Aalto University
School of Electrical
Engineering

AALTO UNIVERSITY
School of Electrical Engineering
Department of Communications and Networking

Sriharsha Kuchimanchi

Bluetooth Low Energy Based Ticketing Systems

Master's Thesis submitted in partial fulfillment of the degree of Master of Science in Technology

Espoo, February 2015

Supervisor: Prof. Riku Jäntti, Aalto University, Finland
Instructor: Shkumbin Hamiti, Nokia Oy, Finland

AALTO UNIVERSITY **ABSTRACT OF THE MASTER'S THESIS**

Author:	Sriharsha Kuchimanchi
Title of the Thesis:	Bluetooth low energy based ticketing systems
Date:	25th February 2015
Department:	Department of Communications and Networking
Professorship:	Radio Communications
Supervisor:	Prof. Riku Jäntti, Aalto University, Finland
Instructor:	Shkumbin Hamiti, Nokia Oy, Finland

This thesis proposes a Bluetooth Low Energy (BLE) based payment solution for public transportation. The thesis first reviews some of the 'Mobile payment solutions´. Traditionally, these services have revolved around technologies like Barcodes, Quick Response (QR) codes, Short Messaging Services (SMS) etc. However, with the advent of Smartphone's equipped with Bluetooth controllers and security chips, a number of innovative payment services are being studied.

Furthermore, this thesis also presents the necessary background to understand the BLE technology. Based on BLE technology, BLE based ticketing protocols are presented. An emphasis on Public transport agency is made. To accomplish the task of designing these protocols, Generic Attribute Profile (GATT) system is extensively studied. GATT defines the concepts of services and profiles into the BLE architecture. Profiles like alert notification, proximity and blood pressure are already defined in the BLE specification. Any new profiles created will be placed on top of GATT and are known as Custom Profiles.

There are Bluetooth controllers manufactured by Nordic semiconductor which is used as reference in designing these protocols. However complete implementation of the system is not performed in this work. A model with necessary protocols is only presented here. This work can also be implemented on various Bluetooth manufacturing controllers.

Keywords: *Bluetooth Low Energy, Ticketing systems, Public Transport System, Mobile Payments*

Language: *English*

Acknowledgements

Riku Jäntti, Head of department (Radio Communication), Espoo has kindly consented me to undergo my master thesis in Nokia, Finland.

Shkumbin Hamiti, was nominated as the guide for the study & training in Nokia on "*Bluetooth Low energy based ticketing system*".

I am highly grateful and appreciate excellent guidance given by Shkumbin Hamiti and also thank our H.O.D Radio communications for his kind consent.

In addition to the above I would like to express my heartfelt gratitude and thanks to other innumerable staff colleagues and friends of Nokia corporation and Aalto University who have helped me in appreciating the role of industry institute interaction using this thesis work (in Alphabetical order)

Arto Palin

Jukka Reunamäki

Juha Salokannel

Kanji Kerai

Markus Isomäki

Otaniemi, Espoo: 25.02.2015

Sriharsha Kuchimanchi

To Nana & Amma

Contents

LIST OF ABBREVATIONS

AES- Advanced Encryption Standard

AMP- Alternative MAC/PHY

APDU- Application Protocol Data Unity

API- Application Programming Interface

BCD- Binary Coded Decimal

BLE- Bluetooth Low Energy

BR- Basic Rate

CA- Certificate Authority

CRC- Cyclic Redundancy Check

EDR- Enhanced Data Rate

GAP- Generic Access Profile

GATT- Generic Attribute Profile

GFSK- Gaussian frequency shift keying

GPIO- General Purpose Input/output

HCI- Host Controller Interface

HS- High Speed

HSL- Helsingin seudun liikenne

ICT- Information and Communication Technology

ID- Identification

IMEI- International Mobile Equipment Identity

IOT- Internet of Things

ISO- International Standards Organization

ISM- Industry, Scientific and Medical

IEC- International Electro technical Commission

L2CAP- Logical Link Control and Adaptation Protocol

LSB- Least Significant Bit

LL- Link Layer

MAP- Message Access Profile

MSB- Most significant Bit

Mbps- Megabit per second

MIC- Message Integrity Check

NFC- Near Field Communication

PAN- Personal Area Network

PC- Personal Computer

PHY- Physical Layer

PKSCS- Public key Cryptography standards

POS – Point of Sale

PTO- Public Telephone Operator

QR- Quick Response

RF- Radio Frequency

RFID- Radio Frequency Identification

RSA- Ron Rivest, Adi Shamir, Leonard Adleman

RSSI- Receive Signal Strength Indicator

SIG- Special Interest Group

SMS- Short Messaging Services

SoC- Silicon on Chip

SPP- Serial Port Profile

TEE- Trusted Execution Environment

UUID- Universally Unique Identifier

VD- Validation Device

List of Figures

TABLES

Chapter 1

Introduction

The field of Information and communication Technology is growing at a rapid rate. People around the world are adopting various technologies and shunning away yesteryear technologies. People are going the extra mile in getting their work done as efficient as possible. This is leading to latest technologies and their development. In the modern days, mobile phone has become a basic need for mankind. It has the power to reach millions of people on the go. These mobile phones and several other gadgets are conquering the present day markets. They are all getting inter connected through various sources which make the Internet of Things (IOT). With the demand various multinational companies in ICT are setting up business and striving for more innovative products. The Technology business is just at the beginning of this whole new journey and yes, it means change. But that means new challenges, new competencies and a chance for everyone to individually impact how it shapes the future programmable world. According to the demand and the needs, organizations are also undergoing transitions and growing industrial corporate organizations need to master these three different technologies.

- **Mastering of design technologies**: Achieving product development and value additions.
- **Mastering of manufacturing technologies**: Reductions in costs, Enhancement of productivity and quality improvement.
- **Mastering of creative technologies**: Innovation, emerging and new products and services.

Once the organization starts focusing its attention and involvement, the organization can not only meet the customer requirements but also can create customer needs. There is plenty of scope for organizational excellence through the improvement of

- Process technology and development.
- Product development.
- Enhancement of capacity utilizations.
- And Employee Involvement.

Nokia corporation earned its pride of place amongst other organizations by its sheer vision, strategic formulation cross fertilizations of design technologies and selective introduction of product mix

through customer orientation and introduction of higher value added products and services. Now this company mainly aims to be a technological company with a vision of the programmable world.

Programmable world according to my own experience can be interpreted as follows, "Everything in this world is getting digitalized, digital data is enormous and getting generated from various sources like mobile, internet, sensors etc. Each of these worldly things is getting the required intelligence to get it connected to every other thing. To make this happen one has to program to make it run. This is the programmable world!

Oldest mobile payment solutions included cellular payments and more recently contactless payments have evolved. In cellular payments, the customers can complete their purchases using mobile communication techniques like Short Messaging Service (SMS) [1]. The user was supposed to contact the point of sale (POS) or the terminal which had the necessary hardware to complete the transaction. This POS used technologies like GSM to complete the transaction. The drawbacks of this technology were that SMS uses GSM network infrastructure. The SMS has a limitation that it is restricted to 160 characters long and it occupies the GSM frequencies which is capacity overloaded for voice and data. Because of this congestion in the network [2], there are chances that the SMS doesn't reach the payee on time or gets delayed. This technique was also not considered to be safe since the payee gets access to confidential personal information. To minimize this problem studies were done on secured SMS transaction. A system which allows users to transmit confidential information to a mobile banking system was investigated [3]. This technique made use of encryption techniques at the mobile end. The SMS once received at the payee ends decrypts it. This study focused on symmetric key AES algorithm. This study though enhanced the security of the SMS payment but couldn't solve the problem of delays in SMS delivery times.

Studies quickly shifted to short range communications. Short range communications enable a user to remain in close vicinity of the POS. this technique also allows a two way mutual authentication to be approved. Technologies like Barcodes, QR codes and NFC were used in many services. In case of QR codes [4] items which had price tags were embedded with QR codes. These codes were considered to be bills, which are supposed to be scanned by a Mobile camera. Companies like Digi cash [3] allows customers to pay for their goods using mobile phones, where a Quick Response (QR) code is scanned. This code has a digital signature, a user having access to this information was directed to merchant's site and had to complete his transactions online. However this solution had problems like long processing times, poor mobile web pages for the merchants. The Barcodes also

functioned similarly and consumers had no advantage to use these systems. There was a feeling that it is as safe as paying money directly to the merchant rather than not using any mobile phone. A similar study has been done by Bhaktiari M.G and Sahajari. M [6], where the authors widely discuss on the architecture of stacked 2D bar codes and Matrix 2D bar codes. They present methods of transaction using a hash chain model. But it doesn't really solve the core issues of Mobile payment.

Radio Frequency Identification systems (RFID) also falls under the contactless payment. This technology is been in use over a long period of time. The definition of RFID is interchangeably used with smart cards. RFID are typically used in applications associated with tracking and tracing, supply chain applications etc. On the other hand smart cards are used in payments, access controls in a room/building. The main differentiation between these two is the operating frequencies of these technologies. RFID is used at 125/135 KHz, 13.56 MHz, 902-928 MHz whereas smart cards are used only at 13.56 MHz. Due to the similar operating frequencies the terms are interchangeably used. Another key difference between the two is the reading distances. RFID can typically read up to 10 meters where as smart cards can read up to 10 cm only.

A company in Canada [3] has already started piloting in Bluetooth payments. They use hardware from Estimote. This is a beacon which transmits context and micro location data from a reader to the nearest Bluetooth terminal. The customer can simply tap on the item which he/she wishes to buy or pay for. And then the transaction is done. Similar services by Digi Cash are also provided in the country of Luxemburg. Globe Sherpa [5] is another company in the USA, which is also rolling out services in the area of payment solutions. They launched a ticketing app which is being collaborated with the Virginian railway express in USA.

Near field communication technology (NFC) is also being widely tested in the payment solutions. In a study [7] conducted in South Korea, operating principles using NFC have been presented. The authors deal with the peculiar problem of power variations both at the transmitter and the receiver. NFC technology is also leveraged by Visa, Master cards and also by communication hand set manufacturers like Microsoft, Samsung. These players are introducing a secure element in to their chips, which enables a secure transaction.

1.1 Research Objectives

This thesis focuses on Bluetooth low energy technology as a mobile payment solution. Bluetooth low energy ticketing protocols are presented. Thus we define the following goals for this research work:

a) Designing the Bluetooth low energy ticketing architecture
b) A model of Bluetooth Ticketing profile is presented
c) Ticketing protocols using BLE are suggested

1.2 Scope of Thesis

The mobile payment solution based on BLE is presented in this thesis. The ticketing protocols are developed from scratch. The model which is presented here is more specifically applicable to Public transport systems. An implementation of the same is however not performed. But the suggested ticketing model can be implemented on Nordic semi-conductor evaluation kit. Nordic semiconductors produce necessary hardware to implement the BLE protocols. So, the ticketing protocols are developed using the technical support of Nordic developer zone. It is also worth mentioning that the BLE protocols suggested here can also be implemented on Texas instrument kits or any other BLE controller manufacturer.

1.3 Structure of Thesis

This thesis is structured as follows: Chapter 2 introduces the existing mobile payment solutions. A more focus is put on RFID and NFC based mobile payment solutions. Chapter 3 briefly introduces the basic concepts, usability's, architecture and applications of Bluetooth low energy (BLE). Chapter 4 introduces the Nordic Semiconductor evaluation kit which is the BLE controller. This chapter focuses on the hardware capability of BLE and the BLE link. This part is used as a reference in the next chapter. Chapter 5 introduces the Ticketing profile that makes use of the GATT layer in the BLE. Chapter 6 concludes this thesis and also discusses the future scope for this work.

Chapter 2

Mobile payment solutions

This chapter briefly describes the techniques and theories that were studied as part of this Thesis. Firstly, various mobile payment solutions are described. Special emphasis is laid on RFID and NFC based ticketing systems. This is followed by a brief description of the challenges faced by the above mentioned ticketing systems.

Mobile payment solutions have existed since a long times. Several different mobile payment solutions are already being tested and used. In a study conducted by Visa [1] it is observed that, there are around 540,000 contact terminals in Europe. These terminals allow users to complete a hassle free payment. This study also opines that, the same customers are coming back to enjoy their services, which they feel is quite good. More and more people are willing to use their mobile phones, contact less cards as a means to pay for their ticket costs. Several payment solutions already exist in the market. The most popular include apple pay, PayPal, Google wallet to name a few.

The present-day Smartphone is equipped with several features that can be utilized in paying bills. These include, camera, NFC, Bluetooth. Camera has been extensively used in scanning barcodes, QR codes [4]. However they pose certain security challenges which the near field technologies like NFC and RFID take care of. These challenges lead to significant decrease in consumers, who used their cameras in smart phones to pay their bills. But thanks to technologies like NFC, RFID and Bluetooth that payment solution have significantly improved their security features. These technologies provide hassle free contactless payments, which are fast, less time consuming and trust worthy to use. The next section focuses on RFID and NFC based payment solutions and highlights some of their features. These typically fall under the Near field communication technologies.

2.1 Existing Ticketing Solutions

Gone are the days when a Bus conductor or a Driver would give a paper ticket to the passengers. People are constantly looking for innovations. They want to travel light, carry less paper and just use mobile to get a ticket for a journey. In pursue for this goal public transport agencies are constantly looking for innovative ideas in the field of payment solutions. With the advent of mobiles and smart phones and their penetrations into the electronic consumer market, a lot of innovations are taking place in the field of mobile payments. Ticketing payment solutions can be broadly divided into two types: Gated systems and Non gated systems.

2.1.1 Gated systems

The idea of gated system is that, customers have a Smart card. This smart card is issued by the local public transport agency over a counter after payment for the same is done. The smart card is typically loaded with money or value, depending on the customers' requirement. Every POS has a device which is a RFID reader. The customer willing to buy a ticket needs to swipe the reader. Swiping in a process where the customer places the smart card in front of the reader, the typical distances range from 5-10 cm. After this the user's ticket will be validated appropriately and the customer is charged for his journey from the balance in his smart card. If the transaction is successful the gates open up. This is known as the gated systems. An unsuccessful transaction will lead to closure or non-opening of the gates. This is either displayed as a message on the readers screen and/or with blinking of Red light with warning sounds. These systems are already in place in cities like London, Istanbul, Hong Kong and Delhi and almost in major international city centers. They are widely used in Underground railway stations, metros.

2.1.2 Non Gated Systems

On the contrary there are systems where the onus is on the passenger to use his smart card, as a valid ticket for his journey in the absence of a gate. The honesty of passenger is trusted and there are sporadic checks during sporadic times to ensure that the passenger is travelling with a valid ticket. Helsinki city transport currently uses these non-gated systems.

2.2 Contactless Smart Card Ticketing system

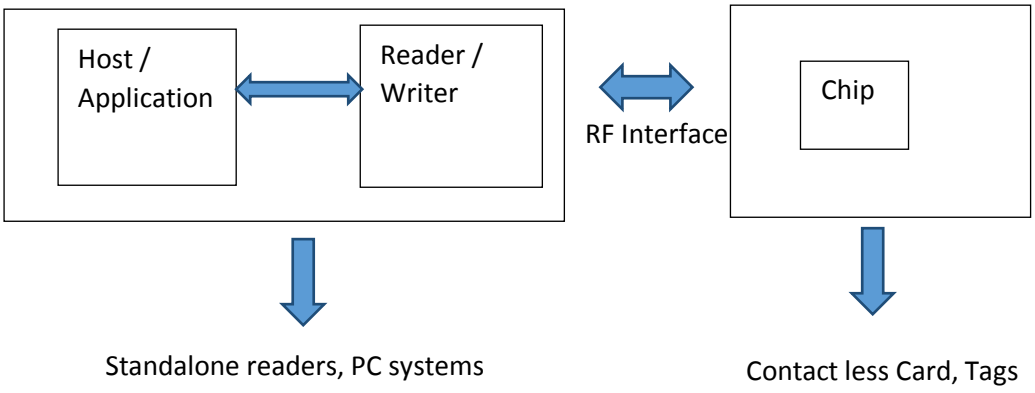

Figure 1: Architecture of contactless Smart card system

Contactless smart card systems consist of a tag or a smart coin. This acts as a transponder. It consists of a chip with a coil in it. This has an antenna which communicates with the readers on the other side. The readers typically consist of a Host/Application embedded onto the reader which has an antenna. The communication between the two entities takes place over the RF interface. Data transfer is typically conducted at 13.56 MHz at a data rate of 106KBps. However these systems are susceptible to security risks like skimming and eavesdropping [10, 11]. Eavesdropping typically has been reported at distances of up to 2 meters but quite many works have been performed in these areas. One study [12] suggested a ticketing model where two tags were placed in the bus. These tags communicated with the reader present outside the bus. A complete detection algorithm was presented to suggest the bus timings to the passengers who are waiting in the next stops. However these systems where enhanced by user authentication systems which used passwords, a card or using a biological trait. Though the security was enhanced, time was a factor which was compromised. Users have to typically either enter their pass codes, for which they must remember or store it a safe place. Or else the users where supposed to get their finger print scanned across the terminal. All these additions though enhanced the security but could not reduce the typical purchasing times. Long queues remained common.

NFC technology on the other hand provided some benefits to the above mentioned problems. In the following section NFC based ticketing systems will be discussed.

2.3 NFC based Ticketing systems

NFC is a wireless radio frequnecy identification technology that is standardised in ISO/IEC 18092 [15] and ISO/IEC 21481 [16]. There is an industry consortium called as NFC forum [17] which decides the standards for this technology. Though this technology is susceptible to evasedropping nad man in the middle attacks various mobile phone manufacturers have embraced this technology. A secure element known as Tusted execution environment (TEE) is uslly embeded in these mobile phones. This enhances the security features of the smart phones.

The architecture of the NFC ticketing systems [18] serves both the gated and non gated terminals. In its architecture it typically consists of a reader which verifies the identity of the passenger. This infrastructure is assumed to be set up by the transport authority system. At the back end user accounts are linked to each of the users. This is used for auditing and fare calculation purposes.

One of the works carried out in NFC [18] provides a protocol framework for combining gated and non-gated ticketing systems into one coherent system. This research though is not fully

implemneted. It has proposed non-gated protocols that leverage the TEE present in mobile phones. This work makes use of signatures and certificates. The certificates are modified versions of X.509 certificates. A extra part of key is padded on to the already existing X.509 certificate. With this additional padding the tickeing system treats the Certificate authority Public key as secret. This public key is decrypted at the back end during a ticketing transation. So the system has a ticket issuer which is known as a certificate authority (CA). This CA has a pair of public/private key, which is correctly known by all others. There is a client and this client could be a ticketing app in a NFC enabled phone. There are the public transport agencies who act as verifiers and they check the public keys of the client. The proposed systems [18] are not implemneted as an entity but are implemneted as parts. Apple pay also uses this technique to certain extent.

2.4 Short comings of the above systems

To a certain extent NFC technologies are found to be more secured compared to other short range communiations. Other than secuirty, typically the problems of these ticketing systems include processing times in billing and the ease with which the user can handle an app. Bar code systems, QR code systems typically take up more time in processing a bill and are inefficient. NFC takes lesser time in transacting but however every time a transaction is to be done than the smart phone must be activated and has to be placed on the readers.

2.5 Requirements for a Ticketing system

Based on previous existing ticketing systems, it is identified that that the emerging ticketing systems must have certain features. These are:

1) A system which is robust to all kinds of digital thefts
2) A system which is easy to use, scalable and efficient enough.
3) A system which can be implemenetd on the existing infrastruatcure without incurring great costs.
4) A system which enables users to conveniently and swiftly purchase their tickets without causing any delays through longer processing times.

To investigate these issues, this thesis makes an attempt to explore into long range communication i.e Bluetooth Technology. With the current existing standard, Bluetooth has significant advantages over its contemporary near field technologies and other systems which are already existing. Being a long range comunication, it is possible to communicate with the

readers through longer reading distances. The next chapter introduces the Bluetooth techology in detail and focusses on the theory of Bluetooth low energy which is Bluetooth 4.0.

Chapter 3

Bluetooth Technology

This chapter describes the evolution of Bluetooth technology leading to the low energy Bluetooth technology that will be used in this thesis. This has been divided into two sections. Section 3.1 discusses Bluetooth technology and Section 3.2 discusses the current state of art version Bluetooth low energy.

3.1 Bluetooth

Bluetooth is a standard for Personal area networks (PAN) which was developed by Ericsson research group in 1994. This technology is considered to be a short range and low power technology. This operates in the Industry Security Medical (ISM) frequency band of 2.4GHz.Bluetooth has been adopted by most of the Information and Communication Technology industry since its acceptance by the Bluetooth special interest group (SIG) in 1998. The SIG board members include mobile manufacturing giants (e.g., Apple, Microsoft and Motorola) and silicon chip manufacturers (e.g., Intel, Nordic semi-conductor). The term Bluetooth has evolved since 2000. First standard was Basic rate which focused on short range networks like Personal Area Networks. It typically had ranges from 10m to 100m. It was using frequency hopping spread spectrum techniques. Data rates of 1Mbps were achieved. The next standard was introduced as Enhanced data rate. This updated standard offered higher data rates of 2-3 Mbps. In 2008, the High Speed was introduced, which offered data rates up to 24 Mbps. The architecture of Bluetooth is shown in the figure 2. This architecture will be explained in the next section.

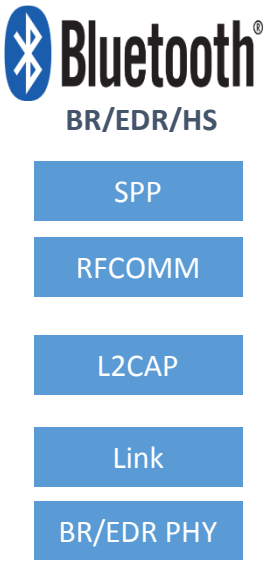

Figure 2: Architecture of Bluetooth

However ICT industry also felt a need for low energy version which would facilitate short distance and low power networks. Also advances in battery technology imposed challenges on these earlier versions of Bluetooth. There was a need for advanced version of Bluetooth which would be used in accessories that uses less battery and required less charging which lead to the latest additions in Bluetooth, known as Bluetooth low energy (BLE). This current version of Bluetooth i.e. Bluetooth 4.0 is employed in this thesis. A detailed description of this technology will be discussed in the next section.

3.2 Key Features of the Bluetooth Low Energy

Bluetooth low energy is the newly designed and a complementary technology to the classic Bluetooth. It is the current lowest possible power wireless technology. This technology borrows its name from its parent which had a basic rate of 1 megabit per second (Mbps) and was known as Basic rate (BR). Enhanced data rate (EDR) was version 2 which had a data rates to 3Mbps. Version 3 which is known as Alternate MAC PHY (AMP) delivered data rates up to hundreds of megabits per second. However, BLE provides lesser data rate compared to AMP but instead optimizes for ultra-low power consumption by virtue of its design which means that the Bluetooth connection can be maintained for a longer duration, say hours or days.

3.2.1 Bluetooth single mode and Bluetooth dual mode

Since Bluetooth devices came into existence in late 90´s. There are already several devices in the market which support versions 1, 2 and 3. These are called the Bluetooth classic only devices. They have architecture as shown in figure 1. Two new devices are also built which are known as dual mode and single mode devices. A single mode device is a Bluetooth device that supports just the BLE. Devices that support both BLE and the classic Bluetooth are called dual mode devices. Their architectures are as shown in the figure 3 respectively.

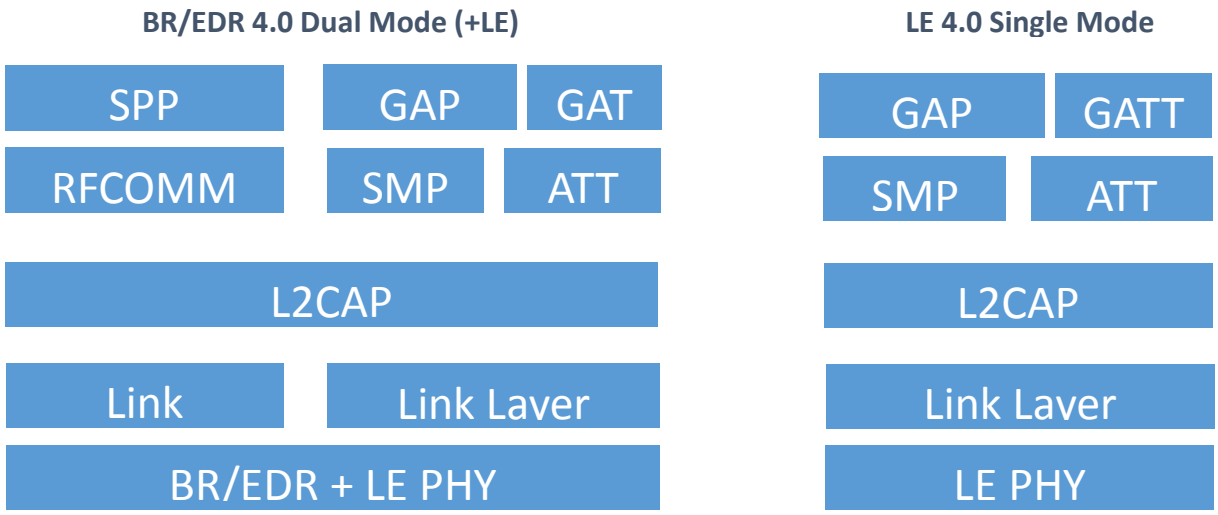

Figure 3: Dual Mode and Single Mode Architecture

In the market the dual mode and single mode devices are sold as Bluetooth smart ready and Bluetooth smart devices. Each of these modes has its own architecture as shown in figure 3. Since Dual mode devices support both classic and LE, these devices can talk with all the versions of Bluetooth. However, single mode also known as Bluetooth smart devices can only communicate with the Bluetooth smart ready also known as dual mode devices.

Figure 4: Bluetooth Smart ready and Bluetooth Smart Logos

Dual mode devices are new in the market and they require new hardware and firmware in the controller and software in the host. It is because of this reason that existing Bluetooth classic controller cannot be upgraded to support low energy. The single mode devices are highly optimized for low power consumption which is powered by button cell batteries. Since applications like public transport systems rely very much on low power and less battery consumption an attempt in made in this thesis to implement the ticketing protocols on the BLE. The following section will introduce the complete architecture of the Bluetooth low energy.

3.3 Bluetooth Low energy Architecture

The BLE architecture can be divided into three main parts -: a Controller, Host and Profiles. The controller is a Radio which has Physical layer (PHY), link layer (LL) and a Host controller Interface (HCI).

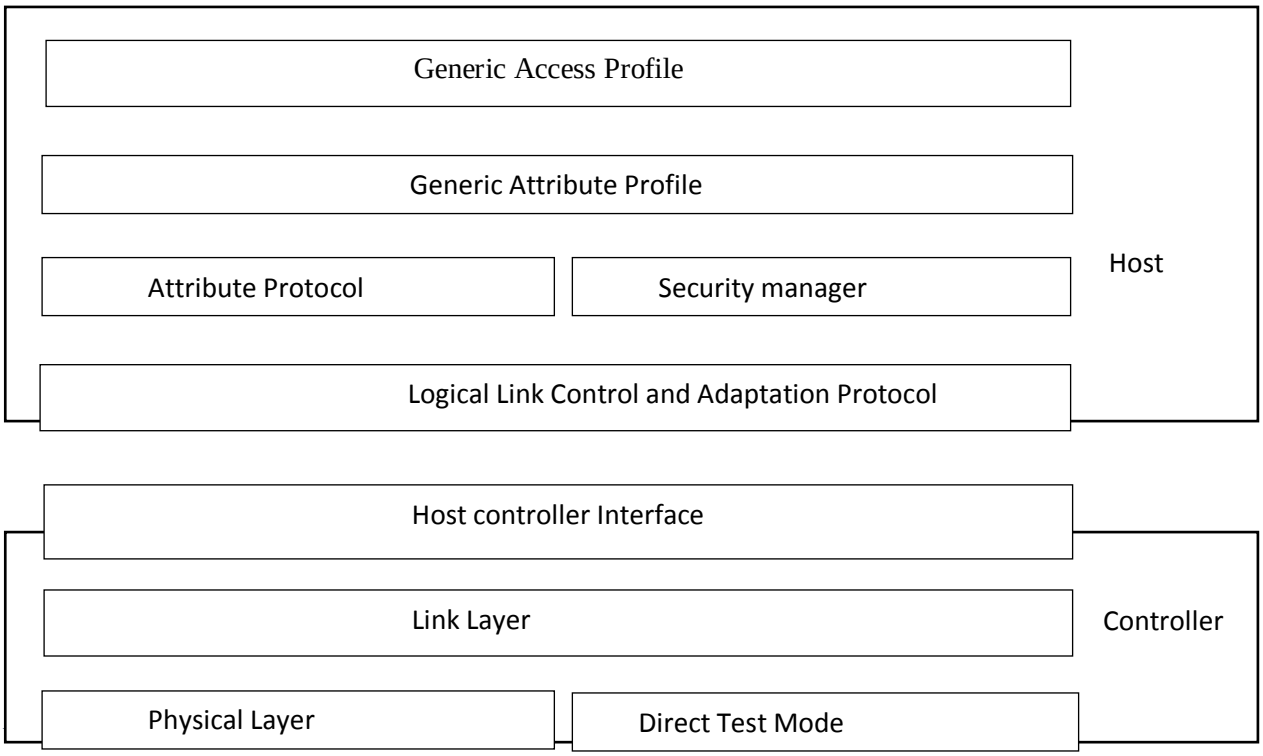

Figure 5: Architecture

Figure 5 represents the layered architecture. It is termed layered because it consists of so many layers which are placed on top of each other. Physical layer is the bottom most layer which receives and transmits bits of information. The link layer considers these bits as packets of data and it controls these packets and sends this data in various procedures and protocols. The Host controller interface (HCI) is the next layer which acts as an interface between the Controller and the Host. The logical link control and adaption protocol also known as L2CAP acts as a multiplexer to the number of channels which are present on top of the controller. The attribute protocol which is on top of L2CAP is the protocol which is used to access the data on a device. It helps in reading, writing and various other functions on the device. The generic access profile provides various services which are present in the device. It gives firsthand information of how things are organized on the device. It also consists of Meta attributes and various characteristics of the device which define the

organization. On top of these lie the applications. Various applications like battery profiles, temperature profiles, proximity, heart rate monitor etc. are defined and developed over this space.

3.3.1 The controller

The controller can physically be represented as a hardware which is a Bluetooth chip or radio. It consists of analog and digital parts which are embedded onto a silicon chip which support the transmission and reception of the data packets. Companies like Nordic semiconductor, Texas instruments manufacture the controller and sell them in the market with various commercial names. An example of the same is the Nordic semiconductor kit. It is to be understood that the physical layer is an nrf51822 Radio.

3.3.2 Physical layer

This layer is responsible for transmitting and receiving the data in the form of bits using the 2.4GHz radio. BLE uses Gaussian frequency shift keying (GFSK) which means ones and zeroes are coded onto the radio by slightly shifting the frequency up and down. Whenever there is an abrupt frequency shift, at that moment a pulse of energy spread's out over a wider range of frequencies. To enable to stop the energy spreading into these high and low end frequencies a Gaussian filter is used. This is called Gaussian because the transfer characteristics of this filter looks like a Gaussian curve. This also implies that low energy signal spreads out more than a standard Bluetooth classic radio signal since it doesn't use a tighter filter. Due to this reason the BLE is governed under the spread spectrum radio regulations as against frequency hopping mechanisms used by its parent technology.

So the modulation index used in the case of BLE is slightly higher than the classic Bluetooth which implies more number of channels to be used. In this case, the 2.4GHz band in case of BLE is split into 40 separate RF channels each of them are 2MHz apart from each other. In 40 RF channels three channels are fixed channels which are used for advertising data. The remaining 37 channels are used for transmitting application data and are dynamic in nature.

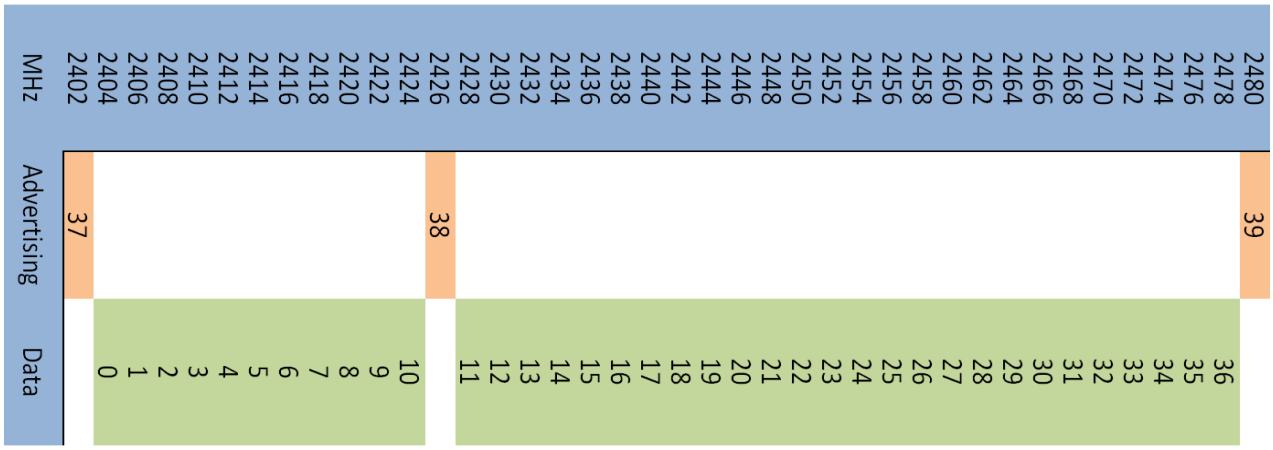

Figure 6: The mapping of Channel Index [21]

Figure 6 represents the ISM band of frequencies. In this band channel 37, channel 38 and channel 39 are the advertising channels and the remaining channels are the data channels used to send data. So when a device is adverting data it implies that the data is being sent over one of the 3 adverting channels. And the reason the advertising channels are placed in this manner is to avoid the Wi-Fi channels which operate on the other frequencies.

One of the main reasons that BLE uses 3 advertising channels is that it allows the devices to be discovered and connectable over a given period of time. It also makes the system more robust which also means it gives low power. Another reason being, if one device needs to be connected than one of the advertising channels is used and the device which is scanning discovers this advertisement than it takes around 1.3 ms to complete this connection. So with 3 channels it gives a very fast connection, which implies its duty cycle is ten to twenty times better than classic Bluetooth which also proves that BLE is more power efficient.

3.3.3 Link Layer

Link layer is the next layer which is on top of the physical layer and is below the L2CAP layer. This is more complex layer which ensures that the packets are structured so that the key functionalities like advertising, scanning and creation and maintenance of the connection is taken care of. To enable the link layer to perform these functionalities channels, packets and procedures are defined in the Bluetooth specifications. Various channels, packets and their structures will be explained in detail in the next chapters.

The best way to visualize the link layer is to understand it as a state machine as shown in figure 3. This state machine has five different states. *Standby state* is the first state where nothing is done. As

soon as a device is switched on, it is assumed that it is in standby state. It is possible to move into advertising, scanning or initiating states from this state. So this is in the Centre state and the most important and inactive state in the state machine.

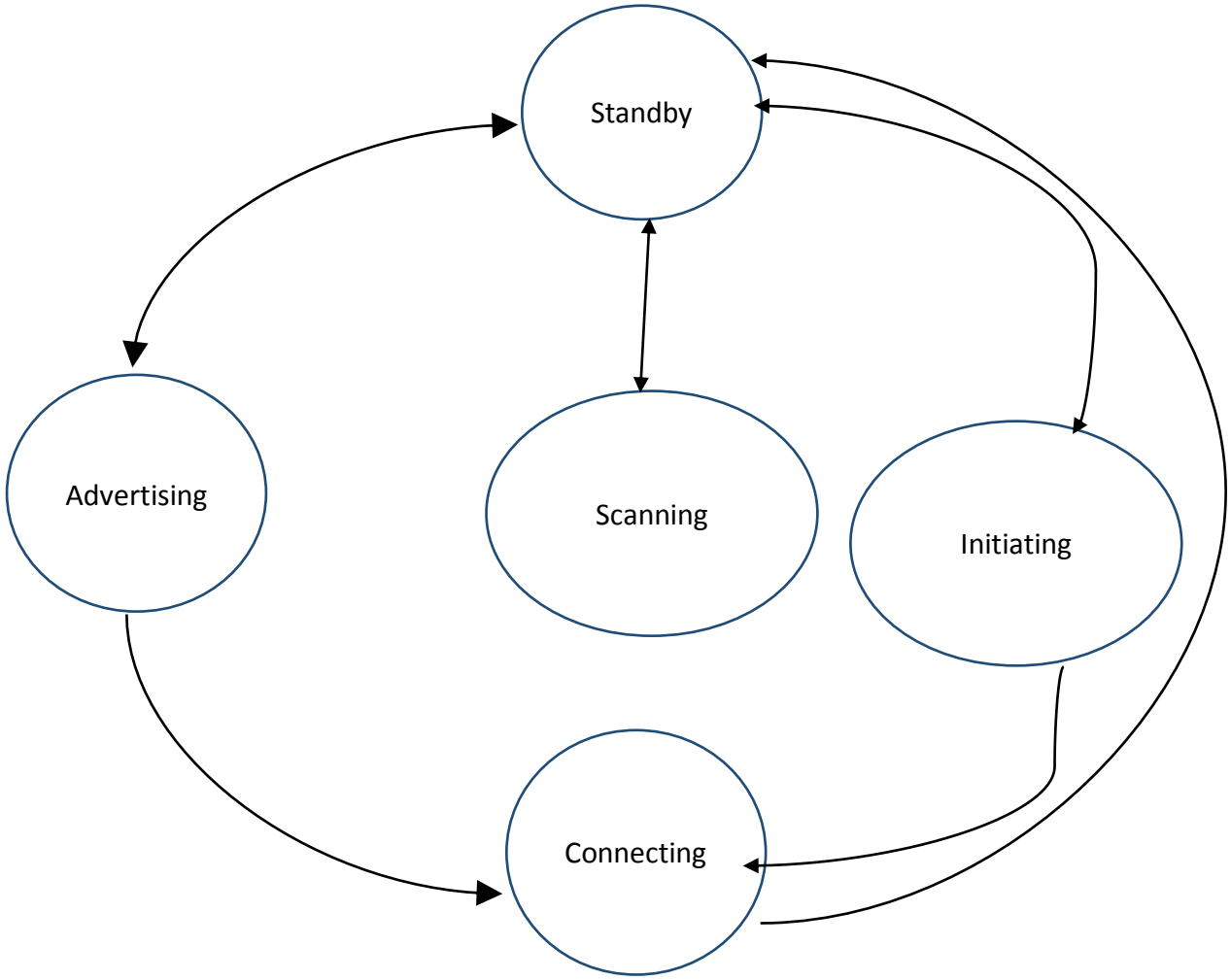

Figure 7: State diagram

Advertising state: From the stand by state it is possible to get into advertising state. By doing this the device transmits advertisement data packets. A clear packet structure will be defined and used in next chapters. If a device needs to be discovered or connectable, the device must get into this state. This state is also mandatory if the device has to broadcast some data. From this state it is also possible to respond to scan requests from devices which are actively scanning the device under test.

A device in scanning state will receive advertising data packets from the advertiser. Passive and active scanning are two types of scanning sub states. In passive scanning it is just possible to hear the advertisements. However in active scanning, scan requests can be sent to obtain additional scan

response data. As the state machine suggests, it is only possible to get back to the stand by state from scanning state.

To initiate a connection with any device, the state machine must be in the initiating state. In this case the device will listen to the initiators message. If an advertisement packet is received from this device than link layer will send a connect request to the advertiser. So the device gets connected. In case the connection is to be dropped than the initiator can stop initiating and get back to stand by state by just stopping a initiating a connection.

The final state of the link layer state machine is the connection state. This can be achieved either via advertising state or through the initiating state. It is performed under the initiator state than the device is said to take the role of master. Once the connection is established through the advertisement, it takes the role of slave. Master and slave are the two sub states in this connection state. Also the connection state is achieved by making use of the data channels. All other states make use of advertising channels.

To send data on any of the above mentioned channels through any of these states, it is done through packets. A packet is a small encapsulation of data that is sent from transmitter to receiver over a short period of time.

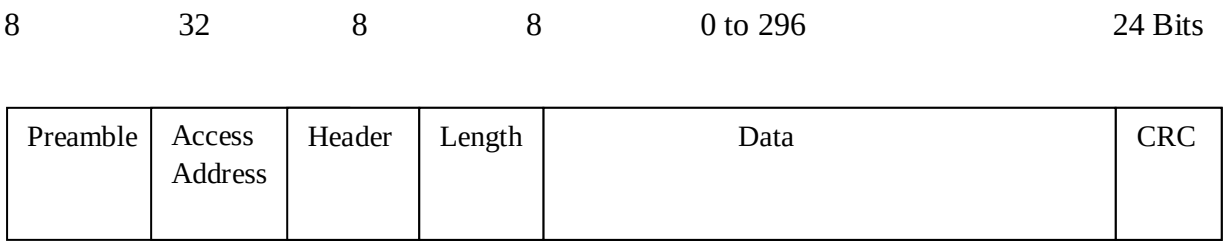

Figure 8: Link Layer Packet Structure

3.3.4 Host Controller interface

This is an interface which connects the host and the controller. It is to be noted that Bluetooth classic had around 60% of Bluetooth controllers which used HCI interfaces. It allows the host to send commands and data to the controller and the controller to send events and data back to the host. So it consists of two separate parts, the logical interface and the physical interface. Logical interface typically include Application programming interface on the controller. Physical interfaces typically include Universal Serial Bus, Secure Digital Input output etc.

3.3.5 Host

The Host part is mainly divided into logical link control and adaptation protocol (L2CAP) layer, Attribute ATT layer, Generic Attribute Protocol (GATT) and Generic Access Profile (GAP). The host performs multiplexing of data, follows protocols and various procedures so that data flows. Each layer of the host will be elaborated in detail in the following section. To simply things host can be any tablet, PC which has an operating system. Or it can be visualized as an environment which exposes host API´s.

3.3.6 Logical Link Control and Adaptation Protocol

This layer acts as a multiplexing layer in BLE. It defines two concepts known as L2CAP channels and L2CAP signaling commands. An L2CAP channel is a bi directional data channel that is terminated on a profile or particular protocol in a peer device. In the case of BLE, fixed channels are used which are in the form of one signaling channel, one security manager and one attribute protocol. So with respect to the packet structure the L2CAP looks like as shown in figure 9.

Length	Channel ID	Information Payload
2 octet	2 octets	Length octets

Figure 9:L2CAP Packet Structure

Each payload in the advertisement packet can be included with the L2CAP packet structure. So each channel ID follows a particular operation which are clearly defines in the Bluetooth specification with separate channel ID´s.

3.3.7 Security manager protocol

To pair up with any device and to retain trust on it, the security manager protocol is used. It provides authentication for the device. Pairing is usually followed by link being encrypted and also by the distribution of the key. Using this scheme, the shared secrets can be distributed from a slave to master for the purpose of reconnection during a later date.

3.3.8 Attribute protocol

The main usage of this protocol is to get the information regarding the device. It defines the rules for accessing data in a device. Attribute is just a piece of data. It can be small piece of data, which has a value that has meaning. Attributes are usually stored in attribute servers, which can be read by an attribute client which can in turn perform reading, writing operations. Attribute usually defines six types of different messages. 1) Requests sent from the client to server 2) responses sent from the server to the client in reply to a request 3) commands sent from the client to the server that have no responses 4) notifications sent from server to client that have no confirmations 5) indications sent from server to client. 6) Confirmations sent from client to server in reply to an indication.

BLE is all about transmitting small piece of data. This data is usually accessed by many devices. Typical data could be signal strength of a mobile device, temperature in a room, current time in a city, the number of times the equipment is opened or closed etc. To represent this data attribute has a value, universally unique identifier and a handle.

Handle is the memory location associated in the BLE controller. It is a 16 bit address which accesses the memory location of a piece of data. Attribute data handles start from 0x0001 to 0xFFFF. Each of these handles represents a memory address, port number or hardware register address for the attribute value.

Value is the type of data that is exposed by a device. It can be of size starting from 0 bytes to a maximum of 512 bytes in length. Attribute also has a type which exposes the type of data. For example it might be temperature, pressure, volume, distance, power, time, charge, Boolean state. Since there are many different types of data, each type of data is exposed by a 128 bit number. These unique identifiers are called universally unique identifier (UUID´s). These require 16 bytes of data to be sent across, so that devices identify the type of data. Bluetooth SIG has defined 128 bit UUID known as Bluetooth Base UUID which can be combined with a 16 bit number. The reason for introducing this 16 bit number which leads to derived UUID is to follow the rules for allocating UUID´s. This also ensures that sending UUID´s across devices is done through short UUID´s and then can be recombined with Bluetooth base UUID when received.

The Bluetooth base UUID is defined as following:

00000000-0000-1000-8000-00805F9B34FB

To understand the concept of attributes consider an example: 22.5 degree Celsius is an attribute. It might be displayed in a BLE device. In order to process this, it is split into the following: 22.5 is value, degree Celsius is represented by a UUID, and the handle would be the memory address which is allocated in the BLE controller. The address would be identified from the look up table where the attribute is present. So the attribute just exposes data on a remote device.

These types of data are usually present in a server which can be represented as shown in figure 10.

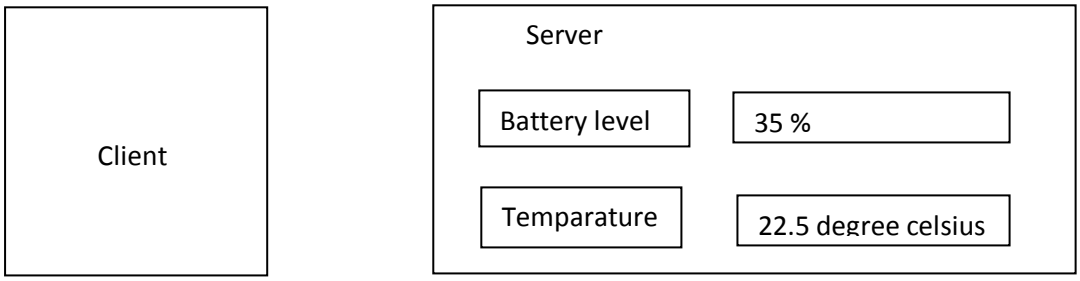

Figure 10: Types of data stored in a server; which are used as clients

Attribute server and attribute client are the two main roles associated with attributes. Attribute server contains all the data; it receives requests, executes those requests and responds to them. It can also indicate values. Attribute client usually communicates with server; it sends requests and waits for response. It can also send confirmations to indications. For the efficient usage of attribute protocol the following operations are usually performed. They are Pull, Push, Set, Broadcast, and Get.

Whenever the data changes in a server, it pushes the data using attribute push. In such case the client can also configure the server to indicate the change. Whenever there is a need for client to request a data from the server, attribute pull is requested. In such case, the client polls the server for the attribute value. This operation might be inefficient when the data doesn't change quite often. Client can also set attributes to the server using attribute set operation. It uses write request command to do so. Typical examples might include setting up a room Temperature to 22 degree Celsius. There is a possibility for the server to broadcast data to any listening devices which can be performed using attribute broadcast. Commands like Read information request can be used to identify if the data exists in the server.

3.3.9 Attribute data base, server and client

A collection of attributes is called a database. Table 1 gives an example of database. Attribute database can be large and complex or it can also be very small.

Table 1:Attribute data base

Handle	Attribute Type	Value
0x0001	«Primary Service»	«GAP»
0x0002	«Characteristic»	{r, 0x0003, «Device Name»}
0x0003	«Device Name»	"Temperature Sensor"
0x0004	«Characteristic»	{r, 0x0006, «Appearance»}
0x0006	«Appearance»	«Thermometer»
0x000F	«Primary Service»	«GATT»
0x0010	«Characteristic»	{r, 0x0012, «Attribute Opcodes Supported»}
0x0012	«Attribute Opcodes Supported»	0x00003FDF
0x0020	«Primary Service»	«Temperature»
0x0021	«Characteristic»	{r, 0x0022, «Temperature Celsius»}
0x0022	«Temperature Celsius»	0x0802

Various operations can be performed to access the attributes. It might include Find requests, read requests, write requests, write command, notifications and indication. To find a particular attribute from a database, find attribute is used by the client. Read request is used to read an attribute from a database. A single attribute, a range of attributes can be read using this command. Write request is used to write a value onto the attribute. Multiple values in a database can also be written using this

operation. In some cases, there might be a situation where a response from write request is not needed, in such cases write command is used. This enables the client to send a write command without any response in return to the client. There is also an option of notifying any changes in the attributes, which can be performed using notification command. Indication also has a same attribute handle and value, but this command causes an attribute confirmation to be sent back. The terms in the attribute database will be explained more clearly in the next part of this document.

3.3.10 Generic attribute Profile

Attribute protocol introduced in the previous section had a flat structure. It consisted of pieces of data which had handles to get them located in the memory. However this attribute table gets quite complex when there's a lot of data which doesn't have any shape or form. So in order to bring a grouping into the data, profiles are created. These profiles have services, sub services and characteristics embedded into them. Typical examples of profiles include a Temperature profile, battery profile, device profile etc. In order to process these profiles, there's a need for hierarchical structure. In order to achieve this, services are used which gives a simple form of structure to them.

In BLE, Generic attribute profile (GATT) defines two forms of grouping known as services and characteristics. Services are synonymous to objects in object oriented systems which is unchanging over a period of time. Services typically include one or more characteristics and also sub services. The characteristic is a unit of data or exposed behavior. These characteristics are self-describing, such that generic clients can read and display these characteristics. To simply things, a service just consists of characteristics and their associated behaviors.

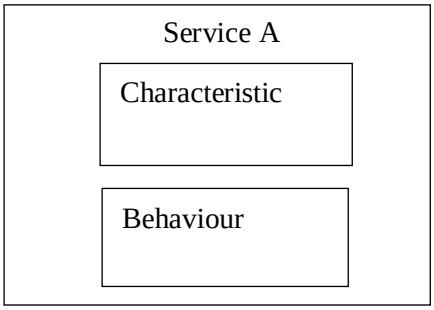

Figure 11: Service consisting of characteristics and behaviors

For the attribute data base represented in table 1, grouping can be performed in the following way. The name of the service is GAP which defines a device called Temperature sensor. It consists of two characteristics which can be read by looking into the memory location as indicated by the handle values. It can be different

Table 2:Grouping

Handle	Attribute Type	Value
0x0001	«Primary Service»	«GAP»
0x0002	«Characteristic»	{r, 0x0003, «Device Name»}
0x0003	«Device Name»	"Temperature Sensor"
0x0004	«Characteristic»	{r, 0x0006, «Appearance»}
0x0006	«Appearance»	«Thermometer»

Characteristic is just a single value. In figure 9 for the corresponding handle of 0x0002 a characteristic is defined. It refers to memory location 0x0003 which has a device name of Temperature sensor. The characteristic in memory location 0x0004 refers to the thermometer in memory location 0x0006. This is how reading of characteristic data takes place.

3.3.11 Generic Access Profile

This is the top most layers in the host and this layer acts as an interface to the application layer. This layer governs the advertising and connection parameters and ascertains roles to the devices.

A BLE device is either given a Central or a Peripheral role. This role is administered based on the initiator of the BLE link. The Central is always the device that initiates the connection, while the Peripheral is the device that is connected to. When the Central and the peripheral are in connection the terms Master and Slave are used extensively. Bluetooth Core Specification also defines Observer and Broadcaster roles. Observers listen to what's happening on the air medium and Broadcasters send but don't receive information.

If the central connects with a peripheral, the Peripheral is said to be advertising state. Peripheral sends advertising packets with a time interval, known as the advertising interval. It is between 20 ms and 10.24 s. The advertising interval defines the time taken to initiate a connection. The Central must receive an advertising packet before it can send a connection request to initiate a connection. The Peripheral only listens for connection requests for a short while after sending an advertising

packet. An advertising packet can contain up to 31 bytes of data. It usually contains a user readable name, preamble, information about the device sending packets, some flags used to know whether the device is connectable or not. The packet structure would like figure 4.

When a Central receives an advertisement packet, it can also send a request for more advertising data, using a Scan Request, if it is configured as an Active Scanner. In such a case, the peripheral might be sending additional data to prove its identity. A Peripheral responds to the request by sending a Scan Response that can contain an additional 31 bytes.

Scanning is used by the Central to listen for advertising packets and to send scan requests. There are two additional timing parameters known as scan window and scan interval. For each scan interval, the Central scans for a time equal to the scan window. This implies that if the scan window is equal to the scan interval, the Central will do a continuous scan. The scan duty cycle (measured in percentage) of the Master is also known as the scan window divided by the scan interval.

When the Central wants to enter a connection, it will use the same procedure as when scanning to listen for advertising packets. When initiating, the Central will send a connection request to the Peripheral when it receives an advertising packet.
By definition it is to be understood that the Central and Peripheral are in a connection from the first data exchange. When in a connection, the Central will request data from the Peripheral at specifically defined intervals. This interval is known as the connection interval. It is decided and applied to the link by the Central. However the peripheral has the authority to send connection parameter updates to the central and can choose its connection interval. According to the Bluetooth Core Specification, the connection interval is between 7.5 ms and 4 s.

If in any case the Peripheral doesn’t respond to data packets from the Central within the time-frame called connection supervision timeout, the BLE link is lost. It is possible to achieve higher data throughput by transmitting multiple packets in each connection interval. Each packet transferred can be up to 20 bytes. There is also a term known as slave latency. It is used when the peripheral chooses to ignore the data sent by the central than the number of ignored intervals is called the slave latency.

Chapter 4

Implementation of BLE ticketing protocol

4.1 Introduction

In the implementation of the BLE ticketing protocols various entities are associated as part of architecture. A comprehensive architecture is not studied or presented here. It is assumed that

a) There is a Transport authority connected to a back-end system. The main role of this body is to install necessary hardware for successful Bluetooth transaction. This hardware is represented in the form of a validation device which will be explained in this chapter.

b) Furthermore all the information exchanged in the ticketing transaction is collected as evidence and forwarded to a back-end processing unit. This unit perhaps can use this information for charging the exact fare for the passenger. This database is assumed to be operated by the transport authority.

c) Transport authority includes security-relevant interactions. These include database of customers, their telephone numbers and all the identity proofs for identifying a user.

This part of the section introduces the ticketing model as shown in figure 12 and is proposed for future implementation.

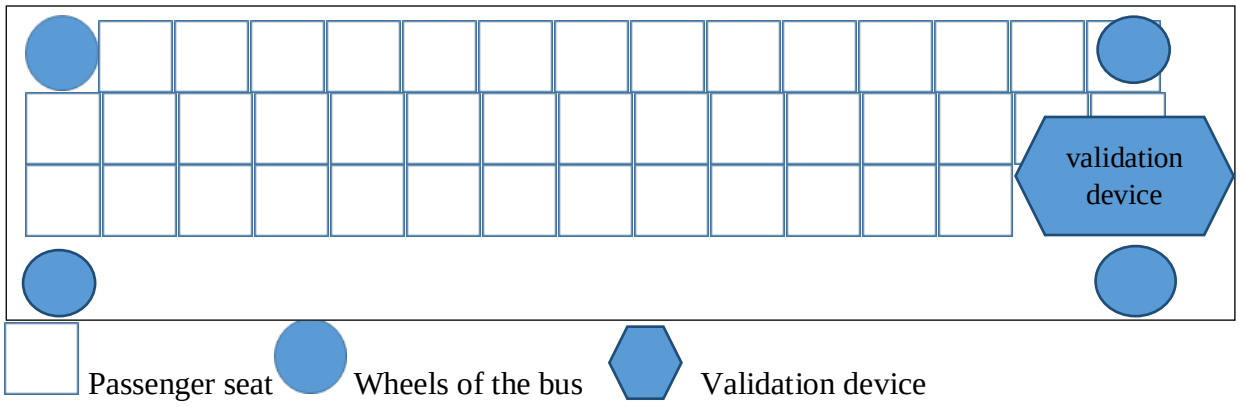

Figure 12: Top view of the Bus

Passengers with BLE enabled mobile phones getting into the bus have a chance to purchase a journey ticket. Validation device allows customers to use their mobile phones and generate a valid record as a proof of entering the Transport Network. Validation Devices are active (self-powered).

The user needs to download a transport app in order to avail this service. Since it is assumed that the handheld devices are low powered devices, they would send the broadcasts to the validation device. The message flow is represented as shown in the figure 13.

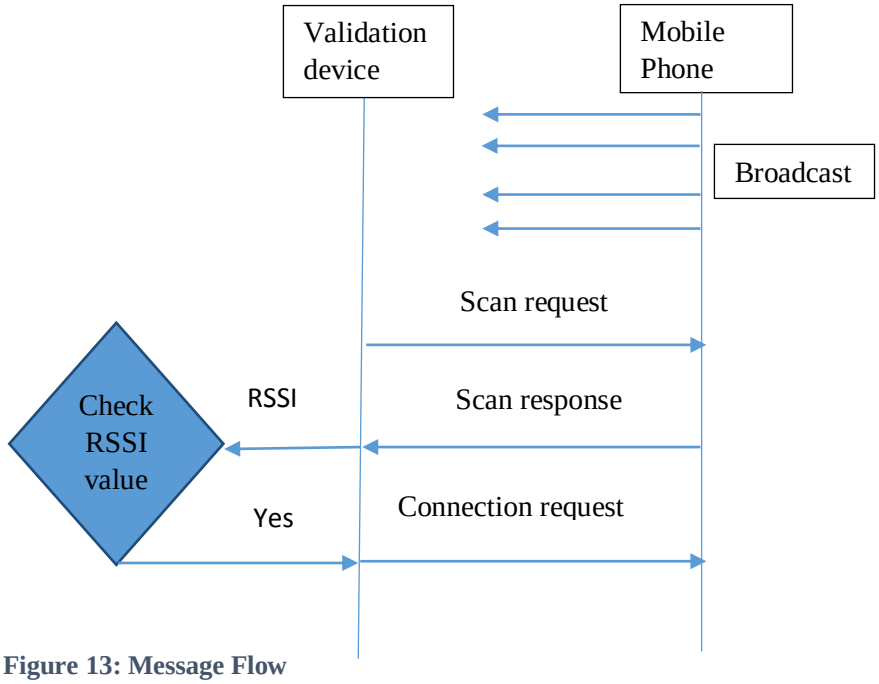

Figure 13: Message Flow

The mobile phone gets into the advertising mode once the ticketing app runs. In this mode the phone constantly sends out advertising packets in the following format:

Preamble	Access address	Header	Length	CRC	Payload

Figure 14: Advertisement data packet

Since the advertising packets are sent by the mobile phone, it assumes the role of a peripheral while the validation device in the bus assumes the role of central. The validation device (VD) after noticing these advertisement packets sends across a scan request.

So VD makes an active scanning, in this state any new device will be discovered by the link layer. Since it is usual that several users/passengers will be willing to enter the bus, the VD makes an active scanning. It is expected that the broadcasting device(s) send a scan response in reply. In the scan response the Bluetooth address, RSSI value, type of data, length and actual payload from the device is transmitted.

The VD compares these values. It performs an identity check for authenticating the passenger. Before this transaction can happen the RSSI value is checked with a threshold value- This action is performed so as to differentiate people who are willing to enter the bus and those who are not willing to get the bus. It implies those passengers who are positioned closer to the VD have higher RSSI value, whereas those staying farther away will have lesser RSSI. After RSSI is checked, a connection request is sent by the VD. At this point of time a BLE link is established between the Mobile phone and the validation device. This will be displayed on the users mobile and he will give his final approval by accepting or rejecting the connection.

The advertising packets shown in Fig 12 will have the following data and they are explained here.

Preamble: The first 8 bits of the packet are 01010101 or 10101010. The determination of the sequence is based on the first bit of the access address. If the first bit of access address is a 0 than 01010101 patterns are used and if it's 1, than 10101010 is used.

Access address: This is 32 bits and when it is the advertising state it uses an advertising access address. In this case the access address is a fixed value i.e. 8E89BED6.

Header: For advertising packet the header includes the advertising packet type. It also some flag bits to specify whether the packet includes public or random address. During the first advertisement sequence the advertisement packet is a general advertisement packet i.e. ADV_IND.

Length: This is 6 bits of data, with valid values starting from 6 to 37

CRC: It is a 3 byte cyclic redundancy check. This is calculated over the header, length and payload fields. The polynomial used is a 24 bit CRC

Payload: This is the actual data that is being transmitted. The ticketing protocols mentioned in this chapter forms the actual pay load in case of ticketing service.

Since we have data in the validation device and the mobile phone, both of these entities form a client and server architecture. Since most of the data is present in the validation device, the VD is made the server while the client is the app in the mobile phone. The validation device stores

a) Identity of the bus
b) Identity of the passenger to authenticate the passenger
c) Identity of the mobile phones in the form of IMEI number

The mobile phone which acts as a client has the following information

a) Identity of data/Application data
b) Certificate on the phone
c) Credit card/Debit card details of the passenger

Since there are lots of attributes, a service specifically for the ticketing is introduced here. It is known as the ticketing service.

4.2 Ticketing service

The idea of the ticketing service is to bring a structure to the data. It includes primary service, characteristic, identity and various other services. A ticketing service would include the following characteristics:

a) Ticketing service name
b) Ticketing Certificate
c) Ticketing ID

Since there are several Bus transport agencies in the world, a separate service is the need of the hour. To ascertain this, ticketing service name is proposed in this parent service. Typical examples include HSL Helsinki, Tampere public transport, London public transport etc. Since each of the Bus transport agencies have their own ticketing prices, terms and conditions etc. ticketing certificate is introduced. This certificate gives us information related to date of authentication of the ticketing service, expiry of the ticketing service, contact details and various other specific details which are in record, which needs to be stored in the passenger database.

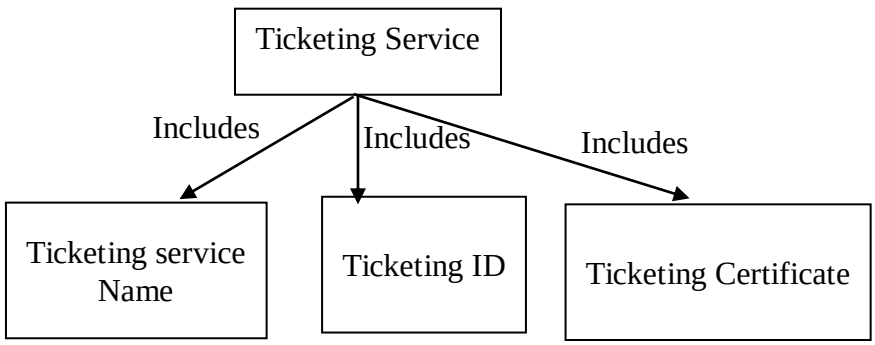

Figure 15: Ticketing service

GATT imposes a structure format and the following table based on the services used for the ticketing profile has to be used. The handle values have been chosen as consequent memory locations to simplify the complexity. These memory locations correspond to the memory locations in a BLE controller.

Table 3: GATT Table

Handle	Type	Value	Permissions
0x0001	«Primary Service»	«GAP»	R
0x0002	«Characteristic»	{r, 0x0003, «identity»}	R
0x0003	«identity»	"IMEI Number" or PAN	R
0x0004	«Characteristic»	{r, 0x0005, «Certificate»}	R
0x0005	«Certificate»	X.509 Certificate	R
0x0006	«Primary Service»	«Ticketing service»	R
0x0007	«Characteristic»	{r, 0x00008, «service name»}	R
0x0008	«service name»	HSL service	R
0x0009	«Characteristic»	{r, 0x0010, «Ticketing ID»}	R
0x0010	«Ticketing ID »	0x0802	R

A GATT Server organizes data in what is called as an attribute table and it is the attribute that contains the actual data. Table 3 is the attribute table. It is to be read as follows:

The BLE controller would have several memory locations in its chip, on one of the memory locations 0x001, the primary service called as Generic access Profile which has the permission to be read is present. It is linked to a characteristic at memory location 0x0002, which has identity details at a particular memory location that is 0x0003. In this case the identity is IMEI number or the PAN number. This service has two characteristics and the next characteristic consists of certificate which is linked to memory location 0x0005. The use of two characteristics is to differentiate both identity and certificate in two separate message flows. It will be clearer in message flow diagram.

The attribute table also has another primary service known as the ticketing service. This ticketing service has two characteristics at memory locations 0x0007 and 0x0009 respectively. Handle 0x0007 links to a service known as HSL service which has the permission to be read. The handle in 0X0009 links to a ticketing ID which has a random value of 0x0802 in this case. As mentioned in the table all the handles are only permitted to be read. Once the attribute table is defined, the following ticketing protocols will be operational.

4.3 Ticketing Protocols

Transaction with a validation device

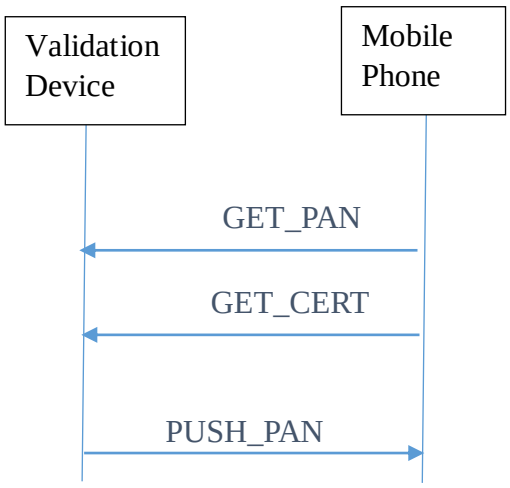

Figure 16: Ticketing Message Flow

This message flow diagram illustrated in figure 16 is the message flow between mobile phone and the validation device. The validation device will query the phone when it is in the range of the VD.

The range can be ascertained by the RSSI value defined in figure 11. A threshold RSSI value is used to match against the mobile phones RSSI value. If the mobile phones RSSI is greater than the Validation devices RSSI, than the validation device begins querying the phone.

Step 1:

The validation device will begin querying the PAN and the certificate from the phone.

GET_PAN

Request data size: 0
Response data size: 10 (Packed BCD, 0xF padded)

The phone is represented by a PAN number, where PAN implies a personal assistant number. This number is a unique number and specific to a mobile phone. So it might also represent the IMEI number of the mobile phone. It is not termed as IMEI number directly and is termed PAN in order to bring all the stakeholders in the ticketing system. Each of these stake holders reserves certain PAN numbers for allocating them to their passengers. So this command returns the PAN identity for the phone. PAN gives the information about the service provider of the passenger. For example: Helsinki transport agency, London bus transport etc. Each of these transport agencies is allocated a suitable numbering system which looks as shown below.

Helsinki transport agency- 001

London bus transport – 002

This command returns a data size of 10bytes which is the format of packed BCD. This forms the payload to the format mentioned in figure 14. So the exact message that goes from the mobile to the validation device is as shown in figure 17.

Preamble	Access address	Header	Length	CRC	Payload
01010101	8E89BED6				GET_PAN

Figure 17: GET_PAN packet

Once the PAN number is cross verified in the database of the VD, a request is also made to get the certificate from the mobile phone.

GET_CERT (Get Certificate)

Request data size: 0

Response data size: TBD

This command returns the certificate that is stored on the phone. The command will succeed only when a certificate has been inserted by the Transport agency into the passengers profile/database. The certificate provides data like the authentication of the passenger, his login credentials, his expiry dates, and his current status for e.g. student, women, pensioner etc. ISO-7816-4 APDU provides a list of certificates that can be used, however for simplicity X.509 certificate is used in this case. There is also a possibility to split the certificate into smaller pieces if needed. Then one byte at a time can be sent across. Once the Certificate is requested the following packet gets transmitted from the mobile to the validation device.

Preamble	Access address	Header	Length	CRC	Payload
01010101	8E89BED6				GET_CERT

Figure 18: GET_CERT packet

To enable two way mutual authentications, validation device must also prove its identity. In order to do that the validation device sends across the PAN to the mobile phone and the following command is used.

Step 2:

The validation device would send the PAN of its own to the phone

PUSH_PAN

Request data size: 10

Response data size: 0

This command sends, in a validation device, its PAN number to the mobile phone. The reason for this is evidence routing, i.e. the phone will report evidence to its home service provider. In this case the home service provider could be any transport agency which the passenger is primarily associated with. This command in turn is used by the transport provider for getting him the fare calculation done. So the process of the Fare calculation starts at this point. The data size requested

by the mobile phone is around 10 bytes and it gets transmitted over the BLE link. The data packet looks as shown in figure 19.

Preamble	Access address	Header	Length	CRC	Payload
01010101	8E89BED6				PUSH_PAN

Figure 19: PUSH_PAN packet

Since the data size is requested with respect to the mobile phone, the response data size is indicated as zero.

All the passengers who wish to enable this transaction done have to get their credit card credentials inside this transaction. For this purpose Trusted Execution environment (TEE) is used, which stores sensitive data in a mobile phone.

Trusted execution environment (TEE)

The TEE is a secure area of the main processor of a smart Phone. It maintains the confidentiality and integrity of the data stored in a mobile phone. TEE in this use case ensures that the consumers can carry out any financial transaction in a safe and trusted environment. All the credit card details are safely stored in this piece of chip.

Step 3:

The validation device would send a challenge to the phone. This is the time when Phone invokes its TEE to compute the bound signature on the challenge.

BOUND_CHALL

Request data size: 20
Response data size: 0
Preconditions: INS_CERT

This command inserts data that will be used as a challenge for the bound signature(s). An entered challenge will apply to only a single execution of BOUND_RSA.

Preamble	Access address	Header	Length	CRC	Payload
01010101	8E89BED6				BOUND_CHALL

Figure 20: BOUND_CHALL packet

The data packet would look like above and this data packet gets transmitted from the mobile to the validation device. In this case a mutual authentication is performed using the RSA algorithm. In reality the hash of the message is calculated by the mobile phone. Message in this case will be certificate which uses a private key for itself. The signature gets attached to the message and both are transferred back to the validation device. The recipient recalculates the hash of the message and then uses the public key to verify the signature it received from the mobile phone.

Step 4: *When the signature has been computed in the TEE it is returned to the Active Validation Device. The Active Validation Device can validate and examine the Certificate and the signature to determine the validity of the Passenger's ticket.*

BOUND_RSA_2 (Bound RSA signature – SHORT)

Request data size: 0

Response data size: 156

Preconditions: INS_CERT, BOUND_CHALL

This command returns a signed string over the following data that is returned together with the signature:

4B: 'I', 'N'. 'C', 'R'

4B: Counter value, in the same format as in GET_CTR

20B: Input challenge as given by BOUND_CHALL

====

28B

It is followed by the 128B PKCS#1.5 signatures, done with the TEE private key.

The short version of the RSA signature is used (in the *Phone* TEE) with the validation devices. The local counter is increased immediately after the operation, i.e. the counter value bound by the signature is the value prior to the update. The Validation Device also has the option to populate the Phone with a ticket inspection record or a failure record in case the TEE signature / Certificate or any other attribute was insufficient for providing access to the PTO's network. The validation device now successfully disconnects the BLE link.

Additional ticketing commands:

GET_CTR

Request data size: 0
Response data size: 4 (MSB)

This command invokes the counter. It asks the counter value that is stored in the phone. Since protocols are with respect to the mobile phone, it responds with a data size of 4 bytes. The phone contains a monotonically increasing counter, which is forwarded by the BOUND_RSA commands. The ability to read the counter is a management service, the presence of which may be detrimental to privacy. The counter is reported in 4 bytes, MSB. Only 100 steps are reported in one byte, i.e. the value 99 is coded as 0x00 0x00 0x00 0x63 and the value 100 as 0x00 0x00 0x01 0x00.

INS_CERT

Request data size: (length as previously provided by INS_CERTLEN)
Response data size: 0
Preconditions: INS_CERTLEN called

This command sets / provisions the certificate. The command may be used several times in sequence. The P1 and P2 bytes of the ISO-7816-4 APDU encode an offset into the certificate buffer, i.e. the certificate can be written in pieces. The protocol can also make use of the X.509 certificate if needed.

INS_CERTLEN

Request data size: 2
Response data size: 0
Preconditions: INS_CERT not called

This command sets / provisions the certificate length for the certificate that is stored on the card. The command can only be given before the certificate has been inserted.
The request bytes are encoded with MSB first, e.g. the data 0x01 0x02 indicates that the certificate length is 258 bytes.

Chapter 5

Experimental setup in Nordic Semiconductor evaluation kit

Nordic Semiconductors Company has been manufacturing ultra-low power and highly efficient wireless solutions in the 2.4GHz space. One of their products nRF51822-EK is introduced in this chapter. This hardware was used as a prototype in testing the BLE link. For ease of convenience this product will also be called as evaluation kit and this term will be used quite extensively.

The evaluation kit can be used as initial prototyping for Bluetooth smart devices. This kit gives access to access to all GPIO pins via pin headers. It incorporates a coin-cell battery holder. The evaluation kit is as shown in figure. The left part of the kit is called as PCA10001 which is the Silicon on Chip (SoC) Integrated Circuit. The Right part of the figure 21 is known as soft device

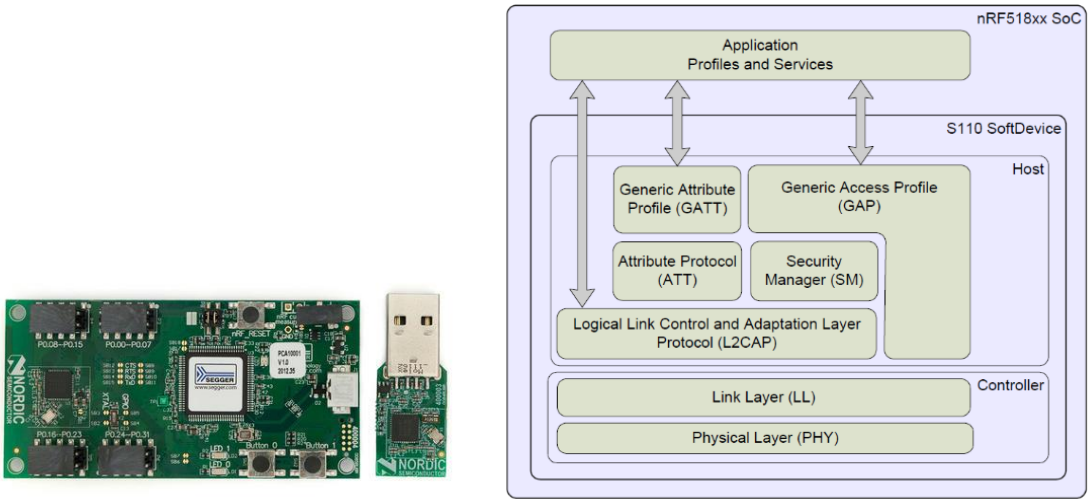

Figure 21: Nordic Evaluation Kit and Architecture (Figure from Nordic Developer zone)

The Soft device is precompiled and linked binary software that integrates a BLE protocol stack on the nRF51822 chip. The Application Programming Interface (API) is a standard C language set of functions and data types that give the application complete compiler and linker independence from the Soft Device implementation. So the soft device is a protocol stack solution that runs on the protected code area. It is accompanies by RAM area which has various memory locations. It can be independently programmed and application will be developed over this.

5.1 Ticketing profile using nRF evaluation kit

As already discussed in chapter 3, the ticketing service can be visualized as server-client architecture. The data that is stored on the validation device will be stored in the GATT server. This data can be accessed using the application programming interface using certain program calls as represented below.

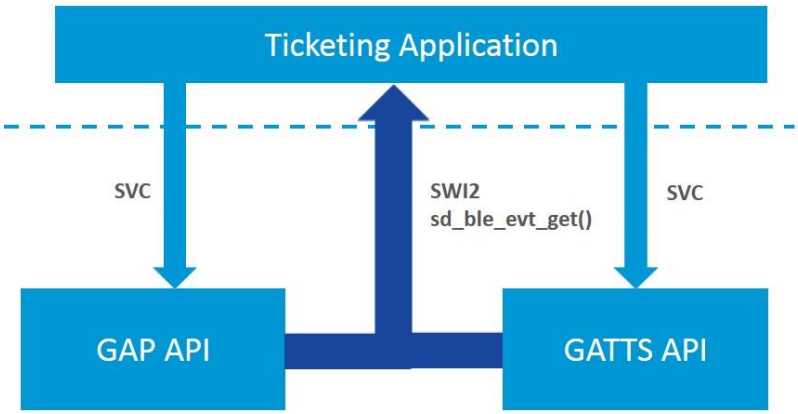

Figure 22: Ticketing application Function calls

The following calls are made by the GATT server to populate the attribute table.

Any changes in the data which is present in the GATT server (GATTS) will be sent as a BLE event when sd_ble_evt_get() is called upon. This event will be acknowledged back by the ticketing application over an API call known as Super Visor calls. All these calls will be numbered to keep a track of the number of updates. Since a ticketing service is built which includes Ticketing service name, Ticketing Certificate, Ticketing ID, an attribute table has be made. To populate the attribute table the function sd_ble_gatts_service_add (type, UUID, out_handle) is called upon. This adds an empty service to the ATT table. So referring to the attribute table the events

The service has number of characteristics as mentioned in the attribute table so the following command is used to add one characteristic at a time:

sd_ble_gatts_characteristic_add (svc_handle, md, value, out_handles)

For attribute table is not always constant and the value gets changed. So to set a value the following function is called

sd_ble_gatts_value_set (handle, offset, len, value)

To get a value from the attribute table the following function is used

sd_ble_gatts_value_get (handle, offset, len, value)

5.2 API calls implemented on the Ticketing protocols

Once the mobile phone sets itself into the advertising mode, the function sd_ble_gap_adv_data_set is called upon. The RSSI value present in the advertising is checked upon a threshold value. Once this value is cleared, the connect request is sent across. The mobile and the validation device are now in connection. As explained in the previous section the GATT server and the attribute table are activated at the validation device end. The ticketing protocols presented in chapter 3, go as advertisement packets in each of the packets which are called as separate functions.

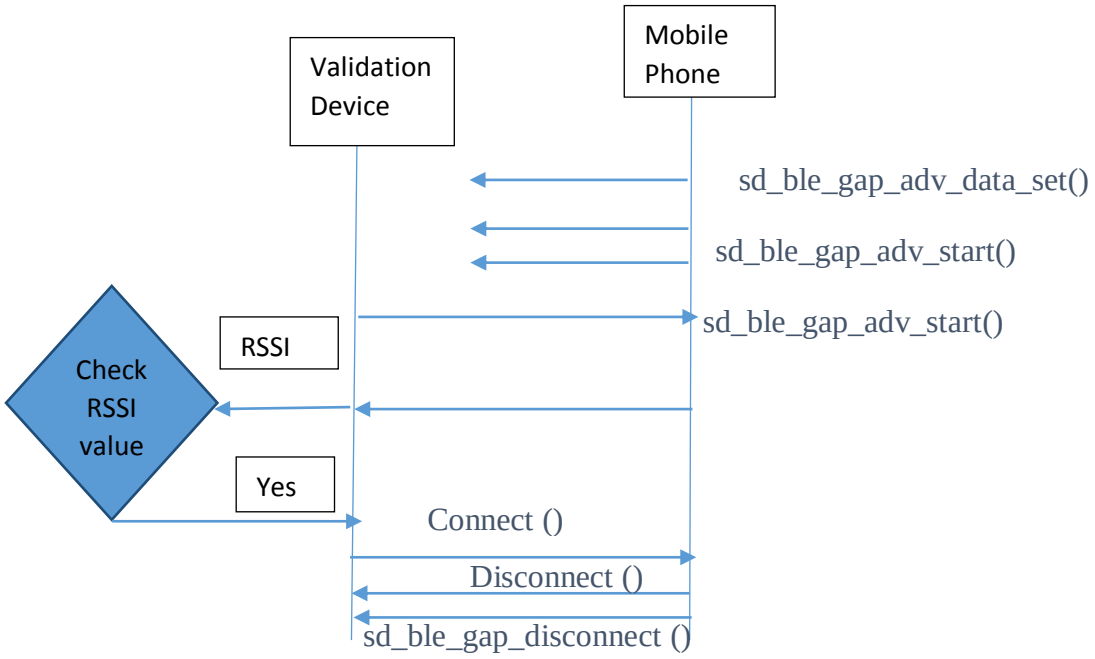

Figure 23: Ticketing Protocols function calls

5.3 Security in BLE ticketing protocols

BLE provides encryption and security features which are similar to Bluetooth classic. It provides features like Man in the middle protection, encryption of payload, authentication of payload and privacy of devices. The suggested BLE ticketing protocols are authenticated using Message integrity checks.

A 32 bit Message Integrity Check (MIC) is added at the end of each payload. This is calculated over the header, length and data payload. Since security is divided between host and controller, it is

assumed that MIC will be validated before sending payload to the host. The packet structure is described in figure 24.

Preamble	Access Address	Header	Length	Data Payload	MIC	CRC
1 octet	4 octets	1 octet	1 octet	0 to 33 octets	4 octets	3 octets

Figure 24: MIC integrated in Adv packet

5.3.1 Phases of Security Setup- First Part of Pairing

During the initial exchange of protocols between the mobile phone and validation device, the advertising packets exchange Input Output capabilities. These are two simple messages, where the first message originated from the VD and gets transmitted to Mobile Device. This event can be requested through a call procedure of BLE_GAP_EVT_SEC_PARAMS_REQUEST.

In phase 2 of this security setup BLE_GAP_EVT_SEC_INFO_REQUEST and BLE_GAP_EVT_CONN_SEC_UPDATE the link is securely established. The details of the mobile device are now stored onto the master which culminates the security in its Phase 3. The signal flow with the corresponding function calls are represented in figure 25.

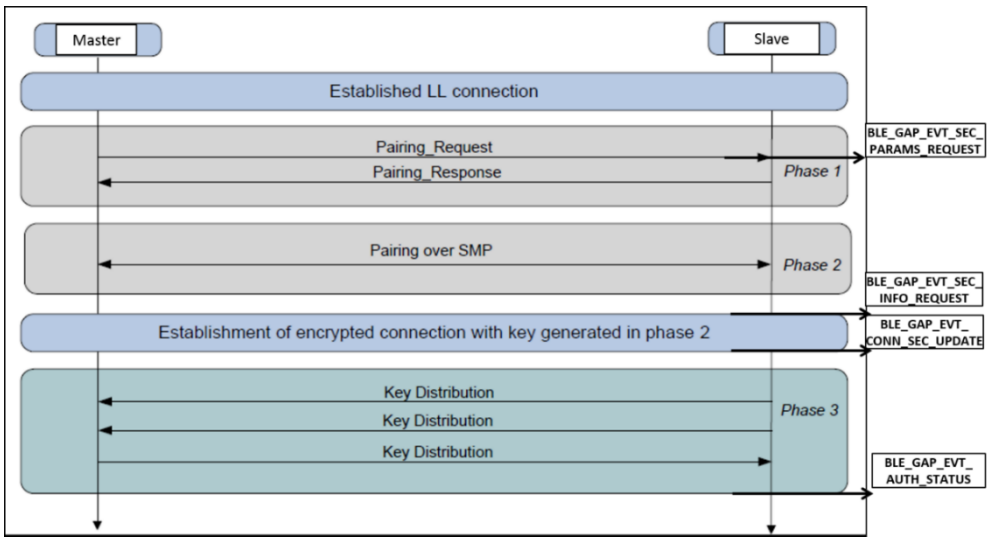

Figure 25: Phases of Security Setup (Figure from Bluetooth Special Interest Group)

Chapter 6

Discussion and Conclusions

Bluetooth low energy is the new global standard. It is a very low power wireless technology. Presently there are around 50 billion devices in the market.

Table 4: Intelligent devices

Phone accessories	>10 billion
Smart energy	~1 billion
Home automation	>5 billion
Health, Wellness, Sports & Fitness	>10 billion
Assisted Living	>5 billion
Animal tagging	~3 billion
Intelligent Transport systems	>1 billion
M2M (Internet connected Devices)	>10 billion

These devices are spread over different markets like health and fitness, consumer electronics and PC, automotive, smart home, industrial automation etc. All over the world average age of human is increasing, healthcare is getting expensive and there are wide spread solutions in the market which uses different proprietary technologies. Low energy can penetrate in these markets and provide solutions to the mass market. In case of smart homes, people are actively looking for technologies such as controllers through which they would like to monitor their houses. Lights, temperature, humidity levels, security locks and windows are getting small computer chips embedded into them. With these new features they are getting controlled. BLE has the potential to make greater impacts in such case. BLE can also be used in the gate way ecosystem. Data that is logged onto a web server can be accessed and analyzed by application software's. All these web servers can be connected with the low cost devices which use this BLE, which means all the things are connected to the internet. As we think the number of applications in which BLE is used enormously increases. However an attempt has been made in this thesis to suggest a ticketing protocol for transport agencies.

This thesis presented GATT based ticketing profile. A simple attribute table was presented considering the attributes from the Mobile phone and validation devices. These devices are assumed to be Bluetooth smart devices. The validation device establishes the connection with the mobile

phone using secured packets from the link layer. This secured feature is discussed in chapter 5, where it is understood that all the message packets arriving at the mobile end are encrypted. These packets make use of the Message Integrity check feature in the BLE specification. Since these packets are highly secure the connection established is considered to be highly secure. This makes this system more robust. The GATT based architecture is presented in chapter 4 which consists of a Ticketing service. Each of these services consists of sub services to accommodate other services which would be used with ticketing service. This feature enables BLE systems to be modeled as simple structures.

Several GATT based custom profiles are being implemented, this thesis made an attempt to make a Ticketing profile in BLE. However, a complete implementation of the same has not been conducted. However, this thesis involves the necessary protocols and the necessary structures to achieve a ticketing profile. It also thoroughly describes the ticketing protocols and introduces the formal details of the packet structure involved in ticketing transaction.

Several existing payment systems demand the users to always keep their phone in close proximity to the validation devices. But BLE has the potential to access these validation devices through a longer range. This would ultimately reduce long queuing times and increase productivity for the end consumers. With specific reference to Transport agencies, these agencies have already used quite many technologies starting from SMS services till RFID based payments. It is highly recommended for them to start using a secure technology which performs a ticketing transaction in a short amount of time. This study will help them to identify the right technology for their future use.

If these get implemented on any of the Bluetooth controllers, than the real potential of BLE will be unleashed. Future work in this area of course includes implementing these suggested protocols. Work can also include setting up constant updates in the form of notifications to the passenger willing to enter the bus. This passenger would get the notifications of the number of vacant seats in the bus, the exact seat number and its position.

References

[1] A.M Alshahrani, S.Walker, "NFC performance in a mobile payment service compared with a SMS based solution," *Int. Conv. On Green Computing, Communication and Conservation of Energy (ICGCE),* Chennai, 2013, pp.282-286

[2] Simon Fong, Edison Lai, "Mobile mini payment scheme using SMS-credit," *Computational Science and Its Applications (ICCSA),* 2005, pp.1106-1114

[3] Manoj V, Bramhe, "SMS based secure mobile banking," *Int. Journal of Engineering and Technology Vol.3 (6),* 2011, pp.472-479

[4] Wen Chuan Wu, "A QR code based on street parking fee payment mechanism," *10th Int. Conf. on Intelligent Information Hiding and Multimedia Signal Processing (IIH-MSP),* 2014, pp.106-109

[5] http://www.globesherpa.com/virginia-railway-express-selects-globesherpa-for-mobile-ticketing-project/

[6] Dinparast S. et all, "A mobile payment scheme using 2D-barcode," *5th Conf. on Information and Knowledge Technology (IKT),* 2013, pp.400-105

[7] Parul Oswal and Michelle Foong, "RFID Vs Contactless smart cards- An unending debate", Industrial Technologies Frost & Sullivan Asia Pacific.

[8] Lacmanović I et all, "Contactless payment systems based on RFID Technology," *Proc. of the 33rd Int. Conv.MIPRO,* 2010, pp.1114-1119

[9] http://www.visa.ie/media/images/cloudberries-55-1718.pdf

[10] Juels A, Molnar D, Wagner D, "Security and privacy issues in e-passports," *1st Int. Conf. on Security and Privacy for Emerging Areas in Communications Networks,* 2005, pp.74-88

[11] Meingast M, King J, Mulligan D.K, "*Embedded RFID and Everyday Things: A case study of the security and privacy risks*," IEEE Int. Conf. on RFID, 2007, pp. 7-14

[12] Hasan, M.F.M et all, *"RFID-based Ticketing for Public Transport System: Perspective Megacity Dhaka,"* 3rd IEEE Int. Conf. on Computer Science and Information Technology (ICCSIT), 2010, pp. 459-462.

[13] Qinghan Xiao, Savastano Mario, "An Exploration on Security and Privacy Issues of Biometric Smart ID Cards," Information Assurance and Security Workshop, 2007, pp.228-233.

[14] Smart Card Alliance, "Transit and retail payment: Opportunities for collaboration and convergence," October 2003.

[15] Information technology – Telecommunications and information exchange between systems Near Field Communication – Interface and Protocol (NFCIP-1), 1st edn, ISO, Geneva, Switzerland (2004)

[16] ISO/IEC 21481:2005. Information technology – Telecommunications and information exchange between systems – Near Field Communication Interface and Protocol -2 (NFCIP-2), 1st edn., Geneva (2005)

[17] NFC Forum. Internet: http://nfc-forum.org/, [5th Feb, 2015]

[18] Jan-Erik Ekberg, Sandeep Tamrakar, *"Mass Transit Ticketing with NFC Mobile Phones."* Trusted Systems Lecture Notes in Computer Science Volume 7222, 2012, pp 48-65

[19] Wei LIU, Feng ZHAO, "The GPRS based Mobile payment system based on RFID," Int. conv. On Communication Technology, 2006

[20] Bluetooth Special Interest Group. Internet: https://www.bluetooth.org/en-us/specification, [5th Feb, 2015]

[21] Robin Heydon, Bluetooth Low Energy: The Developers Handbook, Prentice Hall, 2013