NTNU – Trondheim
Norwegian University of
Science and Technology

Understanding Data Analysis in an End-to-End IoT System

Sindre Schei

Submission date: April 2016
Responsible professor: Frank Alexander Kraemer, ITEM
Supervisor: David Palma, ITEM

Norwegian University of Science and Technology
Department of Telematics

Abstract

Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like “Huardest gefburn”? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

This is the second paragraph. Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like “Huardest gefburn”? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

And after the second paragraph follows the third paragraph. Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like “Huardest gefburn”? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

After this fourth paragraph, we start a new paragraph sequence. Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like “Huardest gefburn”? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of

the original language. There is no need for special content, but the length of words should match the language.

Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like “Huardest gefburn”? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

Sammendrag

Sikkerheten til nesten all offentlig nøkkel-kryptografi er basert på et vanskelig beregnbarhetsproblem. Mest velkjent er problemene med å faktorisere heltall i sine primtallsfaktorer, og å beregne diskrete logaritmer i endelige sykliske grupper. I de to siste tiårene, har det imidlertid dukket opp en rekke andre offentlig nøkkel-systemer, som baserer sin sikkerhet på helt andre type problemer. Et lovende forslag, er å basere sikkerheten på vanskeligheten av å løse store likningsett av flervariable polynomlikninger. En stor utfordring ved å designe slike offentlig nøkkel-systemer, er å integrere en effektiv "falluke" (trapdoor) inn i likningssettet. En ny tilnærming til dette problemet ble nylig foreslått av Gligoroski m.f., hvor de benytter konseptet om kvasigruppe-strengtransformasjoner (quasigroup string transformations). I denne masteroppgaven beskriver vi en metodikk for å identifisere sterke og svake nøkler i det nylig foreslåtte multivariable offentlig nøkkel-signatursystemet MQQ-SIG, som er basert på denne idéen.

Vi har gjennomført et stort antall eksperimenter, basert på Gröbner basis angrep, for å klassifisere de ulike parametrene som bestemmer nøklene i MQQ-SIG. Våre funn viser at det er store forskjeller i viktigheten av disse parametrene. Metodikken består i en klassifisering av de forskjellige parametrene i systemet, i tillegg til en innføring av konkrete kriterier for hvilke nøkler som bør velges. Videre, har vi identifisert et unødvendig krav i den originale spesifikasjonen, som krevde at kvasigruppene måtte oppfylle et bestemt kriterie. Ved å fjerne denne betingelsen, kan nøkkelgenererings-algoritmen potensielt øke ytelsen med en stor faktor. Basert på alt dette, foreslår vi en ny og forbedret nøkkel-genereringsalgoritme for MQQ-SIG, som vil generere sterkere nøkler og være mer effektiv enn den originale nøkkel-genereringsalgoritmen.

Preface

Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like “Huardest gefburn”? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

Contents

List of Figures

List of Tables

List of Algorithms

List of Symbols

A_B^C A dummy symbol.

List of Acronyms

6LoWPAN IPv6 over Low Power Wireless Personal Area Networks.

BLE Bluetooth Low Energy.

CoAP Constrained Application Protocol.

DNS Domaine Name System.

GUI Graphical User Interface.

HCI Host Controller Interface.

I2C Inter-Integrated Circuit.

ICMP Internet Control Message Protocol.

IDE Integrated Development Environment.

IoT Internet of Things.

IPv4 Internet Protocol version 4.

IPv6 Internet Protocol version 6.

LAN Local Area Network.

MQTT Message Queueing Telemetry Transport.

NTNU Norwegian University of Science and Technology.

OS Operating System.

PAN Personal Area Network.

SPI Serial Peripheral Interface.

TCP Transmission Control Protocol.

TFTP Trivial File Transfer Protocol.

UDP User Datagram Protocol.

Chapter 1
Introduction

1.1 Motivation

This is a test in the introduction chapter.

1.2 Methodology

1.3 Structure

Chapter 2
Background

This thesis describes the setup and usage of an End-to-End Internet of Things (IoT) system. In order for this to be set up by others later to perform reproducible tests, a detailed description of components, sensors and protocols used is needed. This chapter will go through the background information of the devices and technologies used, and why these where chosen over other alternatives.

2.1 Bluetooth Low Energy

BLE is a wireless technology for short range communication developed by the Bluetooth Special Interest Group [3]. The idea was to create a low energy single-hop network solution for Personal Area Networks (PANs). A major advantage of this is that Bluetooth 4.0 is already a well established technology in cell phones, laptops and several other devices, and BLE can use several similarities with this. The 6LoWPAN Working Group has recognized the importance if BLE in IoT [4].

The protocol stack of BLE has two main parts, the controller and the host. In the system used in this thesis this represents the Raspberry Pi as the controller (master) and Nordic nRF52 as the host (slave). The communication between these is done through the standard Host Controller Interface (HCI). All slaves are in sleep mode by default, and are woken up by the master when communication is needed. Links are being identified by a randomly generated 32-bit code.

In the case of the network used in this thesis, when the BLE slave has been connected to a master, it stops searching for other masters, and it is not possible to connect to several masters. This means that we are only able to create a *star network*, not a *mesh network*. This could possibly be an idea for improvement later. Other than this, BLE seems like a very good alternative in this project.

2.2 6LoWPAN

6LoWPAN is a defined protocol for using Internet Protocol version 6 (IPv6) in low enerdy networks, to identify sensors and devices, as defined in the IEEE 802.15.4.

To use the Internet Protocol was proposed by Geoff Mulligan and the 6LoWPAN Working Group in [5]. In this paper the advantage of 6LoWPAN is explained as the easiest way to use standard protocols such as User Datagram Protocol (UDP), Transmission Control Protocol (TCP), Internet Control Message Protocol (ICMP), glsdns and Trivial File Transfer Protocol (TFTP), since they can be used directly without the requirement of a translation mechanism. They are already adapted to run over IPv6.

In addition to this, the paper argues that 6LoWPAN was developed to be used in small sensor networks. Implementations can fit into 32Kb flash memory parts, wich is smaller than Zigbee and Zensys (which is the two main comparisons made in the paper). It also uses an impressive header comparison mechanism that allows the transmission of IPv6 packets in 4 bytes, much less than the standard 40 bytes. This is done by using stacked headers, same as in the IPv6 model, rather than defining a specific header as for Internet Protocol version 4 (IPv4). This means that a device can send only the required part of the stack header, and does not need to include header fields for networking and fragmentation [4].

6LoWPAN was therefore a natural tool to use in this project, in the IPv6 and BLE based network.

2.3 Raspberry Pi

Developed by Element 14, the Raspberry Pi has become a central tool for many people wanting to get started using microcomputers. This was therefore a natural starting point for us as well.

The Raspberry Pi 2 model B+ is the second generation of its kind, and its *ARMv7 processor* has approximately six times the performance of the predecessor [2]. With a USB Bluetooth dongle connected, it is quite simple to enable both 6LoWPAN and BLE, given that the right Unix kernel has been used in the Operating System (OS) of the Pi.

Later in this thesis, the processing power of the Pi compared to performing calculations in the end-points will be a central topic for discussion.

2.3.1 Ubuntu Mate

Along with the Raspberry Pi, we needed a good and stable operating system with a kernel that supported the 6LoWPAN architecture. For this, Ubuntu Mate version 15.10 with kernel version 4.15 was chosen, and used on the Raspberry Pi. As other versions of Ubuntu this is Unix based, and has a complete Graphical User Interface (GUI) of a full OS.

2.4 nRF52

The nRF52 is developed by Nordic Semiconductor, and is being described as a family of highly flexible, multi-protocol system on chip.

[1]

Running at 64MHz it racks up impressive stats: EEMBC Coremark® score of 215 and 58 Coremark®/mA. Built in a cutting edge 55nm process technology the nRF52 Series is architected for todays need for speed, speed to carry out increasingly complex tasks in the shortest possible time and return to sleep, conserving precious battery power. Introducing a Cortex-M4F processor at its heart, it is the most capable Bluetooth Smart SoC on the market today. With a brand new multiprotocol radio architecture limits are pushed even further for Bluetooth Smart: A total link budget of 100dBm, -96dBm sensitivity, +4dBm output power and -42dBm selectivity, it is built to exist reliably and effectively in a busy 2.4GHz band. Couple this RF resilience with outstanding power consumption: 5.3mA at 0dBm TX output power and 5.4mA RX it is miserly with power when getting things done on-air.

Designed with wealth of digital and analog interfaces and peripherals, there's something to cover every design requirement. The demands of today's Bluetooth smart single chip designs are increasingly varied and complex, they can range from digital audio processing, to FFT/FIR and security algorithms, to driving device displays and UIs. Whatever the task, nRF52 series has it covered.

We've truly made low power easy. We haven't just just great datasheet numbers, but an automatic and adaptive power management system which combined with extensive EasyDMA and PPI makes the nRF52 Series positively frugal with power.

nRF52 Series brings NFC for the first time to a Bluetooth Smart SoC. The on-chip NFC-A tag allows developers to take advantage of NFC 'touch-to-pair' functionality in their designs.

Maintaining the total flexibility philosophy of the nRF51 Series, it is a flash device. With 512kB flash + 64kB RAM on chip, developers keep all the software

stack flexibility and Over-The-Air Device Firmware Upgrade (OTA-DFU) features they enjoyed with the nRF51 Series.

We understand that in the world of wearables, size really is everything, so we built a device that is the smallest Bluetooth Smart SoC to date measuring as small as 3.0 x 3.2mm (CSP). With an on-chip balun and a minimum of external components it has the smallest design footprint out there.

Tomorrow's Bluetooth Smart Solutions are going to ask a lot. The nRF52 Series delivers.

2.5 Adafruit ADXL345 Accelerometer

In order to do collect data, a sensor needed to be connected to the Nordic Semiconductor nRF52. The main thought behind the thesis was to measure vibrations, and a good accelerometer was needed. The Adafruit ADXL345 accelerometer was chosen for several reasons.

- It is the same accelerometer built in on the Zolertia Z1 microcontroller
- It can measure acceleration in the axis
- It sends digital data directly
- It supports both

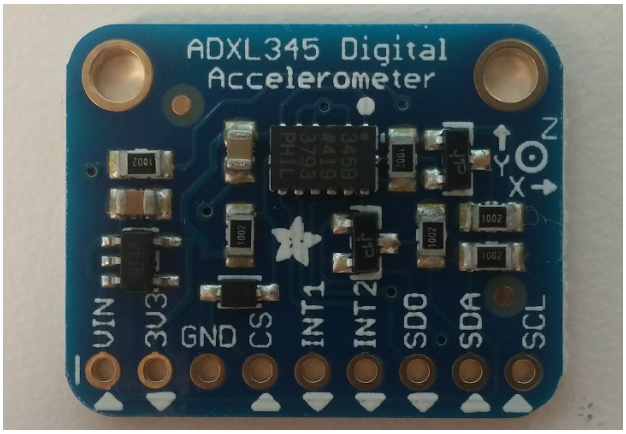

Figure 2.1: ADXL345 Accelerometer

2.6 Transport protocols

2.6.1 CoAP

CoAP is a transport protocol designed to be used in constrained networks for Machine-to-machine (M2M) communication, and works good in low-power and lossy networks. It can be used with microcontrollers, and with IPv6 and 6LoWPAN. Both GET and PUSH functionality can be used, as well as *observable* GET. This means that a server can "subscribe" to end notes in the network, and get updates either after a given timespan or when changes have been made. This therefore looked like a promising protocol to use, and was chosen as the first transport protocol to test the network.

2.6.2 AioCoAP

2.6.3 MQTT

2.7 Software tools

As Integrated Development Environment (IDE), the *KEIL Vision* was used, as recommended by Nordic Semiconductor (where?), for writing C programming. For other programming languages (for instance Python 3.1) *Sublime Text 2* for Windows and Linux was used, as well as *Pluma* for Ubuntu Mate on the Raspberry Pi.

Chapter 3
Architecture

This chapter will describe in detail how the different components of the system has been connected together, and how the different protocols has been configured to read and transfer data efficiently.

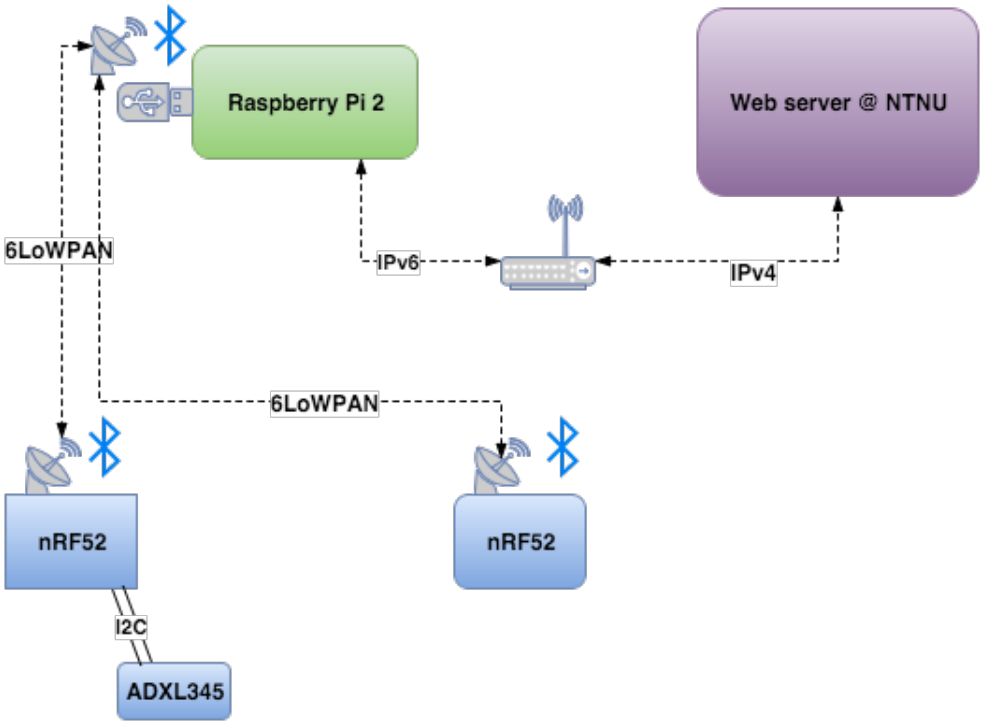

Figure 3.1: System architecture

Figure 3.1 shows how the complete End-to-End system of this thesis is set up. In short terms, the *ADXL345 Accelerometer* is connected to a *Nordic Semiconductor nRF52* using the I2C interface. This requires four cables (noted from nRF52 to ADXL345):

- 5V – VIN
- GND – GND
- P0.27 – SDA
- P0.26 – SCL

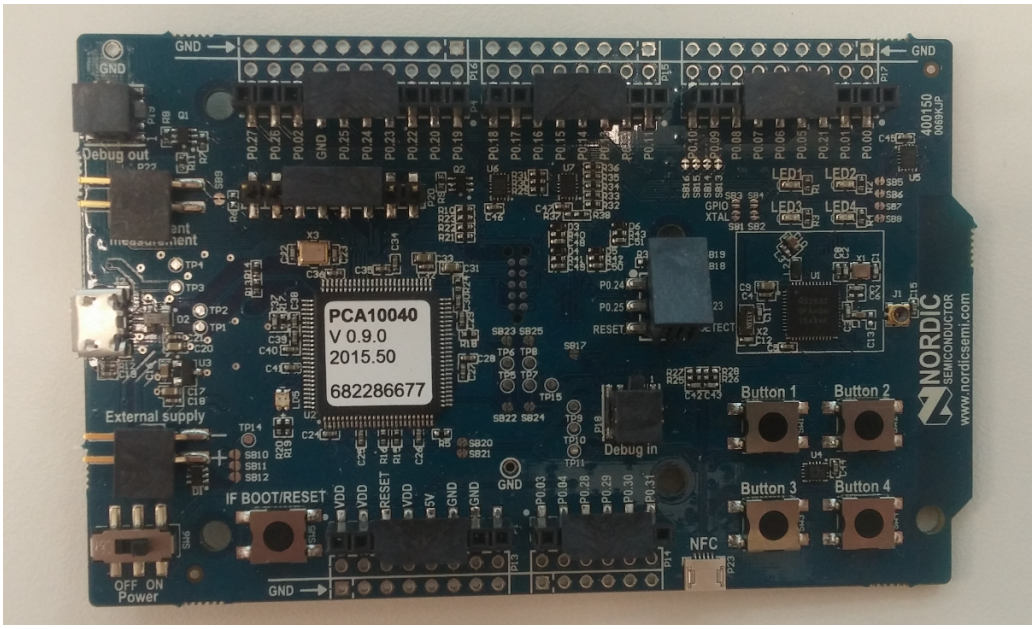

Figure 3.2: Nordic Semiconductor NRF52

3.1 Enabling 6LoWPAN and BLE

To enable these things.

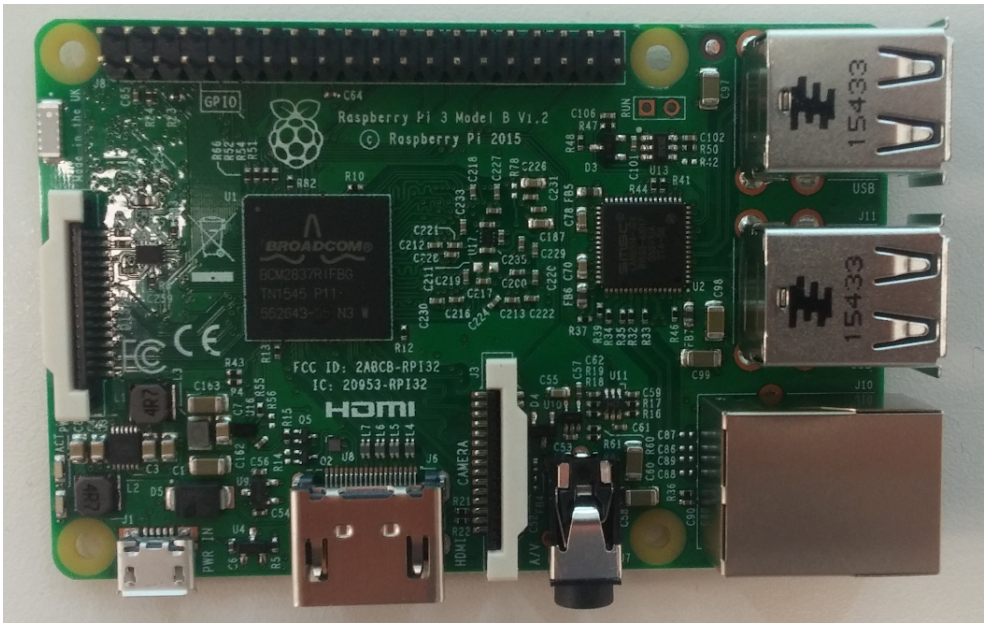

Figure 3.3: Raspberry Pi 3

3.2 Connecting Raspberry Pi and nRF52

3.3 Connecting nRF52 and ADXL345

3.3.1 I2C

3.3.2 Problems

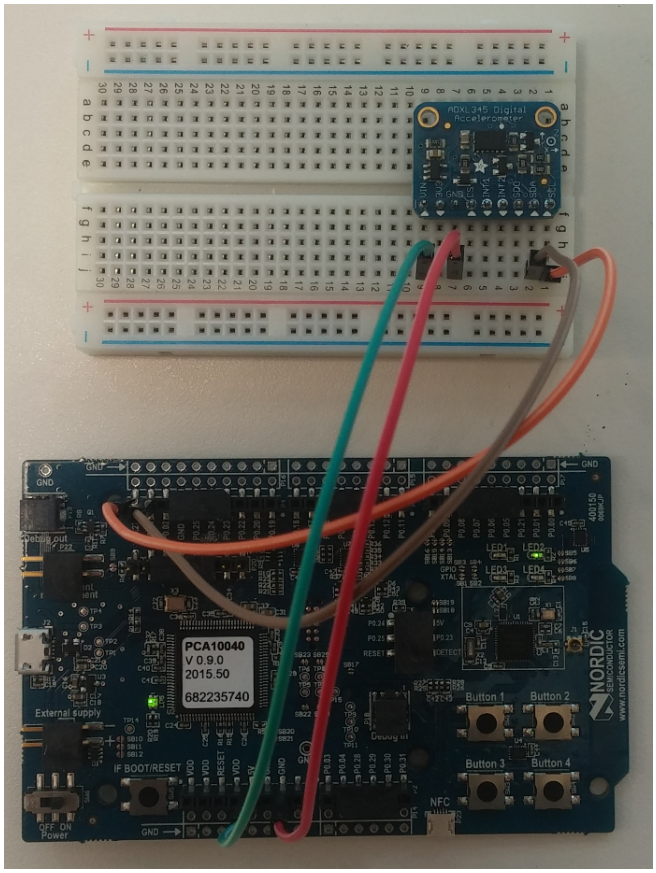

Figure 3.4: Connected nRF52 – ADXL345

Chapter 4
Conclusion

Metaphore: Electric car

References

[1] Nordic Semiconductor: nrf52 series soc. https://www.nordicsemi.com/Products/nRF52-Series-SoC. Accessed: 25-03-2015.

[2] Raspberry Pi: raspberry pi 2 model b. https://www.raspberrypi.org/products/raspberry-pi-2-model-b/. Accessed: 27-03-2015.

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

[4] Hui, J. W. and D. E. Culler (2008). Extending ip to low-power, wireless personal area networks. *Internet Computing, IEEE 12*(4), 37–45.

[5] Mulligan, G. (2007). The 6lowpan architecture. In *Proceedings of the 4th workshop on Embedded networked sensors*, pp. 78–82. ACM.

[6] Shelby, Z., K. Hartke, and C. Bormann (2014). The constrained application protocol (coap).