personal communications*, 7(2), 8-18.

[57] Hasan, K. S., Rahman, M., Haque, A. L., Rahman, M. A., Rahman, T., & Rasheed, M. M. (2009, March). Cost effective GPS-GPRS based object tracking system. In *Proceedings of the international multiconference of engineers and computer scientists* (Vol. 1, pp. 18-20).

[58] Samfat, D., Molva, R., & Asokan, N. (1995, December). Untraceability in mobile networks. In *Proceedings of the 1st annual international conference on Mobile computing and networking* (pp. 26-36). ACM.

[59] Kramp, T. & Sornin, N. (2015, March). *Technical Specification*. LoRa Specification. Retrieved from: `https://www.lora-alliance.org/portals/0/specs/LoRaWAN%20Specification%201R0.pdf`.

[60] Mijnen, J. (2014, January). *Blog post, Dutch*. Baggage afhandeling. Retrieved from: `https://roundlet.tweakblogs.net/blog/9864/bagage-afhandeling.html`.

[61] Collins, J. (2006). RFID-Zapper shoots to kill. *RFID Journal*. Retrieved from: `http://www.rfidjournal.com/articles/view?2098`.

Appendices

.1 RFID Tags and Readers

The tags and readers used were the following: Neuftech ID Card Reader USB 125KHZ RFID EM4100/EM4001/TK4100 Contactless Card Reader Plug and Play With 5pcs keys. Available at `https://www.amazon.co.uk/Neuftech-Reader-125KHZ-EM4100-Contactless/dp/B018OYOR3E/ref=sr_1_8?s=computers&ie=UTF8&qid=1481276939&sr=1-8&keywords=rfid`.

Easy to use plug and play reader and Low-Frequency tags, for a very reasonable price. Reader acts as keyboard and inputs the read ID plus an enter. Reader offers no real other options, contains an audio cue upon reading a tag (has to disconnected to disable). Tags can (with some difficulty) be opened in order to repackage them. Security-wise this set offers next to nothing, only identification, so not recommend for real-world implementation, but it is great for these kinds of projects.

.2 Code

The code for the reader application, central application and user interface (website) can be made available upon request. The reader and central applications concern Java code, while the website is written in PHP with Ajax scripts.