Delft University of Technology
Master's Thesis in Embedded Systems

# From Check-in/Check-out to Be-in/Be-out: BLE-based Automated Journey Payment in Public Transportation

**Jan Jaap Treurniet**

Technolution

embedded software

TU Delft Delft University of Technology

# From Check-in/Check-out to Be-in/Be-out: BLE-based Automated Journey Payment in Public Transportation

Master's Thesis in Embedded Systems

Embedded Software Section
Faculty of Electrical Engineering, Mathematics and Computer Science
Delft University of Technology
Mekelweg 4, 2628 CD Delft, The Netherlands

Jan Jaap Treurniet (1308351)
j.j.treurniet@student.tudelft.nl

19th January 2015

**Author**
Jan Jaap Treurniet (j.j.treurniet@student.tudelft.nl)
**Title**
From Check-in/Check-out to Be-in/Be-out:
BLE-based Automated Journey Payment in Public Transportation
**MSc presentation**
23rd January 2015

**Graduation Committee**

| | |
|---|---|
| prof. dr. K.G. Langendoen (Chair) | Delft University of Technology |
| dr. R.R. Venkatesha Prasad (Supervisor) | Delft University of Technology |
| dr. Z. Erkin | Delft University of Technology |
| ir. W. de Boer | Technolution B.V. |

**Abstract**

The current Dutch *OV-chipkaart* payment system for public transportation is based on RFID cards and requires a lot of manual actions from travellers. In this thesis, a novel automated payment system based on Bluetooth Low Energy (BLE) is proposed.

Energy is an important criterion for smartphone applications. Where multiple wireless connections co-exist, interference may influence performance. These factors must be considered when implementing the proposed system. Simulations and experiments with BLE devices are performed to derive an energy model. This model predicts energy consumption and latency for BLE under influence of interference.

The proposed system must be secure against abuse and protect the traveller's privacy. A performance analysis model and a secure, energy-efficient communication protocol are proposed. The model is applied to find that the daily energy requirement for a typical traveller is 12.6 J or less than 0.1% of a smartphone's battery capacity.

During the conducted research, no threats to the feasibility of the proposed system were found. The proposed protocol can be implemented to prove the practical feasibility. Furthermore, the energy model can be applied to predict performance for other BLE applications.

# Preface

This thesis marks the end of my time as a student in Delft. It reports on the work I performed at Technolution BV for the past 8 months. I enjoyed working on this project, that was security-related, versatile, practical, recognizable and relevant. I hope the results contribute to the development of energy-efficient and secure Bluetooth Low Energy systems and that the day will come that we pay in a completely automated way for our public transportation journeys.

Although the road to a final thesis often appears to be a lonely one, this result could not have been achieved without the help and support of others. I would like to mention some of them here. First I would like to thank my thesis advisors. Willem at Technolution for his guidance and valuable experience, Chayan and VP from the TU Delft for the academical guidance. Furthermore, I would like to thank the colleagues at Technolution, in particular *gewaardeerde collega's* Erik, Michel and Olaf for allowing me in their office. Finally, I would like to thank friends, family and anyone else who provided either support or distraction during this period.

Jan Jaap Treurniet

Delft, The Netherlands
19th January 2015

# Contents

**Preface** **v**

**1 Introduction** **1**
1.1 Issues and challenges . . . . . 2
1.2 Organisation . . . . . 2

**2 System architecture and problem analysis** **3**
2.1 Previous work . . . . . 3
2.2 Proposed system architecture . . . . . 4
2.3 Challenges . . . . . 5
2.3.1 Energy consumption . . . . . 5
2.3.2 Security and privacy . . . . . 5
2.3.3 Scalability and accuracy . . . . . 6
2.3.4 Verification . . . . . 6
2.3.5 Localization . . . . . 6
2.4 Focus in project . . . . . 7
2.5 Wireless technology . . . . . 7
2.5.1 Low energy aspects . . . . . 8
2.5.2 Physical and link layer . . . . . 8
2.5.3 Application layer . . . . . 8
2.5.4 Security and privacy . . . . . 9

**3 Energy consumption and interference** **11**
3.1 Previous work . . . . . 11
3.2 Simulations . . . . . 12
3.2.1 Implementation details . . . . . 12
3.2.2 Simulation results . . . . . 13
3.3 Experiments . . . . . 14
3.3.1 Experimental setup . . . . . 14
3.3.2 Implementation problems . . . . . 18
3.3.3 Experimental results . . . . . 21
3.4 Model development . . . . . 24
3.4.1 Advertising . . . . . 25

3.4.2 Discovery and connection . . . 26
3.4.3 Data transfer . . . 26
3.5 Discussion . . . 26
3.5.1 Model evaluation . . . 27
3.5.2 Implications for proposed system . . . 27
3.5.3 Implications for application developers . . . 27
3.6 Other discoveries . . . 28
3.6.1 Clock drift influence . . . 28
3.6.2 Adherence to specification . . . 28

**4 Communication protocol** **29**
4.1 Functional requirements . . . 29
4.1.1 Use cases . . . 29
4.1.2 Operational requirements . . . 31
4.2 Performance aspects . . . 31
4.2.1 Requirements . . . 31
4.2.2 Performance model . . . 32
4.3 Security and privacy aspects . . . 34
4.3.1 Requirements . . . 34
4.3.2 Introduction to cryptographic techniques . . . 35
4.4 Protocol design . . . 36
4.4.1 Global design . . . 36
4.4.2 Previous work . . . 39
4.4.3 Proposed protocol details . . . 41
4.4.4 Security analysis . . . 45
4.4.5 Cryptographic details . . . 46
4.4.6 Protocol implementation . . . 49
4.4.7 Protocol performance . . . 52

**5 Conclusions and Future Work** **53**
5.1 Future Work . . . 54

**A Measurement results** **59**

# Chapter 1

# Introduction

In 2005, the *OV-chipkaart* payment system was first introduced in The Netherlands. From 2011 this system completely replaced all old ticketing systems and the paper *Strippenkaart*. The main goal of the card was to improve information supply for public transportation companies. Furthermore, it would allow more flexible pricing and more ease of use for travellers. Since the introduction, a lot of articles like the one in Fig. 1.1 have been published. This article quotes the chairman of the Dutch railway company, who says only one check-in should be enough for every journey.

The main critiques to the OV-chipkaart system are the number of check-in/out actions that have to be performed and the way these actions have to be performed. If a traveller takes the bus to the train station, then the train to another city, and finally another bus to his destination, the traveller needs to perform three check-in and three check-out actions just to reach one destination. All these actions have to be performed at different devices, placed at different locations. If one of these actions fails, the traveller either pays too much for the journey, or does not pay at all and risks a fine.

“Eén keer inchecken voldoende in het ov”

NS. Het moet op het spoor snel eenvoudiger en eenduidiger worden voor de reiziger. Dat zegt NS-topman Timo Huges in een exclusief gesprek met Metro. De een jaar geleden aangetreden president-directeur pleit voor simpeler gebruik van de ov-chipkaart, het liefst in het gehele openbaar vervoer. “Dat constante in- en uitchecken kun je bijna niet maken naar de reiziger”, vindt hij. PAGINA 2-3

Figure 1.1: Newspaper article: ‘One check-in sufficient in public transportation.’ (Metro, 17th June 2014).

At the same time, more and more people are using a smartphone that is capable of wireless communication using Bluetooth Low Energy. In this thesis, we propose a 'be-in/be-out' system, an alternative for the traditional 'check-in/check-out' system, where a traveller only has to carry a smartphone and get into a public transportation vehicle. The system automatically takes care of billing for the journey.

## 1.1 Issues and challenges

We performed a preliminary feasibility study of the proposed system and identified a number of research challenges. Within the scope of this thesis, we focus on two main aspects. As the system uses wireless communication and smartphones have a limited energy supply, energy consumption is an important factor while designing it. As many people will potentially use the system at the same time in crowded environments, interference might cause problems. We perform simulations and experiments regarding latency and energy consumption to determine the impact of these factors. Furthermore, the public transportation companies need ensure that they bill only those travellers who actually travelled. At the same time, they need to make sure that all travelling passengers actually pay. All this must be achieved without jeopardizing the traveller's privacy. We propose a protocol to make sure the system meets all these requirements.

## 1.2 Organisation

In Chapter 2, we describe the system architecture and give an introductory description of Bluetooth Low Energy. In Chapter 3, we describe the energy-related simulations and experiments performed followed by their results and we propose a model based on these results. In Chapter 4, we propose a model to assess the performance of the system, a secure protocol for the system and a way to implement this protocol. Our final conclusions can be found in Chapter 5.

# Chapter 2

# System architecture and problem analysis

In this chapter we describe the proposed system architecture and perform a problem analysis for the system. We explain our choice for Bluetooth Low Energy as the wireless technology used in the system and give a description of BLE to provide the background that is necessary to understand the rest of this thesis.

## 2.1 Previous work

We searched for previous work on automated public transportation payment systems and relevant work on other payment systems. Most prevalent systems use NFC-based communication, which is also the mode of operation for the OV-chipkaart. We enlist some of the works that are relevant to our proposed system.

McDaniel *et al.* [40] studied a fully automated system for fare payment in public transportation in 1993. Based on the available RF smartcard technology at that time, they call such a system too costly to implement, but they expect future technology improvements to make it feasible.

Caulfield *et al.* [15] performed a study on the requirements passengers have for a public transportation ticketing system. The study is based on a system with ticket-vending machines, but a relevant conclusion is that one of the important wishes from passengers is to be informed about cost, routes and estimated arrival times.

Benelli and Pozzebon [6, 7] proposed intelligent payment systems for car parks. These systems are however based on either RFID (requiring manual action from the user), location entry by the user or a specialized device built into the car. None of these solutions are feasible for a fully automated public transportation payment system.

In 2013 the OV-chipkaart Graduation Lab took place at the Delft University of Technology. This lab consisted of an initial analysis of the OV-chipkaart [33], followed by three MSc theses on subjects resulting from the analysis. Niermeijer [43] proposed an improved way of presenting check-in devices to travellers. Niks [44] developed and evaluated methods to present travelling and payment information to a traveller real-time during the journey. Joppien [32] proposed improvements to the available card types and procedures.

## 2.2 Proposed system architecture

We present the architecture of the proposed system. A schematic overview of the system can be found in Fig. 2.1.

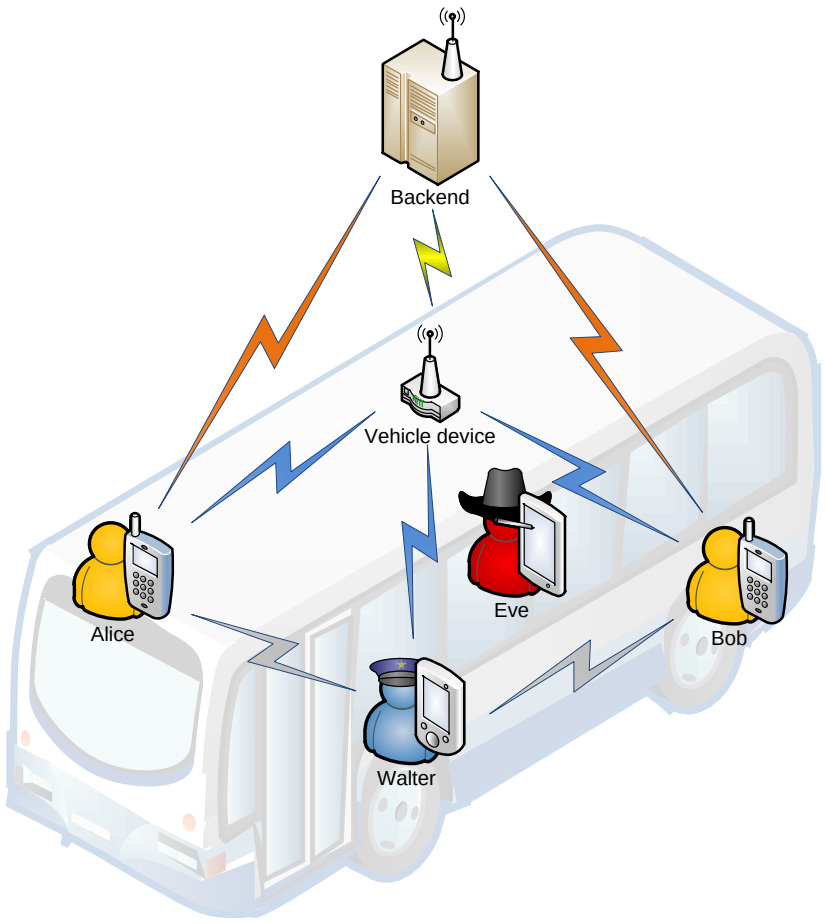



Figure 2.1: Proposed system overview.

The central back-end server keeps a database of all vehicles, railway guards, travellers and performed journeys. The vehicle device checks travellers in and out. In Fig. 2.1, there are two travellers (Alice and Bob), a railway guard (Walter) and a probable attackers (Eve) in the vehicle.

Before the journey, the travellers have to install an application on their smartphone and register themselves with the back-end. They also need to communicate regularly with the back-end to receive authentication information. While travelling, the vehicle device identifies the travellers and makes a local check-in for the authenticated traveller. Furthermore, it may supply information about the schedule and possible changes in the schedule to the traveller's devices. The guard's device connects to the vehicle device to query it about the check-in status of travellers and connects to traveller's smartphones to identify travellers. The attacker can eavesdrop communication or perform any transmission to try to break the system or gain information about travellers. To register journey details, the vehicle device can either connect to the back-end throughout the day (for example via a 3G mobile data connection), or do this at the end of the day.

## 2.3 Challenges

In this section we describe the major research challenges associated with the proposed system. For each challenge we present possible solutions and pointers to previous work where relevant.

### 2.3.1 Energy consumption

Smartphone batteries have a limited capacity, while wireless data transmission tends to be heavy in terms of energy consumption. An increasing number of concurrent travellers might influence energy consumption. Calculations, simulations and experiments are needed to determine how much energy consumption is caused by the proposed system.

### 2.3.2 Security and privacy

Since wireless communication happens through the air, everyone can eavesdrop the communication or even send signals pretending to be a legit device. This means security and privacy measures have to be taken to prevent abuse of the system. A user must only be able to claim to be checked in when he is actually checked in, and only the user himself must be able to perform a check-in. Attacks to the system's availability must be prevented and/or detected. The traveller's privacy must be ensured. A protocol must be designed which implements all features listed above.

### 2.3.3 Scalability and accuracy

Since the system takes care of journey payments, the accuracy of the check-in process is important: if people are billed for journeys they did not make they will complain and if people are not billed for journeys they did make the public transportation companies will miss part of their income. The system will potentially be used by a lot of travellers at the same time, so accuracy for an increasing number of concurrent travellers is important as well. Calculations, simulations and experiments are necessary to determine the accuracy of the system in terms of false-positive and false-negative check-in rates for an increasing number of travellers.

### 2.3.4 Verification

A railway guard needs to be able to verify if a traveller claiming to pay using be-in/be-out is indeed using the system and if he is successfully checked in. This means the guard's device will need to communicate with the traveller's device to exchange data. For this communication we found the following solutions:

- Near-Field Communication (NFC)
- BLE with signal strength analysis
- Optical or audible communication between the devices (exchanged via display/camera and speaker/microphone)
- Manual comparison of some identifier on the two devices

We believe a combination BLE with signal strength analysis and manual comparison of an identifier will result in a feasible system.

### 2.3.5 Localization

To provide accurate billing information, the system needs to be able to make the distinction between a traveller who travels in the vehicle and a person who is within reach of the vehicle device, but not in the vehicle. People can be, for example, waiting at a bus stop for another bus or driving next to the bus in a car. An analysis of the mechanism has to be performed to determine the robustness of this fencing system in terms of false positive and false negative rates. We found the following solutions for the localization problem. One of these or a combination could be used.

- Combination of calibrated vehicle devices and analysis of the moment a person appears for the first time
- Localization using multiple (calibrated) vehicle devices [62]
- Accelerometer-based transportation mode detection [26, 45]
- Other localization methods, such as Cell-ID, WiFi and GPS

We suppose a simple signal strength analysis combined with information about the vehicle position and the moment a traveller is detected for the first time gives enough precision. Real-life experiments are necessary to validate this solution. If this method turns out to be inaccurate, a more intelligent localization method must be implemented.

## 2.4 Focus in project

We made a selection from these challenges and defined the scope for the rest of the research project. We decided to investigate two parts of the system. Energy and interference of Bluetooth Low Energy are covered in Chapter 3. With the results of simulations and experiments performed we are able to assess scalability and accuracy and energy consumption. A secure communication protocol is developed in Chapter 4. This protocol implements all security features that are required. We apply the results from the energy study to determine the actual energy consumption of the system.

## 2.5 Wireless technology

For implementing the automated payment system a wireless communication technology has to be chosen. One of the possible protocols is Bluetooth Low Energy (BLE). Another interesting candidate would be WiFi, which is supported by virtually any smartphone and already present in some trains. One of the goals of this MSc thesis project was to gain experience with BLE. For this reason we selected BLE as the wireless technology for implementing the proposed payment system. In the remainder of this section we describe the Bluetooth Low Energy protocol. The goal is to give the reader a basic understanding of BLE and provide enough background to understand the rest of this thesis. This chapter is by no means an exhaustive description of the protocol. For details the reader is referred to [13, 27, 55].

BLE was first introduced in 2010 with version 4.0 of the Bluetooth Core Specification [12]. BLE is not compatible with Bluetooth Classic (BR/EDR) but many devices, such as smartphones and tablets, are compatible with both BR/EDR and BLE. In 2013 version 4.1 of this specification [13] was released. This update allows devices to function in central and peripheral roles for multiple connections at the same time in any configuration. In 2014 version 4.2 of the specification [14] was released with an increased maximum message size and better privacy protection when using random devices addresses. Currently most devices only support Bluetooth 4.0, so in the remainder of this thesis we will assume this version of the standard. Where changes are relevant we will mention this.

BLE is meant for typical Internet of Things (IoT) applications, where both the available resources and amounts of data are limited. This means BLE is not designed to replace BR/EDR. Both protocols will be used, because both protocols have their own purpose.

### 2.5.1 Low energy aspects

As the name suggests, the goal of Bluetooth Low Energy is to have a low energy consumption. To achieve this goal, a number of measures were taken. We list the most important properties which ensure low energy consumption:

- The protocol is designed asymmetrically. In many cases where two devices are communicating, only one has a limited amount of energy available. In our public transportation case, the user's mobile phone has limited energy available while for the vehicle device energy consumption is less an issue.
- The protocol parameters, such as advertising and connection intervals, can be tuned specifically for an application to achieve the lowest energy consumption, while still offering acceptable latencies.

### 2.5.2 Physical and link layer

At the physical layer BLE uses the unlicensed 2.4GHz ISM band. Many other protocols like WiFi and ZigBee are also using this band. Data is transmitted at a rate of 1 Mbit/s. The band is divided in 40 channels. 3 channels are used for advertising messages and the other 37 for data transmission. During a connection the devices communicate with a constant interval which is called the *connection interval*. This value can be set anywhere between 7.5 ms and 4 s. A channel hopping scheme changes the communication channel every connection interval to mitigate interference.

### 2.5.3 Application layer

We describe how applications can use BLE. We do this for the different phases in a connection: device discovery, connection setup and data transfer.

**Discovery and connection:** An advertising device transmits advertisement packets. These packets are sent with an interval that is incremented with a random delay to avoid repeated collision. A listening device listens for advertising packets to discover devices. Advertisement packets may contain information about the device name or supported services. After transmitting an advertisement packet the peripheral device listens for a short period in which the central device can send a connection request to set up a connection.

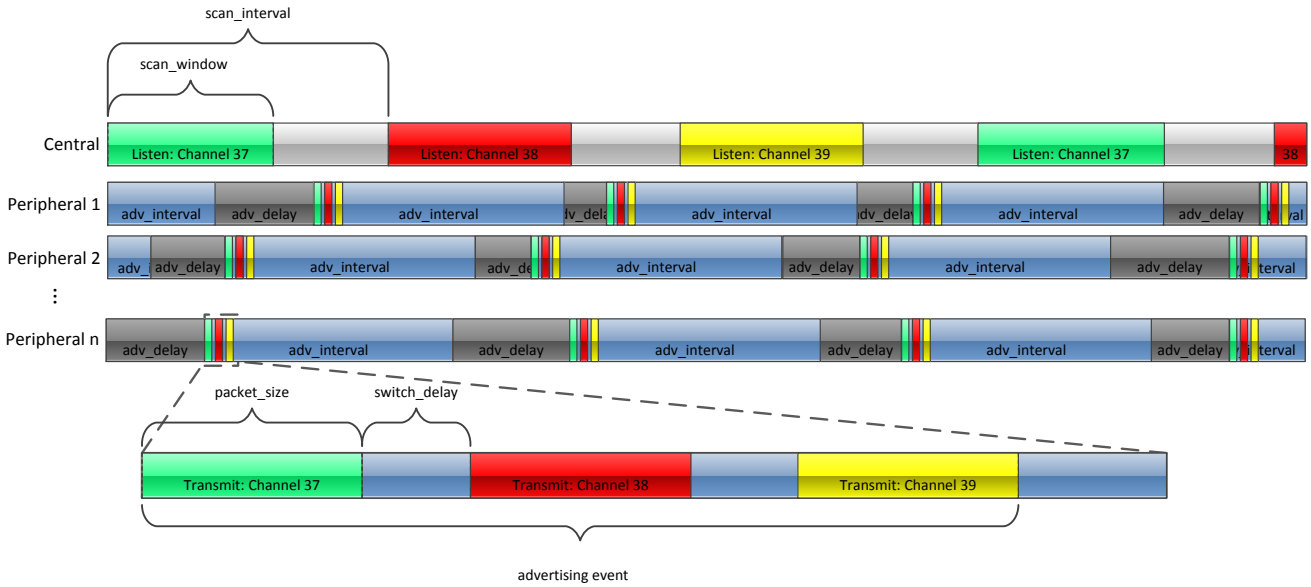



Figure 2.2: Timeline of the BLE device discovery process.

**Connected phase:** During a connection, one device (mostly but not necessarily the peripheral) functions as a *GATT server* and offers *services* containing *characteristics* to other devices. These characteristics can be used to transfer data packets containing a maximum of 20 bytes of payload, initiated by either the GATT server or client (depending on the characteristic's properties). In version 4.2 of the Bluetooth specification the maximum payload of packets was increased to 244 bytes to increase the achievable data transfer rate.

### 2.5.4 Security and privacy

BLE provides optional link layer security based on AES-128 [42] in Counter with CBC-MAC (CCM) mode [18], providing both data authentication and confidentiality. These algorithms are recommended by the US government and considered to be safe. There are however some weaknesses in the way the algorithms are implemented. Ryan [49] describes how to break all security modes, except for the OOB mode which requires a 128-bit key to be exchanged over a secure channel in advance. Rosa [48] shows another attack for the same security modes.

Any BLE device needs a device address. There are two available types of device addresses. A **public** device address is determined by the manufacturer of the device and does not change over the lifetime of the device. A **random** device address is generated by the device and can be changed at any time, but not without disconnecting active connections. In version 4.2 of the Bluetooth Specification functionality was added to change addresses without disconnecting.

# Chapter 3

# Energy consumption and interference

Energy consumption by various applications is a major issue for smartphone users. As a result, the proposed automated payment system will be less appealing for users if the energy consumption by the application is not kept minimal. Additionally, parallel Bluetooth-based communication (e.g. headphones, wristbands etc.) can cause interference and thus increased energy consumption for the smartphones.

In this chapter, we study the energy consumption and latency of BLE devices for various operations. First, we perform a simulation of the device discovery process. Secondly, we validate the simulation, investigate data transfer and measure energy consumption by performing experiments with up to 30 devices. Finally, we develop a model to apply the results of the simulations and experiments in the development of the proposed payment system.

## 3.1 Previous work

We briefly discuss some existing work on energy consumption and interference for Bluetooth Low Energy.

Liu *et al.* [37] developed a model for the analysis of device discovery in BLE in terms of latency and validated this with a simulation. This is done for a situation with only one advertising device. Later, the authors developed a model and did extensive energy measurements for the BLE device discovery process while varying advertising and scanning parameters [38], but they still did not take interference into account. Chong *et al.* [16] developed a model for throughput and energy consumption for a ZigBee Network under the presence of Bluetooth Classic interference and validated the model with a simulation. Stranne *et al.* [53] performed experiments to validate the

model from [52] for the influence of mutual interference on the throughput of Bluetooth Classic. Howitt [28] developed a model for interference between independent Bluetooth Classic connections and verified the model for cases with one interferer by an experiment. Goldenbaum *et al.* [22] performed a study of the general trade-off between energy consumption and robustness in multi-antenna sensor networks with interference but do not perform any experiments. Gomez *et al.* [23] developed a model for BLE throughput based on the bit-error rate. The model is verified using a simulation. Kindt *et al.* [35] presented a very extensive energy model covering all operating modes of BLE and verified it using an experiment. Again, interference is not taken into account. Siekkinen *et al.* [51] measured energy consumption for BLE and ZigBee. Some experiments with WiFi interference are performed.

Summarizing the existing work, we conclude that energy consumption of Bluetooth Classic devices under mutual interference has been measured, as well as the energy consumption of BLE under WiFi interference. No previous work considered the influence of mutual interference from BLE devices on energy consumption and latency. To our best knowledge, we are the first to conduct research regarding Bluetooth Low Energy who take interference into account in modelling, simulations or experiments.

## 3.2 Simulations

In a simulation we can explore situations with lots of devices without actually needing those devices. The device discovery process for BLE, as described in Section 2.5, is fairly simple and we wanted to be able to explore this for a large number of devices. In the next subsections, we describe how we implemented a simulation for this process, followed by its results.

### 3.2.1 Implementation details

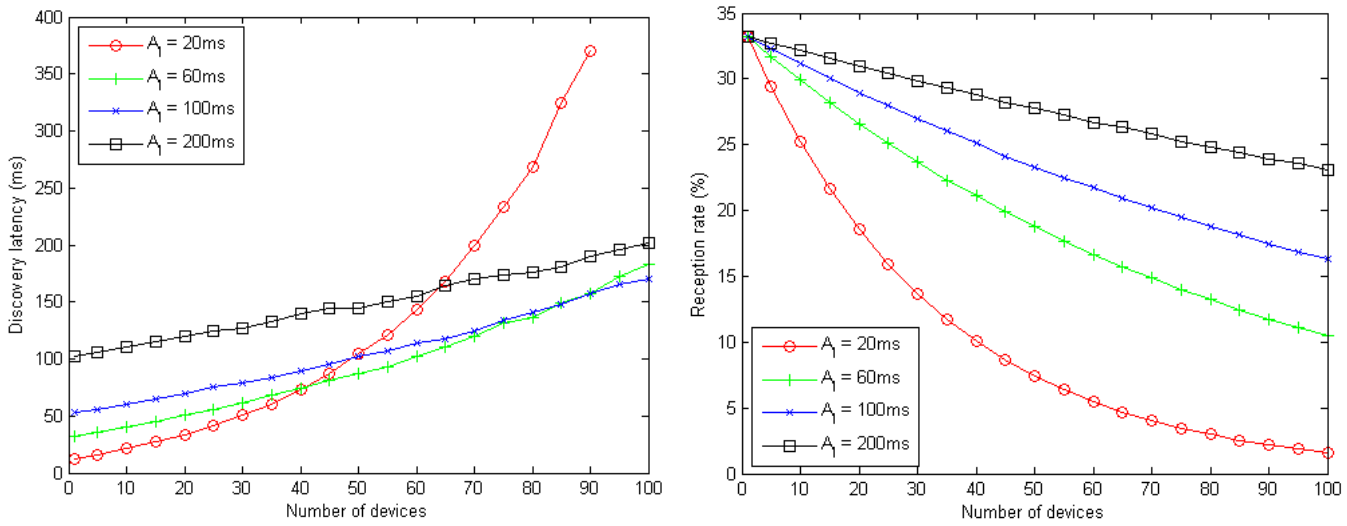
The simulation is implemented in MATLAB. First, a list of transmitting intervals is generated for every advertising device. Overlapping intervals between devices are removed from the list of intervals. For the scanning device, a list of scan intervals is generated. The transmitting intervals that are contained in a scan interval are then filtered out.

From the resulting intervals two output figures are calculated. The average latency is the time from the start of the first scan window until the moment the first packet from a certain device is received. The receive rate is the percentage of transmitted packets that is actually received by the scanning device.

The implementation assumes that there is one scanning device and a number of peripheral devices that are all advertising with the same interval and

| Variable | Value |
|---|---|
| Advertising interval | $\{20, 60, 100, 200\}$ ms |
| Packet length | 376 $\mu$s |
| Switch delay | 150 $\mu$s |
| Scan interval | 100 ms |
| Scan window | 100 ms |
| Number of devices | [1 5:5:100] |
| Number of runs | 10000 |
| Simulation length | 5000 ms |

Table 3.1: Parameters used in simulation runs.




(a) Latency (average)

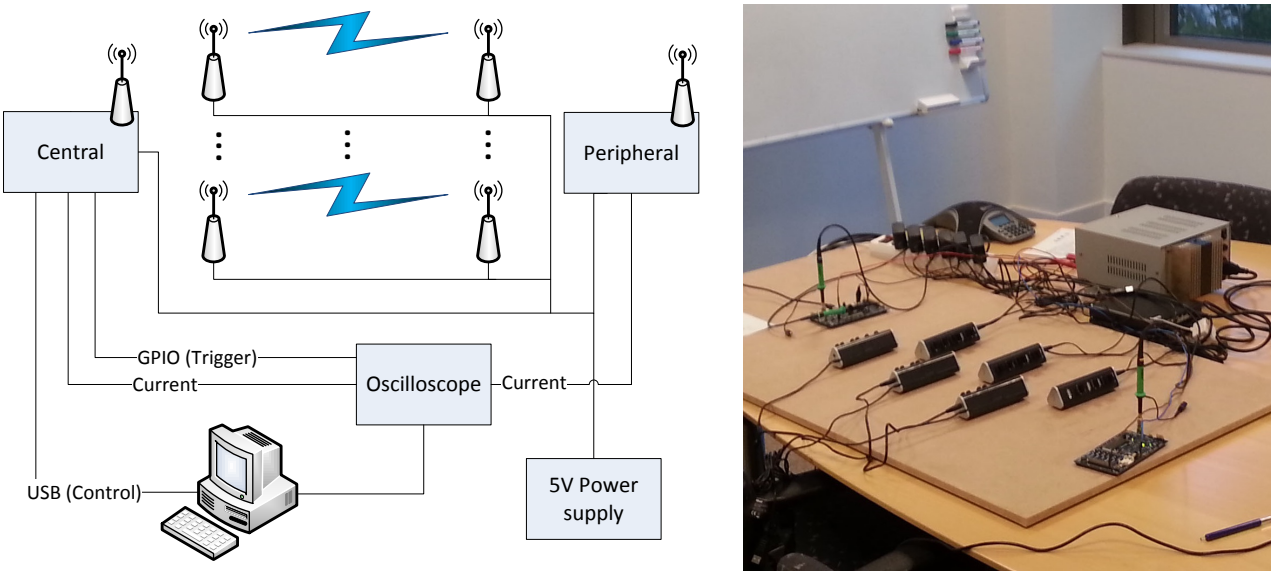
(b) Receive rate (average)

Figure 3.1: Result of simulation runs.

packet size. If two packets collide on the same channel, both packets are discarded. If a packet does not collide and it is contained in a scan interval, it is always assumed to be received.

### 3.2.2 Simulation results

We ran a simulation using the settings as found in Table 3.1. The results can be found in Fig. 3.1. From these graphs, we can see how the discovery latency increases with the number of simultaneously advertising devices. One important aspect to notice is that in a situation with over 40 devices, the shortest advertising interval no longer results in the fastest average discovery.




(a) Schematic

(b) Photo

Figure 3.2: Experimental setup.

## 3.3 Experiments

Wireless data transmission is a complex phenomenon in which all kinds of physical effects play a role. This means that it is impossible to be sure about its behaviour without a real-life validation. For this reason, we performed experiments with real BLE devices.

### 3.3.1 Experimental setup

The experiments were performed using BLE modules from Bluegiga. These modules combine an 8051 microcontroller with a Bluetooth transceiver. They are available in the form of a relatively cheap USB stick and as a development kit allowing for easy energy measurements. The modules can be programmed using BGscript, which is a module-specific scripting language. Fig. 3.2 shows how the devices used in the experiment were connected. There are two 'devices under test': a central and a peripheral device. Both are powered via a power supply and connected to an oscilloscope for current measurement. The central device is in addition connected to the MATLAB application via USB, and via a GPIO pin to a separate oscilloscope channel for sending trigger pulses.

#### Interference-generating devices

The interference-generating devices work autonomously and are only connected to a power supply. By default all devices are in advertising mode. A control application is used to switch them to data transfer mode when

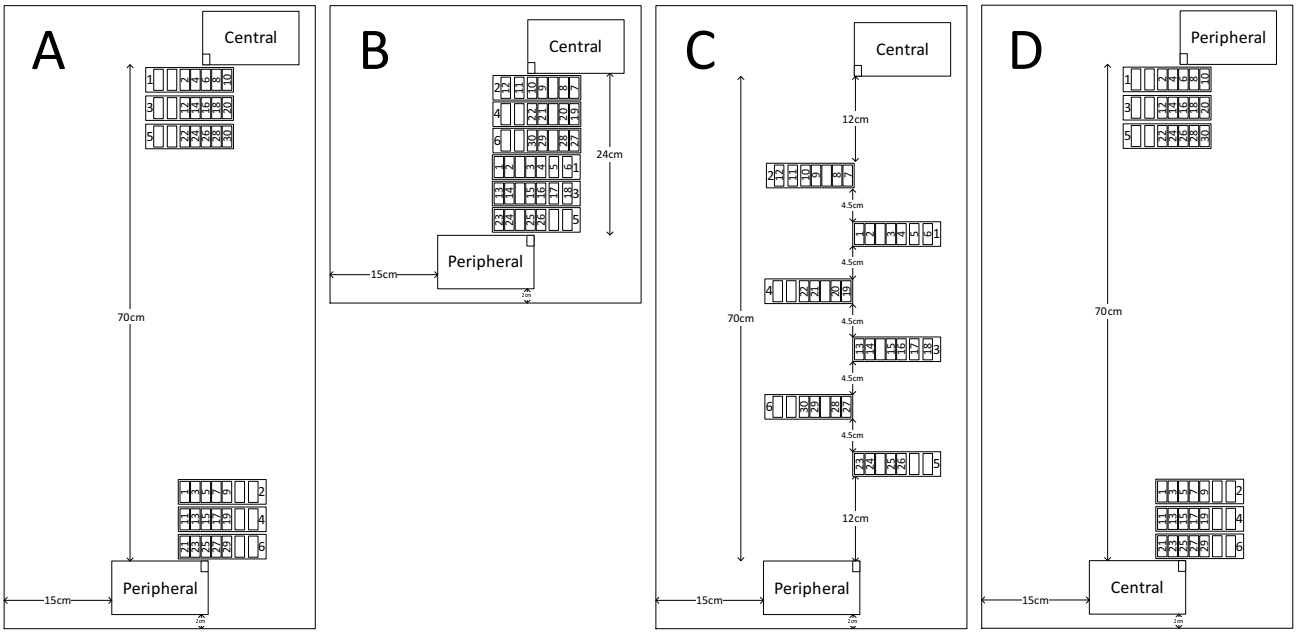



Figure 3.3: Device layouts used in the experiments.

needed. In data transfer mode the devices are transmitting at the maximum achievable throughput, which is around 100 kb/s [35, 10].

### Energy measurement

The BLE113 development board is equipped with a current measurement circuit [8]. This circuit, consisting of a shunt resistor and an instrumentation amplifier, measures the current flowing to the module and outputs this as a voltage. This voltage is measured and the current can be calculated from this voltage using the following equation:

$$I = \frac{3.3 - V_0}{30},$$

where $I$ is the current flowing to the module and $V_0$ the measured voltage. The module is powered via a 3.3 V LDO, which means its voltage is constant at 3.3 V. Energy consumption is calculated by trapezoidal numerical integration over the current measurements, multiplied by the voltage.

### Device layouts

In the experiments the four layouts as found in Fig. 3.3 were used. The interfering devices are the small, numbered devices. In transfer mode, odd numbered devices function as central and even numbered devices as peripheral. Device $N$ communicates with device $N + 1$. The devices marked as Central and Peripheral are the devices for which energy consumption is measured.

- In **Layout A**, all pairs of devices communicate over an approximately

equal distance. This means all received signals will have around the same signal strength.

- In **Layout B**, the interfering pairs communicate over a very small distance and all devices are very close to each other. This represents a situation where multiple interfering devices are communicating over a short distance, while situated between two devices communicating over a longer distance. An example of a similar situation is a smartphone communicating with a smartwatch, while another smartphone is communicating with a beacon in a vehicle.
- **Layout C** is similar to Layout B, but with the devices spread out over a larger area.
- **Layout D** is similar to Layout A, but with the Central measured device close to the Peripheral interfering devices, and vice versa.

### Channel limitation

During the experiments, we only used channels 1-8 for data transfer instead of all 37 channels that normally would be used. This way we increase the influence of interference for the measurements and we make sure that we can achieve a significant impact on latency and energy consumption without requiring an excessive amount of interfering devices. The number of advertising channels is not limited, i.e. all three advertising channels are used.

### Measurement procedure

For determining the energy consumption of different operations, a measurement sequence was developed. This sequence contains 8 phases, which will be explained in detail below. A schematic view of this sequence can be found in Fig. 3.4. The sequence is started by a command from the MATLAB script to the central device via USB. The central node controls the rest of the sequence and sends trigger pulses to the oscilloscope. After the measurement sequence, the oscilloscope data is sent to the MATLAB script, which splits the current measurement data using the trigger pulse data.

- When no measurement is active, sleep mode of the central node is disabled, to be able to receive the start command. When the sequence is started, sleep is enabled to minimize the current consumption during measurement.
- In the **Discover** phase, the central device is listening for advertisement messages from the peripheral device. This phase ends when the first advertisement packet is received.
- In the **Connect** phase, the central device waits for another advertisement message and sends a connection request immediately after receiving this. To check if the connection is really established, a read


Figure 3.4: Schematic view of the measurement sequence.

request is sent to the peripheral. When a response to this request is received, the connect phase ends. When no message is received after 6 connection intervals, the connection is considered to be lost. The probable reason is that the connection request packet was not received by the peripheral, for example because a collision occurred. In this case a new connection request is sent.

- In the **Update** phase, the connection parameters are updated to use only channels 1-8 as described before.
- In the **Transfer** phase, a read request is sent to the peripheral device. The peripheral responds with a value of 20 bytes. As soon as this value is received by the central, the phase ends.
- In the **Idle** phase, the connection is kept active for 500 ms without transmitting any data.
- In the **Disconnect** phase, a disconnect request is sent to the peripheral device. The phase ends when the disconnection is acknowledged by the peripheral. However, there is no way to report if the disconnection request times out. When this happens no trigger will be sent and the result of the complete measurement cycle will be discarded.
- In the **Sleep** phase, both devices are kept in sleep mode to perform the current calibration as explained in Section 3.3.2.
- In the **Advertising** phase, the central device is continuously scanning and the peripheral device is continuously advertising.

For each configuration, this cycle is repeated 500 times[1]. The presented results are the average of these measurements.

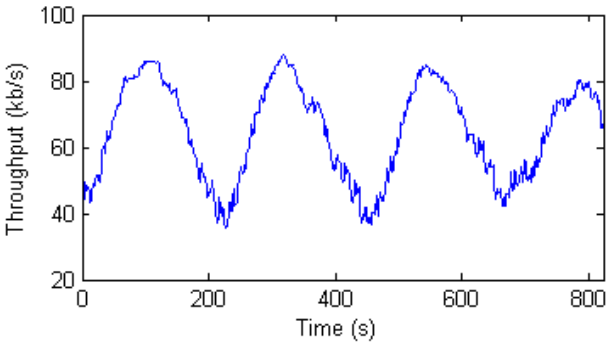
### 3.3.2 Implementation problems

During the implementation of the software for the measurement devices and the execution of the experiments, we encountered some problems. This section describes these problems and the solutions we found.

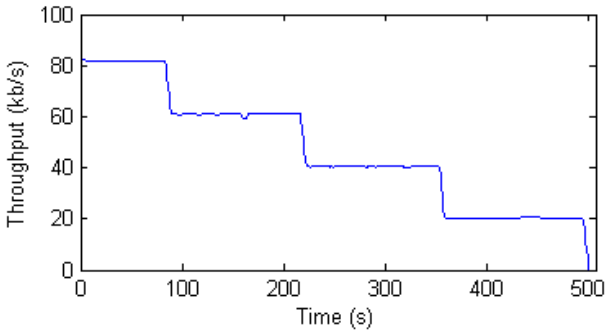
#### Clock Drift

In the process of developing the software for the interference-generating devices, we discovered an interesting effect. In a setup with 2 pairs of devices continuously trying to achieve maximum throughput, we measured the result as shown in Fig. 3.5a. To investigate this further, we limited the connections to only 1 channel, which resulted in the throughput as shown in Fig. 3.5b. This led us to the following hypothesis for the cause of these effects: the clocks in the different devices have a (very slightly) different

[1]Due to time constraints, some of the measurements were repeated only 100 times. This is mentioned in Table A.1.




(a) Channel 1-8 enabled.




(b) Only 1 channel enabled.

Figure 3.5: Throughput with 1 interfering pair (averaged over 20 values).

frequency. Within a connection the BLE protocol takes care of small differences between clocks, however, in this case there are two independent connections.

When data is transmitted using notification packets with the chosen settings, normally 4 packets are transmitted in every connection interval. When the connection is set up (for the 1-channel case as displayed), the connection intervals are out of sync. However, if one of the connections has a very slightly shorter connection interval the packets will start colliding over time. In the 8-channel case, some packets will still arrive because of the channel hopping.

In our measurement setup we wanted to eliminate these long-period effects. To achieve this, we adopted a periodic reconnection strategy with a random interval for the interference-generating devices.

Unfortunately the BGscript platform does not have a random function available. After some experiments we discovered that reading the 5 least significant bits of the ADC value for a not-connected pin yielded a close to uniform random distribution, which is sufficient for our application.

### Current Measurement

To measure the Bluetooth module's energy consumption, we measure its current as explained in Section 3.3.1. This current varies from 0.9 $\mu$A in Power Mode 2 (which is the lowest possible sleep mode used) to 27.0 mA when the radio is receiving [9, 11]. Measuring the lowest sleep current of 0.9 $\mu$A would require an infeasible measurement accuracy. Besides, some inaccuracy turned out to be present in the measurement circuit.

We decided to assume that the current consumption when sleeping is 0.9 $\mu$A, as specified by the manufacturer. We put the module in sleep mode for 500 ms and calculate the parameter $V_C$ for the current formula in Section 3.3.1 using the following equation:

$$V_C = V_M + 30 \cdot 0.0000009,$$

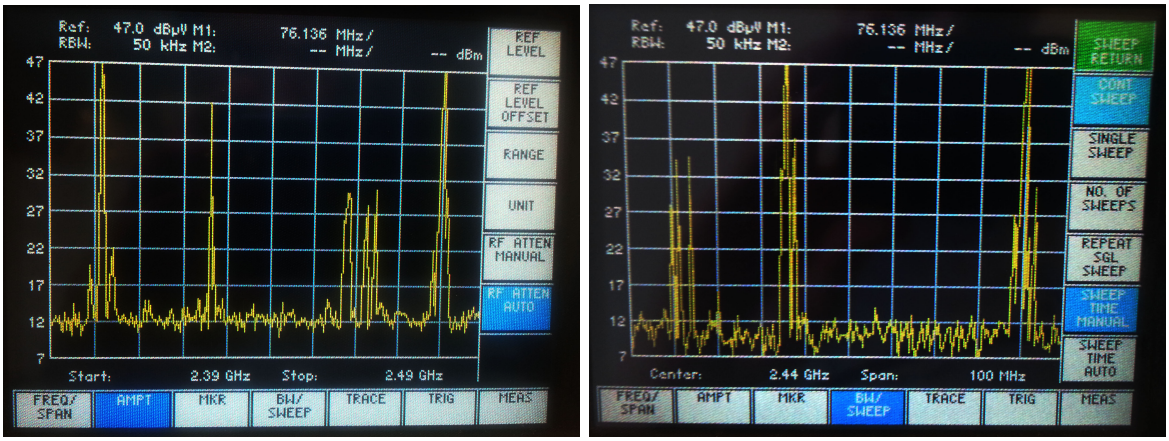
where $V_M$ is the mean measured voltage during the sleep period.

### Measurement Method

We tried to collect the energy measurement data from the oscilloscope directly via USB streaming mode. In this mode, the MATLAB application was informed by the Bluegiga module via markers over UART and controlled the scope. However, the timing was not precise enough. We solved this by adding an extra measurement channel, send the triggers over GPIO and analyzing the data after the measurement was completed.

### Testing environment

Because the 2.4 GHz ISM band is also used by other wireless protocols, the most important being WiFi, we had to find a suitable location to perform the experiments. First we used a spectrum analyzer to analyze the interference present in a normal office environment. The result can be found in Fig. 3.6a. The peaks around 2402, 2426 and 2480 MHz are the advertising BLE devices, but the smaller peaks around 2460 MHz are interference from WiFi.




(a) Office environment (b) EMC cage

Figure 3.6: Spectrum at different locations.

We tried performing the experiments in an EMC testing cage, which blocks all signals from outside. This resulted in a very clear spectrum as seen in Figure 3.6b. However, during the experiments we observed that a very small change in the setup, like moving the complete setup a few centimeters, could result in completely different measurements. We expect this to be caused by the fact that the cage was not equipped with proper damping material, so all kinds of reflections of the signals could occur and attenuate or amplify each other.

Eventually we performed measurements in a meeting room in a corner of the building which was only covered by only one WiFi accesspoint, which was turned off for the time of the measurements. We regularly used the spectrum

analyzer to check if interference from any source was present during the measurements.

### 3.3.3 Experimental results

In this section, we present the measurement results. First, we present two detailed measurements as examples of the results. We show the influence of interference with some graphs. The complete measurement results can be found in Appendix A. As mentioned earlier, all measurements only use channels 1-8 during data transfer.

#### Detailed measurements

Fig. 3.7 shows the current consumption from a single measurement cycle for two configurations. In Fig. 3.7a, it can be seen that interfering devices in advertising mode cause a longer discovery and connect process, while Fig. 3.7b shows that a lot of interfering devices in transferring mode can dramatically increase the time needed for data transfer.

#### Interference measurements

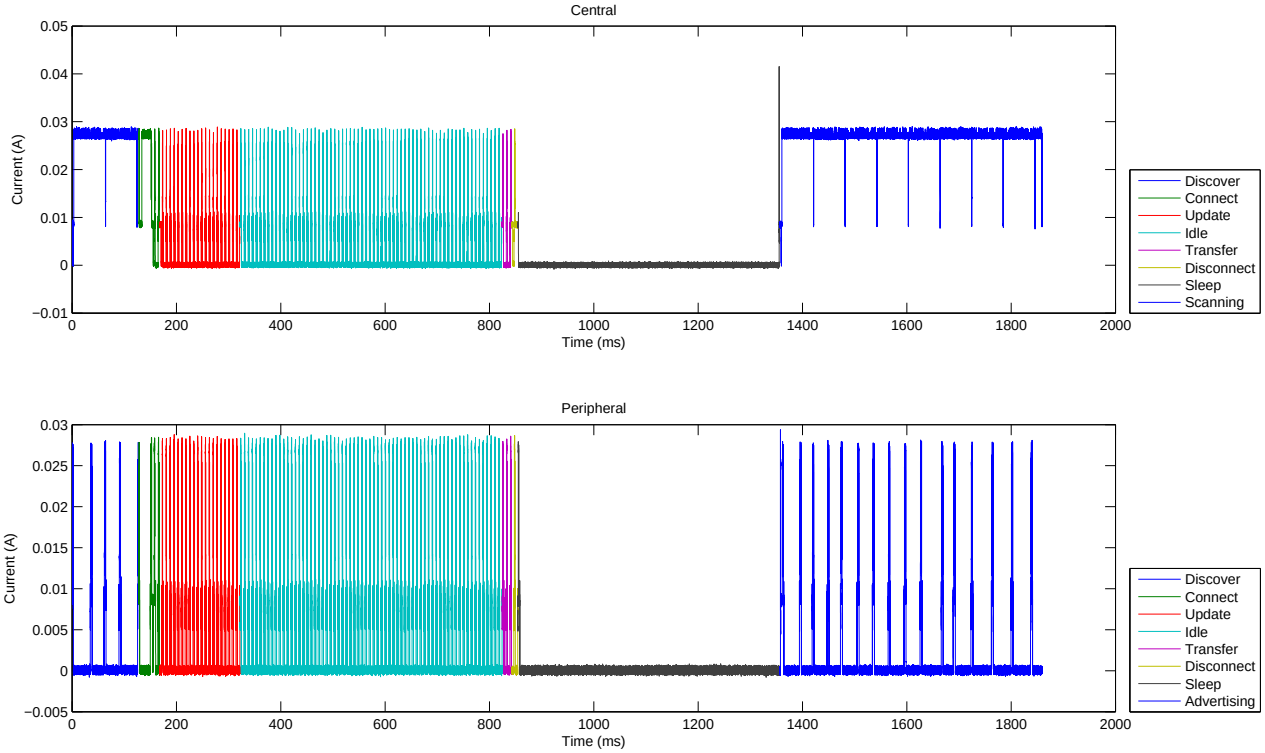
For various configurations, we measured the influence of an increasing number of interfering devices on the latency and energy consumption.

Fig. 3.8 shows the latency and energy consumption measurements for device discovery under interference from advertising devices with layouts B and C. We see that the maximum energy consumption increase for the peripheral from 0 to 30 interferers is 2.2× for layout B. The increase is greater for layout B than for layout C. When comparing the simulation results with experiment results, we see that the shape of the graph is equal, but that there is a constant difference.

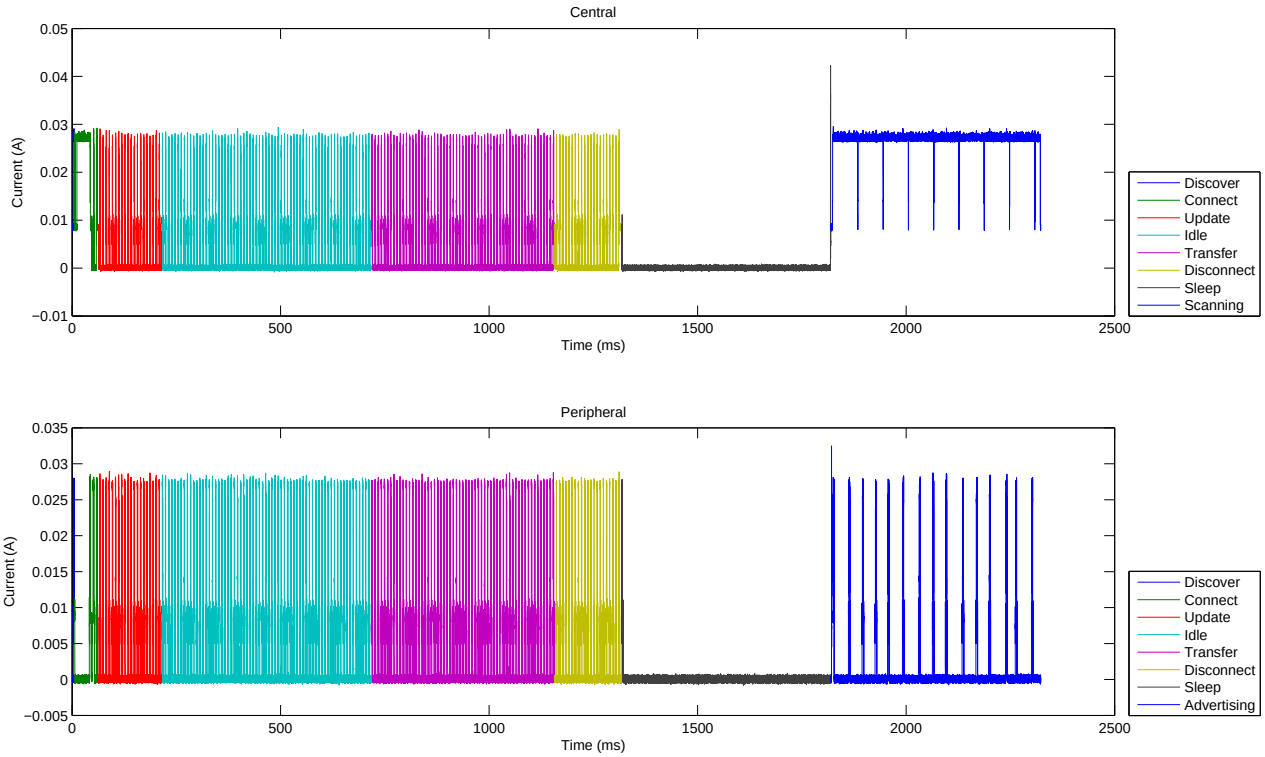
Fig. 3.9 shows the latency and energy consumption measurements for the connection setup operation under interference from advertising devices with layouts B and C. Similar to the discovery results, we see a maximum energy consumption increase of 2.2× from 0 to 30 devices and a larger increase for layout B than for layout C.

Fig. 3.10 shows the measurement results for data transfer under interference of transferring devices. In this graph, the net transfer energy is the amount of energy spent during the transfer of a data packet, reduced by the energy spent to keep the connection idle for the same period. In other words, the amount of energy spent on the transfer of the data packet, assuming that a connection would have been active anyway. The net transfer energy is calculated as follows:

$$E_{net} = E_{transfer} - L_{transfer} \cdot \frac{E_{idle}}{L_{idle}},$$

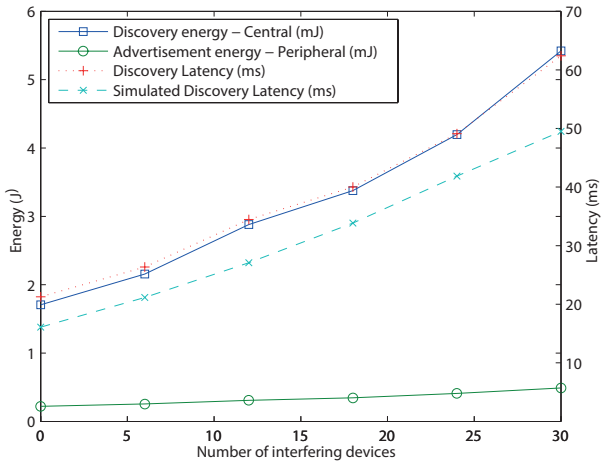



(a) Layout C with 24 devices advertising

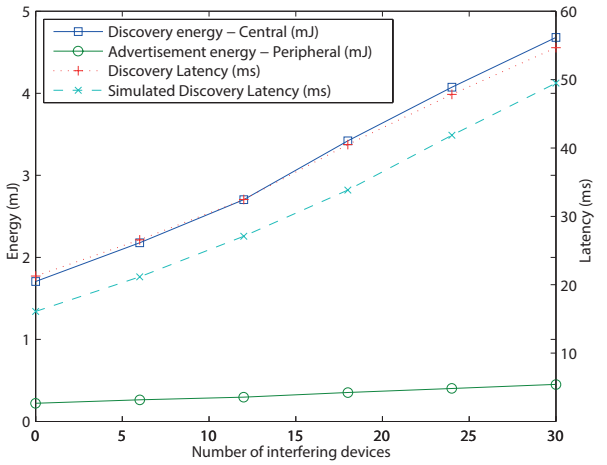



(b) Layout A with 30 devices transferring

Figure 3.7: Results of a single measurement sequence for various configurations

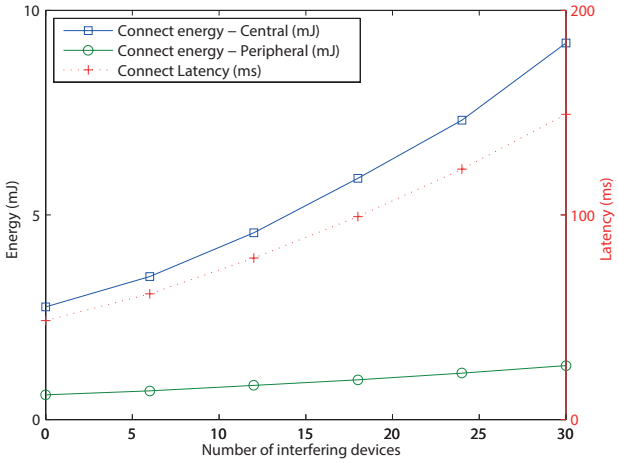



(a) Layout B, advertising mode

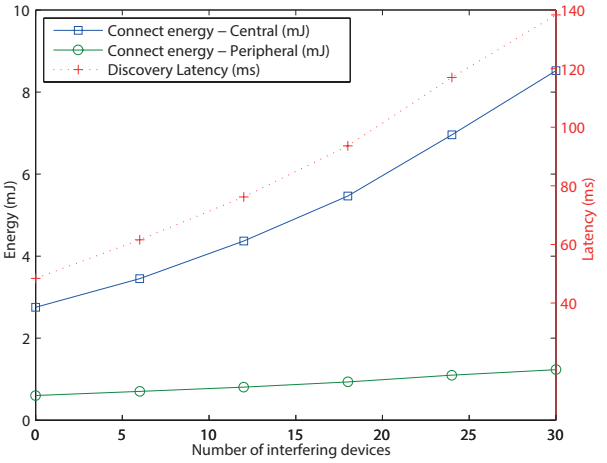



(b) Layout C, advertising mode

Figure 3.8: Measurements for device discovery.

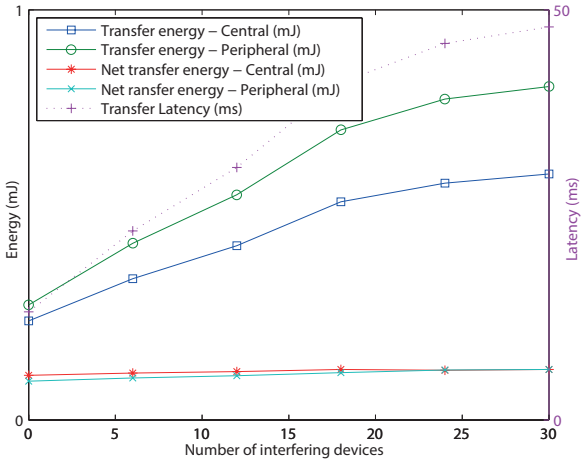



(a) Layout B, advertising mode

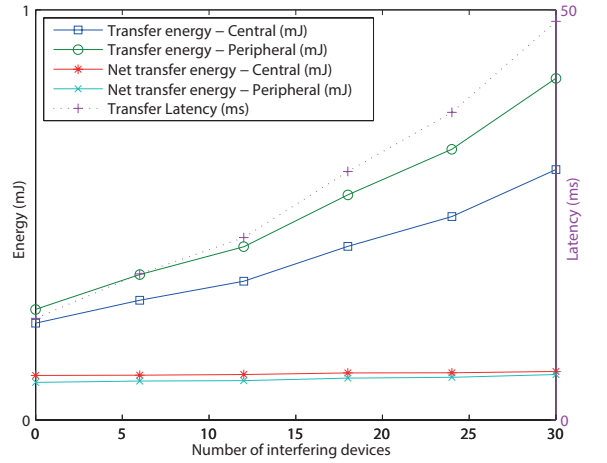



(b) Layout C, advertising mode

Figure 3.9: Measurements for connection setup.




(a) Layout A, transfer mode




(b) Layout C, transfer mode

Figure 3.10: Measurements for data transfer.

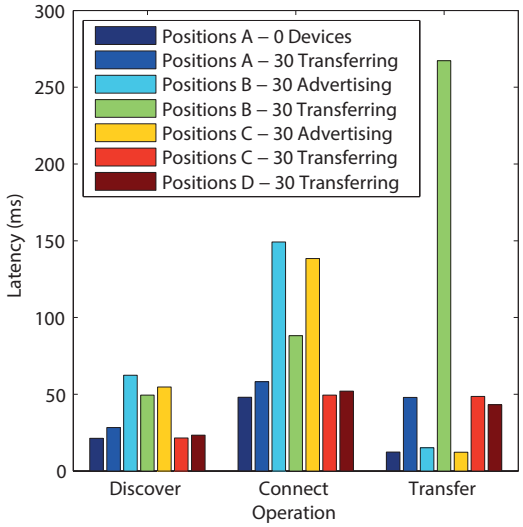



Figure 3.11: Discover, connect and transfer latencies for various configurations.

where $E_{net}$ is the net transfer energy, $E_{transfer}$ the measured transfer energy, $L_{transfer}$ the transfer latency, $E_{idle}$ the idle energy and $L_{idle}$ the time for which the idle energy is measured.
From these graphs, we see that for layout A the energy consumption increases more for a small number of devices than for layout C. The net energy consumption increases only 1.3× from 0 to 30 devices.

### Latency comparison

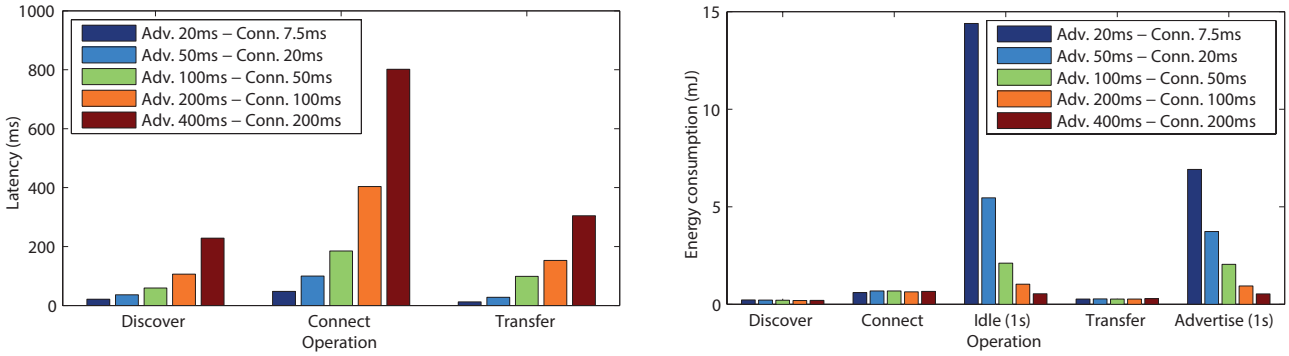
Fig. 3.11 shows latencies for the maximum number of interfering devices for all device layouts. This shows that the influence of interfering devices in advertising mode on the devices transferring (and vice versa) is very small. The largest latency increase for transferring devices is 3.9× from 0 to 30 devices with layout C.

### Connection and advertising intervals

To determine the influence of connection parameters, we experimented with longer advertising and connection intervals. These measurements were done without any interfering devices. Fig. 3.12 shows discovery, connect and transfer latency and peripheral energy consumption for different intervals. From these measurements, we can conclude that longer advertising intervals mean a lower energy consumption but also a higher latency.

## 3.4 Model development

Based on the simulation and experiment results, we developed a model for the energy consumption and latency of BLE. We did this only for the parts




(a) Latency

(b) Peripheral energy consumption

Figure 3.12: Measurements for various advertising and connection intervals.

that we need later on in the protocol development. In Table 3.2 we list the functions, parameters and symbols used in the model.

| **Inputs** | | |
|---|---|---|
| Symbol | Description | Unit |
| $I_{adv}$ | Advertisement interval | ms |
| $I_{conn}$ | Connection interval | ms |
| $N_{packets}$ | Number of packets | # |
| $N_{adv}$ | Number of advertising devices | # |
| **Model parameters** | | |
| Symbol | Description | Unit |
| $E_{advertisement}$ | Energy requirement for 1 advertisement | mJ |
| $E_{packet}$ | Energy requirement for 1 data packet | mJ |
| $T_{proc,disc}$ | Processing time for device discovery | ms |
| **Outputs** | | |
| Symbol | Description | Unit |
| $E_{advertising}(I_{adv})$ | Advertising energy consumption | mJ/s |
| $T_{discover}(I_{adv}, N_{adv})$ | Discovery latency | ms |
| $E_{connect}()$ | Connect energy | mJ |
| $T_{connect}(I_{adv}, I_{conn}, N_{adv})$ | Connect latency | ms |
| $E_{transfer}(N_{packets})$ | Transfer energy (per packet) | mJ |
| $T_{transfer}(I_{conn}, N_{packets})$ | Transfer latency | ms |

Table 3.2: Parameters used in the energy model.

### 3.4.1 Advertising

The advertising energy (in J/s) depends on the advertising interval and is given by the following equation:

$$E_{advertising}(I_{adv}) = E_{advertisement} \cdot \frac{1000}{I_{adv} + I_{random}}, \tag{3.1}$$

where $I_{random}$ is the average random interval. From the experiments we found the value of 0.22 for $E_{advertisement}$ and 10 for $I_{random}$.

### 3.4.2 Discovery and connection

The discovery latency for a device depends on the advertising interval and the number of advertising devices. We assume the advertising interval is equal for all devices. The discovery delay (in ms) is given by the following equation:

$$T_{discover}(I_{adv}, N_{adv}) = (0.5 \cdot I_{adv} + T_{proc,disc}) \cdot e^{\frac{2N_{adv}}{3(0.5 \cdot I_{adv} + T_{proc,disc})}} \quad (3.2)$$

From the simulation results we found a value of 9.7 for $T_{proc,disc}$.

We model the connection latency (in ms) as the discovery time plus 3 times the connection interval. This is the connection latency including the check if the connection really succeeded as explained in Section 3.3.1. The connection latency is given by the equation:

$$T_{connect}(I_{adv}, I_{conn}, N_{adv}) = T_{discover}(I_{adv}, N_{adv}) + 3 \cdot I_{conn} \quad (3.3)$$

The connection energy for the peripheral (in J) is modelled without taking interfering advertisers into account and is thus constant and given by the following equation:

$$E_{connect}() = 0.6 \quad (3.4)$$

### 3.4.3 Data transfer

We model the transfer latency for a packet with the maximum supported payload of 20 bytes. The transfer latency (in ms) is given by the equation:

$$T_{transfer}(I_{conn}, N_{packets}) = \frac{3}{2} \cdot I_{conn} \cdot N_{packets} \quad (3.5)$$

We model the transfer energy for a packet with the maximum payload of 20 bytes for the peripheral device. The transfer energy in J is given by the following equation:

$$E_{transfer}(N_{packets}) = E_{packet} \cdot N_{packets} \quad (3.6)$$

From the experiments we found the value of $E_{packet}$ to be 0.27.

## 3.5 Discussion

In this section we discuss the results of the simulations, experiments and modelling. We evaluate the developed model and discuss the implications of the results for the proposed payment system and for application developers in general.

### 3.5.1 Model evaluation

To evaluate the accuracy of our model, we used it to calculate part of the values that resulted from the experiments. Table 3.3 shows the experiment values with the corresponding model outcomes. From these results we can see that the model is in general pretty accurate. For the connection energy and latency we see that the experiment values are higher than the model values in situations with interfering devices. For the energy this is caused by the fact that we do not consider the interfering devices in the model. For the connection latency this is caused by the fact that reconnections (as described in Section 3.3.1) are not accounted for in the model.

| Lay. | Interf. (#) | Intervals (ms) | | Conn. en. (mJ) | | Transf. en. (mJ) | | Adv. en. (mJ/s) | | Discov. lat. (ms) | | Conn. lat. (ms) | | Transf. lat. (ms) | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Adv. | Conn. | Adv. | Mod. | Exp. | Mod. | Exp. | Mod. | Exp. | Mod. | Exp. | Mod. | Exp. | Mod. | Exp. |
| A | 0 | 7.5 | 20 | 0.60 | 0.60 | 0.27 | 0.27 | 7.33 | 6.91 | 20.4 | 21.3 | 42.9 | 48.0 | 11.3 | 12.4 |
| C | 6 | 7.5 | 20 | 0.60 | 0.70 | 0.27 | 0.28 | 7.33 | 6.96 | 25.0 | 26.6 | 47.5 | 61.6 | 11.3 | 12.7 |
| C | 12 | 7.5 | 20 | 0.60 | 0.81 | 0.27 | 0.27 | 7.33 | 6.90 | 30.6 | 32.5 | 53.1 | 76.2 | 11.3 | 12.4 |
| C | 18 | 7.5 | 20 | 0.60 | 0.93 | 0.27 | 0.28 | 7.33 | 6.96 | 37.5 | 40.5 | 57.5 | 93.7 | 11.3 | 12.8 |
| C | 24 | 7.5 | 20 | 0.60 | 1.10 | 0.27 | 0.28 | 7.33 | 6.93 | 45.9 | 47.8 | 68.4 | 117.1 | 11.3 | 12.6 |
| C | 30 | 7.5 | 20 | 0.60 | 1.23 | 0.27 | 0.27 | 7.33 | 6.95 | 56.2 | 54.7 | 78.7 | 138.4 | 11.3 | 12.3 |
| A | 0 | 20 | 50 | 0.60 | 0.68 | 0.27 | 0.28 | 3.67 | 3.73 | 35.4 | 36.5 | 95.4 | 99.9 | 30.0 | 27.6 |
| A | 0 | 50 | 100 | 0.60 | 0.68 | 0.27 | 0.27 | 2.00 | 2.05 | 60.4 | 59.3 | 210.4 | 185.1 | 75.0 | 99.0 |
| A | 0 | 100 | 200 | 0.60 | 0.64 | 0.27 | 0.27 | 1.04 | 0.94 | 110.4 | 106.6 | 410.4 | 403.6 | 150.0 | 152.9 |
| A | 0 | 200 | 400 | 0.60 | 0.66 | 0.27 | 0.29 | 0.54 | 0.53 | 210.4 | 228.6 | 810.4 | 801.7 | 300.0 | 304.6 |

Table 3.3: Compared values for model and experiments.

### 3.5.2 Implications for proposed system

We can conclude that the influence of interfering devices that are advertising on connected devices that are transferring data is very small. Also, the influence of connected devices that are transferring on devices that are discovering is very small.

For the more realistic device layouts (A, C and D), the maximum increase of energy consumption and latency is about 3×. For layout B, we see a much higher number, especially for data transfer mode. However, in this layout, all devices are placed in close proximity which means near-field effects will probably be a large factor in these results.

### 3.5.3 Implications for application developers

Developers who are working on applications that use BLE can use the results presented to conclude that there is only a limited impact of mutual

interference. Furthermore, they can use the energy model to predict the energy consumption of an application before implementing it.

## 3.6 Other discoveries

During the experiments, we did some other interesting discoveries regarding Bluetooth Low Energy. We describe them in this section.

### 3.6.1 Clock drift influence

As described in Section 3.3.2, we found that BLE connections can show some interesting effects that can probably be explained by clock drift. If clock drift is indeed the cause of these problems, it will happen that two independent connections with the same settings choose the same randomly chosen channel hop increment value. In that case the hop sequences will be exactly synchronized at some point. When this happens, all connection intervals will interfere and the connections will time-out eventually. We did not investigate this further.

### 3.6.2 Adherence to specification

According to the Bluetooth specification [12], a pseudo-random delay between 0 and 10 ms should be added to every advertisement interval. During the experiments, we found delays larger than 10 ms for the Bluegiga modules. We recorded over 17500 advertisement messages for 1 module using a BLE sniffer and found the advertising intervals as shown in Fig. 3.13. From these numbers, it appears that the modules uses a delay with a $U(0, 22)$ distribution, which is not according to the specification.

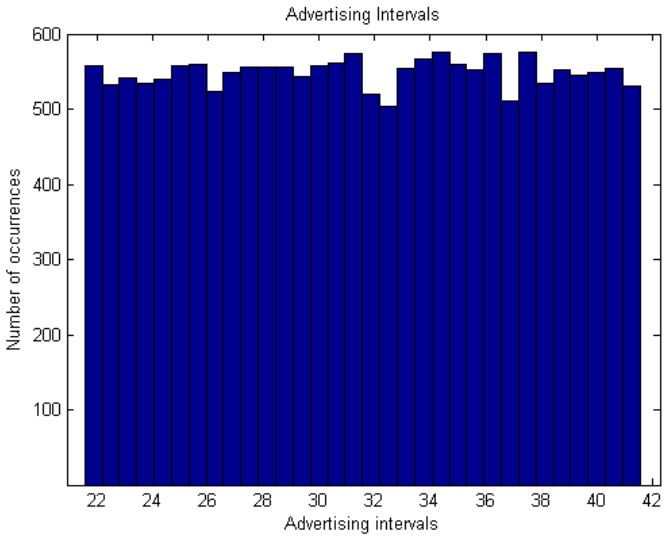



Figure 3.13: Advertising intervals for BLED112 module with advertising interval set to 20 ms.

# Chapter 4

# Communication protocol

In this chapter, we describe the protocol for communication between the devices involved in the automated public transport payment system. We start analysing the functional requirements of the system in Section 4.1. Then we analyse the two important aspects of this protocol: the performance aspects in Section 4.2 and the security aspects in Section 4.3. Finally, we propose a protocol and discuss the implications for energy consumption in Section 4.4.

## 4.1 Functional requirements

This section describes what functions the protocol will have to perform. We start with defining the use cases for the system and list the operational requirements.

We distinguish two kinds of payment and verification 'patterns' in the Dutch public transportation system. In a **train**, the traveller needs a valid ticket, but this is only checked incidentally by a railway guard. This also applies to subway and trams. In a **bus**, the driver checks the traveller's ticket when he gets on the bus. We handle these cases separately, because they result in different timing constraints.

### 4.1.1 Use cases

We describe the use cases for the system for travellers in both train and bus.

#### Traveller (in a train)

In a train, the traveller is checked in and out automatically. No further action from the traveller is required.

#### Check-in

1. The traveller enters the train.
2. The vehicle device detects that the passenger has entered the train and registers him as checked in.

#### Check-out

1. The traveller leaves the train.
2. The vehicle device detects that the passenger left the train and registers this as a check-out.

### Traveller (in a bus)

When used in a bus, a slightly different operation is needed because the bus driver needs to check immediately if a traveller checks in successfully.

#### Check-in

1. The traveller enters the bus.
2. The traveller holds his phone in close proximity to the check-in device.
3. The check-in device detects the phone, registers the check-in and notifies the driver of a successful check-in.

#### Check-out

1. The traveller leaves the bus.
2. The vehicle device detects that the passenger left the bus and registers this as a check-out.

### Guard

A railway guard can check if a passenger paid for his journey in the following way:

1. The guard asks a traveller for his ticket.
2. The traveller shows his phone.
3. The guard device checks if the traveller's phone is registered and if the phone is successfully checked in

### 4.1.2 Operational requirements

In addition to the functional requirements as defined, there are some operational requirements for the system. We define them below. As explained in Section 2.5, we select Bluetooth Low Energy as wireless communication protocol and implement the system according to version 4.0 of the Bluetooth Core Specification. All travelling information (moment and location of check-ins and check-outs for all travellers) needs to be collected at the central back-end every day. The system needs to be able to function without a continuous connection between the vehicle and the back-end.

## 4.2 Performance aspects

In this section, we analyse the performance aspects of the system. These are quantitative properties of the system that ensure it is working sufficiently fast and scalable. We start by formulating the requirements and then develop a model that will be used later on to determine if these requirements can be met in a later section.

### 4.2.1 Requirements

Below we describe the performance requirements for the system. The requirements are summarized in Table 4.1.

To determine the minimum time between two stations, we analysed the timetable of HTM[1]. This public transportation in The Hague operates both buses and trams. We analysed the schedule in terms of stations per minute. We found the tightest schedule for the following lines: bus 28 with 7 stops in 7 min, and tram 11 with 18 stops in 22 min. There are never two stops in the same minute, so we assume the minimum time between two stations to be 30 s.

A check-out needs to be processed at least before the vehicle is at the next station to make sure the right check-out location is registered. As the shortest time between two stations is 30 s, a regular check-in or check-out needs to be registered within these 30 s.

When a passenger enters the bus, he needs to check in at the driver to allow the driver to check if every passenger checked in. To avoid delays when multiple passengers enter the bus, this type of check-in needs to be processed within 250 ms.

When the guard checks a passenger, the devices must be ready for the visual verification within 1 s.

[1] We used the schedule as found on `https://www.htm.nl/reisinformatie/`, visited 2014-11-10.

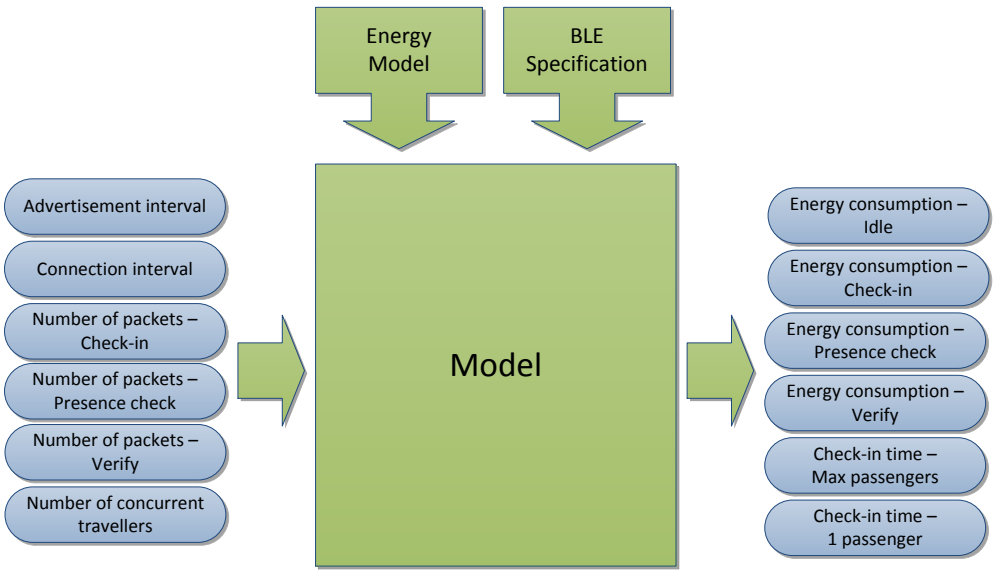



Figure 4.1: Overview of the performance model.

We limit the number of passengers within range of a vehicle device to 50. Mostly one device will be used in every compartment, but in case of larger compartments multiple devices may be used.

| Description | Symbol | Maximum value | Unit |
|---|---|---|---|
| Check-in - Train (Automated) | $t_{ci,a}$ | 30 | s |
| Check-in - Bus (Manual) | $t_{ci,m}$ | 250 | ms |
| Check-out | $t_{co}$ | 30 | s |
| Verification by Guard | $t_v$ | 1000 | ms |
| Concurrent travellers | $n_t$ | 50 | travellers |

Table 4.1: Overview of quantitative requirements.

### 4.2.2 Performance model

In this section, we present our performance model. This model takes the system parameters as inputs and gives the performance parameters as defined above as outputs. The energy model as presented in Section 3.4 is also used as an input to this model. A schematic overview of the model can be found in Fig. 4.1. The model inputs, outputs, internal variables and symbols used in the equations are listed in Table 4.2.

In the model, a **check-in** is the initial authentication of a traveller in a journey, where a temporary journey key is negotiated. A **presence check** is performed periodically to determine if a traveller is still present. A **verification** is an action performed by a railway guard to check whether a traveller is successfully checked in.

| **Inputs** | | |
|---|---|---|
| Symbol | Description | Unit |
| $I_{adv}$ | Advertisement interval | ms |
| $I_{conn}$ | Connection interval | ms |
| $P_{checkin}$ | Number of packets - Check-in | # |
| $P_{presence}$ | Number of packets - Presence check | # |
| $P_{verify}$ | Number of packets - Verify | # |
| $N_{trav}$ | Number of concurrent travellers | # |
| **Energy model functions** | | |
| Symbol | Description | Unit |
| $E_{advertising}(I_{adv})$ | Advertising energy consumption | mJ/s |
| $T_{discover}(I_{adv}, N_{adv})$ | Discovery latency | ms |
| $E_{connect}()$ | Connect energy | mJ |
| $T_{connect}(I_{adv}, I_{conn}, N_{adv})$ | Connect latency | ms |
| $E_{transfer}(N_{packets})$ | Transfer energy (per packet) | mJ |
| $T_{transfer}(I_{conn}, N_{packets})$ | Transfer latency | ms |
| **BLE specification inputs** | | |
| Symbol | Description | Unit |
| $T_{packet}$ | Maximum packet length | ms |
| $T_{ifs}$ | Inter frame space | ms |
| **Internal variables** | | |
| Symbol | Description | Unit |
| $N_{conn}(I_{conn})$ | Number of concurrent connections | # |
| **Outputs** | | |
| Symbol | Description | Unit |
| $E_{idle}(I_{adv})$ | Energy consumption - Idle | mJ/s |
| $E_{checkin}(P_{checkin})$ | Energy consumption - Check-in | mJ |
| $E_{presence}(P_{presence})$ | Energy consumption - Presence check | mJ |
| $E_{verify}(P_{verify})$ | Energy consumption - Verify | mJ |
| $T_{checkin}(N_{trav}, P_{checkin}, I_{adv}, I_{conn})$ | Check-in time for N travellers | ms |

Table 4.2: Parameters used in the performance model.

For determining the maximum number of concurrent connections, we assume that both devices send a packet with the maximum allowed packet size once in a connection interval, both followed by the inter frame space. The number of concurrent connections is given by the equation:

$$N_{conn}(T_{packet}, T_{ifs}, I_{conn}) = \left\lfloor \frac{I_{conn}}{2 \cdot (T_{packet} + T_{ifs})} \right\rfloor . \tag{4.1}$$

When the traveller's device is idle, the only energy consumption is caused

by advertising messages transmitted:

$$E_{idle}(I_{adv}) = E_{advertising}(I_{adv}). \tag{4.2}$$

The energy required by the traveller's device to check-in is the energy required to connect plus the energy required to exchange the presence check packets:

$$E_{checkin}(P_{checkin}) = E_{connect}() + E_{transfer}(P_{checkin}). \tag{4.3}$$

The energy required by the traveller's device for a presence check is the energy required to connect plus the energy required to exchange the presence check packets:

$$E_{presence}(P_{presence}) = E_{connect}() + E_{transfer}(P_{presence}). \tag{4.4}$$

The energy required by the traveller's device for a verification is the energy required to connect plus the energy required to exchange the verification packets:

$$E_{verify}(P_{verify}) = E_{connect}() + E_{transfer}(P_{verify}). \tag{4.5}$$

To check-in $N$ travellers, we need to discover them, connect to them and perform the check-in procedure. However, this can be done in parallel for as many travellers as we can keep concurrent connections.

$$T_{checkin}(N_{trav}, P_{checkin}, I_{adv}, I_{conn}) = (T_{discover}(I_{adv}, N_{trav}) + T_{connect}(I_{adv}, I_{conn}, N_{trav}) + T_{transfer}(I_{conn}, P_{checkin})) \cdot \left\lceil \frac{N_{trav}}{N_{conn}(I_{conn})} \right\rceil. \tag{4.6}$$

## 4.3 Security and privacy aspects

The proposed system must be secure against abuse and must preserve the traveller's privacy. In this section we deal with these aspects. We start with defining the requirements with regard to security and privacy. Then we introduce some cryptographic techniques that will be used later on, when the protocol is developed.

### 4.3.1 Requirements

Below we define the security and privacy requirements for the system.

**Offline operation:** As explained in Section 4.1, the traveller, vehicle and guard's device might not have a continuous connection with the central back-end. This means a traveller will need to be authenticated using just authentication information that is available locally.

**Storing long-term keys:** The traveller, vehicle and guard's device devices are used in the field and might get stolen. The thief might be able to extract secret keys stored in these devices. If this happens the impact should be limited to a certain period. So, we do not allow any long-term secret authentication information to be stored in these devices.

**Mutual authentication:** The protocol must achieve mutual authentication between devices: the traveller needs to be sure that he is dealing with a registered vehicle or guard and the vehicle or guard needs to be sure to be dealing with a registered traveller.

**Anonymity:** The traveller's identity must be protected against anyone but the vehicle and guard's device. For this requirement we assume an active adversary [41], who is not only able to eavesdrop communication but also to alter this communication.

**Unlinkability:** In addition to the anonymity, as described earlier, we also require unlinkability. This means an adversary who is active during two journeys must not be able to determine whether these journeys are made by the same traveller. We only require unlinkability between separate journeys, not between multiple communication rounds within a journey.

**Perfect forward secrecy (PFS):** When a vehicle or guard's device is disposed of, an attacker could extract the secret authentication information from the device. If this happens, we require that the attacker is not able to decrypt any previously recorded communications sessions. This requirement is also known as *perfect forward secrecy* [41].

### 4.3.2 Introduction to cryptographic techniques

To implement a protocol fulfilling the requirements as formulated we need some cryptographic techniques. These techniques are introduced below. We only give a basic functional description. For mathematical background and implementation details the reader is referred to [56, 41, 46].

**Encryption:** The goal of encryption is to protect the confidentiality of data. When data is encrypted using a secret key, only someone who knows this secret key can decrypt the data. This is also referred to as symmetric key encryption.

**Public/Private key encryption:** The goal of public/private key encryption is the same as for symmetric key encryption: to protect data confidentiality. However, in this case, there are two keys: a public key that can

be used to encrypt data and a private key that can be used to decrypt the data. This is also referred to as asymmetric key encryption.

**Message authentication codes (MAC)/Digital signatures:** The goal of a message authentication code and a digital signature is the same: to prevent data integrity, or in other words, to be sure about the source of a certain packet of data. In case of a MAC, this is done using a symmetric key, which is necessary for both generating the MAC and verifying it. In case of a digital signature, there are two keys: a private key to generate the signature and a public key to verify it.

**Certificates:** A certificate is a digital document that contains an authenticity proof for its contents. These contents can for example include a public key. A certificate is issued by a Certification Authority (CA), which generates a digital signature for the document using its private key. Anyone who knows the public key of the CA can use this public key to verify the signature.

**Diffie-Hellman key exchange:** The goal of a Diffie-Hellman key exchange is to negotiate a (symmetric) encryption key between two parties over an insecure channel. Both parties transmit a partial key and use it to deduce a secret key, but an adversary who knows both transmitted parts of the key is unable to reconstruct the secret key.

## 4.4 Protocol design

In this section, we propose a protocol that implements all required features in terms of functionality, performance and security as described before. First, we present the global design of the protocol. Then, we analyse previous work to see if any suitable protocol exists. We describe our proposed protocol in more detail and analyse its security. We present the cryptographic details of the protocol and show how it can be implemented using BLE. Finally, we analyse the performance of the protocol in terms of energy and latency.

### 4.4.1 Global design

An overview of the protocol is shown in Fig. 4.2.

**Central Back-End:** In a central back-end, information about all journeys is collected. Based on this information travellers are billed.

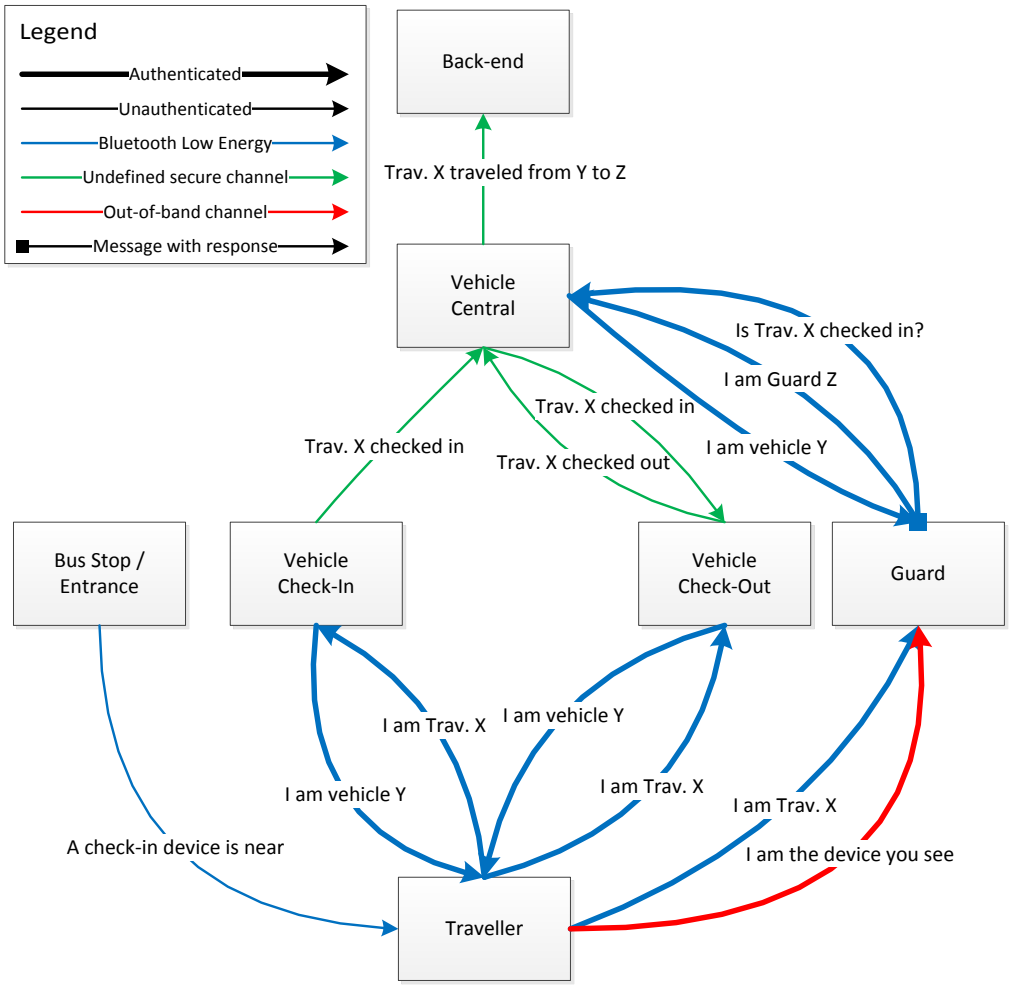



Figure 4.2: Global design scheme of the proposed protocol.

**Vehicle Central:** The vehicle central device keeps track of all travellers present in a vehicle. The device is connected (by a secure mobile data connection, which is not considered in further detail) to the central back-end, but it can operate without this connection. When this is the case, journeys will be saved locally and sent to the back-end as soon as the connection is restored.

**Vehicle Check-in:** In a bus, the check-in device will be placed close to the entrance and the driver. It will be configured to check-in devices only in a range of about 20cm. This means an explicit action of the traveller is needed. When a successful check-in is performed, the driver will be notified by a visual and audible signal. To make sure a check-in can be registered fast enough, an additional device will be placed near the bus entrance and/or at the bus stop. This device notifies any traveller's device within range to change its advertising settings for a certain period.

In a train, the check-in device will be configured to check in any device within the range that is considered to be inside the train.

**Vehicle Check-out:** The check-out device regularly checks if a traveller who checked in is still present. If the traveller's device is not in range for a certain period, the traveller will be checked out. Note that a single device can act as both check-in, check-out and vehicle central device. However, in a larger vehicle, there can be multiple check-out devices that cooperate to determine whether a traveller moved out of the vehicle or changed location within the vehicle.

**Bus Stop/Entrance:** As explained before, a device close to the bus entrance or bus stop can be configured to send a request to all near traveller's devices to decrease their advertising interval for a short period.

**Traveller:** The traveller's device will be a smartphone owned by the traveller. The device accepts connections from three kinds of devices:

- From the check-in/out devices - when checked in, the device will show the current check-in status.
- From the guard's device - when requested, the device will show the verification code from the guard device as explained below.
- From the bus stop/entrance device - on request, the advertising parameters will be changed. This will result in slightly higher energy consumption in this period for the traveller's device, but also in a lower latency for checking in.

**Guard:** The guard's device is connected to the vehicle central device and only connects to traveller's device in a small range (chosen by signal strength). The guard's device checks with the vehicle central device if the traveller is indeed checked in, and the guard can verify manually (for example by a random code displayed on both the guard and traveller device) if the connected device is the device shown by the traveller. When a lower connection latency for connecting to the traveller's device is needed, the guard's device can be configured, in the same way as the bus stop/entrance device, to request a change in the advertising settings of the traveller's device when it is near.

To save energy for the traveller's device, the wireless connectivity could be turned off by default and turned on only when the accelerometer data suggests that the user is travelling with a public transportation vehicle. Several researchers proposed methods to detect transportation mode based on accelerometer data. [26, 45]

| | Authentication | Anonymity | Unlinkability | Perfect forward secrecy | Key recency | Key revocation | Public key known in advance | Number of messages | Computation cost (user) |
|---|---|---|---|---|---|---|---|---|---|
| Yang *et al.* [61][a] | Mutual | Weak | No[b] | Yes | No | No | Yes | 3 | 4.25SM |
| He *et al.* [24] | Mutual | Strong | Yes | Yes | No | Yes | Yes | 3 | 15.75SM |
| Wang *et al.* [57] | User | No[c] | No[d] | Yes | No | No | Yes | 1 | 4SM |
| Almuhaideb *et al.* [3] | Mutual | Weak | No | No | Yes | No | Yes | 2 | 3SM |
| Li *et al.* [36] | Mutual | Weak | Yes | Yes | No | No | Yes | 3 | 4SM |
| Liu *et al.* [39] | Mutual | Strong | Yes | Yes | Yes | Yes | Yes | 3 | -[e] |
| Required protocol | Mutual | Weak | Yes | Yes | Yes | Yes | No | - | - |

[a] Protocol 1
[b] Only unlinkable within session
[c] Only after authentication phase
[d] Only unlinkable within session
[e] Not mentioned in ECSM, but more complex then [24]

Table 4.3: Overview of properties for existing roaming protocols.

### 4.4.2 Previous work

We searched for existing literature regarding protocols that fit the requirements as formulated. We found no protocol specifically developed for the proposed application in public transportation payment. We did find a series of literature about roaming authentication protocols. These protocols are developed for a situation where a mobile device (for example a mobile phone) can use services (for example an internet connection) provided by a foreign server based on his subscription with his home server. This can be applied to our application in the following way: the traveller's device is the mobile device, the vehicle device the foreign server and the back-end the home server.

These protocols can provide two kinds of anonymity: *weak anonymity* meaning that the identity of the user is not revealed to eavesdroppers and *strong anonymity* which means the identity is not revealed to the foreign server as well. For our application, we only require weak anonymity because we need the identity of the traveller to be able to bill for the journey.

A lot of these protocols require the home server to be involved in every authentication [59, 58, 50, 60]. Because there is no continuous connection

between the vehicle and the back-end, these protocols are not suitable.

There are some protocols that do not require this continuous connection. Yang *et al.* [61] described two protocols: one providing weak and one providing strong anonymity. The weak anonymity protocol provides perfect forward secrecy, but it provides unlinkability only within a session, not between different sessions. They showed how a billing system can be added even while maintaining strong anonymity. He *et al.* [24] proposed *Priauth*, a protocol providing strong anonymity, PFS and unlinkability. Weak anonymity is not possible in their case. A revocation mechanism is used for the mobile device. Wang *et al.* [57] proposed a protocol that exchanges only 2 messages with strong anonymity and PFS. No authentication is included and unlinkability is only implemented within a foreign device, not between different foreign devices. Almuhaideb *et al.* [3] introduced a protocol which is recency-evidence-based for the mobile device. Recency-evidence, originally introduced by [47], means that a user regularly gets proof from his home-server to prove his subscription is still valid. Though anonymity is ensured, unlinkability is not provided. Li *et al.* [36] presented a protocol with weak or strong anonymity at choice with PFS. Unlinkability only applies with strong anonymity. No recency evidence or revocation mechanism is implemented. He *et al.* [25] reviewed *Priauth* [24] and proposed an alternative revocation mechanism based on recency evidence and show how this could be implemented in *Priauth*. They performed an analysis of its resistance to Denial of Service (DoS) attacks. Liu *et al.* [39] presented a protocol with an extensive security analysis and detailed implementation details. They combine recency-evidence with revocation lists. The protocol provides strong anonymity and unlinkability. The public key for all foreign servers is assumed to be known by all mobile devices.

Apart from these, the IETF specified Transport Layer Security (TLS) 1.2 in RFC 5246 [17]. This protocol can be used with a number of algorithms. The protocol is widely used on the internet and commonly accepted. It is however not optimized for environments with little resources available. PFS can be achieved, but anonymous and unlinkable authentication is not implemented. The ISO 9798-3 standard [29] describes a protocol for mutual authentication. This protocol is based on certificates for both parties and does not provide anonymity or unlinkability for the user.

To the best of our knowledge, none of the existing protocols implement expiration or revocation for the foreign server's key, and none of the protocols implement the required combination of weak anonymity, full unlinkability between sessions and PFS, which is a strong requirement for our use-case.

The properties of the analysed protocols are summarized in Table 4.3. With respect to the computation complexity all computationally efficient computations like hashing and symmetric encryption are neglected. The complexity is measured in Elliptic Curve Scalar Multiplications (ECSM) [61].

### 4.4.3 Proposed protocol details

We describe the proposed protocol below. We describe the key distribution in the system and present the authentication procedures for the connections in the system.

#### Key distribution

Fig. 4.3 shows an overview of the keys in the system and how they are distributed.

The **back-end** functions as Certification Authority (CA), so it has a persistent CA public/private key pair. This private key is used to sign certificates and tokens that are issued to vehicles, guards and travellers. The back-end keeps track of a revocation lists for issued certificates and tokens. These keys are the only keys in the system that will never change.

The **vehicle** has the CA-Public key to verify signatures for tokens and certificates. The vehicle regularly generates an own public/private key pair. The public key is sent to the back-end, to receive a public key certificate, which is valid for a limited period. It receives revocation lists for guard certificates and traveller tokens.

The **guard** has the CA-Public key to verify signatures for tokens and certificates. The guard regularly generates an own public/private key pair. The public key is sent to the back-end, to receive a public key certificate, which is valid for a limited period. He regularly receives revocation lists for vehicle certificates and traveller tokens.

The **traveller** has the CA-Public key to verify signatures for certificates. The traveller regularly receives a token which is valid for a limited period, and receives revocation lists for vehicle and guard certificates.

Note that we use a public key certificate for the guard, but only a (simpler) token for the traveller. The reason for this choice is that a token has a smaller size, so this results in less data to be transmitted. The protocol will make sure that only trusted vehicles and guards can obtain the traveller's token to make sure it can not be used for malicious authentications.

We set the following values for key lifetime and revocation distribution. For the vehicle and guard devices, we require a connection to the back-end at least every day. At this moment, the journeys registered within the last 24 hours are be uploaded, and a new public key with certificate is issued. This means the certificates issued should have a validity of two days from the moment of distribution. The revocation list for travellers is distributed with at least the same frequency.

For the travellers, we choose to be more conservative, because a traveller might not have a data connection at every given moment and we do not

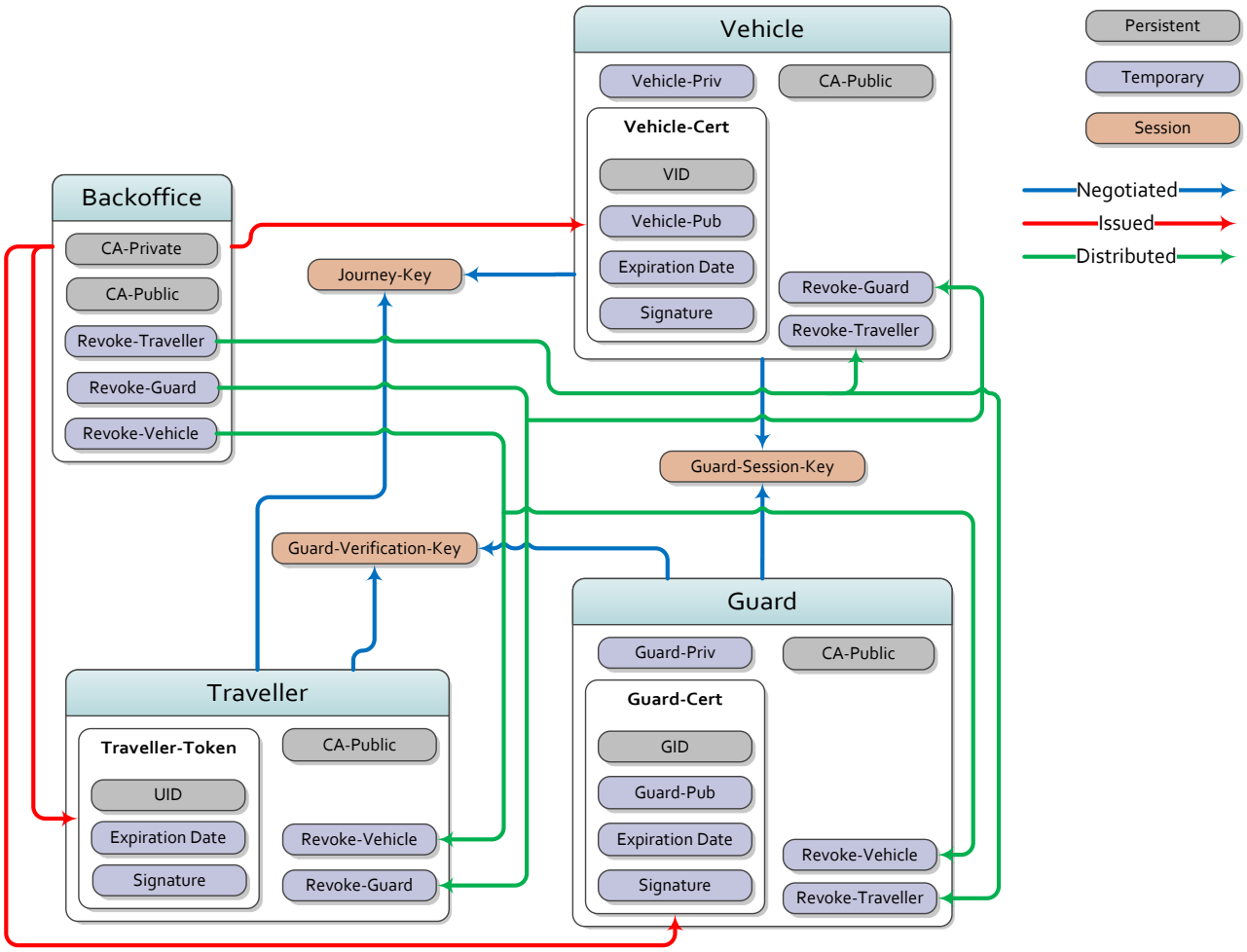



Figure 4.3: Key distribution scheme.

want to use this connection more than needed. We suggest to issue tokens with a validity of two weeks, and try to update these certificate when the left-over validity is one week. This way, a traveller will only be unable to check in when he does not have an internet connection for more than a week. The revocation lists for vehicles and guards will be distributed once a day. When a traveller does not receive this list on a certain day, he will still be able to use the system using the old revocation list. In this case, however, the risk of connecting to a malicious vehicle will slightly increase.

### Authentication procedures

We describe the authentication and key exchange procedures for the different connections in the system. An overview of all symbols used in these procedures can be found in Table 4.4. The sizes in this table will be explained later.

| Symbol | Description | Size (bits) |
|---|---|---|
| $PubCA$ | CA Public Key | 512 |
| $PrivCA$ | CA Private Key | 256 |
| $VID$ | Vehicle ID | 32 |
| $PubV$ | Vehicle Public Key | 384 |
| $PrivV$ | Vehicle Private Key | 192 |
| $CertV$ | Vehicle Certificate | 992 |
| $GID$ | Guard ID | 32 |
| $PubG$ | Guard Public Key | 384 |
| $PrivG$ | Guard Private Key | 192 |
| $CertG$ | Guard Certificate | 992 |
| $DH$-1 | First Diffie-Hellman 'half' key | 288 |
| $DH$-2 | Second Diffie-Hellman 'half' key | 288 |
| $UID$ | Traveller ID | 32 |
| $Token$ | Traveller Token | 608 |
| $KeyJ$ | Journey Key | 72 |
| $KeyG$ | Guard Session Key | 72 |
| $R$ | Temporary Authentication Key | 64 |
| $Verif$ | Guard Verification Key | 128 |
| $Chal$ | Random challenge | 72 |
| $MAC_{Key}(Data)$ | Message Authentication Code for $Data$ using $Key$ | 64 |
| $E_{Key}(Data)$ | Result of encrypting $Data$ using $Key$ | - |
| $E_{Key}(Data)$ | Result of signing $Data$ using $Key$ | - |

Table 4.4: Overview of symbols used in protocol descriptions.

**Traveller - Vehicle:** Fig. 4.4 shows the key establishment and authentication protocol between the traveller and the vehicle. The vehicle sends a certificate with his public key ($CertV$) to the traveller. The traveller uses this key to encrypt a randomly chosen key ($R$) and sends this to the vehicle, together with the first part of the Diffie-Hellman key ($DH$-1) and a message authentication code (MAC) created with key $R$. The vehicle responds with the second part of the DH key ($DH$-2) and a MAC for this key created with $R$. Both parties can now construct the journey key ($KeyJ$). The traveller uses this key to encrypt his secret token and sends this to the vehicle.

In Fig. 4.5 the presence check procedure between the vehicle and the traveller can be found. The vehicle sends a random message, the challenge ($Chal$) to the traveller, encrypted with the journey key ($KeyJ$). To prove he knows the journey key, the traveller decrypts the challenge, adds an OK message, encrypts the results and sends this back to the vehicle.

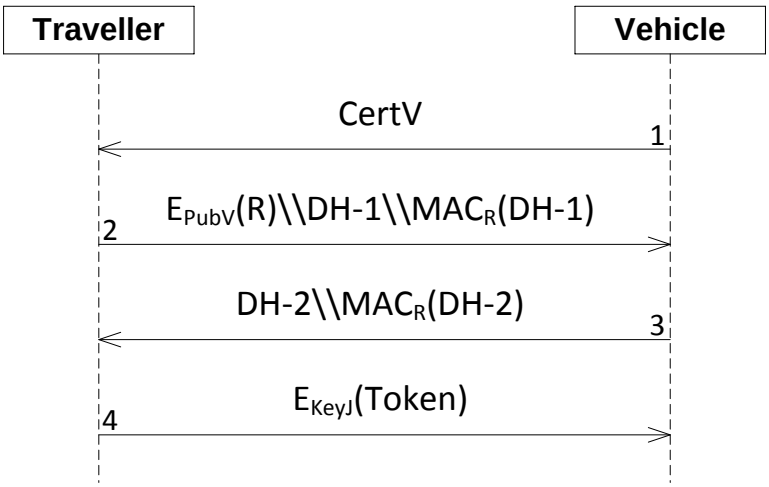



Figure 4.4: Check-in procedure between traveller and vehicle.

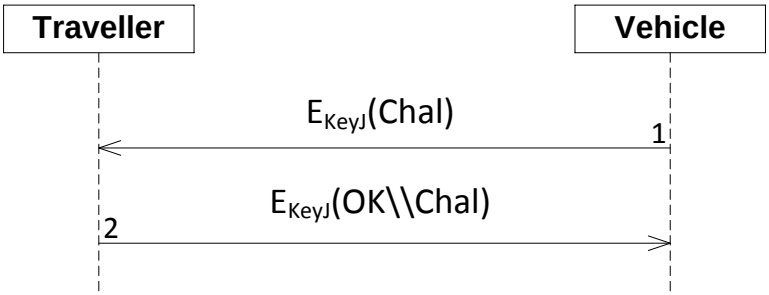



Figure 4.5: Presence check between traveller and vehicle, after check-in.

**Traveller - Guard:** The key establishment and authentication procedure between the traveller and guard for verifying a traveller is shown in Fig. 4.6. This procedure is very similar to the procedure between the traveller and the vehicle, except for the verification key. In the third step, the journey key is known to the guard already. So, the verification key can be sent immediately to allow the guard to check it.

**Vehicle - Guard:** For the authentication procedure between the guard and the vehicle, the requirements are different. We do need to establish a session key and perform mutual authentication, but anonymity and unlinkability are not required. Furthermore, this procedure should work using just the guard and vehicle certificates, without any additional tokens or keys distributed in advance.

We developed a key exchange and authentication procedure based on ISO 9798-3 [29]. This procedure can be found in Fig. 4.7. As this procedure does not involve the traveller's device, energy consumption is not as important as for the previous procedures. For this reason, we will not consider it in the further analysis.

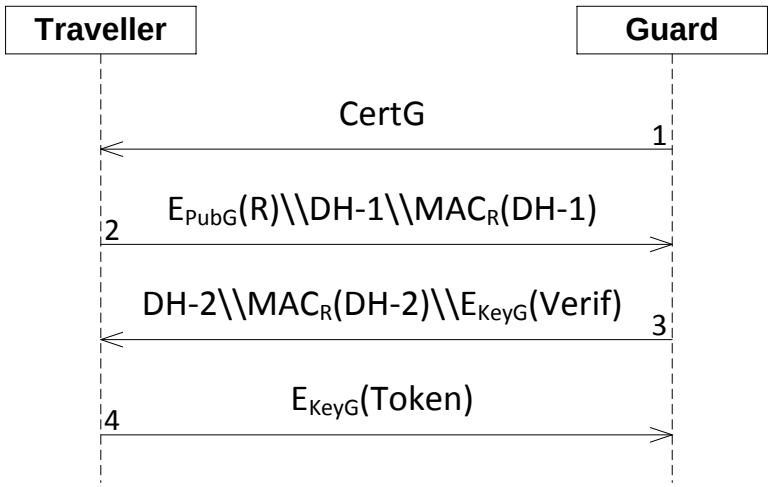



Figure 4.6: Key establishment and authentication procedure between the traveller and guard.

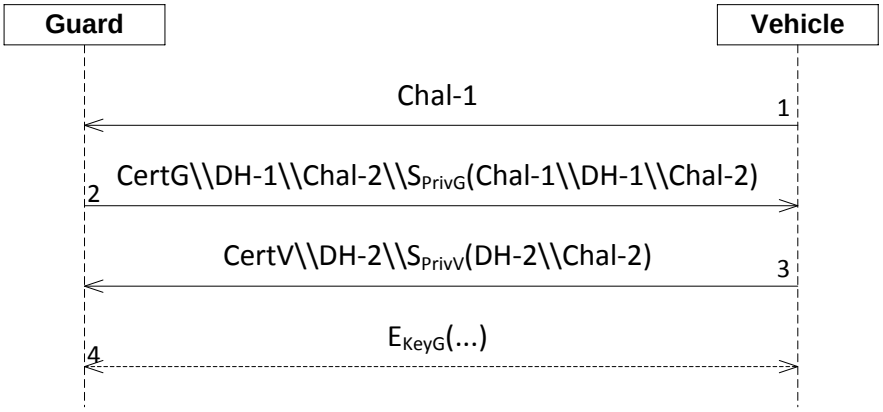



Figure 4.7: Key establishment and authentication procedure between the vehicle and guard.

### 4.4.4 Security analysis

There exist formal techniques to prove security of protocols, for example as proposed by [54]. We will not apply those here because of time constraints in this project. We give an intuitive explanation why the protocol meets the requirements and why it is secure against a number of known attack types. As there is a large similarity between the traveller-vehicle and traveller-guard procedures, we only do this for the traveller-vehicle procedure.

**Mutual authentication:** The vehicle reveals its identity in the public key certificate. Only a device knowing the secret private key corresponding with the certificate can decrypt the secret session key $R$. When the vehicle sends the MAC that it generated using $R$ in step 3, it proves to the traveller that it indeed is the vehicle that it claims to be.

The traveller's token is secret and only known by the traveller. The vehicle validates the signature of the token. If the signature is valid, the traveller is authenticated.

**Key establishment:** As a Diffie-Hellman key exchange mechanism is used to calculate the secret journey key $KeyJ$, an eavesdropper knowing both $DH$-1 and $DH$-2 is still unable to calculate $KeyJ$.

**Anonymity:** The first time the traveller exposes information revealing his/her identity is in step 4, by sending his token. However, this token is encrypted with the secret journey key before transmission. This means the anonymity of the traveller is guaranteed.

**Unlinkability:** As stated above, the first information based on the traveller's identity is exchanged in step 4. This information is encrypted using the secret journey key which changes every journey. Consequently, the ciphertext will change. This means an eavesdropper is unable to link two journeys made by the same traveller.

**Man-in-the-middle attacks:** In step 1, an attacker is unable to modify the message because this would invalidate the signature of the certificate. In step 2, an attacker could replace $R$ by another value, but this would invalidate the MAC that is sent back in step 3. An attacker can not change $DH$-1 because this would invalidate the MAC in the same step. In step 3, an attacker is unable to make any changes, because this would invalidate the MAC. In step 4, an attacker is unable to make any changes, because he does not know the secret journey key $KeyJ$.

**Replay attacks:** The parameters for the Diffie-Hellman key exchange are generated randomly for every execution of the procedure by both parties. This means that any replay attack will fail, because the recorded messages are based on a different DH parameter.

### 4.4.5 Cryptographic details

In this section, we describe the cryptographic details for implementation of the proposed protocol. We start defining for all keys how much security is required. Then we select suitable algorithms and determine the resulting key lengths.

#### Required security levels

We determine the required minimum security levels for the keys used in the system based on the ECRYPT II Yearly Report on Algorithms and Key-

sizes [20]. This report gives advice on key lengths and encryption algorithms, based on the required period of the security and the available budget of possible attackers. The possible attackers range from 'Hacker', with a budget of $400, to 'Intelligence agency', with a budget of $400M.

The CA key, used for signing certificates and traveller tokens, is constant and will not change over time. This means very long-term protection is needed, so we require at least 128 bits security. For the vehicle and guard keys, that are valid for a limited time only, a short-term protection against agencies and a long-term protection against small organizations is sufficient. So, we require 80 bits of security. The session key is renewed every session, thus breaking this key means only information concerning 1 journey is at risk. Short-term protection against medium organizations requires a minimum security level of 72 bits. The temporary authentication key used in the authentication procedure is valid for only a very small time. So, this will only need to be safe against real-time attacks. We select a minimum level of 64 bits security for this key. The required security levels are summarized in Table 4.5

**Algorithm choices**

In this section, we select the algorithms to be used for the implementation of the protocol.

**Hash functions:** For applying digital signatures, a hash function is needed. We select the SHA-256 hash algorithm as defined by [21] and recommended by [20].

**Symmetric encryption:** Where symmetric encryption is needed, we select the Advanced Encryption Standard (AES) [42] in CCM mode [18]. Both are recommended by [20] and support is included in the Bluetooth Core Specification. AES has a minimum key length of 128 bits. When the key exchange results in a shorter key, the selected hash algorithm will be used to inflate the key to 128 bits.

**Asymmetric encryption:** The most well-known asymmetric encryption protocol is RSA[31]. A big disadvantage of RSA is the key size. For the required security level of 128 bits, a key length of over 3072 bits is required.

A relatively new class of asymmetric encryption algorithms are those based on Elliptic Curve Cryptography (ECC). These systems are approved by the NIST for US government encryption [5]. Systems based on elliptic curves have the advantage of relatively small key sizes: twice the size of the asymmetric equivalent key length is advised by [20]. For this reason, we select ECC for our system. When using ECC, the following relations hold

for the sizes of keys, blocks of information to be encrypted and the resulting encrypted blocks of information [56]: the private key has the length of the keysize. The public key is twice the keysize. A plaintext block is twice the keysize, and the resulting ciphertext is twice the plaintext size.

**Message authentication codes (MAC):** For message authentication codes in the protocol, we choose CMAC as described by NIST [19] and recommended by [20]. This MAC is AES-based and has a minimum recommended tag size of 64 bits.

**Digital signatures** In the certificates and tokens, digital signatures are used. We use the Elliptic Curve Digital Signature Algorithm (ECDSA) as described and approved by the NIST [34]. This algorithm results in a private key of twice the required security level, a public key twice the size of the private key and a generated signature of twice the size of the public key [2, 30].

**Key exchange** For secure key exchange, we need a key exchange protocol. The classic approach is the Diffie-Hellman protocol [41], but this has the same disadvantage as RSA: large key sizes and high computational complexity. The NIST describes Ephemeral Unified Model Elliptic Curve Cryptography Cofactor Diffie-Hellman [5]. This protocol uses ECC to reduce key size and complexity. For the key exchange, both parties need to send an ephemeral public key with a double the size of the key length [1].

### Key sizes

For the keys where asymmetric encryption is used, we need to convert the symmetric equivalent security level to an asymmetric key length. When Elliptic Curve Cryptography is used, the recommendation from [20] is to use a key with twice the size of the symmetric equivalent.

For the certificate and token signatures, we select the NIST-suggested P-256 curve [34]. Private keys for this curve have a size of 256 bits, public keys are 512 bits bytes long and generated signatures are 512 bits long as well.

For the vehicle and guard keys, we need a key length of at least 160 bits. We use the NIST-suggested P-192 curve here, which results in a private key size of 192 bits and a public key size of 384 bits.

For ECDHE the NIST [4] suggests a key size of at least 160 bits when a security level of 80 bits is required. We choose to use the NIST P-192 here as well. This results in a private key size of 192 bits and a public key size of 384 bits. This means that the effective key size of the negotiated session key will be 96 bits.

The resulting key sizes are summarized in Table 4.5.

| Key | Sec. | Key type | Min. key size | Key size |
|---|---|---|---|---|
| Certification Authority | 128 bits | ECC | 256 bits | 256 bits |
| Vehicle key | 80 bits | ECC | 160 bits | 192 bits |
| Guard key | 80 bits | ECC | 160 bits | 192 bits |
| Session key | 72 bits | Symmetric | 72 bits | 96 bits |
| Temp. authentication key | 64 bits | Symmetric | 64 bits | 64 bits |

Table 4.5: Overview of required security levels and resulting key sizes

### 4.4.6 Protocol implementation

In this section, we describe how the protocol as proposed in the sections before can be implemented using Bluetooth Low Energy. First, we split the messages exchanged in a protocol run in BLE packets. We make choices for a how often connections are made, BLE device roles and the way the BLE security modes are used. Finally the performance model from Section 4.2.2 is applied.

#### Message sizes

We combine the information collected above to determine the actual size of the messages transmitted in the authentication, presence check and verification procedures. Table 4.6 lists the contents of certificates and tokens. Table 4.4 shows the size of transmitted parts in the protocol, based on the algorithms selected above.

| **Bits** | **Field** |
|---|---|
| 32 | Certificate type |
| 32 | Expiration date |
| 32 | Vehicle/Guard ID |
| 384 | Public key |
| 512 | Signature |
| 992 | **Total** |

(a) Vehicle/Guard certificate.

| **Bits** | **Field** |
|---|---|
| 32 | Token type |
| 32 | Expiration date |
| 32 | Traveller ID |
| | |
| 512 | Signature |
| 608 | **Total** |

(b) Traveller token.

Table 4.6: Contents of certificates and tokens.

The resulting message sizes for each step can be found in Table 4.7.

#### BLE packet distribution

As BLE only supports relatively small-sized packets, we need to split all messages in packets with a maximum size of 20 bytes. We calculate the number of messages for each step using the following equation:

| S | Bits |
|---|---|
| 1 | 992 |
| 2 | $512 + 384 + 64 = 960$ |
| 3 | $384 + 64 = 448$ |
| 4 | 608 |

(a) Check-in.

| S | Bits |
|---|---|
| 1 | 72 |
| 2 | $72 + 32 = 104$ |

(b) Presence check.

| S | Bits |
|---|---|
| 1 | 992 |
| 2 | $512 + 384 + 64 = 960$ |
| 3 | $384 + 64 + 128 = 576$ |
| 4 | 608 |

(c) Guard verification.

Table 4.7: Message sizes for all protocol steps.

$$N = \left\lceil \frac{M}{160} \right\rceil,$$

where $N$ is the number of packets and $M$ the message size.

The required number of packets for every protocol step can be found in Table 4.8. In the next section we apply the performance model to these amounts of packets.

| S | Bits | Packets |
|---|---|---|
| 1 | 992 | 7 |
| 2 | 960 | 6 |
| 3 | 448 | 3 |
| 4 | 608 | 4 |
| **Total** | | 20 |

(a) Check-in.

| S | Bits | Packets |
|---|---|---|
| 1 | 72 | 1 |
| 2 | 104 | 1 |
| **Total** | | 2 |

(b) Presence check.

| S | Bits | Packets |
|---|---|---|
| 1 | 992 | 7 |
| 2 | 960 | 6 |
| 3 | 576 | 4 |
| 4 | 608 | 4 |
| **Total** | | 21 |

(c) Guard verification.

Table 4.8: Message sizes for all protocol steps.

Note that in version 4.2 of the Bluetooth Core Specification, the maximum packet size has been increased. This means no packets have to be split up any more, which will lead to simpler implementation and a performance gain.

### Continuous vs. ad-hoc connection

There are two options for using BLE connections in our system. The first option is to make a connection when a traveller is checked in and keep this connection active until the traveller leaves the vehicle again. The second option is to set-up a connection every time data needs to be exchanged, and disconnect immediately after.

In the previous section, we found that 21 packets need to be transmitted for a verification action. To perform this action within the required time of 1 s, a short connection interval would be necessary. For this reason, we choose to make a connection ad-hoc when needed.

### Role of BLE devices

As explained in Section 2.5, there are two roles in a BLE connection: one device is the peripheral device, that is advertising and the other is the central device, that listens to advertising messages. In our protocol implementation, the traveller device will either need to have the central or peripheral role.

With the traveller as central, the traveller's device is completely in control and is not required to accept any connections from other devices. Certificates can be broadcast by the vehicle and picked up. Furthermore, there is only one device advertising, which means no collisions between advertisement messages will occur.

If we choose the traveller as peripheral device, we can support the Bluetooth 4.0 Specification, which allows only one central device to be connected to every peripheral device. The vehicle can completely determine his connection schedule, and incorporate data about the location of the vehicle in this schedule (between two consecutive stations only one presence check has to be performed). Lastly, this will not cause message collisions from connection requests within the system.

We summarize the advantages of both options below:

**Traveller as Central**

- More control for traveller
- Possibility to broadcast certificates
- Traveller's responsibility to connect
- Less collision in advertising messages

**Traveller as Peripheral**

- Possible using BLE 4.0
- Vehicle controls connection schedule
- Vehicle can incorporate location in schedule
- Less collision in connection requests

As we chose to build the system according to the Bluetooh 4.0 Standard, we chose the central role for the vehicle device and the peripheral role for the traveller's device for our implementation.

In a future version of the protocol, these roles could be changed. In this case, more research is needed regarding the performance and energy consumption for a situation where the traveller's device fulfils the peripheral role.

### Encryption implementation

To ensure unlinkability for the traveller, his/her device should use an address of the *random* type (see Section 2.5). For the authentication and key exchange procedure, it is not possible to use the BLE link layer security functions, because these are either insecure or need a symmetric key exchange in advance (see Section 2.5). From the moment when the symmetric session key is established there are two possibilities: start using the BLE link layer encryption or keep using encryption in the application layer. By using the BLE link layer encryption, we make the implementation of the

protocol easier since, according to the standard, this encryption is already supported by all devices. So, we choose to use the built-in BLE encryption.

### 4.4.7 Protocol performance

In this section, we analyse the performance of the proposed protocol. This way we evaluate whether the performance requirements from Section 4.2 can be met and what the energy consumption for the traveller's device when using the system will be. To do this, we use the proposed energy model. To evaluate the resulting impact for travellers, we consider a 'typical traveller'. For this traveller the system is idle for 16 hours. The person travels twice for 30 min in which his/her presence is checked every minute. The traveller is verified once by a railway guard. The energy consumption $E_{daily}$ for this traveller is calculated using the following equation:

$$E_{daily} = \frac{1}{1000} \cdot 16 \cdot 3600 \cdot E_{idle} + 2 \cdot E_{checkin} + 2 \cdot 30 \cdot E_{presence} + E_{verify} \quad (4.7)$$

We applied the performance model for two cases - first for 'normal operation', where the device is advertising with a low rate, and then for the situation where the device is advertising at an increased rate, for checking in at a bus. The results can be found in Table 4.9.

| | **Param.** | **Unit** | Normal operation | Bus check-in |
|---|---|---|---|---|
| Inputs | $I_{adv}$ | ms | 1000 | 20 |
| | $I_{conn}$ | ms | 7.5 | 7.5 |
| | $P_{checkin}$ | # | 20 | 20 |
| | $P_{presence}$ | # | 2 | 2 |
| | $P_{verify}$ | # | 21 | 21 |
| | $N_{trav}$ | # | 50 | 1 |
| Outputs | $E_{idle}$ | mJ/s | 0.22 | 7.3 |
| | $E_{checkin}$ | mJ | 6.0 | 6.0 |
| | $E_{presence}$ | mJ | 1.14 | 1.14 |
| | $E_{verify}$ | mJ | 6.27 | 6.27 |
| | $T_{checkin}$ | s | 10.7 | 0.29 |
| | $E_{daily}$ | J | 12.6 | - |

Table 4.9: Results of the application of the performance model.

From these results, we can conclude that the typical daily energy consumption of 12.6 J is negligible for an average smartphone battery, which has a capacity of tens of kJ's. The check-in time with increased advertising rate is 290 ms. The maximum allowed time was 250 ms, so this requirement is not met. However, the value is probably close enough to result in a working system.

# Chapter 5

# Conclusions and Future Work

In this work, we have studied a Bluetooth Low Energy based automated payment system for public transportation system. We analysed the system requirements for such a system and listed a number of research problems related with such a system. Within the scope of this Master's thesis, we focused our study on two major points: (i) energy consumption and interference aspect of a BLE device and (ii) a protocol design for the payment system with emphasize on security and privacy aspects.

### Energy and interference

We performed a latency simulation of the BLE device discovery process and experiments for the energy consumption and latency under mutual interference. We developed a model based on the data gathered to predict performance of BLE in certain situations.

We can conclude that BLE is pretty robust in terms of interference resistance. The developed model can be helpful to developers of applications that are using BLE and want to predict energy consumption and the influence of interference. With respect to the be-in/be-out system, we used the model as an input for the development of the communication protocol.

### Communication protocol

We developed a communication protocol that is suitable for the be-in/be-out system. The protocol implements all the operational and performance requirements found. We developed a performance model and used this to find feasible configuration parameters for the system.

The protocol contains a novel authentication procedure that implements a combination of weak anonymity, unlinkability and energy efficiency. This

can be useful in other applications as well. With respect to the be-in/be-out system, the protocol is the basis of the actual implementation of the system.

#### Demo implementation

We have implemented a demo of the system. This demo is helpful to show in practice how the be-in/be-out system functions. During the implementation of the demo, no problems were encountered that are a threat for the feasibility of the concept.

#### Complete system

Combining the conclusions from the studies above, we believe it is possible to implement the be-in/be-out system with the currently existing technology. For a typical user, who has the system activated for 16 hours on a day and actively travels for 1 hour, we find a total energy consumption of 12.6 J, or less than 0.1 % of the capacity of a typical smartphone battery. With the latest version of the Bluetooth standard, this number can even be decreased further.

## 5.1 Future Work

We have identified a number of directions for future work. With regard to energy we only performed experiments for certain scenarios and developed a model based on simulation and experimental data. Further research could include developing a theoretical analytical model and validating this using experiment data. We only considered interference from other BLE devices. Further research could be done to the influence of interference from other sources transmitting in the 2.4 GHz ISM band.

The protocol developed is currently supported by a very brief security analysis. An extensive security analysis would be necessary before using it in practice. We made an estimate of the protocol performance, but a real-life evaluation is required to validate this estimate.

For the complete be-in/be-out system we only researched the aspects mentioned above in detail. The accuracy and verification of the system must be studied in more detail. The proposed solution for the localization problem must be validated and the complete system must be subjected to a thorough field test.

# Bibliography


[1] National Security Agency. Suite b implementer's guide to nist sp800-56a. Technical report, 2009.

[2] National Security Agency. Suite b implementer's guide to fips 186-3 (ecdsa). Technical report, 2010.

[3] A. Almuhaideb, Phu Dung Le, and B. Srinivasan. Two-party mobile authentication protocols for wireless roaming networks. In *Network Computing and Applications (NCA), 2011 10th IEEE International Symposium on*, pages 285–288, Aug 2011.

[4] Elaine B. Barker, William C. Barker, William E. Burr, W. Timothy Polk, and Miles E. Smid. Sp 800-57. recommendation for key management, part 1: General (revision 3). Technical report, Gaithersburg, MD, United States, 2012.

[5] Elaine B. Barker, Don Johnson, and Miles E. Smid. Sp 800-56a. recommendation for pair-wise key establishment schemes using discrete logarithm cryptography (revised). Technical report, Gaithersburg, MD, United States, 2007.

[6] G. Benelli and A. Pozzebon. An automated payment system for car parks based on near field communication technology. In *Internet Technology and Secured Transactions (ICITST), 2010 International Conference for*, pages 1–6, Nov 2010.

[7] Giuliano Benelli and Alessandro Pozzebon. Innovative solutions for the automatic payment of car parks. *International Journal for Infonomics (IJI)*, 1, 2013.

[8] Bluegiga. BLE113 Development Kit 2.0 Datasheet. `https://www.bluegiga.com/`, 2013. Accessed: 2014-10-10.

[9] Bluegiga. Knowledgebase: Ble module low power and sleep modes. `https://bluegiga.zendesk.com/entries/23173106--REFERENCE-BLE-module-low-power-and-sleep-modes`, 2013. Accessed: 2014-10-13.

[10] Bluegiga. Knowledgebase: Throughput with bluetooth smart technology. `https://bluegiga.zendesk.com/entries/24646818-Throughput-with-Bluetooth-Smart-technology`, 2013. Accessed: 2014-07-21.

[11] Bluegiga. BLE113 Datasheet. `https://www.bluegiga.com/`, 2014. Accessed: 2014-10-14.

[12] Bluetooth SIG. Bluetooth specification version 4.0. *Bluetooth SIG*, 2010.

[13] Bluetooth SIG. Bluetooth specification version 4.1. *Bluetooth SIG*, 2013.

[14] Bluetooth SIG. Bluetooth specification version 4.2. *Bluetooth SIG*, 2014.

[15] B. Caulfield and M. O'Mahony. Passenger requirements of a public transport ticketing system. In *Intelligent Transportation Systems, 2005. Proceedings. 2005 IEEE*, pages 119–124, Sept 2005.

[16] Jo Woon Chong, Ho Young Hwang, Chang Yong Jung, and Dan Keun Sung. Analysis of throughput and energy consumption in a zigbee network under the presence of bluetooth interference. In *Global Telecommunications Conference, 2007. GLOBECOM'07. IEEE*, pages 4749–4753. IEEE, 2007.

[17] T. Dierks and E. Rescorla. RFC 5246 - The Transport Layer Security (TLS) Protocol Version 1.2. Technical report, Aug. 2008.

[18] Morris Dworkin. Sp 800-38c. recommendation for block cipter modes of operation: The ccm mode for authentication and confidentiality. Technical report, Gaithersburg, MD, United States, 2004.

[19] Morris J. Dworkin. Sp 800-38b. recommendation for block cipher modes of operation: The cmac mode for authentication. Technical report, Gaithersburg, MD, United States, 2005.

[20] ECRYPT II Consortium. ECRYPT II Yearly Report on Algorithms and Keysizes (2011-2012). 2012.

[21] Patrick Gallagher. Fips pub 180-4. secure hash standard (shs). Technical report, Gaithersburg, MD, United States, 2012.

[22] Mario Goldenbaum and Sławomir Stanczak. On multiantenna sensor networks with interference: Energy consumption vs. robustness. In *Smart Antennas (WSA), 2012 International ITG Workshop on*, pages 125–132. IEEE, 2012.

[23] Carles Gomez, Ilker Demirkol, and Josep Paradells. Modeling the maximum throughput of bluetooth low energy in an error-prone link. *Communications Letters, IEEE*, 15(11):1187–1189, 2011.

[24] Daojing He, Jiajun Bu, S. Chan, Chun Chen, and Mingjian Yin. Privacy-preserving universal authentication protocol for wireless communications. *Wireless Communications, IEEE Transactions on*, 10(2):431–436, February 2011.

[25] Daojing He, Chun Chen, Sammy Chan, and Jiajun Bu. Strong roaming authentication technique for wireless and mobile networks. *International Journal of Communication Systems*, 26(8):1028–1037, 2013.

[26] Samuli Hemminki, Petteri Nurmi, and Sasu Tarkoma. Accelerometer-based transportation mode detection on smartphones. In *Proceedings of the 11th ACM Conference on Embedded Networked Sensor Systems*, SenSys '13, pages 13:1–13:14, New York, NY, USA, 2013. ACM.

[27] R. Heydon. *Bluetooth Low Energy: The Developer's Handbook*. Pearson Always Learning. Prentice Hall, 2012.

[28] Ivan Howitt. Mutual interference between independent bluetooth piconets. *Vehicular Technology, IEEE Transactions on*, 52(3):708–718, 2003.

[29] ISO/IEC. 9798-3:1998(E): Information technology – Security techniques – Entity authentication – Part 3: Mechanisms using digital signature techniques. Standard, International Organization for Standardization, Geneva, CH, October 1998.

[30] ISO/IEC. 14888-3:2006: Information technology - Security techniques - Digital signatures with appendix - Part 3: Discrete logarithm based mechanisms (ISO/IEC. Standard, International Organization for Standardization, Geneva, CH, December 2006.

[31] J. Jonsson and B. Kaliski. Public-key cryptography standards (pkcs) #1: Rsa cryptography specifications version 2.1, 2003.

[32] J. Joppien. Improving system adaptation of the ov-chipkaart: Linking organisational requiremetns to a user-centered travel experience. Master's thesis, Delft University of Technology, June 2013.

[33] J. Joppien, G. Niermeijer, M.C. Niks, and J.I. van Kuijk. Analysis report: Exploring new possibilities for user-centred e-ticketing, March 2013.

[34] Cameron F. Kerry, Acting Secretary, and Charles R. Director. FIPS PUB 186-4 Federal Information Processing Standarts Publication Digital Signature Standard (DSS). 2013.

[35] Philipp Kindt, Daniel Yunge, Robert Diemer, and Samarjit Chakraborty. Precise energy modeling for the bluetooth low energy protocol. *arXiv preprint arXiv:1403.2919*, 2014.

[36] Xiaowei Li, Yuqing Zhang, Xuefeng Liu, Jin Cao, and Qianqian Zhao. A lightweight roaming authentication protocol for anonymous wireless communication. In *Global Communications Conference (GLOBECOM), 2012 IEEE*, pages 1029–1034, Dec 2012.

[37] Jia Liu, Canfeng Chen, and Yan Ma. Modeling and performance analysis of device discovery in bluetooth low energy networks. In *Global Communications Conference (GLOBECOM), 2012 IEEE*, pages 1538–1543. IEEE, 2012.

[38] Jia Liu, Canfeng Chen, Yan Ma, and Ying Xu. Energy analysis of device discovery for bluetooth low energy. In *Vehicular Technology Conference (VTC Fall), 2013 IEEE 78th*, pages 1–5. IEEE, 2013.

[39] J.K. Liu, Cheng-Kang Chu, S.M. Chow, X. Huang, M.H. Au, and J. Zhou. Anonymous authentication for roaming networks with efficient revocation for large scale networks. *Information Forensics and Security, IEEE Transactions on*, PP(99):1–1, 2014.

[40] T.L. McDaniel and F. Haendler. Advanced rf cards for fare collection. In *Telesystems Conference, 1993. 'Commercial Applications and Dual-Use Technology', Conference Proceedings., National*, pages 31–35, Jun 1993.

[41] Alfred J. Menezes, Scott A. Vanstone, and Paul C. Van Oorschot. *Handbook of Applied Cryptography*. CRC Press, Inc., Boca Raton, FL, USA, 1st edition, 1996.

[42] National Institute of Standards and Technology. FIPS 197: Announcing the Advanced Encryption Standard (AES). Technical report, 2001.

[43] G. Niermeyer. Design of a user-centered, open payment border for the ov-chipkaart. Master's thesis, Delft University of Technology, November 2013.

[44] M.C. Niks. Making the invisible visible: Increasing traveller's trust in electronic ticketing for public transport by making ticket information visible during the journey. Master's thesis, Delft University of Technology, September 2013.

[45] Philippe Nitsche, Peter Widhalm, Simon Breuss, Norbert Brndle, and Peter Maurer. Supporting large-scale travel surveys with smartphones a practical approach. *Transportation Research Part C: Emerging Technologies*, 43, Part 2(0):212 – 221, 2014. Special Issue with Selected Papers from Transport Research Arena.

[46] Harsh Kupwade Patil and Stephen A. Szygenda. *Security for Wireless Sensor Networks Using Identity-Based Cryptography*. Auerbach Publications, Boston, MA, USA, 1st edition, 2012.

[47] RonaldL. Rivest. Can we eliminate certificate revocation lists? In Rafael Hirchfeld, editor, *Financial Cryptography*, volume 1465 of *Lecture Notes in Computer Science*, pages 178–183. Springer Berlin Heidelberg, 1998.

[48] Tomas Rosa. Bypassing passkey authentication in bluetooth low energy. Cryptology ePrint Archive, Report 2013/309, 2013. `http://eprint.iacr.org/`.

[49] Mike Ryan. Bluetooth: With low energy comes low security. In *Proceedings of the 7th USENIX Conference on Offensive Technologies*, WOOT'13, pages 4–4, Berkeley, CA, USA, 2013. USENIX Association.

[50] Eun-Kyung Ryu, Gil-Je Lee, and Kee-Young Yoo. Unlinkable authentication for roaming user in heterogeneous wireless networks. In *Connected Vehicles and Expo (ICCVE), 2013 International Conference on*, pages 629–634, Dec 2013.

[51] M. Siekkinen, M. Hiienkari, J.K. Nurminen, and J. Nieminen. How low energy is bluetooth low energy? comparative measurements with zigbee/802.15.4. In *Wireless Communications and Networking Conference Workshops (WCNCW), 2012 IEEE*, pages 232–237, April 2012.

[52] A. Stranne, O. Edfors, and B.-A. Molin. Energy-based interference analysis of heterogeneous packet radio networks. *Communications, IEEE Transactions on*, 54(7):1299–1309, July 2006.

[53] André Stranne, Ove Edfors, and B-A Molin. Experimental verification of an analytical interference model for bluetooth networks. In *Personal, Indoor and Mobile Radio Communications, 2006 IEEE 17th International Symposium on*, pages 1–5. IEEE, 2006.

[54] F.J. Thayer Fabrega, J.C. Herzog, and J.D. Guttman. Strand spaces: why is a security protocol correct? In *Security and Privacy, 1998. Proceedings. 1998 IEEE Symposium on*, pages 160–171, May 1998.

[55] K. Townsend, C. Cufí, and R. Davidson. *Getting Started with Bluetooth Low Energy: Tools and Techniques for Low-Power Networking*. O'Reilly Media, Incorporated, 2014.

[56] J.C.A. van der Lubbe. *Basismethoden Cryptografie*. Delftse Universitaire Pers, 1994.

[57] Yuan Wang, Duncan S. Wong, and Liusheng Huang. A one-pass key establishment protocol for anonymous wireless roaming with pfs. In *Communications (ICC), 2011 IEEE International Conference on*, pages 1–5, June 2011.

[58] Peng Xiao, Jingsha He, and Yingfang Fu. A new authentication protocol for roaming in wireless mesh networks based on three-party key agreement. In *Multimedia Information Networking and Security (MINES), 2010 International Conference on*, pages 418–422, Nov 2010.

[59] Peng Xiao, Jingsha He, and Yingfang Fu. A secure mutual authentication protocol for roaming in wireless mesh networks. *Journal of Networks*, 7(2), 2012.

[60] Qi Xie, Mengjie Bao, Na Dong, Bin Hu, and D.S. Wong. Secure mobile user authentication and key agreement protocol with privacy protection in global mobility networks. In *Biometrics and Security Technologies (ISBAST), 2013 International Symposium on*, pages 124–129, July 2013.

[61] Guomin Yang, Qiong Huang, Duncan S. Wong, and Xiaotie Deng. Universal authentication protocols for anonymous wireless communications. *Wireless Communications, IEEE Transactions on*, 9(1):168–174, January 2010.

[62] Xiaojie Zhao, Zhuoling Xiao, Andrew Markham, Niki Trigoni, and Yong Ren. Does btle measure up against wifi? a comparison of indoor location performance. In *European Wireless 2014; 20th European Wireless Conference; Proceedings of*, pages 1–6, May 2014.

# Appendix A

# Measurement results

| Layout | Interferers (#) | | Intervals (ms) | | Runs | Discov. energy (mJ) | | Conn. energy (mJ) | | Transf. en. (mJ) | | Idle en. (mJ/s) | | Energy (mJ/s) | | Latency (ms) | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Adv. | Transf. | Conn. | Adv. | | Cent. | Periph. | Cent. | Periph. | Cent. | Periph. | Cent. | Periph. | Discov. | Adv. | Discov. | Conn. | Transf. |
| A | 0 | 0 | 7.5 | 20 | 500 | 1.709 | 0.222 | 2.746 | 0.601 | 0.236 | 0.270 | 10.313 | 14.394 | 88.470 | 6.910 | 21.3 | 48.0 | 12.4 |
| A | 0 | 6 | 7.5 | 20 | 100 | 1.949 | 0.220 | 2.680 | 0.662 | 0.345 | 0.431 | 9.988 | 14.246 | 88.665 | 6.697 | 24.0 | 48.6 | 23.0 |
| A | 0 | 12 | 7.5 | 20 | 100 | 1.931 | 0.222 | 2.728 | 0.696 | 0.425 | 0.549 | 9.963 | 14.334 | 88.614 | 6.685 | 23.8 | 49.2 | 30.8 |
| A | 0 | 18 | 7.5 | 20 | 100 | 2.083 | 0.241 | 3.071 | 0.702 | 0.532 | 0.707 | 9.941 | 14.388 | 88.824 | 6.785 | 25.4 | 52.9 | 41.1 |
| A | 0 | 24 | 7.5 | 20 | 100 | 2.209 | 0.256 | 3.249 | 0.758 | 0.577 | 0.783 | 9.937 | 14.391 | 88.779 | 6.682 | 26.9 | 57.6 | 45.9 |
| A | 0 | 30 | 7.5 | 20 | 100 | 2.343 | 0.261 | 3.353 | 0.724 | 0.600 | 0.813 | 9.946 | 14.394 | 88.774 | 6.729 | 28.4 | 58.1 | 47.9 |
| B | 0 | 0 | 7.5 | 20 | 500 | 1.706 | 0.221 | 2.753 | 0.603 | 0.236 | 0.270 | 10.292 | 14.417 | 88.543 | 6.938 | 21.3 | 48.4 | 12.4 |
| B | 6 | 0 | 7.5 | 20 | 500 | 2.157 | 0.255 | 3.494 | 0.700 | 0.239 | 0.275 | 10.310 | 14.436 | 88.507 | 6.941 | 26.4 | 61.4 | 12.7 |
| B | 12 | 0 | 7.5 | 20 | 500 | 2.881 | 0.308 | 4.563 | 0.833 | 0.247 | 0.287 | 10.289 | 14.463 | 88.531 | 6.939 | 34.5 | 78.9 | 13.5 |
| B | 18 | 0 | 7.5 | 20 | 500 | 3.376 | 0.346 | 5.895 | 0.971 | 0.256 | 0.299 | 10.269 | 14.447 | 88.629 | 6.918 | 40.0 | 99.2 | 14.3 |
| B | 24 | 0 | 7.5 | 20 | 500 | 4.196 | 0.409 | 7.315 | 1.132 | 0.259 | 0.303 | 10.274 | 14.434 | 88.716 | 6.905 | 49.1 | 122.4 | 14.6 |
| B | 30 | 0 | 7.5 | 20 | 500 | 5.420 | 0.490 | 9.202 | 1.319 | 0.266 | 0.313 | 10.292 | 14.464 | 89.359 | 6.901 | 62.4 | 149.2 | 15.2 |
| B | 0 | 30 | 7.5 | 20 | 500 | 4.329 | 0.419 | 5.323 | 0.957 | 2.870 | 4.284 | 10.042 | 15.043 | 90.856 | 6.917 | 49.5 | 88.1 | 267.3 |
| C | 6 | 0 | 7.5 | 20 | 500 | 2.177 | 0.263 | 3.452 | 0.704 | 0.239 | 0.276 | 10.311 | 14.441 | 88.535 | 6.964 | 26.6 | 61.6 | 12.7 |
| C | 12 | 0 | 7.5 | 20 | 500 | 2.704 | 0.297 | 4.370 | 0.806 | 0.236 | 0.272 | 10.306 | 14.443 | 88.563 | 6.904 | 32.5 | 76.2 | 12.4 |
| C | 18 | 0 | 7.5 | 20 | 500 | 3.420 | 0.352 | 5.466 | 0.933 | 0.240 | 0.277 | 10.304 | 14.453 | 88.575 | 6.955 | 40.5 | 93.7 | 12.8 |
| C | 24 | 0 | 7.5 | 20 | 500 | 4.075 | 0.402 | 6.957 | 1.099 | 0.239 | 0.275 | 10.307 | 14.452 | 88.596 | 6.933 | 47.8 | 117.0 | 12.6 |
| C | 30 | 0 | 7.5 | 20 | 500 | 4.681 | 0.451 | 8.521 | 1.230 | 0.235 | 0.269 | 10.313 | 14.440 | 88.532 | 6.948 | 54.7 | 138.4 | 12.3 |
| C | 0 | 6 | 7.5 | 20 | 500 | 1.712 | 0.226 | 2.751 | 0.620 | 0.292 | 0.355 | 10.259 | 14.586 | 88.501 | 7.056 | 21.4 | 48.4 | 17.8 |
| C | 0 | 12 | 7.5 | 20 | 500 | 1.644 | 0.221 | 2.736 | 0.634 | 0.338 | 0.422 | 10.220 | 14.679 | 88.510 | 6.993 | 20.6 | 48.7 | 22.2 |
| C | 0 | 18 | 7.5 | 20 | 500 | 1.706 | 0.220 | 2.741 | 0.631 | 0.423 | 0.549 | 10.192 | 14.748 | 88.505 | 6.959 | 21.3 | 48.7 | 30.3 |
| C | 0 | 24 | 7.5 | 20 | 500 | 1.711 | 0.220 | 2.771 | 0.654 | 0.496 | 0.660 | 10.155 | 14.827 | 88.472 | 6.948 | 21.3 | 49.6 | 37.5 |
| C | 0 | 30 | 7.5 | 20 | 500 | 1.721 | 0.224 | 2.740 | 0.655 | 0.611 | 0.833 | 10.132 | 14.868 | 88.465 | 6.954 | 21.5 | 49.4 | 48.6 |
| D | 0 | 30 | 7.5 | 20 | 500 | 1.894 | 0.233 | 2.954 | 0.667 | 0.562 | 0.751 | 10.179 | 14.707 | 88.728 | 6.904 | 23.3 | 52.1 | 43.2 |
| A | 0 | 0 | 20 | 50 | 500 | 3.048 | 0.213 | 5.576 | 0.680 | 0.245 | 0.279 | 4.107 | 5.453 | 88.169 | 3.730 | 36.5 | 99.9 | 27.6 |
| A | 0 | 0 | 50 | 100 | 500 | 5.062 | 0.204 | 9.762 | 0.680 | 0.269 | 0.266 | 1.686 | 2.105 | 87.981 | 2.049 | 59.3 | 185.1 | 99.0 |
| A | 0 | 0 | 100 | 200 | 500 | 9.248 | 0.191 | 18.495 | 0.639 | 0.277 | 0.269 | 0.947 | 1.028 | 88.023 | 0.935 | 106.6 | 403.6 | 152.9 |
| A | 0 | 0 | 200 | 400 | 100 | 20.057 | 0.199 | 36.013 | 0.663 | 0.286 | 0.292 | 0.420 | 0.540 | 88.450 | 0.529 | 228.6 | 801.7 | 304.6 |

Table A.1: Complete measurement results from the energy experiments.