NTNU – Trondheim
Norwegian University of
Science and Technology

Bluetooth Low Energy - privacy enhancement for advertisement

Ping Wang

Master of Telematics - Communication Networks and Networked Services (2
Submission date: June 2014
Supervisor: Stig Frode Mjølsnes, ITEM
Co-supervisor: Sandeep Choudhary, Nordic Semiconductor

Norwegian University of Science and Technology
Department of Telematics

Problem Description

Bluetooth Low Energy (BLE) is an emerging low-power wireless technology developed for short-range control and monitoring applications. It links wireless sensors via radio channels to cell phones and computers, leveraging a ready-built infrastructure for data transmission.

Advertisement is a method to utilize a Bluetooth device to deliver messages to other Bluetooth devices in the way of non-connection. There are two typical roles involved in BLE connectionless advertisement: advertiser and scanner. Advertiser periodically broadcast advertising packets over BLE radio channels and may respond with more service data upon request from peering devices. Scanner listen to advertising messages from advertiser and may choose to either request further information from the advertiser or ignore the messages. However, connectionless advertisement over open radio channels has the risks of privacy leakage and device tracking.

The concern of this project is that the BLE devices are expected to utilize privacy enhancement for confidential advertisement and anti-tracking of devices. All the work will build upon Bluetooth core specification version 4.1, investigating the existing BLE connection-oriented privacy mechanisms and properties where essential, and analyzing their strengths and weaknesses. In addition to that, privacy enhancement for connectionless advertisement will then be proposed. The enhancement will focus on both cryptographic analysis and practical implementation.

Assignment Given: February, 2014
Professor: Stig Frode Mjølsnes
Supervisor: Sandeep Choudhary

Abstract

The aim of this project is to design, simulate, and implement a privacy enhancement protocol over BLE advertising channels. The design of the privacy enhancement is generic and modular. Due to the risk of privacy disclosure and device tracking by adversary, the main focus will be put on designing and implementing message confidentiality, replay prevention, and anti-tracking of device over BLE advertising channels. Bluetooth core specification 4.1 is used as baseline for design and implementation.

In order to provide resistance against replay attacks and device tracking, this project has taken counter approach. It proposes a 3-way handshake protocol for nonce Rs deployment. There are two nonces Ra and Rs involved in the 3-way handshake protocol. The advertiser generates a nonce Ra as challenge sent to the scanner, which assures of freshness of the advertising session. Then the scanner generates a nonce Rs for advertising confidentiality and replay prevention. After the nonce Rs is deployed successfully from the scanner to the advertiser, the local counters Receiving (RX) and Transmitting (TX) on both sides are initialized to be Rs which protects all the following advertisement in the advertising session.

To accommodate to open BLE advertising channels a handling mechanism of counter out-of-synchronization is given in system design. Moreover, to avoid unnecessary power consumption in the BLE devices then mitigation for Denial-of-service (DoS) is also proposed.

In addition, advertising confidentiality, replay prevention, and anti-tracking of device have been simulated in Scyther and also been integrated into the code. The functional tests have been done in a realistic testing environment. The results show that the added functionalities work as designed.

Preface

This report is written as a part of my master thesis in Information Security within Telematics at Norwegian University of Science and Technology (NTNU) 2014. The project has been carried out at the department of Telematics in cooperation with Nordic Semiconductor ASA under the guidance and supervision of Professor Stig Frode Mjølsnes (Telematics department) and Mr. Sandeep Choudhary (Senior R&D Engineer Nordic Semiconductor ASA).

I would like to thank Professor Stig Frode Mjølsnes. His enthusiasm for my topic and tremendous expertise is highly appreciated. Multiple discussion are very time-consuming but I got a lot of valuable feedback from professor for both system design and project report. I would also like to thank my supervisor Mr. Sandeep Choudhary. He has provided much help in the practical areas, including hardware support as well as feasibility analysis of system design from a practical perspective. His continuously ongoing enthusiasm throughout this project and insightful ideas have been of great value to me. Their comments have helped me a lot to accomplish this report with fruitful outcome.

Ping Wang
Trondheim, June 2014

Contents

List of Figures

List of Tables

List of Acronyms

AAR Address Accelerator Resolve.

BLE Bluetooth Low Energy.

CID Channel Identifier.

CRC Cyclic Redundancy Check.

DoS Denial-of-service.

FDMA Frequency Division Multiple Access.

GAP Generic Access Profile.

GUI Graphical User Interface.

HCI Host Controller Interface.

IDE Integrated Development Environment.

IRK Identity Resolving Key.

IRKa Identity Resolving Key of Advertiser.

IRKs Identity Resolving Key of Scanner.

L2CAP Logical Link Control and Adaptation Protocol.

LL Linker Layer.

LTK Long Term Key.

MIC Message Integrity Code.

NRPA Non-resolvable Private Address.

NTNU Norwegian University of Science and Technology.

PDU Protocol Data Unit.

RAM Random Access Memories.

RFU Reserved for Future Use.

RPA Resolvable Private Address.

RX Receiving.

SDU Service Data Units.

SMP Security Manager Protocol.

TDMA Time Division Multiple Access.

TIFS Time Inter Frame Space.

TX Transmitting.

XOR Bitwise Exclusive OR.

Chapter 1
Introduction

Bluetooth Low Energy (BLE) is an emerging low-power wireless technology developed for short-range control and monitoring applications. It links wireless sensors via radio channels to cell phones and computers, leveraging a ready-built infrastructure for data transmission.

Advertisement is a method to utilize a Bluetooth device to deliver messages to other Bluetooth devices in connectionless mode. The recipient device can choose to request further information from the advertiser or ignore the messages. As BLE becomes more and more popular, BLE devices have received some attention in privacy with regards to their advertisement.

BLE device address and service data within advertisement over open radio channels gives adversary opportunity to disclose user privacy and track the communicating devices. In some cases the adversary can collect data within advertisement about device information and user habit without the users' consent, which leak users' privacy and may cause more consequence. All the potential risks motivate a built-in mechanism to protect user privacy. BLE devices are, thus, supposed to provide an enhanced privacy protection to advertisement.

1.1 Privacy on BLE

Bluetooth core specification 4.1 is aware of the potential privacy risks during connection mode and connection procedures [Gro13c]. In order to use private address over data channels to reconnect to known BLE devices, there are different mechanisms for device anonymity in the specification, which consists of NRPA and RPA. Among the core specification, four of the privacy features state that [Gro13d]:

- A privacy-enabled peripheral device should use a RPA in undirected connectable mode, and use a RPA or NRPA in scanning answer.

- A privacy-enabled central device should use a RPA to create a connection, and use a RPA or NRPA in active scanning request.

- A privacy-enabled broadcaster shall use either a RPA or NRPA in the broadcast mode.

- A privacy-enabled observer can use either a RPA or NRPA in an active scanning request.

If the privacy features are enabled, BLE devices use private addresses to send connection request to a peer device. The peer BLE devices has to retrieve a pre-stored bonding database in order to resolve the private addresses. The goal of the process is to associate a private address with a legitimate communicating device.

1.2 Thesis motivation

However, the existing privacy features in section 1.1 are not used to advertise in the Generic Access Profile (GAP) discovery mode and procedures [Gro13c]. Since in open radio environment the advertising packets can be eavesdropped by adversary, there will also be a concern with BLE device privacy within BLE connectionless advertisement. This concern motivates the research of this thesis.

The first step of this thesis is to use the Bluetooth core specification 4.1 as baseline, and put the existing privacy features in section 1.1 to connectionless advertising context to evaluate if the existing connection-oriented privacy features can be re-used directly to connectionless BLE advertisement.

Then the vulnerabilities of the existing privacy features are identified by analysis of advertisement. Due to no protection of counter or timestamp, the procedure with RPA or NRPA has suffered from problems with both replay attacking and device tracking. An adversary can replay advertisement with RPA or NRPA, and attempt to elicit response from a peer BLE device which is then traceable. Furthermore, service data within advertisement will not be protected by the existing privacy features. It means that the adversary is able to capture service data in plaintext over the open advertising channels. If the service data is static, it can also be used to track the sending BLE device.

With intention to solve the vulnerabilities mentioned before, this thesis will provide a solution that enhances user privacy over BLE advertising channels. It prevents replaying attack and device tracking within BLE advertisement.

1.3 Thesis objective

1.3.1 Main objective

The main objective of this thesis is to enhance the privacy features in Bluetooth core specification 4.1 by strengthening confidential advertisement and countermeasures against replay attacking and device tracking.

Privacy concerns arise whenever private information within advertisement is transmitted. In order to meet these concerns, this thesis will focus on private BLE advertisement about how to establish attacking models, how to transmit the private information in an insecure environment, and how to resist on various attacks. Other aspect of privacy, such as after connection is established, will not be covered to the same extent.

1.3.2 Research questions

In terms of the main objective there are two research questions I would like to investigate:

- How can we protect private information within advertisement, including BLE device address and static service data.
- Can we propose a system design that can prevent replay attacking and keep adversary from tracking BLE devices.

1.4 Research method

In order to address the research questions, the following steps will be followed in this project:

- Review existing privacy features in the Bluetooth core specification 4.1.
- List most possible scenarios for privacy leakage and tracking BLE devices.
- Make system design to enhance BLE privacy.
- Simulate the privacy enhancement protocol and verify security claims of the protocol.
- Implement the system design and demonstrate the enhanced privacy protocol.

The internal design cycle is one of the most important of any design research project. After defining research objectives, the research activities of this thesis will iterate

among the system design, implementation, evaluation, and improvement based on the feedback of the evaluation [HMPR04]. In terms of project requirements and constraints, the whole project will follow the internal design cycle iteratively until a satisfactory design is achieved.

In this project, in order to evaluate implementation of the system design, functionality and integration test will be performed in a realistic environment. All test scenarios will be relevant to privacy disclosure and device tracking.

1.5 Thesis contribution

Several contributions are made by this thesis to the field of BLE privacy protection over advertising channels. There are existing privacy protection mechanisms for connection-oriented communication in Bluetooth core specification 4.1. However, the contributions made by this thesis focus on connectionless advertisement and haven't yet been defined in the standard specification:

- Identifying attacking models for BLE advertisement, both with respect to user privacy requirements and nature of open BLE advertising channels, and performing the classification.
- Pre-study existing and relevant BLE knowledge, identifying their core characteristics and grouping them together.
- Making system requirements for BLE privacy enhancement protocol over advertising channels.
- Proposing a complete privacy enhancement protocol over advertising channels.
- Developing analytical models for simulating and analyzing security properties of the BLE privacy enhancement protocol over advertising channels.
- Implementing and testing a demonstration code, to make sure feasibility of the privacy enhanced protocol in real BLE advertising environment.

1.6 Outline

In Chapter 2, it gives an overview of BLE architecture, along with a detailed description of advertisement and security features defined in Bluetooth core specification 4.1. If the reader already has competence about BLE system structure, advertisement, and security features, the reader can jump to Chapter 3 directly.

In Chapter 3, an overview of current research process of BLE privacy, and privacy challenge in this research area is given.

In Chapter 4, it describes system design for BLE privacy enhancement.

In Chapter 5, the privacy enhancement protocol will be simulated and verified in Scyther security protocol verification tool.

In Chapter 6, the implementation of system design will be described. Functional and integration test is used as criteria to evaluate privacy level and performance of the system design.

In Chapter 7, future work will be covered. In this chapter, the focus will be put on potential improvements within this project which can be made in the future.

In Chapter 8, a conclusion for this project is made.

Chapter 2
BLE background

In the first section of this chapter the BLE overview and layer structure relevant to this project is described. In the second section the mechanism of data transportation in BLE is presented. The third section details characteristics of the BLE advertisement. The last section outlines underlying BLE security mechanisms.

2.1 Core system architecture

BLE is a low-power wireless technology for short-range control and monitoring applications. It operates in the 2.4 GHz ISM band and is able to link wireless sensors to cell phones or computers via 40 radio channels in the way of ad-hoc [Gro13a]. BLE devices can utilize a ready-built BLE infrastructure for data transmission within an effective range. Figure 2.1 illustrates BLE core system architecture. In following, different perspectives of BLE relevant to this project will be introduced.

2.2 Data transport

In BLE, a layer structure will be utilized to transport data and the structure is illustrated by figure 2.2. L2CAP layer can mimic an integration from transport layer down to data link layer in OSI model, which establishes a connection between two devices. It can process variable-sized user data from application layer, and encapsulate user data into frame [Gro13b]. The second layer, link layer, provides interfaces between L2CAP layer and physical layer. It is controlled by a state machine and provides flow control of data transmission. The Physical layer contains interfaces for many physical interfaces and conducts data transmission through radio system.

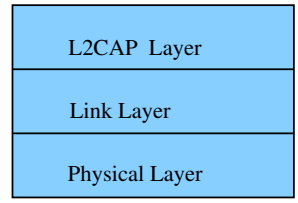


Figure 2.2: BLE data transport structure, adopted from [Gro13b]

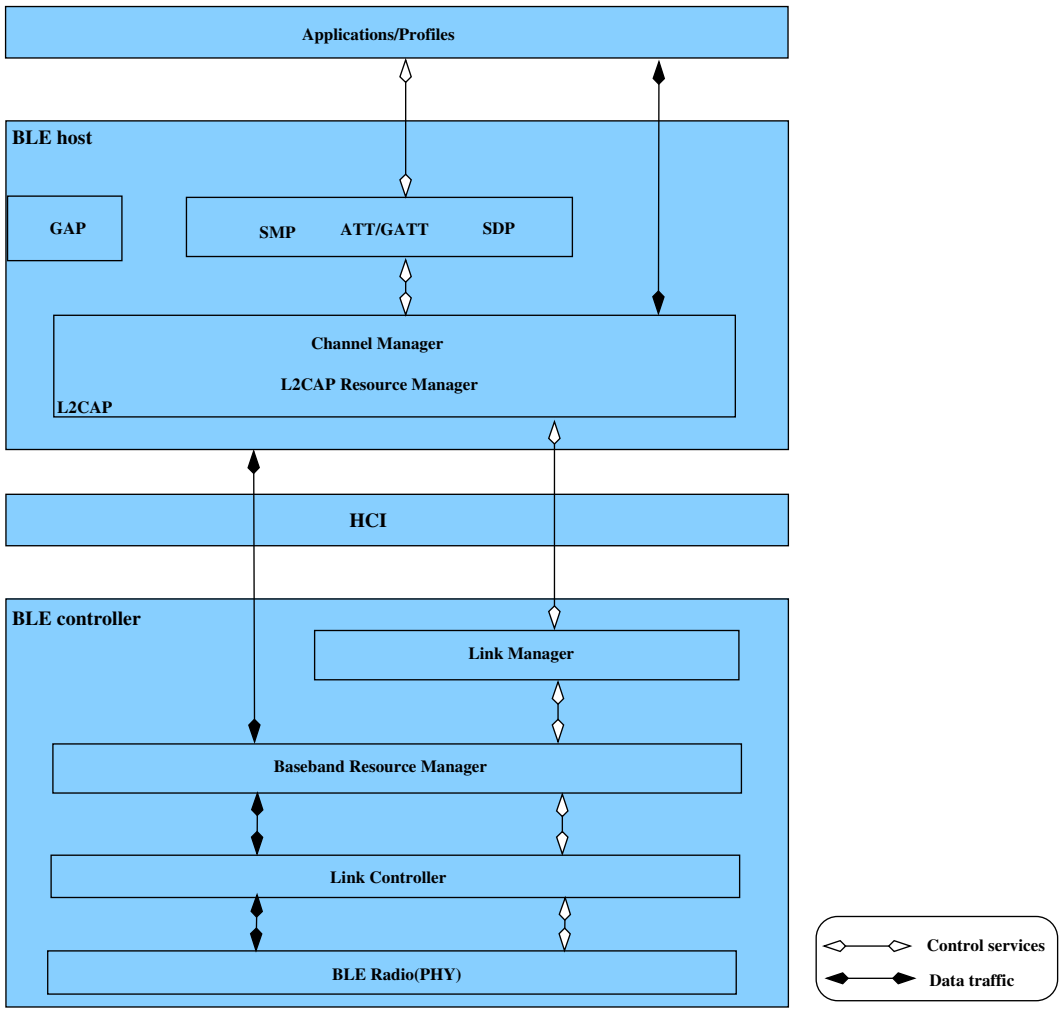


Figure 2.1: BLE core system architecture, adopted from [Gro13a]

2.2.1 Physical layer

The physical layer is also named driver-layer and contains many physical interfaces. In this project, the main focus will be put on BLE physical advertising channels. The advertising channels are a set of three fixed physical channels spread evenly across the BLE frequency spectrum. One complete advertising session must be operated on the same physical channel between two communicating devices within their effective range. Due to a strong likelihood of physical channel collision caused by multiple BLE tuning their transceivers to the same physical channels, a random access address will be generated by BLE devices to identify a physical link [Gro13b].

2.2.2 Link layer

The link layer of a BLE device can be operated in terms of a state machine with the following states: Idle, Advertising, Initiating, Scanning, and Connection. Non-connectable advertisement will use the states: Idle, Advertising, Scanning. Connectable advertisement uses Idle, Initiating, and Connection states [Gro13h]. The state change of link layer will take place according to various input at a given timeslot. Figure 2.3 illustrates the link layer state machine.

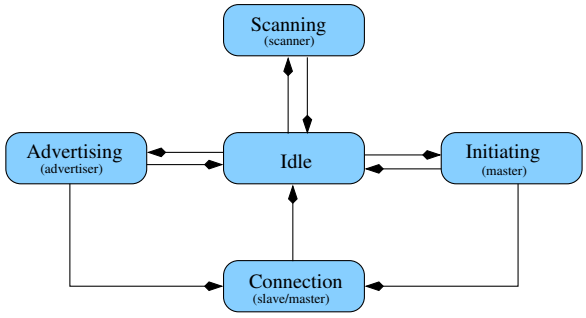


Figure 2.3: BLE link layer state machine, adopted from [Gro13h]

2.2.3 L2CAP layer

L2CAP provides transport interface to applications and services. It can create connection to peer devices in equivalent entities. The endpoints of L2CAP channel can be identified by a Channel Identifier (CID). The main role of L2CAP is to create, configure, and tear down channels. In addition, it is responsible for Service Data Units (SDU) multiplexing and segmentation, flow control [Gro13b].

As to the detail of encapsulation, L2CAP layer encapsulates asynchronous and isochronous user data into frame. Unicast user data between two BLE devices will be transported by connection-oriented L2CAP channels. Broadcasting data will be transported by connectionless L2CAP channels. However, if it is not required to deliver user data in frames, a baseband logical link instead of L2CAP channel will be used. The baseband logical link provides a constant transport rate and fixed intervals to transport data which is locked to the piconet clock [Gro13b].

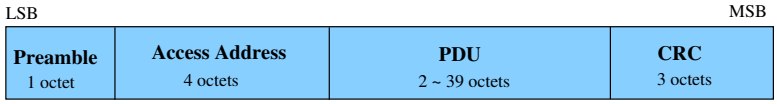


Figure 2.4: BLE link layer packet format, adopted from [Gro13f]

2.2.4 Packet format

There is only one packet format in the BLE link layer for both advertising and data communication. Figure 2.4 illustrates the link layer packet format.

- **Preamble:** the BLE receiver will use the preamble as frequency synchronization.

  Advertising channel: 10101010b.
  Data channel: 10101010b or 01010101b.

- **Access Address:** it is used for BLE piconet communication derived from the initiator's address.

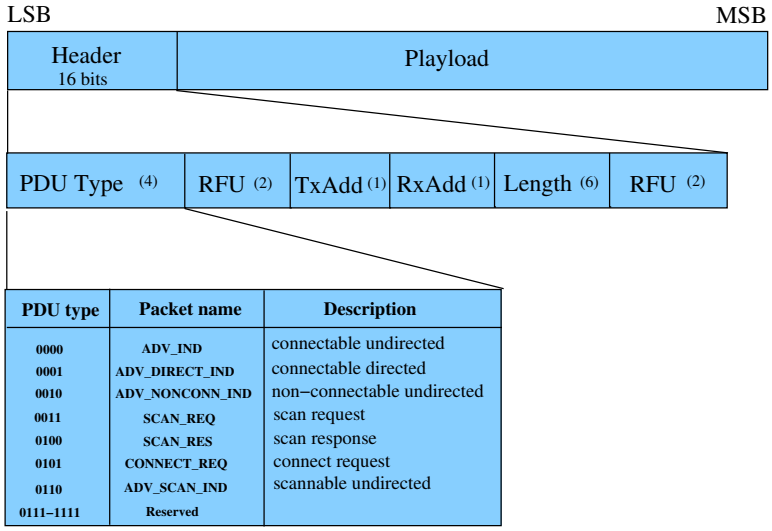
  Advertising channel: Ox8E89BED6.
  Data channel: A 32-bit random value generated by BLE device generated by the initiator and transmitted in a connection request.

- **PDU:** There are two types of PDU in BLE, advertising channel PDU and data channel PDU, which are illustrated by Figure 2.5 and 2.6.

- **Cyclic Redundancy Check (CRC):** It will be calculated on the PDU in all link layer packets and used to detect transmission errors. A packet with an incorrect CRC should be rejected by the receiving device.

2.3 Advertisement

BLE communication has 40 physical radio channels and three of those are used as advertising channels. In order to allow multiple BLE devices to share the same advertising channels, a Time Division Multiple Access (TDMA) based polling scheme is used by the devices to access a advertising channel at different time slots. In more detail, physical channels of BLE advertising can be sub-divided into time units as events, which include Advertising and Connection events. Consecutive events start from the first advertising channel and then occur at regular intervals with a random delay to count against transmission interference [Gro13i].



PDU type	Packet name	Description
0000	ADV_IND	connectable undirected
0001	ADV_DIRECT_IND	connectable directed
0010	ADV_NONCONN_IND	non−connectable undirected
0011	SCAN_REQ	scan request
0100	SCAN_RES	scan response
0101	CONNECT_REQ	connect request
0110	ADV_SCAN_IND	scannable undirected
0111–1111	Reserved	

Figure 2.5: BLE advertising PDU packet format, adopted from [Gro13f]

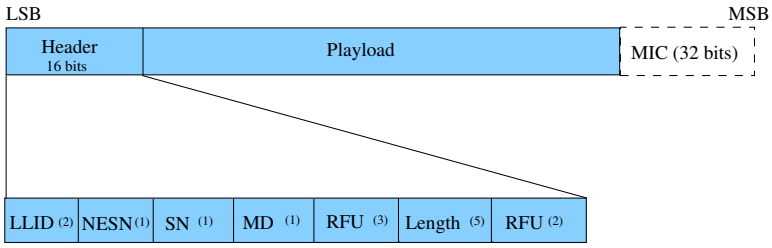


Figure 2.6: BLE data PDU packet format, adopted from [Gro13f]

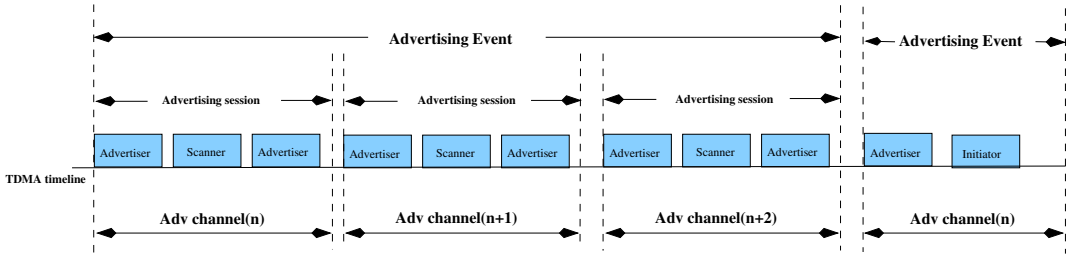


Figure 2.7: BLE advertising event, adopted from [Gro13i]

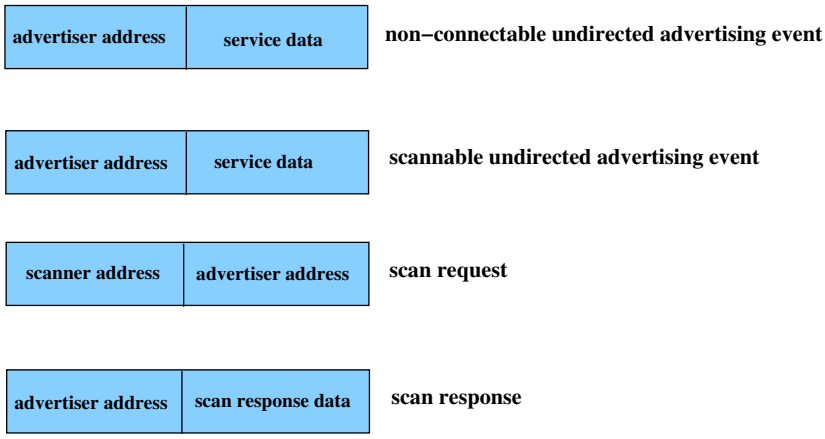


Figure 2.8: BLE PDU for non-connectable advertising events, adopted from [Gro13g]

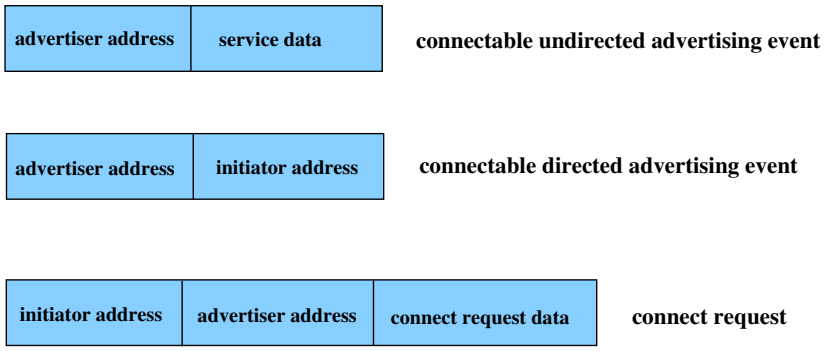


Figure 2.9: BLE PDU for connectable advertising events, adopted from [Gro13g]

Figure 2.7 illustrates the typical BLE advertise events, two typical roles can be involved in the events: advertiser and scanner. An advertiser can transmit advertising packets about its general service on advertising PHY channels. The devices that receive advertising packets without the intention to connect to the advertiser are referred as scanner. Over the advertising channel, the scanner can send scan requests to the advertiser for more detailed service information. The advertiser will then respond to the requests by scan responses. Figure 2.8 illustrates the different advertising PDU of a non-connectable advertising event.

As to connection events, a BLE device that listens for connectable advertisement and then initiates a connection is referred to as an Initiator. Figure 2.9 illustrates the different advertising PDU of connection event.

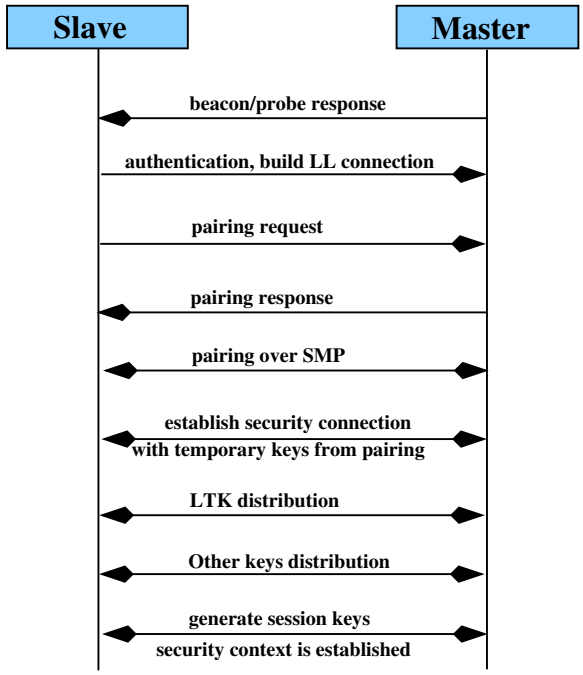


Figure 2.10: SMP in nRF51822 Evaluation chip, adopted from [Gro13l]

From figure 2.8 and figure 2.9, it can be seen that the adversary can fetch the BLE addresses and service data from unprotected PDUs. This gives a big risk of privacy disclosure and device tracking.

2.4 BLE security overview

In Bluetooth core specification 4.1, all the following BLE security mechanisms are only used in connection mode and procedures. So the role of master and slave will be used in following description. Even though the security mechanisms in the connection mode can not directly apply to privacy enhancement over BLE advertising channels, it still gives us the overview of BLE security mechanisms and helps to conduct the following research for this project. To setup an encryption session, the master and slave need to build up an encrypted link through pairing, then execute key distribution in the encrypted link. Figure 2.10 illustrates a typical example about how to set up a security context in BLE communication. Typical steps to establish a bond between the master and slave can be boiled down to [Gro13l]:

- The BLE slave requests the master to initiate a pairing.

- The master and slave exchange security properties.
- If the master supports the encryption mode required by the slave, security manager of the master device will initiate security pairing.
- The output of pairing will be used to build up an encrypted link between the master and slave.
- In the encrypted link, the key distribution will be executed between master and slave devices.
- After getting the distributed keys, the master and the slave will update the local bond database.
- Derive session keys from the distributed Long Term Key (LTK) and establish security context for communication.

If there is a successful bond stored in the database locally for the master and slave, reconnection can be encrypted, random addresses can be resolved, or the BLE device can verify signed data between the master and slave. Table 2.1 shows key definition in BLE and its usage. Identity Resolving Key (IRK) will be used for privacy enhancement in section 4 system design.

Key	Name	Size(bit)	Description
IRK	identity resolving key	128	Generate and resolve random addresses, reduce the ability to track a LE device
CSRK	connection signature resolving key	128	Sign data and verify the signature on the receiving device
LTK	long term key	128	Generate the contributory session key for an encrypted connection
EDIV	encrypted diversifier	16	Identify the long term key, a new EDIV is generated corresponding to a unique LTK
Rand	random number	64	Identify the long term key, a new Rand is generated corresponding to a unique LTK.

Table 2.1: Keys in BLE, adopted from [Gro13m]

After all the keys are available for both the master and slave, the Linker Layer (LL) of the devices provides the mechanism of encryption, decryption, and authentication

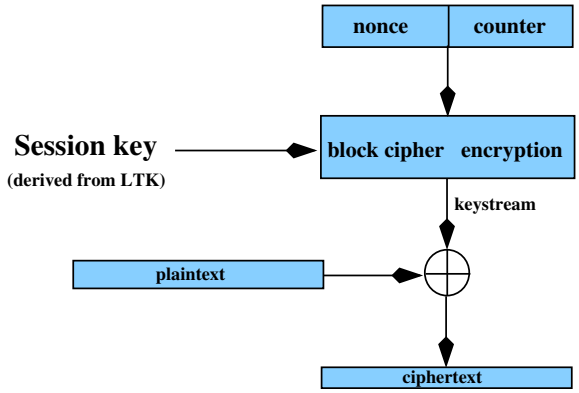


Figure 2.11: BLE PDU encryption, adopted from [Sta01]

by applying AES-CCM to Data Channel PDU with a non-zero length Payload. Then 24 bit CRC is appended on all packets for the purpose of verifying transmission errors [Gro13j]. All the mechanisms ensure the maximum robustness for BLE against malicious tampering and interference.

AES-CCM algorithm utilizes combination of nonce and counter to assure of uniqueness of keystream for each PDU. Encryption of PDU is shown in figure 2.11, the encryption blocks encrypt combination of a nonce and a counter with a session key derived from the LTK to provide the keystream which is used to encrypt the payload. The nonce will be deployed prior to the session and it is same for the whole session. The local counter will be increased by one after encrypting a PDU. So it is similar with a one-time pad stream encryption with higher security level. It means different PDU are encypted by different keystreams.

As to decryption in the peer BLE device, the same operation will be done. The peer device will encrypt combination of the nonce and counter with the negotiated session key to output the keystream. And then the keystream will be Bitwise Exclusive OR (XOR) with the received ciphertext. As encryption will consume less computational resources compared with decryption, the way of decryption in BLE device gives good performance to the whole BLE security system.

It is well known that encryption does not protect against malicious modification of PDU. Thus, the counter-mode CBC-MAC are used to calculate the Message Integrity Code (MIC) for integrity protection. The rules of CBC-MAC to construct MIC is Ci = E_K(Ci-1 $\oplus$ mi). If plaintext is n blocks long, then the last 64 bits is used as MIC. However, CBC-MAC is only secure for a fixed-length message. For variable length messages, the following mechanism will be used : MIC = CBC-MAC E_{K1}(CBC-MACE_K(variable-length message)). It means that final MIC is obtained

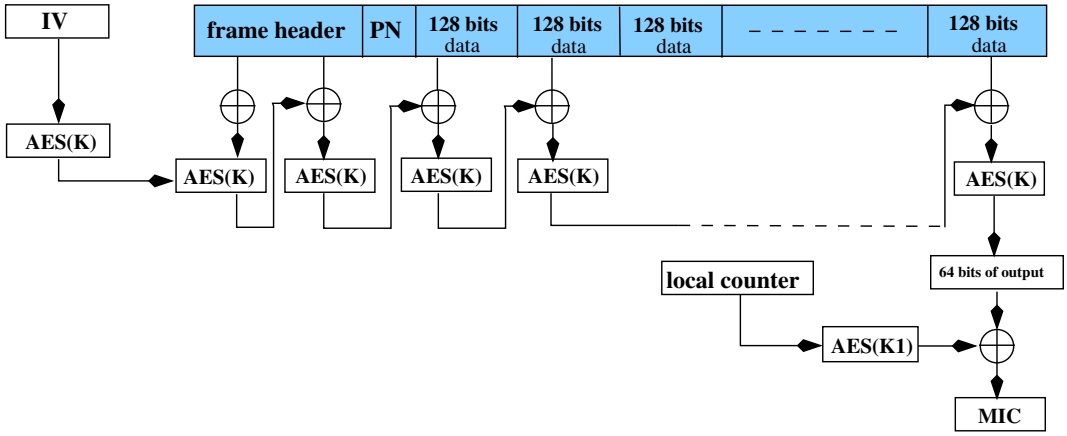


Figure 2.12: BLE PDU integrity protection, adopted from [Sta01]

by encrypting the output of CBC-MAC value with a new key K1. In terms of the research of CBC-MAC [KI04], the mechanism of MIC is proved secure if K and K1 are two random keys. The MIC generation is shown in figure 2.12.

Chapter 3
Related work

In this chapter, firstly existing privacy features in Bluetooth core specification 4.1 will be described and analyzed. In connection mode and procedures, the existing privacy features are used to reduce possibility of a BLE device being tracked. It is very relevant to this project and within the scope of this research. Even though the objective of this project is about connectionless advertisement, the existing privacy features still give us basic understanding about privacy concern and handling mechanism.

In following, next section will analyze the challenges to directly apply the existing privacy features to BLE connectionless advertisement.

3.1 Existing BLE privacy features RPA and NRPA

In order to reconnect to a known BLE device and lower the risk of device being tracked, privacy features have been defined in Bluetooth core specification 4.1. The privacy features will change the device address on a frequent basis over a period of time, such that adversary can not connect a static BLE address to a specific device. However, the private features are not used in connectionless GAP mode, it is only used in connection mode or during connection procedures [Gro13d].

3.1.1 Private address

The existing privacy features are only supported in connection mode and procedures. It mainly focuses on address protection by using private address over the radio channels. The private address is changed on a frequent basis and can be classified into two types: NRPA and RPA [Gro13d].

In connection mode, a BLE device can use NRPA or RPA to reconnect to a known device. NRPA and RPA looks random and is unpredictable, so it is impossible for

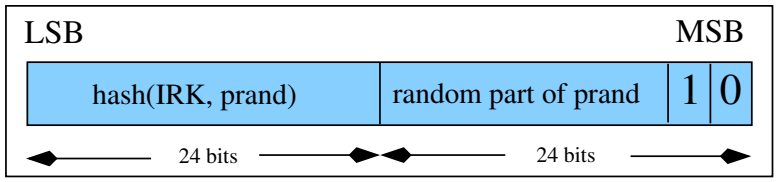


Figure 3.1: BLE RPA format, adopted from [Gro13e]

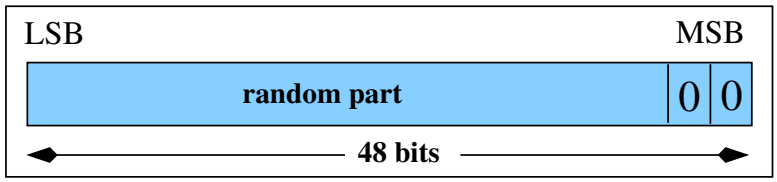


Figure 3.2: BLE NRPA format, adopted from [Gro13e]

adversary to listen on the radio channels and link the random address to a specific BLE device.

RPA

In Bluetooth core specification 4.1, the host of BLE device can generate 48-bit RPA with the IRK and a 24-bit random number known as prand. After receiving the generated RPA, the peer device will do exhaustive search in bond database for a specific IRK. The IRK is used to identify the device which generated the RPA. The rule to generate a RPA is shown by RPA = hash(IRK, prand) || prand. The format of RPA is illustrated by figure 3.1.

NRPA

In connection mode and procedure, a BLE device can follow a specific rule to generate a 48-bit NRPA and distribute to a peer device. Next time, the peer device can use the NRPA to reconnect to the known BLE device. The format of NRPA is illustrated by figure 3.2.

3.2 RPA and NRPA in BLE advertising context

As objective of this project is about privacy enhancement over the advertising channels, the existing concept of RPA and NRPA will be put into BLE advertising context and discussed. Prior to putting existing privacy features into BLE advertising context, a typical BLE advertising scenario will be introduced. In following, how to utilize the existing privacy features to protect BLE advertisement will be detailed.

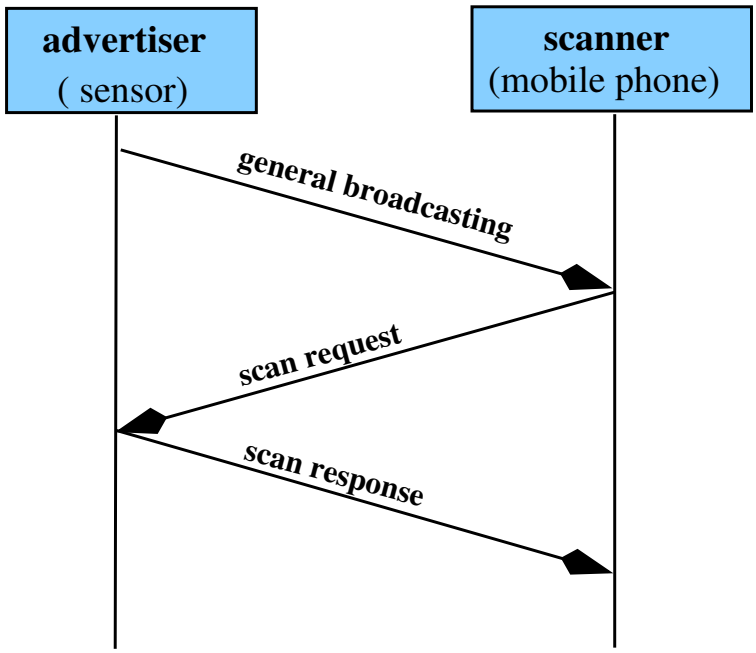


Figure 3.3: A BLE advertising scenario

3.2.1 A typical BLE advertising scenario

Figure 3.3 shows a typical BLE advertising scenario, two roles are involved: advertiser(sensor) and scanner(mobile phone). The first message in the figure is that sensor broadcasts general information over a BLE advertising channel. If the scanner wants more service data from the advertiser, the second message in the figure are sent out by the scanner. Lastly, the advertiser answers the scanner by a scan response.

3.2.2 Usage of RPA and NRPA in BLE advertisement

This section will make use of the advertising scenario of figure 3.3 to analyze the usage of RPA and NRPA. The prerequisite of analysis and discussion is there must be pre-stored bond database in both the advertiser and scanner.

- Privacy protection within BLE advertisement by RPA.
    1. Following the rule in figure 3.1, the host of advertiser generates a RPA as advertiser's device address.
    2. Then the advertiser can use the generated RPA to broadcast information to the scanner over the advertising channels.

3. After the host of scanner receives a advertising message with the RPA, it will do exhaustive search to retrieve its local bond database for the IRKa of the advertiser.
4. As soon as the scanner gets the IRKa, it can link the RPA with the advertiser.

- Privacy protection within BLE advertisement by NRPA.
  1. The host of scanner device generates a random NRPA and distributes it to the advertiser in a connection session.
  2. The advertiser stores the NRPA from the scanner in the local bond database.
  3. For a connectionless advertising session, the advertiser can use the stored NRPA as its random address to advertise message to the scanner.
  4. After the host of scanner receives an advertising message with the NRPA, it will resolve the NRPA based on its local bond, and link the NRPA with the advertiser.

In the same way, the privacy of advertising message from the scanner to the advertiser can also be protected by RPA and NRPA. The scanner can generate a RPA by itself and send advertising message with its RPA to the advertiser. As soon as receiving the message with a RPA, the advertiser can resolve the RPA by exhaustive searching IRKs of the scanner. As to NRPA protection for the scanner, the advertiser can also distribute a NRPA to the scanner in a connection session, the scanner can use the distributed NRPA to communicate with the advertiser during advertisement.

3.3 BLE privacy challenges

In terms of study in section 3.1.1, the existing privacy protection is on the basis of BLE private address. The usage of private address does reduce the possibility of device being tracked to some extent. However, device privacy still faces some challenges over advertising channels. The following lists the privacy challenges within BLE advertising channels under the protection of existing privacy features of RPA and NRPA.

- The adversary can impersonate as the legitimate advertiser, replay the general broadcasting message to the scanner, and elicit the scanner to send scan request. The consequence is that the adversary can observe the advertising channels and track the scanner in terms of responding location.
- The adversary can impersonate as the legitimate scanner, replay the scan request to elicit answer from the advertiser, and track the advertiser.

- service data within advertisement is plaintext. The adversary can also track device by means of linking static service data with a specific BLE device.
- The adversary can do brute-force attacking by unlimitedly replaying the captured advertisement. In this way, he can consume bandwidth and power of both the advertiser and the scanner unlimitedly.

The study of the Bluetooth core specification 4.1 conducted in this project has given background knowledge, current research process, and privacy challenges in this research area. Knowledge accumulated in this chapter is therefore used as the basis of system design, with the intention to solve the challenges mentioned in section 3.3 and enhance the BLE privacy over the advertising channels.

Chapter 4
System design

Following the project objectives mentioned in section 1.3, this chapter confines the overall system design to a complete privacy enhancement protocol over BLE advertising channels. Section 4.1 defines privacy violation models based on the communication analysis. Then section 4.2 determines functional requirements. In following, the complete system design is proposed in section 4.3 and 4.4 on the basis of the functional requirements. Lastly, optional functions for various BLE advertising scenarios are given and explained in section 4.5.

4.1 Privacy violation models

In this section, prior to defining violation models, de facto communication over BLE advertising channels will be presented. Violation models will be followed in the light of analyzing advertising channels and message interaction over the channels.

4.1.1 Advertising analysis

As mentioned in Bluetooth core specification 4.1, there are 3 fixed physical channels for BLE advertisement. Various advertising sessions can be conducted over BLE advertising channels. In this project, a typical advertising session will be used as example to define violation models. However, the violation models can be applied to different kinds of advertising sessions.

A typical advertising session can be composed of 3 message exchanges, as illustrated in figure 4.1. Firstly, the BLE advertiser device broadcasts service data over any of the fixed physical advertising channels. After getting the broadcast message, the scanner device can send out a scan request for further detail information from the advertiser device over the same advertising channel. Then the advertiser device responds to the scan request by a scan response via the same advertising channel.

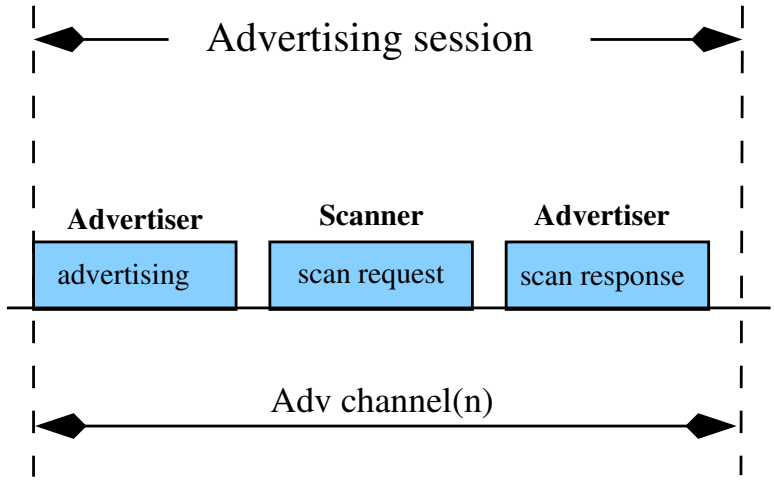


Figure 4.1: A typical BLE advertising session, adopted from [Gro13k]

Prior to defining violation models, there are assumptions described which are relevant to the threats imposed to advertising channels by adversary. Advertising messages will be exchanged via open radio channels and no connection is established between the advertiser and scanner. Thus, in the BLE privacy violation models we assume that adversary control all communication over physical advertising channels between advertiser and scanner.

From a security perspective, it means that the adversary can not only observe all the messages over the channels. But also he can replay, alter, forge advertising messages and impersonate to be a legitimate communicating device.

Furthermore, advertising message may cause the leakage of the device identities, position information and secret service data. The privacy leakage makes it possible for the adversary to track the BLE devices with the help of the device identities and other service data, e.g the adversary is able to track a mobile phone(scanner) or a pacemaker(advertiser) in the light of device addresses within advertisement.

4.1.2 Violation models

In violation models, a distinction can be made between typical roles of BLE advertisement: advertiser privacy violation and scanner privacy violation.

In the former case there are two different approaches for the adversary to violate BLE advertiser's privacy: privacy disclosure and device tracking.

- Disclosing advertiser's privacy by the following way:

  1. The adversary passively listens to advertising channels to capture messages with address and static service data of the advertiser. Then he analyzes the captured messages and discloses the privacy of the advertiser, but not much more than this.

- Tracking the advertiser by the following ways:

  1. The adversary passively observes the channels for address and static service data, connects the observed information with a specific BLE advertiser device, and then tracks the advertiser.
  2. The adversary actively pretends to be a legitimate scanner and replays a captured scan request, elicits a scan response from the advertiser. Then the adversary is able to track the responding advertiser by responding location.

In the latter case, there are also two ways to violate BLE scanner's privacy: privacy disclosure and device tracking.

- Disclosing scanner's privacy by the following way:

  1. The adversary passively listens to the advertising channels to capture messages with address and static service data of the scanner. Then he analyzes the captured messages and discloses the privacy of the scanner, but not much more than this.

- Tracking scanner by the following ways:

  1. The adversary passively observes the channels for address and static service data, connects the observed information with a specific BLE scanner device, and then tracks the scanner.
  2. The adversary with its stronger transmitter can do more than just observing all messages over the physical advertising channels. He can broadcast the observed advertisement and pretend to be a legitimate advertiser in its immediate vicinity, elicit the scanner to send a scan request. As soon as the scanner device sends scan request to the fake advertiser, the adversary need not crack content of the scan request and is able to track the scanner in terms of responding location.

4.2 Functional requirements

As the violation models mentioned in section 4.1.2, all the data communication over the advertising channels in response will have risks of privacy leakage and device tracking. The adversary can disclose the privacy of BLE devices by passive observation. He can also track the devices by passive observation or actively replaying the captured message to elicit response from the victim device.

In this project, it is desired to do two types of privacy protection: protection against passively attacking, mainly focuses on advertisement confidentiality which can be applied to address and static service data; protection against actively attacking, mainly focuses on countermeasures against BLE device tracking.

It is expected, that existing countermeasures can be used to protect device privacy and resist against device tracking, where unfortunately there is no such mechanism in Bluetooth core specification 4.1. This leads to the necessity of functions which can be applied to protect sensitive information and to resist against device tracking over the advertising channels. Thus, the main functional requirements to BLE privacy enhancement can be specified as:

- BLE advertiser's privacy protection from the message confidentiality perspective.
- Keep BLE advertiser from being tracked, focus on the avoidance of static ciphertext and countermeasures against replay attacking.
- BLE scanner's privacy protection from the message confidentiality perspective.
- Keep BLE scanner from being tracked, focus on the avoidance of static ciphertext and countermeasures against replay attacking.

The following section will detail how to solve and mitigate various privacy violations in the light of functional requirements in section 4.2.

4.3 System architecture

This section will elaborate the system design from both stack perspective in overview and separate sub-functional perspectives. Due to various BLE advertising scenarios, some optional functions will be described later on.

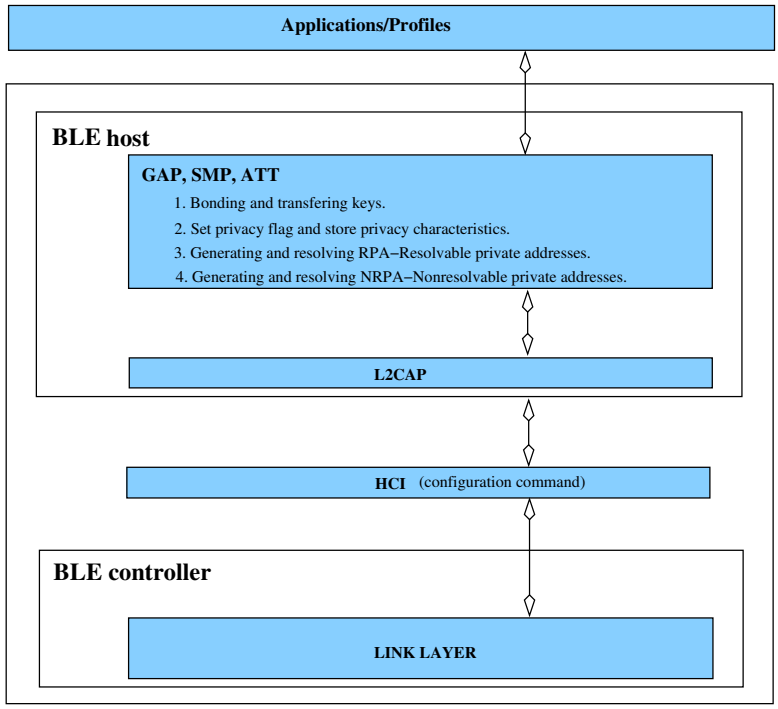


Figure 4.2: Existing BLE privacy software stack in Bluetooth core specification 4.1, adopted from [Gro13m]

4.3.1 Protocol stack design

Existing privacy stack

In Bluetooth core specification 4.1, current privacy solution only involves BLE host to protect device address, as illustrated in Figure 4.2. BLE host is responsible for key distribution, privacy flag setting, and generating and resolving RPA and NRPA.

Enhanced privacy stack

The enhanced privacy protocol consists of an application interface and a background service. The application provides interface to end-user who can enable or disable the enhanced privacy protocol and select privacy level among different supported privacy options. As to the background service, it involve two important parts: BLE host and BLE controller, as illustrated in figure 4.3.

In the privacy enhancement protocol BLE host is responsible for privacy flag setting, key distribution, and nonce deployment. The BLE controller will execute all the concrete tasks for privacy protection, e.g. encryption and decryption of

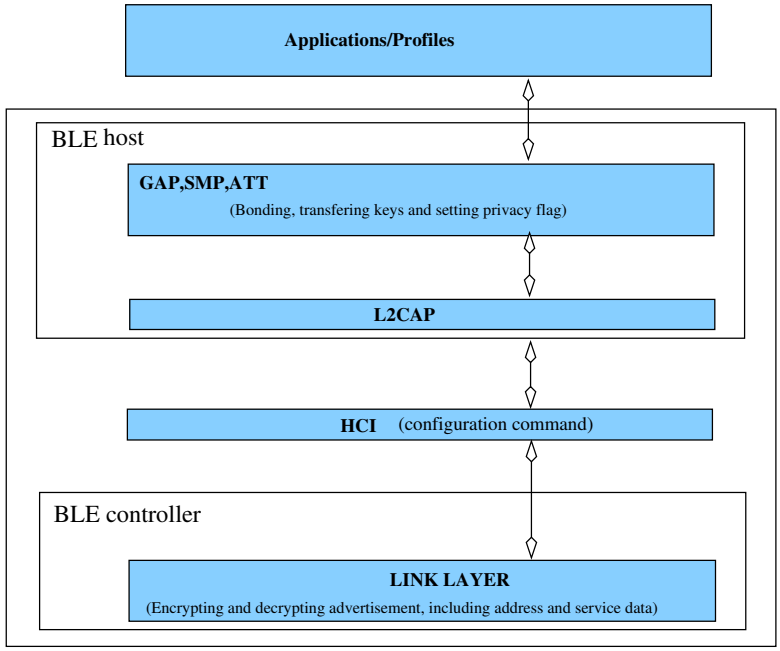


Figure 4.3: BLE software stack of enhanced privacy feature, adopted from [Gro13m]

BLE advertisement, generating and resolving private address. The Host Controller Interface (HCI) is responsible for connecting BLE host with controller.

In more detail, the privacy application will present a list of privacy options to end-user when the BLE device is pairing with a peer device. If the enhanced privacy is selected and supported by both communicating parties, all the following messages over BLE advertising channels will be protected by the enhanced privacy protocol. It means that the new protocol will protect all BLE advertising messages including device addresses and service data. However, if either the BLE advertiser or the BLE scanner stops the enhanced privacy protocol during advertisement, there will be a warning message to end-user through the application interface and end-user need to select other privacy options which can be supported by both the BLE advertiser and scanner.

Difference between the existing and enhanced stack

The main difference of software stack between the enhanced privacy protocol and existing privacy features of Bluetooth core specification 4.1, is significant improvement for energy efficiency. The function of generating and resolving ciphertext is moved from BLE host to controller. BLE host will be not waken up for generating and

resolving ciphertext.

Moreover, the existing privacy protection in Bluetooth core specification 4.1 mainly focuses on the protection of device address. The enhanced privacy protocol is able to protect the whole advertising message consisting of device address and service data.

4.3.2 Processing flow

For the enhanced privacy feature, figure 4.4 demonstrates a flow chart of a complete session over BLE advertising channels. During pairing with a peer device, a list of possible privacy options presents to end-user who can choose a privacy protocol in terms of communication requirements. If the peer device does not support a specific privacy option, they will be warning of privacy not supported to the end-user. After a successful pairing procedure, the BLE device is set to support a specific privacy option. The reason for having different privacy options is to accommodate to various peer devices with different privacy supports.

As shown in the figure 4.4, the pre-shared bidirectional symmetric keys IRKa and IRKs, nonce Ra and Rs are used to resist on privacy disclosure and device tracking over BLE advertising channels as mentioned in section 4.1.2. Nonce Ra assures freshness of an advertising session. 3-way handshake protocol is utilized to deploy the nonce Rs from the scanner to the advertiser. Then Rs is used to initialize the local counters RX and TX on both the advertiser and scanner.

The first message in the 3-way handshake mechanism is a challenge Ra concatenated with its encryption by IRKa from the advertiser. Prior to handling the challenge from the advertiser, the scanner will do exhaustive search to find correct IRKa for the advertiser and then deploy a nonce Rs to the advertiser with the protection of IRKs. The third message is confirmation to a successful deployment of the nonce Rs. As soon as the nonce Rs is deployed successfully, the local counters RX and TX of the advertiser and scanner devices can be initialized to be the nonce Rs. Then all the following advertisement between the scanner and advertiser will be protected by both the local counters RX and TX and the pre-shared keys IRKa and IRKs. In default, a BLE device supporting enhanced privacy protocol will encrypt both address and service data for any type of advertising messages.

4.3.3 The message structure

In most cases, it is quite realistic to build a new solution upon existing mechanisms. The same to this project, the privacy enhancement protocol will also utilize the existing BLE advertisement packet format. The overall packet format consists the same fields as Bluetooth core specification 4.1, which consists of four fields, preamble,

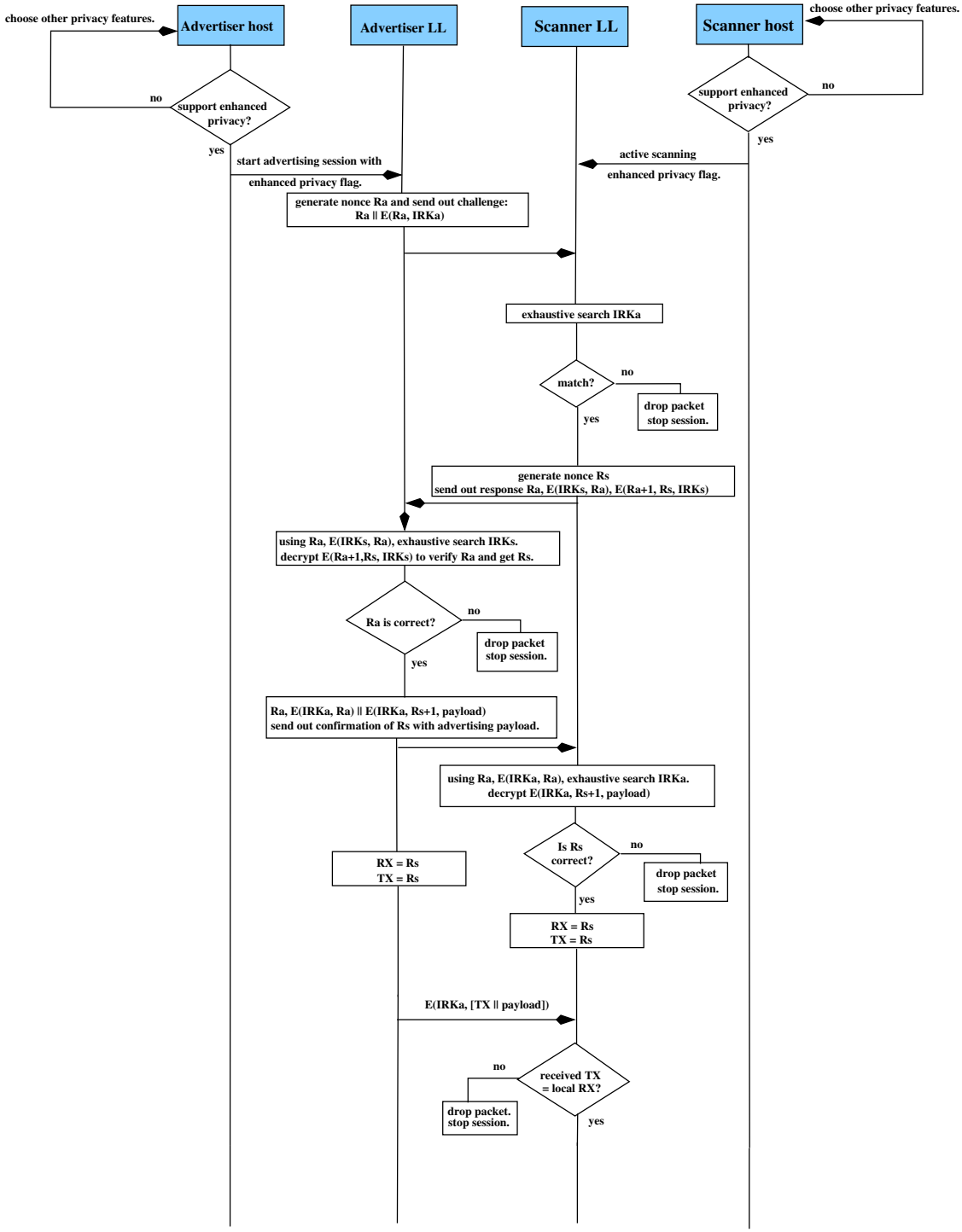


Figure 4.4: BLE devices interaction of enhanced privacy protocol

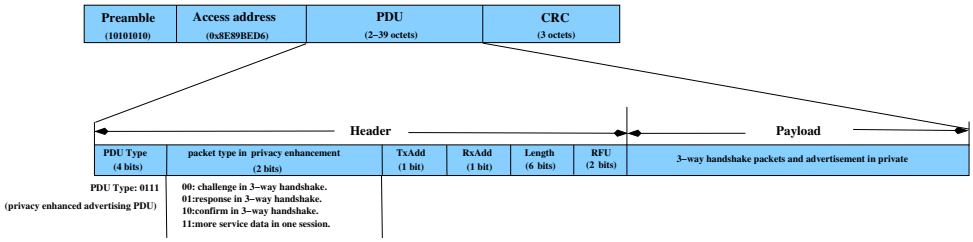


Figure 4.5: Packet structure of enhanced privacy feature, adopted from [Gro13g]

access address, PDU, and CRC. The fields preamble and access address have fixed values for all the advertising messages. So, the focus of this project will be put on the field PDU, which is divided into header and payload fields.

There are some modification in this enhanced privacy protocol compared to the existing packet format. First, a new PDU type in the header is defined in this project to indicate if the privacy enhancement feature is supported. Second, two bits Reserved for Future Use (RFU) in the header are used to indicate different packet types for the BLE privacy enhancement. The new packet types consist of 3-way handshake mechanism and advertising payload under the protection of privacy enhancement. As soon as 3-way handshake protocol is conducted successfully to deploy a nonce Rs, all the subsequent advertisement will then be put in the new packet format under the protection of both the local counters and pre-shared IRK. Figure 4.5 illustrates packet structure of advertisement in the privacy enhancement protocol.

Table 4.1 details the value and description of different fields in the header of the PDU. The enhanced PDU type is defined to be 0111 which indicates to support the privacy enhancement protocol. The fields of packet type in the privacy enhancement will be used to decide the specific packet types in the enhancement protocol, including 3-way handshake and more service data.(* [1] is defined by this project)

[1] new added flags in the PDU header

Field name	Value	Definition
PDU Type	0111	PRIVACY_ENHANCEMENT_PDU *
packet type in privacy enhancement	00	challenge in 3-way handshake *
	01	response in 3-way handshake *
	10	confirm in 3-way handshake *
	11	more service data in one session *
TxAdd	0	public address
	1	private address
RxAdd	0	public address
	1	private address
Length	sizeof(payload)	octets length of PDU payload
RFU	00	no definition

Table 4.1: Header definition of enhanced advertisement PDU, modified from [Gro13g].

4.4 Functional design

4.4.1 3-way handshake for counter deploy

There is no replay prevention over advertising channels in Bluetooth core specification 4.1. Normally a solution with counter or timestamp can be used against replay attacking. In terms of the specification all BLE advertisement will be conducted in connectionless state. It means that there is no central clock for BLE devices to be synchronized to. Thus, in this project a counter solution is selected against replay attacking.

This project has proposed a solution with two counters. Both the advertiser and scanner devices have two counters. One counter is for RX and another is for TX. Figure 4.6 illustrates the 3-way handshake protocol designed by this project, which is used to deploy a nonce Rs from the scanner to advertiser device. After successful deployment of the nonce Rs, Rs will be used to initialize the counters RX and TX of the advertiser and scanner.

As to detail in figure 4.6, the first message is a challenge Ra with its encryption E(IRKa, Ra). The nonce Ra is used to assure freshness of the advertising session. In principle, It should be always a new Ra to identify a new advertising session and Ra should never be repeated.

The second message "Ra, E(IRKs, Ra), E(IRKs, Ra+1, Rs)" is used by the

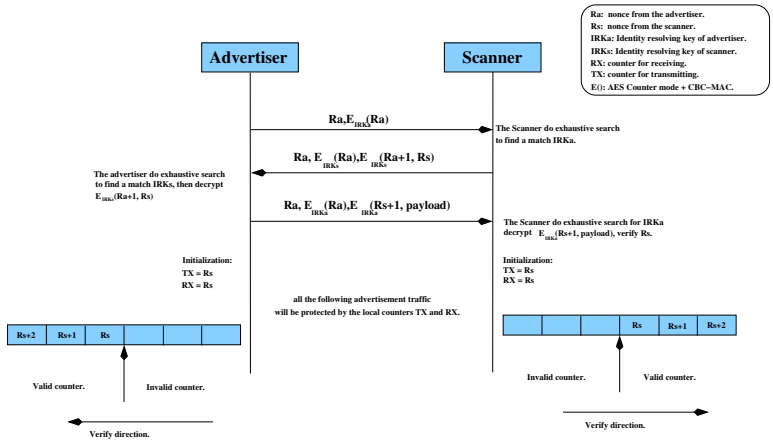


Figure 4.6: 3-way handshake protocol to deploy nonce Rs

scanner to deploy a nonce Rs to the advertiser. The reason to use "Ra, E(IRKs, Ra)" in the second message is that there is no address information for the advertiser to identify the scanner in this phase. The advertiser will utilize "Ra, E(IRKs, Ra)" to do exhaustive search for the correct IRKs, then decrypt "E(IRKs, Ra+1, Rs)" to verify the Ra and extract the nonce Rs. The third message "Ra, E(IRKa, Ra), E(IRKa, Rs+1, payload)" is used by the advertiser to confirm the nonce Rs to the scanner. The same as the second message, the scanner will firstly do exhaustive search for the correct IRKa in terms of Ra, E(IRKa, Ra) and then decrypt the cipher "E(IRKa, Rs+1, payload)" to verify the nonce Rs. If Rs passes the verification, the scanner can extract payload including service data from the advertiser.

Following a successful deployment of the nonce Rs, the TX and RX counters on both sides are initialized to be Rs. The subsequent advertisement will be protected by the counters from replay attacking and device tracking. As to counter store and verification for advertiser and scanner, section 4.4.2 has detailed description.

Security provided by nonce Ra and Rs

In order to utilize existing RPA mechanism which can conduct exhaustive search for correct pre-shared key of peer devices based on a plaintext and its corresponding ciphertext, 24 bits has been chosen to be length of the nonce Ra and Rs in this project. As mentioned before, Ra assures of freshness of an advertising session and every new advertising session should use a new Ra. 24 bits Ra and Rs in this project are able to make sure the security of the 3-way handshake protocol in BLE advertising context.

Prior to analysis security of the nonce Ra and Rs, typical non-connectable BLE

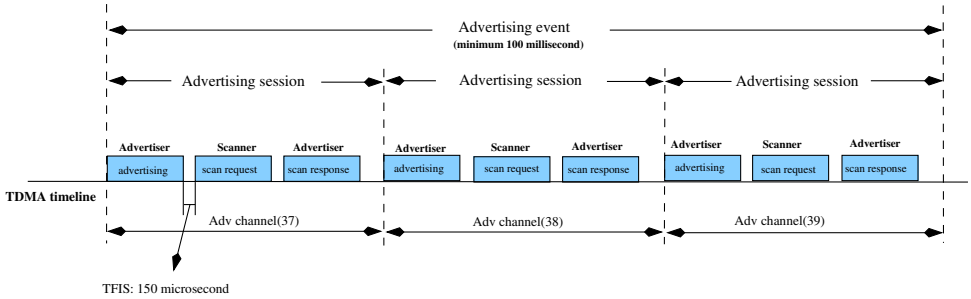


Figure 4.7: BLE advertising interval, adopted from [Gro13k]

advertising interval shown in figure 4.7 will be introduced. The advertising session in figure 4.7 is illustrated by figure 3.3. As to connectable advertising event, there will be different advertising interval (20 millisecond) and advertising transactions, which means that relevant parameters of following analysis shall be adjusted for connectable advertising event. However, the mechanism how to evaluate security of the nonce Ra and Rs is exactly same for non-connectable advertising event and connectable advertising event. Influence of the nonces to non-connectable advertising event will be elaborated in detail. The same mechanism can be applied to connectable advertising event.

In figure 4.7, there are 3 fixed physical BLE advertising channels, 37, 38 and 39. An advertising event maximally consists of 3 advertising sessions belonged to different advertising physical channels. The same advertising session must be conducted in the same physical channels. As to interval of advertisement, the minimum interval of an advertising event is 100 millisecond. In extreme case there will be 3 advertising sessions during 100 millisecond. Put 24 bits nonce Ra into the extreme advertising case, Ra will be repeated around 155 hours. However, combination of 24 bits Ra and Rs makes it infeasible for an adversary to find vulnerability to attack the communicating BLE devices. The reasons are elaborated by following description.

The adversary can eavesdrop a whole advertising session of 3-way handshake protocol over an advertising channel. Assuming that the advertiser and scanner conduct advertising session continuously up to 155 hours, the same challenge Ra, E(IRKa, Ra) shows up again within an advertising session. Then the adversary can replay the eavesdropped second message "Ra, E(IRKs, Ra), E(IRKs, Ra+1, Rs)". The legitimate advertiser will answer the replayed message by "Ra, E(IRKa, Ra), E(IRKa, Rs+1, payload)". In advertiser's answer, payload is normally dynamic service data and the pre-shared key IRKa is kept from the adversary. It is infeasible for the adversary to crack it and make further attack. Furthermore, BLE communication is in the way of ad-hoc and it is not rare situation that the peer device moves out of

range. In most cases it is impractical for the adversary to track continuous advertising session up to 155 hours and replay the eavesdropped message in a correct time point to the legitimate advertiser. Thus, 24 bits Ra will not lower the security of the whole system.

On the other hand, the adversary can also replay the first message "Ra, E(IRKa, Ra)", then the legitimate scanner will newly generate a 24-bit nonce Rs, encrypt "(Ra+1, Rs)" by IRKs, and send to the advertiser. Due to no knowledge of IRKs and dynamic Rs, the adversary can not do further attack. Therefore, combination of 24 bits Ra and 24 bits Rs is safe in BLE advertising context.

4.4.2 Counter store and verification

In this project, 24-bit has been used as counter length. The reason is that the 24-bit counter can easily accommodate to BLE advertisement in ad-hoc and connectionless mode. The two counters RX and TX are utilized on both the advertiser and scanner devices. Combining the characteristics of BLE advertising communication with processing speed, this project has used Random Access Memories (RAM) to store the counters TX and RX of both advertiser and scanner. The disadvantage is that the counters will disappear once power off. The advantage is that it is faster to update the counters during advertisement compared with flash memory. Considering that BLE advertisement is connectionless communication, it is not necessary to store the counters from the previous session and every new advertising session will re-negotiate counter. Therefore, the choice of RAM memory accommodates to BLE advertisement and project requirements.

Here the counter verification will only be discussed in a perfect BLE advertising environment. Section 4.4.3 will discuss how to verify counter in an advertising environment with the potential risk of replay attack and packet loss. This project defines the counters TX and RX as two global variables in the RAM of BLE device. Real-time counter updating will be reflected by updating of variables. The variable TX will be increased by one whenever a packet is sent out. Prior to handling a packet, the variable RX will be used to verify the uniqueness of the received packet. Counter TX in the received packet which is equal to local counter RX of the receiving device is considered to be a valid counter. The local counter RX will be increased by one after receiving a valid packet. However, if there is an invalid counter in the received packet, the BLE device in receiving state will stop the advertising session immediately.

4.4.3 Handle of out of synchronization

The two copies of the counters in the advertising devices can become unsynchronized due to open radio communication channels. It might be replay attack from adversary

or packet loss due to radio interference. The BLE advertisement is prone to influence from other radio interference. Moreover, Counter out of synchronization is not rare case when an advertising device sends message to a peer device which is out of range. In order to make the BLE advertising communication as secure as possible this project has designed a mechanism to handle the counters out of synchronization.

Characteristics of BLE radio system:

- In order to make solution more clear, the characteristics of BLE radio system will be given firstly. The radio system of BLE is half-duplex system, which means a BLE device will transfer between sending and receiving states and can not do both at the same time. If a BLE device is in receiving state, its radio system is strictly controlled by a timer to receive a packet over BLE advertising channels. In terms of Bluetooth core specification, there is a fixed duration Time Inter Frame Space (TIFS) 150 microsecond between messages within an advertising session [Gro13k]. It means that if a BLE device is in receiving state, its radio system will be turn on to receive packet every TIFS (150 microsecond). If a packet arrives in another time point, the packet will be ignored.

Solution of out of synchronization:

- Taking the BLE radio and advertising characteristics into consideration, the solution in this project on how to verify counter in an BLE advertising environment with the potential risk of replay attack and packet loss will be elaborated. As soon as the nonce Rs is deployed from the scanner to advertiser, the local counters RX and TX of the advertiser and scanner will be initialized to be Rs. Counter TX is used to send out a packet and counter RX is used to verify the received packet. As how to prevent replay attack, if the counter TX in the arrived packet is less than the local counter RX in the receiving device, the packet will be treated as a replayed packet and the receiving device will stop the advertising session immediately. On the other hand, in order to accommodate packet loss, after the period TIFS if the BLE device in receiving state has not received a packet, the device will automatically update counter RX by increasing one. However, this accommodation will be limited to some extent, without receiving a packet the counter RX of the device in receiving state will be updated based on TIFS up to a pre-defined times (y). After (y) times, if the device in receiving state still has not received a packet with counter TX which is consistent with local counter RX, the device will stop the advertising session directly.

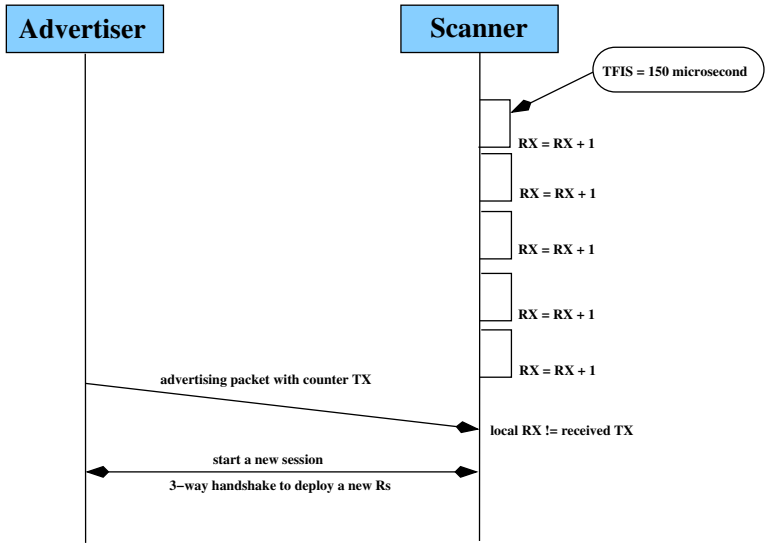


Figure 4.8: Handle of counter out-of-synchronization within BLE advertisement session

Figure 4.8 illustrates if packets from the advertiser are lost less than (x<y) times, the scanner will automatically increase its local counter RX by one. As soon as packet loss is up to (y) times, the scanner will expect to receive a packet with counter which is consistent with the local counter RX of the scanner. If not, the scanner will stop the adverting session. A new 3-way handshake protocol will be invoked to deploy a new Rs. The mechanism to handle counter out-of-synchronization can also be applied to another advertising direction, it means that the advertiser is in the receiving state.

4.4.4 Countermeasure against device tracking

Normally, there are two typical ways to track a BLE device. One is passively listening the communication channels to figure out the connection between the observed static data and specific devices. Another way of device tracking is that the adversary can actively replay the eavesdropped data over the BLE advertising channels, elicit a response from the victim device, and get the position to track it.

To put actively device tracking in the advertising scenario illustrated by figure 4.1, the adversary can impersonate as a scanner device and replay a scan request to elicit a scan response from the advertiser. Then the adversary can track the victim advertiser which responds to the replayed message immediately. Vice versa, the adversary can also impersonate as the advertiser to elicit response from the scanner.

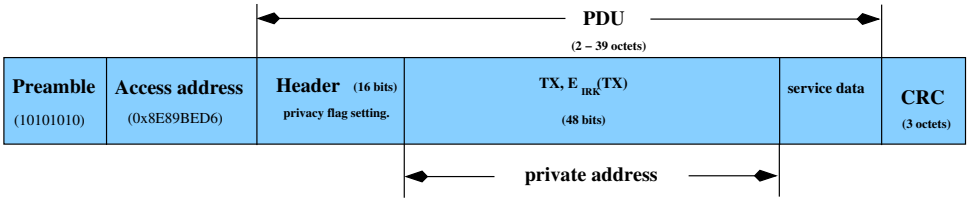


Figure 4.9: Enhanced BLE RPA format, modified from [Gro13e]

In this project, encryption of counter and advertisement message avoids passively device tracking by static address and service data. As to device tracking by eliciting response from the communicating device, there is no such risk for the advertiser device. As figure 4.6 shows that the advertiser will either use dynamic challenge or be protected by both counter and corresponding IRK. Prior to responding every message, the advertiser will assure authenticity of the communicating party. Tracking a scanner device by eliciting a response will be detailed in section 4.5.2.

4.5 Optional protection

4.5.1 Protection only for device address

In some advertising scenarios, it is not necessary to protect data area of PDU. As PDU type "connect request" shown in figure 2.9, it is no need to encrypt the field "connect request data". Thus, the project also proposes how to only protect the address field of the advertising device.

Enhance solution

When enhanced privacy protocol is selected in BLE device, the LL of the BLE device will generate a RPA with the IRK and a 24-bit local counter TX which is updating for every advertising message within the advertising session. The local counter TX is initialized by the nonce Rs, which is deployed by 3-way handshake protocol from the scanner to the advertiser. The rule of generating BLE private address in the privacy enhancement protocol can be RPA = E(IRK, TX) || TX. The format of RPA is illustrated in the figure 4.9.

Difference between existing and enhanced solution

Compared with existing private address feature explained in section 3.1.1, the main difference of private address between the enhanced privacy and existing privacy feature of Bluetooth core specification 4.1, is enhancement against replay attacking. Local counter TX of BLE devices can be used to detect replayed message over the BLE advertising channels. Due to local counter TX is updating for every advertising

message within the advertising session, the address field in the advertising message will be dynamic over the advertising channels. It is not feasible for the adversary to replay the captured advertisement or track the BLE device by the address field.

4.5.2 Mitigation for scanner tracking

As mentioned in section 4.4.4, the discussion of this section will be limited to scanner device tracking by eliciting a response from the scanner. For BLE advertising, in most cases sensor will play the role of advertiser to broadcast service data and mobile phone plays the role as scanner to get service data from a advertiser. If the adversary can track a mobile phone, he can probably get more private information of the phone and maybe track the person who owns the phone. Therefore, it is very important to do study in detail about scanner tracking.

Problem description:

In 3-way handshake protocol of the enhanced privacy, the first packet "Ra, $E_{IRKa}(Ra)$" is just a challenge and there is no deployed counter to prevent replay attacking yet in this phase. The adversary can eavesdrop it and replay it many times over advertising channels. The scanner will answer the replayed message by sending Ra, $E_{IRKs}(Ra)$, $E_{IRKs}(Ra+1, Rs)$. The adversary just observes the advertising channels, figures out which device has responded to the replayed challenge, and then tracks it by responding location. In this way, the adversary can track the scanner device without cracking the encrypted message $E_{IRKs}(Ra+1, Rs)$.

Mitigation:

To lower the possibility of the scanner being tracked, the bonding procedure can deploy a pre-defined number of times (z). It controls how many challenges in the form of "Ra, $E_{IRKa}(Ra)$" must be sent to the scanner in a specific order continuously before the scanner deploys a nonce Rs to the advertiser. In this way, prior to sending the message "Ra, $E_{IRKs}(Ra)$, $E_{IRKs}(Ra+1, Rs)$" the scanner need to check multiple challenges from the advertiser and those challenges must be in a pre-defined order format. Figure 4.10 illustrates schematics of the mitigation for scanner tracking.

4.5.3 Mitigation of DoS

In a well-architected system, it is not reasonable to use bandwidth and power capability for presenting replayed and forged message and determine if they are valid. Unlimited verification for message authenticity at both advertiser and scanner will have influence on power and radio system, which will lead to unpredictable consequences.

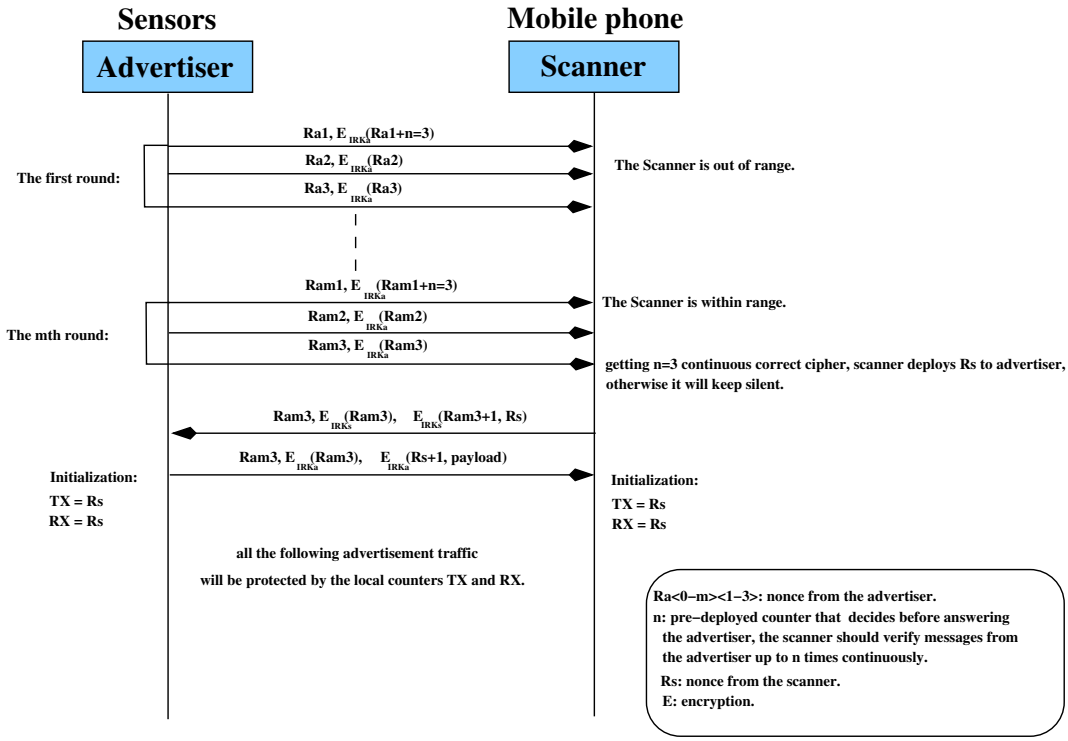


Figure 4.10: Mitigation for scanner tracking

Problem description:

As mentioned in section 4.3, the BLE privacy enhancement protocol makes use of a 3-way handshake protocol to deploy a nonce Rs from the scanner to advertiser. The first message in 3-way handshake protocol is a nonce Ra concatenated by its encryption, so it is possible for an adversary to capture the first message and replay it unlimitedly over the advertising channels. As soon as receiving the replayed message, the scanner will check the validity of the replayed message and maybe respond to it. Even though the adversary has no knowledge of pre-shared IRKs and can not disclose the message from the scanner, the adversary can dramatically consume bandwidth and power capability of the scanner device. Thus, there is a potential risk of DoS for the scanner device which may be not able to handle other legitimate advertisers as it should be.

Mitigation:

Unlimited replay will cause high workload and unnecessary power consumption. Low power usage is critical in a BLE device and therefore a solution is proposed

to mitigate the influence from unlimited replay. If the scanner device receives same challenge messages up to a pre-defined number of times (n) [2] within a fixed period, the radio system can choose to blackout for a short period in order to save power.

[2] n will be decided by experimental result.

Chapter 5
Simulation – modeling and protocol validation

Simulation can be defined as *the imitative representation of the functioning of one system by means of the functioning of another system* [Kha08]. It is a useful methodology when we want to capture all the threats adversary might pose to security protocol. Simulation model can be used to analyze and evaluate vulnerabilities of a security protocol. Normally it is conducted by establishing analytical models, verifying, and analyzing the model with respect to security requirements.

In this project, in addition to system design for the BLE privacy enhancement protocol over advertising channels, a protocol simulation based study is also performed. Prior to simulation, necessary assumptions will be made in order to conduct protocol analysis. Then, simulator is utilized to simulate communicating scenarios and verify the protocol in different granularities.

Firstly, this chapter gives an introduction for the simulation tool used in this project, and method how to establish and verify security model. Then following part covers protocol simulation and validation of the BLE privacy enhancement over advertising channels.

5.1 Choice of simulation tool - Scyther

The goal of protocol simulation is to *provide a methodology for the formal analysis and verification of abstract protocols. In particular, simulation requires a simulator with formal semantics, intuitive definitions of security properties, and efficient tool support* [CM12d]. The simulator should *allow us to formally model black-box security protocols, and define all possible behaviours of such protocols* [CM12d].

For the sake of this project, the attention is confined to security protocol verification. Scyther security protocol verification tool is selected in this project as simulator. The reasons for using Scyther are that it separates concerns of the protocol,

distinguishes protocol description from their dynamic behavior, and explicitly defines knowledge of adversary.

Support for formal definitions of both existing and new security properties with minimum efforts makes Scyther very attractive for using. Verification of security properties is original purpose of Scyther, combining with mechanism to customize security properties for requirements makes Scyther even more powerful.

5.2 Scyther introduction

Scyther is a tool for security analysis of protocols under perfect cryptography assumption. It presents simulation model by defining protocol behaviors and claiming security properties. Protocol is executed in the model explicitly by means of runs of role instances named agents and message interaction among the agents. Then claiming event is able to integrate security properties into the protocol.

During simulating Scyther binds values to agents and handles parallel execution of the security protocol. Claiming event is used to verify if the security protocol satisfies all the claimed security properties according to the messages they received from a local perspective on the state of the system. If there is vulnerability in the protocol and the role in the protocol does not fulfill the claimed security properties, threat model is presented to illustrate the weakness.

Scyther is also capable of modeling security protocols in black-box by defining all possible behaviors of an adversary [CM12d]. It means that there is already a perfect cryptographic scheme in the underlying system. The perfect cryptographic scheme is used as baseline to build up a secure protocol. The communicating message in the protocol are treated as just abstract terms, the adversary can gain the complete message by having a secret key or he learns nothing [CM12d].

The following part of this section will introduce the semantics, working mechanism, and ways of model establishment and verification of Scyther.

5.2.1 Scyther runs

Scyther can describe message communication among a set of roles by run which guide the interaction of actual agents in a system. When a protocol is executed, each agent can execute any role any number of times in parallel. We call each such a single execution of a role a run. In Scyther model, two runs executed by the same agent are independent and no shared variables. Roles can be instantiated multiple times in several runs [CM12c]. In order to instantiate a role we have to bind the role names to the names of actual agents, and we have to ensure that the freshly generated values are unique for each run.

5.2.2 Security properties

Scyther integrates the security properties into the protocol specification by means of claim events. The main idea behind claim events is locality: role instance agents have a local view on the system state in terms of the messages they receive. The protocol should guarantee that, the agent can be sure about some properties of the global state of the system in a local perspective [CM12e], e.g. a secret key is not in the adversary knowledge, or that a certain agent is active.

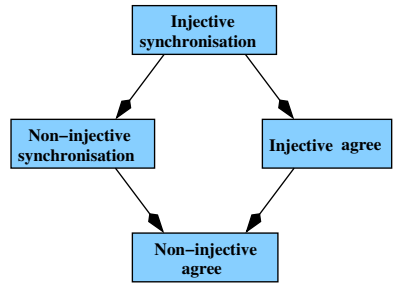


Figure 5.1: Security property hierarchy, adopted from [CM12e]

The following part of this section will present several important security properties of Scyther, which will be used in the BLE privacy enhancement protocol. Figure 5.1 depicts the hierarchy of authentication properties [CM12e].

- **Secrecy** : an adversary can not reveal information over an untrusted network.
- **Authentication** : a way to verify user's identity as a communication partner claims to be.
    1. **Aliveness** : the property has requirement for the communication partner, without putting restrictions on the contents of the communication.
    2. **Synchronisation** : the property has requirement for the communication contents which are indeed sent and received by the communication partners as specified by the protocol description.
    3. **Non-injective Synchronisation** : everything intended to happen in the protocol description happens in the trace. It makes sure that the protocol will be executed as expected if no adversary were present. Agents may communicate with the same agents in multiple runs and later the adversary may be able to replay packets from one session in another session. Thus, a protocol with non-injective synchronization is vulnerable to replay attacks.
    4. **Injective Synchronisation** : multiple protocol executions are performed as expected. There will be an injective mapping from claiming initiator runs to corresponding responder runs. More precisely, different instances of the initiator claim must correspond to different runs of the responder.
    5. **Message Agreement** : requires that the contents of the received packets will be exactly same with the sent packets as specified by the protocol after the execution of the protocol.

5.2.3 Threat model

Scyther assumes the adversary control all communication between the parties. The adversary can eavesdrop all the messages over the communication channels. He can also replay, reroute, and forge any messages. In a worse case, the adversary can compromise a valid agent and participate communication of the protocol on behalf of the valid agent [CM12b].

Scyther uses protocol assumption as baseline, analyzes security protocol, and evaluates if the protocol fulfills all the claimed security properties. If there is any vulnerability in the protocol, threat model will be used to demonstrate the message flow how the adversary crack the communication among the legitimate parties.

5.3 BLE privacy enhancement

In terms of system design in chapter 4, a protocol simulation will be conducted in the following section. Firstly, the assumption of the protocol simulation will be introduced as baseline of the whole simulation. Then protocol simulation, analysis and validation will be shown in detail.

5.3.1 Model assumptions

In the following, the assumptions are made in order to confine simulation and analysis of the BLE privacy enhancement protocol in a stable and reliable circumstance:

- The underlying cryptographic mechanism between BLE advertiser and scanner device is secure, e.g AES-CCM encryption and decryption. In reality this assumption is depending upon the number of real-time messages the adversary can eavesdrop over the radio channels and the computational resources.
- Bidirectional symmetric keys IRKa and IRKs are only shared between BLE advertiser and scanner device and are kept from the adversary.
- The BLE communicating channels is purely open channels and is under full control of the adversary.

In following, the assumptions will be used as baseline to do protocol simulation and validation.

5.3.2 Simulation

Figure 5.2 depicts the protocol simulation for BLE privacy enhancement. Prior to message exchange there will be bidirectional symmetric keys IRKa and IRKs shared

between the advertiser A and scanner S. The message encryption is denoted by {|m|}IRK.

The message exchange of protocol simulation will follow the definition of system design in section 4.3. Generally speaking, the BLE privacy enhancement utilizes a 3-way handshake protocol, which assures freshness of advertising session and deploys a freshly generated nonce Rs from BLE scanner to advertiser. The nonce Rs will be used to initialize local counters which can protect following message exchange between advertiser and scanner.

As illustrated in figure 5.2, in order to simulate the protocol, two agents of role advertiser and scanner are defined and initiated. Firstly, the advertiser agent generates a fresh nonce Ra and sends it together with its encryption by the pre-shared key IRKa to the scanner. As soon as the scanner receives the message which passes a local checking, he will generates a fresh nonce Rs, then send Rs and confirmation of Ra encrypted by the pre-shared IRKs, to the advertiser. The next step will be that the advertiser extracts the nonce Rs and confirms the received Rs to the scanner. The result of the protocol running is both the agent advertiser and agent scanner have the knowledge of the nonce Rs and assure its authenticity.

During protocol execution, the agent advertiser and agent scanner satisfy all the security claims showed in the figure 5.2. In particular, the properties ni-agree, ni-synch, and alive have been hold, which stands for authentication in the form of non-injective synchronization, non-injective agreement and alive. The detail of the security claims has been explained in section 5.2.2.

5.3.3 Analysis of security property

The protocol in figure 5.2 deploys a nonce Rs securely from the scanner to advertiser and satisfies the security properties ni-synch and ni-agree.

In the protocol, every receive event of message is preceded by a corresponding send event. Without sending request for Rs, the advertiser device will not accept and handle the message from the scanner device which consists of a Nonce Rs. In the same way, the agent of scanner also receives and sends out messages in a fixed order as specified in the protocol. If a message arrives out of order, the agents of the protocol will just ignore it. As the ni-synch property requires the corresponding send and receive messages executed in the expected order of a protocol, so the agents of the advertiser and scanner satisfy the security claim of ni-synch.

As to ni-agree property, it requires the content of the corresponding send same with the receive messages. In the protocol, the message exchange of 3-way handshake protocol is protected by the pre-shared keys IRKa and IRKs. As mentioned in the

assumption of the protocol in section 5.3.1, the keys are known only by the advertiser and scanner. So the adversary can only observe the messages over the BLE advertiser channels and can not modify the messages to resend them. After deploying the nonce Rs from the scanner to advertiser successfully, all the advertiser message will be protected by both the pre-shared keys and local counters. It is also infeasible for the adversary to tamper the captured message and resend it.

Therefore, the protocol is both ni-synch and ni-agree. As to injectivity property, it is a higher-order property. That means that it is not possible to identify all attacks on injectivity by a finite set of patterns. Scyther uses a different approach for the verification of injectivity. In particular, *Scyther has syntactically established and verified that LOOP is a sufficient condition to guarantee injectivity for protocols that satisfy NI-SYNCH* [CM12f].

The following will analyze injectivity of the BLE privacy enhancement over advertising channels. As 24-bit nonce Ra and Rs is used in the system design of this project, 24-bit Ra will be repeated around 155 hours in the worst case of BLE advertisement. In order to make protocol analysis we can assume that there are continuously advertising sessions up to 155 hours between a BLE advertiser and scanner. If an adversary can capture all the advertisement and wait for a repeated challenge "Ra, E(IRKa, Ra)" from the advertiser, the adversary can then replay the captured "Ra, E(IRKs, Ra), E(IRKs, Ra+1, Rs)" to the advertiser. From the advertiser's local perspective, the replayed message is legitimate and the advertiser will answer the replayed message. Therefore, in principle 24-bit nonce Ra makes the protocol not fulfill injectivity. It means the adversary can capture all the advertising sessions, wait for Ra to be repeated, and then replay an eavesdropped message from previous session which the legitimate device can not distinguish. However, it is infeasible for adversary to attack BLE advertising devices in ad-hoc mode, the detailed reasons are elaborated in section 4.4.1.

5.3.4 Validation

Generally there will be infinitely possible protocol traces. However, there are a lot of similar protocol traces from the perspective of security verification. Thus, the simulation in this protocol will use trace pattern to represent a class of traces which is defined as a set of message exchange in order.

Scyther tool already implements an algorithm to analyze security properties based on trace pattern of a protocol. The algorithm is constructed upon the analysis of trace patterns, which represent sets of message exchange in order. So the necessary trace properties will be used to evaluate the claimed security properties. e.g. Order of message exchange, occurrence of agent events, the adversary knowledge and so on.

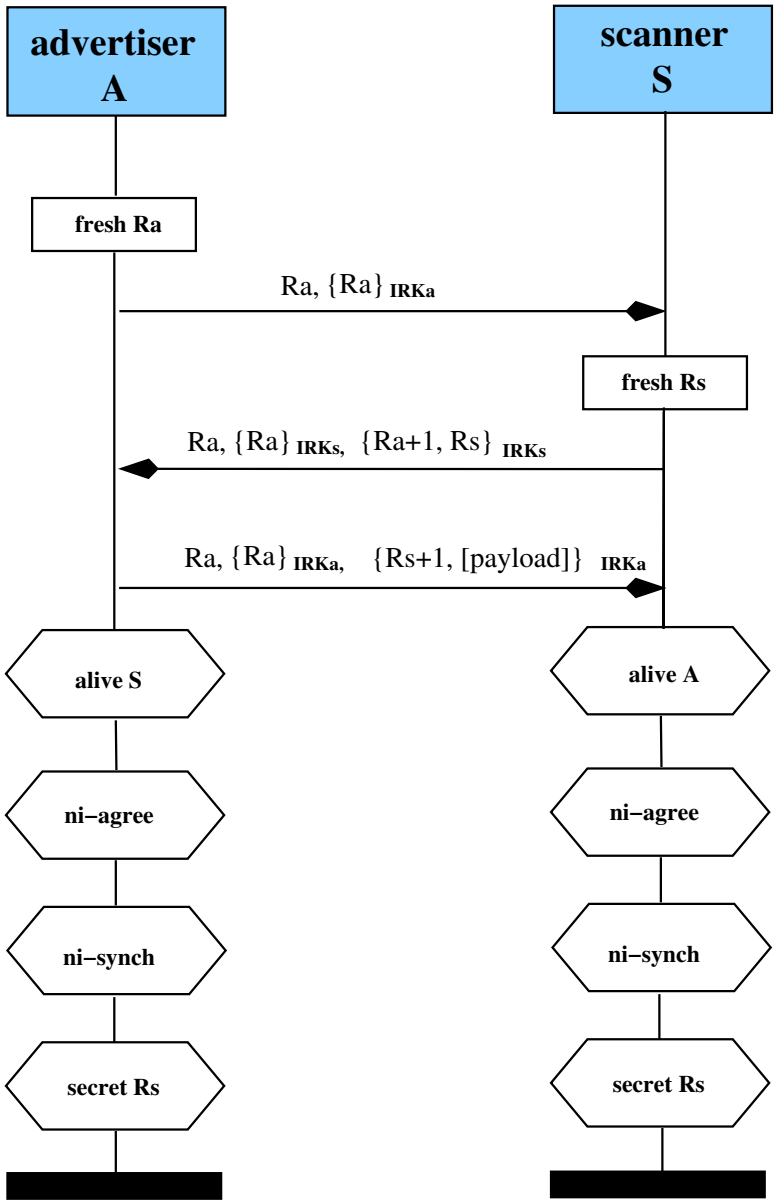


Figure 5.2: BLE privacy enhancement protocol, adopted from [CM12a]

Scyther results : verify

Claim				Status	Comments
protocol2	I	protocol2,i1	Secret Rs	Ok	No attacks within bounds.
		protocol2,i2	Niagree	Ok	No attacks within bounds.
		protocol2,i3	Nisynch	Ok	No attacks within bounds.
		protocol2,r4	Alive	Ok	No attacks within bounds.
	R	protocol2,r1	Secret Rs	Ok	No attacks within bounds.
		protocol2,r2	Niagree	Ok	No attacks within bounds.
		protocol2,r3	Nisynch	Ok	No attacks within bounds.
		protocol2,R1	Alive	Ok	No attacks within bounds.

Done.

Figure 5.3: BLE privacy enhancement protocol verification

As figure 5.3 illustrates that following the specified protocol trace the privacy enhancement protocol satisfies all the claimed security properties. The nonce Rs is deployed from the scanner to advertiser securely and fulfill all the claimed security properties. The message exchange between the advertiser and scanner fulfills the Non-injective Synchronization as well as Non-injective Agreement. That indicates that the protocol is executed exactly as it would be and both the advertiser and scanner agree on that the content of the received message correspond to the sent messages which is specified by the protocol.

In the following chapter protocol implementation in Cortex-M0 platform will be described. The whole chapter covers how to establish developing environment for practical experiments, and also complete coding implementation. The system design and simulation model of BLE privacy enhancement protocol will be examined in more depth.

Chapter 6
Implementation

One of the research methods stated in section 1.4 is to implement the system design and demonstrate the enhanced privacy protocol. This chapter gives a detailed description of implementation for the privacy enhancement protocol. First, it is essential to have a thorough overview about the experimental environment of the implementation, including software environment setup and hardware connection used for this project. Second, functionalities described in system design section 4.4 will be implemented in detail. In order to make implementation more understandable, some code snippet from the most important features have been included.

6.1 Environment setup

6.1.1 Hardware setup

The laboratory used during this project consisted of two identical nRF51822 BLE chips to implement the system design as scanner and advertiser in Chapter 4. Both nRF51822 chips are built around a 32-bit ARM® Cortex™ M0 CPU with 256kB flash + 16kB RAM [Sem13b]. As figure 6.1 shows USB cables will be connected to both boards for power supply and serial terminal readout. J-Link was used to program the nRF51822 BLE chips.

In this project there are two ways to assist program and debug. Firstly debug Graphical User Interface (GUI) running in the development computer shows running result in user terminal and helps user to debug. Secondly Bluetooth sniffer BPA500 is able to capture the packets in the air between the advertiser and scanner, and it can disassembler the captured packets to a a readable format. Figure 6.2 illustrates the overview of experimental setup for this project implementation.

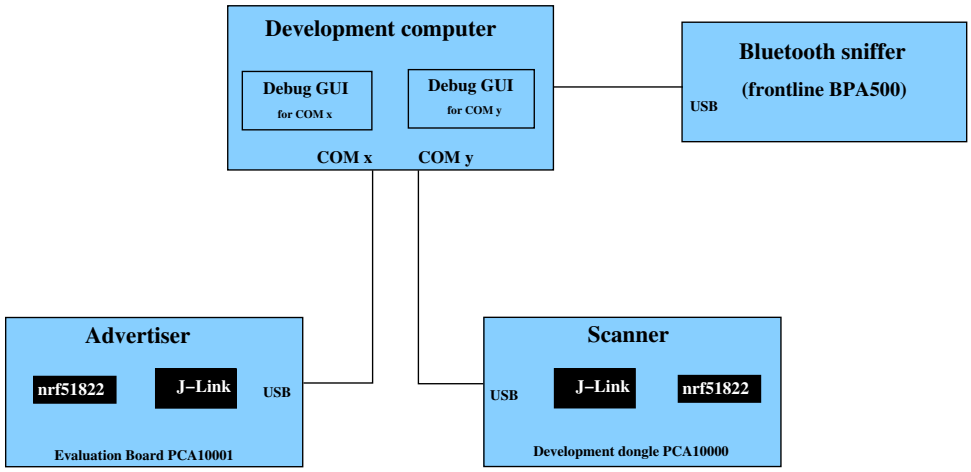


Figure 6.1: The schematics of the boards connection, adopted from [Sem13b]

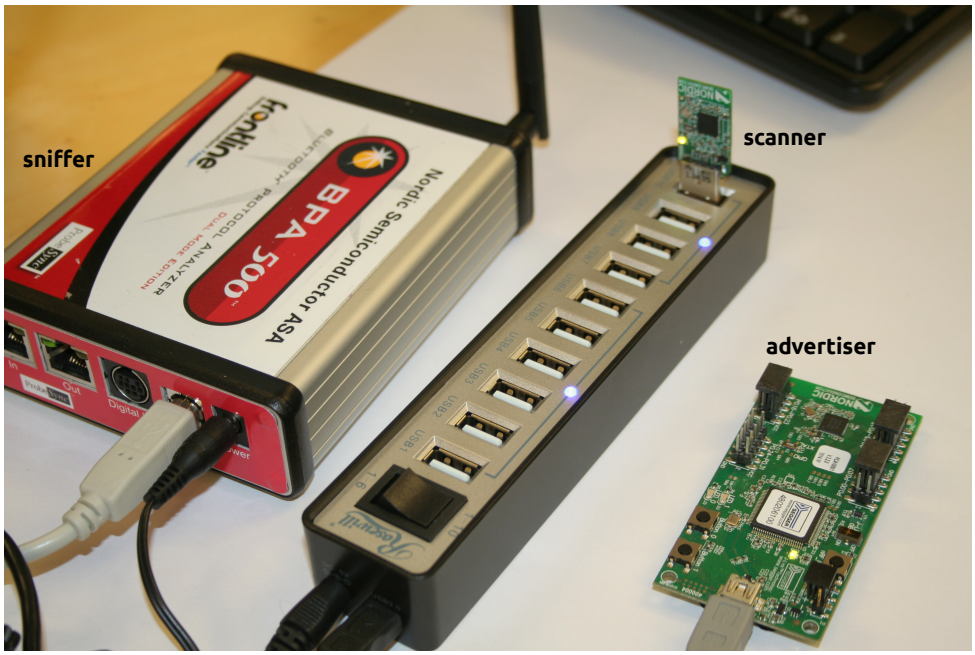


Figure 6.2: Experimental setup

6.1.2 Software installation

All the software development has been done in windows 7, this section will introduce how to set up developing environment for nRF51822 BLE chip in the windows platform.

- Keil MDK-ARM Lite v4.54 : nRF51822 BLE chips programming Integrated Development Environment (IDE).
- J-Link Software v4.56 : upload program into nRF51822 BLE chips.
- nRF51822 Software Development Kit : develop firmware on application microcontrollers used with the nRF51822.
- S110 nRF51822 SoftDevice : BLE stack.
- S110 SoftDevice programming tools : upload program into nRF51822 BLE as stack code.

6.2 Implementation

Following functional design of the BLE privacy enhancement in section 4.4, detailed implementation will be elaborated in this section.

6.2.1 Nonce Rs deploy

Section 4.4.1 described system design for 3-way handshake protocol which deploys a nonce Rs from scanner to advertiser. Here will focus on packet structure and detail the deployment of nonce Rs in three steps. The 3-way handshake packets will be constructed based on the packet definition in section 4.3.3. In particular, the fields "PDU type" and "packet type in privacy enhancement" will be used to determine if a packet is a 3-way handshake packet.

First message - $Ra, E_{IRKa}(Ra)$

Figure 6.3 illustrates how to construct the first message in 3-way handshake protocol. The first message is a nonce Ra and its encryption from the advertiser. In nRF51822 stack, there is already existing RPA mechanism to do exhaustive search for the correct IRK in the field "AdvA" based on a plaintext and corresponding ciphertext. Thus, the first message "$Ra, E_{IRKa}(Ra)$" will be put into the field of "AdvA" in PDU payload directly. As soon as receiving a message is privacy enhancement PDU 0111 and packet type indicates a challenge of 3-way handshake protocol, the scanner device will do exhaustive search for IRKa. If there is a IRK in the bonding database of scanner, the scanner will prepare an answer to the advertiser.

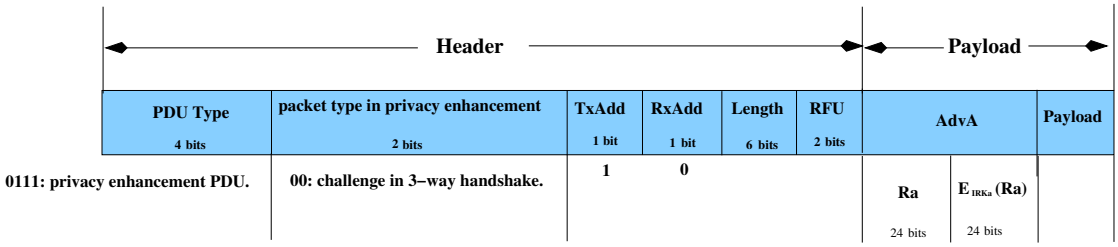


Figure 6.3: Packet structure of 3-way handshake challenge, modified from [Gro13g]

```
bool ll_adv_3way_handshake_challenge_generate()
{

  static nrf_ecb_hal_data_t block_config;

  (void)soc_rand_lowerstack_vector_get(&challeng_in_3way_handshake[0],
                                       LL_ADV_RANDOM_LENGTH_PRA_ENHANCE);

  m_revcpy( &block_config.key[0], &irk_in_advertiser[0], SOC_ECB_KEY_LENGTH);
  m_revcpy( &block_config.cleartext[0], random_number_PRA,
  SOC_ECB_CLEARTEXT_LENGTH);
  soc_ecb_block_encrypt(&block_config);

  memcpy(challeng_in_3way_handshake, random_number_PRA,
         LL_ADV_RANDOM_LENGTH_PRA_ENHANCE );
  memcpy(challeng_in_3way_handshake+LL_ADV_CIPHER_LENGTH_PRA_ENHANCE,
         &block_config.ciphertext[0], LL_ADV_CIPHER_LENGTH_PRA_ENHANCE);
  return true;
}
```

Second message - $Ra, E_{IRKs}(Ra), E_{IRKs}(Ra+1, Rs)$

As soon as the first message passes the verification in the scanner device, the scanner will construct the message "$Ra, E_{IRKs}(Ra), E_{IRKs}(Ra+1, Rs)$" to the advertiser device. The nonce Rs is deployed to the advertiser and used as replay prevention on both sides over the advertising channels. In order to make it possible for the advertiser device to make use of existing RPA mechanism to exhaustively search for $IRKs$ of the scanner, the scanner device puts $Ra, E_{IRKs}(Ra)$ in the field "ScanA". In following, $E_{IRKs}(Ra+1, Rs)$ in the field of "payload".

As soon as receiving the message, the advertiser device will do exhaustive search for IRKs of the scanner in terms of $Ra, E_{IRKs}(Ra)$. If correct IRKs is found in the bonding database of advertiser, the advertiser will use the IRKs to decrypt the ciphertext in the field of "payload" and initialize the local counter RX and TX by the nonce Rs. Figure 6.4 illustrates how to construct the second message in 3-way handshake protocol.

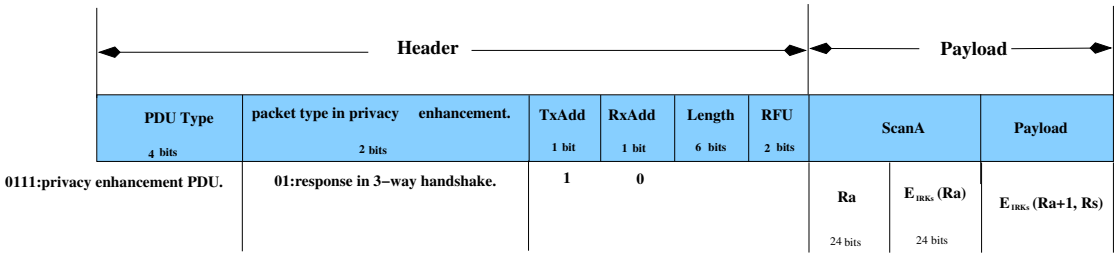


Figure 6.4: Packet structure of 3-way handshake response, modified from [Gro13g]

```
bool ll_scan_3way_handshake_Rs_deploy_cipher_generate(void)
{
  static nrf_ecb_hal_data_t block_config;
  uint8_t concatenation_Ra1_RS_scanner[LL_SCAN_RA1_RS_LENGTH];

  (void)soc_rand_lowerstack_vector_get(&deployed_counter_from_scanner[0],
                                       LL_SCAN_DEPLOY_RS_LENGTH);

  memcpy( concatenation_Ra1_RS_scanner, challeng_from_advertiser,
          LL_CHALLENGE_FROM_ADV_LENGTH );
  memcpy( concatenation_Ra1_RS_scanner + LL_CHALLENGE_FROM_ADV_LENGTH,
          deployed_counter_from_scanner, LL_SCAN_DEPLOY_RS_LENGTH );

  ll_xor_encrypt_fast(&concatenation_Ra1_RS_scanner[0], SOC_ECB_CIPHERTEXT_LENGTH,
                      (void*)&keystreamScan_of_Ra_session);
  return true;
}
```

Third message - $Ra, E_{IRKa}(Ra), E_{IRKa}(Rs+1, [servicedata])$

As to the third message in 3-way handshake protocol, it is confirmation of Rs from the advertiser to scanner device. The field "AdvA" is used to put the $Ra, E_{IRKa}(Ra)$, which can be used by the scanner device to exhaustively search for the correct $IRKa$ of the advertiser device. As soon as finding the correct $IRKa$, the scanner device decrypt the ciphertext in the field "payload". After decryption, the scanner device will first verify correctness of the Rs. If Rs is correct, the scanner device will read the following service data.

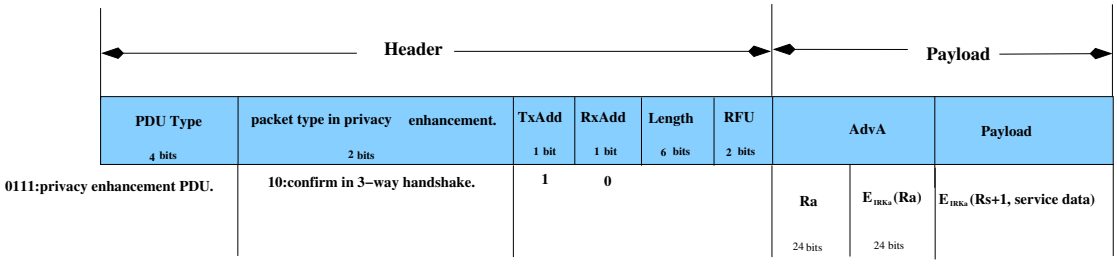


Figure 6.5: Packet structure of 3-way handshake confirm, modified from [Gro13g]

```
bool ll_scan_3way_confirm_from_advertiser_generate(void)
{
  ll_xor_encrypt_fast(&deployment_from_scanner[0] ,
        SOC_ECB_CIPHERTEXT_LENGTH, &keystreamScan_of_Ra_session[0]);
  if(memcmp(&deployment_from_scanner[0], &challeng_in_3way_handshake[0],
   LL_ADV_RANDOM_LENGTH_PRA_ENHANCE) == 0)
  {
    memcpy(&deployed_Rs_from_scanner[0],
    &deployment_from_scanner[LL_ADV_RANDOM_LENGTH_PRA_ENHANCE],
           LL_ADV_RANDOM_LENGTH_PRA_ENHANCE);
  }
  memcpy(&advertiser_payload[0], &deployed_Rs_from_scanner[0], RS_LENGTH);
  ll_xor_encrypt_fast(&advertiser_payload[0], SOC_ECB_CIPHERTEXT_LENGTH,
                      (void*)&keystreamAdv_of_Ra_session);
  return true;
}
```

6.2.2 Counter store and update

The nRF51822 chip has has a 32-bit memory space and separate memory types for program and data connected with distinct buses, and each memory type has its own address space. Such a memory architecture allows the processor to access both program and data memories at the same time [Sem13a]. In more detail, there are three main categories of memory in nRF51822 : internal flash (code memory), RAM, and peripheral registers. Internal flash is non-volatile memory, which will not lose its contents when power-off. RAM and peripheral registers are volatile memory.

In section 4.4.2 counter's location and update has also been designed. Combining the executing performance and BLE advertising characteristics, RAM memory is selected. Figure 6.6 shows that real-time updating of the local counter TX and RX will be executed in the RAM area. The counter RX and TX will be defined as two global variables in the program, which will be located in the RAM memory. Prior to handling a packet, the device will firstly check the packet number in the received

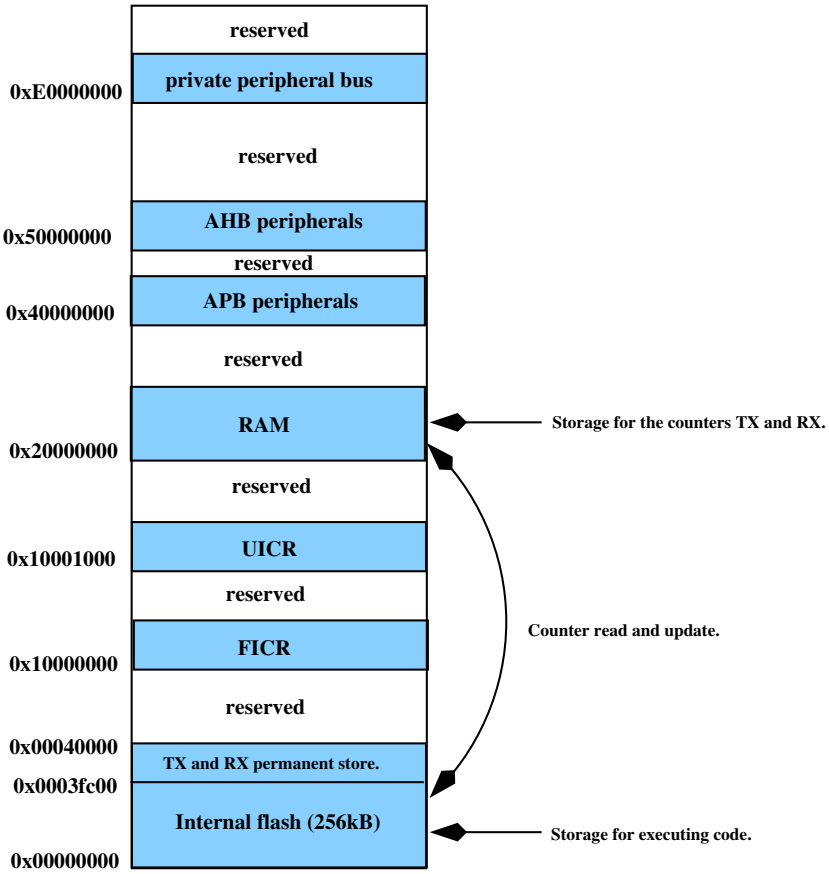


Figure 6.6: Memory map for counter store, adopted from [Sem13a]

packet by the variable RX value. On the other hand, the variable TX value will also be appended to a packet before sending out.

6.2.3 Handle of out-of-synchronization

The simplest way to prevent replay attacks is to discard any received packet in which the counter has not increased by one compared to the last successfully received packet. If this simplest way is used as countermeasure against replay attack in the context of BLE advertisement, every single packet loss will invoke a 3-way handshake protocol. However, packet loss is not rare situation when a advertiser sends packets to a scanner out of range. Therefore, in terms of system design in section 4.4.3 a mitigation is implemented to accommodates BLE advertisement to normal packet loss.

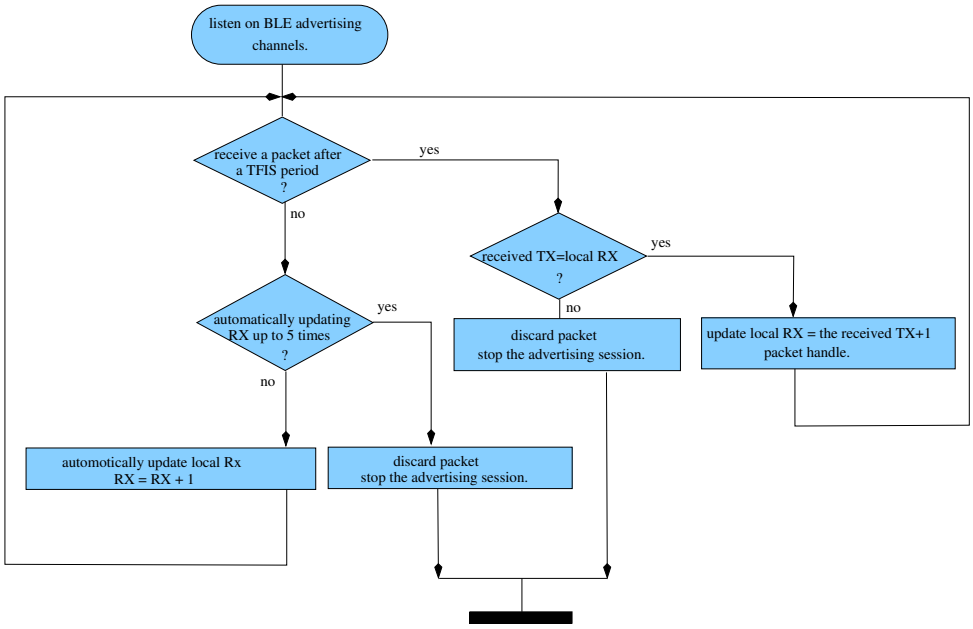


Figure 6.7: Schematics to mitigate packet loss over the BLE advertising channels

As mentioned in section 4.4.3, BLE radio system is half-duplex system which means that a BLE device will in a receiving state or sending state, but not both at the same time. The device in the sending state will concatenate a local counter TX to send a packet and then update the TX by increasing by 1. On the other side, the device in the receiving state will firstly verify the received packet with its local counter RX and then process the packet in terms of result of the verification. In order to accommodate to the packet loss over BLE advertising channels, a mitigation will be implemented for BLE device in receiving state. Detailed scheme is illustrated by the figure 6.7.

As soon as the nonce Rs is deployed from the scanner to advertiser, the local counter RX and TX of the advertiser and scanner will be initialized to be Rs. Counter TX is used to send out a packet and counter RX is used to verify the received packet. Taking the characteristics of BLE radio in section 4.4.3 into consideration, if the BLE device in the receiving state has not received a packet after the period TIFS, the device will automatically update counter RX by increasing one. This behavior will be done automatically up to (5) [1] times. After (5) times, if the device in the receiving state still has not received a packet with a counter which is consistent with

[1] pre-defined number 5 is used in experimental environment of this project, it can be adjusted in terms of different requirements

the local counter RX, the device in receiving state will stop the advertising session immediately. The control processing will return and the device in the receiving state stop to listen to the advertising channel. If a new advertising session is required, a 3-way handshake protocol will be invoked and a new nonce Rs will be deployed to initialize local counters.

Chapter 7
Future work

This project implemented a demonstration of the privacy enhancement protocol in nRF51822 chips, which prevents privacy leakage and device tracking of BLE devices over advertising channels. All the security implementations are limited to the demonstration of the protocol designed. Thus, stable functions are more important than other aspects. Next phase of the privacy enhancement can have further improvements and provide more user-friendly GUI.

7.1 Future functionality

The demonstration of the system design proposed in Chapter 4 has focused on the most critical functionalities which are sufficient for the privacy enhancement protocol. The optional functionalities in section 4.5.2 and 4.5.3 have been omitted.

First of all, mitigation against DoS in section 4.5.3 is an improvement for performance of the system. The consideration in this phase of the project will be put more weight on a functional and secure protocol in a realistic environment. Thus, the mitigation will be postponed to the future. Moreover, the current system design has also proposed the mitigation for scanner tracking proposed in section 4.5.2. If some applications have higher security requirement against scanner tracking, the mitigation for scanner tracking can be implemented in the specific applications. However, the ability to completely prevent scanner tracking has to be designed and implemented in the future.

7.2 User interface

There must be some adjustment and improvement to be made in current demonstration with regard to user-friendly GUI. In order to be more modularize and integrated into the existing BLE stack easily, all the functionalities of this demonstration have been merged into the link layer. In current phase of the project, all the programming

debug and function scheduling has been done through a dummy host. However, the code of the dummy host is only accessible for the engineers who can access the internal source code. For a user-friendly protocol, the whole stack should be implemented to uphold a seamless end-user experience and implementation detail should be kept away from the end-user. So, the next phase of this project should add more interfaces in end-user's GUI, and also more commands in HCI layer. The end-user just need to select privacy options in the user GUI, then commands will go through HCI and are directed to the link layer for execution of privacy protection over the advertising channels.

7.3 Thorough test

As mentioned before, it is hard to predict and emulate all the attacking scenarios over the BLE advertising channels precisely. So a more comprehensive test environment should be set up and thorough tests are needed, which will improve the system reliability and performance. Based on the thorough tests in the future, the parameters in the program may be adjusted, including pre-defined number of times to allow packet loss without re-invoking 3-way handshake protocol.

Chapter 8
Conclusion

This chapter summarizes the work accomplished through the whole project. The project has focused on design, simulation and implementation of a BLE privacy enhancement over the BLE advertising channels. The main focus has been put on the message confidentiality, replay prevention, and anti-tracking of devices over the advertising channels. Due to lack of central clock, the privacy enhancement utilizes a counter solution instead of timestamp as countermeasure against replay attack and device tracking.

All security features designed and implemented as part of this project are using Bluetooth core specification 4.1 as baseline, which already includes bonding, key distribution, and AES-CCM mechanisms. The project starts with a background chapter. In following, system design of the enhanced privacy protocol has been conducted. Then Scyther security protocol verification tool is selected as simulator to verify the protocol and claimed security properties. In order to implement and test functionalities, a realistic environment has been set up. It consists of two nRF51822 chips which work as advertiser and scanner to communicate over BLE advertising channels. All the implemented functionalities have been verified based on system design in section 4.

In this thesis work, a number of achievements have been made:

- First of all, a pre-study of BLE has been performed from different perspectives, which are: core system architecture, data transport, advertisement, and the overview of existing security mechanisms. Related techniques are grouped together by the pre-study to give hints to functional requirements which are critical for system design. Furthermore, the pre-study also provides overview about how to integrate the new designed protocol into the existing BLE stack.

- Secondly, privacy violation models and functional requirements has been made in terms of BLE pre-study. In following, system design has been elaborated

from stack, processing flow and message structure perspectives. The detailed functions cover a 3-way handshake protocol to deploy a nonce Rs from the scanner to the advertiser, which is used to initialize local counters RX and TX on both the advertiser and scanner. Furthermore, handling of out-of-synchronization, countermeasure against device tracking, and mitigation for DoS has also been designed.

- The third achievement made in this work is simulation of the privacy enhancement protocol. Scyther simulates the privacy enhancement protocol and establishes analytical models. It is very useful especially in this case because the analytical models could verify the privacy enhancement under various dynamic attacking scenarios.
- After performing protocol simulation and verification, a realistic environment has been setup with two nRF51822 chips as advertiser and scanner to implement and test a demonstration of the system design. The whole implementation has been added in the link layer of BLE stack. The advantage in its favor is it being more modularized and has less influence on other subsystems. It eases implementation, maintenance, functional integration, and test.

The most important areas that can be improved in the future are mentioned in chapter 7. In accordance with system design, multiple of the goals were reached. The output of code is a complete privacy enhancement demonstration with confidentiality protection, replay prevention, and anti-tracking of device. I feel that the whole research has been conducted following the research method in section 1.4 and the goals of this project have been reached.

References

[blu13] *Specification of the Bluetooth System, core version 4.1*, volume 8 of *BLuetooth*, Kirkland, WA 98033 USA, 2013. Bluetooth SIG, Inc.

[CM12a] Cas Cremers and Sjouke Mauw. Authentication. In *Operational Semantics and Verification of Security Protocols* [CM12d], pages 41–50.

[CM12b] Cas Cremers and Sjouke Mauw. Communication and threat model. In *Operational Semantics and Verification of Security Protocols* [CM12d], page 16.

[CM12c] Cas Cremers and Sjouke Mauw. Describing protocol execution. In *Operational Semantics and Verification of Security Protocols* [CM12d], pages 25, 26.

[CM12d] Cas Cremers and Sjouke Mauw. *Operational Semantics and Verification of Security Protocols*. Information Security and Cryptography. Springer, 2012.

[CM12e] Cas Cremers and Sjouke Mauw. Security properties. In *Operational Semantics and Verification of Security Protocols* [CM12d], pages 41–52.

[CM12f] Cas Cremers and Sjouke Mauw. Verifying injectivity. In *Operational Semantics and Verification of Security Protocols* [CM12d], pages 96–103.

[Gro13a] B. S. I. Group. Architecture, core system architecture. In *Specification of the Bluetooth System, core version 4.1* [blu13], pages 144–157.

[Gro13b] B. S. I. Group. Architecture, data transport architecture. In *Specification of the Bluetooth System, core version 4.1* [blu13], pages 163,174–199.

[Gro13c] B. S. I. Group. Architecture, security overview. In *Specification of the Bluetooth System, core version 4.1* [blu13], page 221.

[Gro13d] B. S. I. Group. Generic access profile, privacy feature. In *Specification of the Bluetooth System, core version 4.1* [blu13], pages 2017–2018.

[Gro13e] B. S. I. Group. Generic access profile, random device addres. In *Specification of the Bluetooth System, core version 4.1* [blu13], pages 2020–2022.

[Gro13f] B. S. I. Group. Link layer specification, air interface packets. In *Specification of the Bluetooth System, core version 4.1* [blu13], pages 2503–2521.

[Gro13g] B. S. I. Group. Link layer specification, air interface packets. In *Specification of the Bluetooth System, core version 4.1* [blu13], pages 2504–2510.

[Gro13h] B. S. I. Group. Link layer specification, link layer states. In *Specification of the Bluetooth System, core version 4.1* [blu13], pages 2498–2501.

[Gro13i] B. S. I. Group. Link layer specification, non-connected states. In *Specification of the Bluetooth System, core version 4.1* [blu13], pages 2526–2538.

[Gro13j] B. S. I. Group. Low energy link layer security. In *Specification of the Bluetooth System, core version 4.1* [blu13], pages 2611–2616.

[Gro13k] B. S. I. Group. Non-connected states, connectable undirected event type. In *Specification of the Bluetooth System, core version 4.1* [blu13], pages 2529–2530.

[Gro13l] B. S. I. Group. Security manager specification, pairing methods. In *Specification of the Bluetooth System, core version 4.1* [blu13], pages 2249–2257.

[Gro13m] B. S. I. Group. Security manager specification, security in bluetooth low energy. In *Specification of the Bluetooth System, core version 4.1* [blu13], pages 2258–2280.

[HMPR04] Alan R. Hevner, Salvatore T. March, Jinsoo Park, and Sudha Ram. Design science in information systems research. *MIS Quarterly*, 28(1):75–105, 2004.

[Kha08] Ravindra Khare. Simulation - a powerful technique to improve quality and productivity. *Simulation article: Symphony Technologies*, page 7, 2008.

[KI04] Kaoru Kurosawa and Tetsu Iwata. Tmac: Two-key cbc mac. *IEICE Transactions*, 87-A(1):46–52, 2004.

[Sem13a] Nordic Semiconductor. nrf51 series reference manual, version 2.1. pages 11–13, 2013.

[Sem13b] Nordic Semiconductor. nrf51822 evaluation kit, user guide v1.2. page 13, 2013.

[Sta01] William Stallings. Message encryption codes. In *Cryptography and network security - principles and practice (5. ed.)*, pages 376–380. Prentice Hall, 2001.

Chapter A

Appendix

A.1 Modeling protocol-BLE privacy enhancement

In this appendix, I have included code of Scyther for this project.

```
/*
   Nonce Rs deployment protocol
*/


// User type declaration
const increaseByOne: Function;

// bidirectional symmetric keys between advertiser and scanner
macro kir = k(I,R);
macro ksr = k(R,I);


// Protocol description
protocol ns3(I,R)
{
        role I
        {
                fresh Ra: Nonce;
                fresh serviceData: Nonce;
                var Rs: Nonce;

                send_1(I,R, Ra, {Ra}kir );
                recv_2(R,I, Ra, {Ra}ksr, {increaseByOne(Ra), Rs}ksr);
                send_3(I,R, Ra, {Ra}kir, {increaseByOne(Rs),serviceData}kir);
```

```
26
27                  claim(I,Secret,Rs);
28                  claim(I,Alive);
29                  claim(I,Niagree);
30                  claim(I,Nisynch);
31          }
32
33          role R
34          {
35                  fresh Rs: Nonce;
36                  var Ra: Nonce;
37                  var serviceData: Nonce;
38
39                  recv_1(I,R, Ra, {Ra}kir );
40                  send_2(R,I, Ra, {Ra}ksr, {increaseByOne(Ra), Rs}ksr);
41                  recv_3(I,R, Ra, {Ra}kir, {increaseByOne(Rs),serviceData}kir );
42
43                  claim(R,Secret,Rs);
44                  claim(R,Alive);
45                  claim(R,Niagree);
46                  claim(R,Nisynch);
47          }
48  }
```

A.2 Code

In order to comply with non-disclosure agreement with Nordic Semiconductor, this appendix has only included part of code of this project.

A.2.1 Scanner

```
bool le_initilize_keystream_to_decrypt_advertiser()
{
  static nrf_ecb_hal_data_t block_config;
  memset(&challeng_from_advertiser[0], 0xab, 3);
  m_revcpy( &block_config.key[0], &irk_peer_advertiser[0],
                SOC_ECB_KEY_LENGTH);
  m_revcpy( &block_config.cleartext[0], challeng_from_advertiser,
                LL_CHALLENGE_FROM_ADV_LENGTH);
  soc_ecb_block_encrypt(&block_config);
  memcpy(&key_stream_for_advertiser_decryption[0],
        (void*)&block_config.ciphertext[0],
        LL_ADVERTISER_KEYSTREAM_LENGTH);
  return true;
}

void le_xor_encrypt_fast(uint8_t *nonce, size_t len,
        uint8_t key[KEY_LENGTH])
{
    // key is a 16-byte xor key
    size_t i;
    for(i = 0; i < len; i++)
        nonce[i] ^= key[i % 16];

}

bool scanner_3way_handshake_scan_address_generate(void)
{
  static nrf_ecb_hal_data_t block_config;

  m_revcpy( &block_config.key[0], &irk_in_scanner[0], SOC_ECB_KEY_LENGTH);
  m_revcpy( &block_config.cleartext[0], challeng_from_advertiser,
                LL_CHALLENGE_FROM_ADV_LENGTH);
  soc_ecb_block_encrypt(&block_config);

  memcpy(response_address_in_3way_handshake, challeng_from_advertiser,
```

```
36          LL_CHALLENGE_FROM_ADV_LENGTH );
37    memcpy(response_address_in_3way_handshake+LL_CHALLENGE_FROM_ADV_LENGTH,
38          &block_config.ciphertext[0], LL_SCAN_PRA_CIPHER_LENGTH_ENHANCE);
39    return true;
40  }
41
42  bool scanner_3way_handshake_Rs_deploy_cipher_generate(void)
43  {
44    static nrf_ecb_hal_data_t block_config;
45    uint8_t concatenation_Ra1_RS_scanner[LL_SCAN_RA1_RS_LENGTH];
46    memset(&concatenation_Ra1_RS_scanner[0], 0x00, LL_SCAN_RA1_RS_LENGTH);
47
48    (void)soc_rand_lowerstack_vector_get(&deployed_counter_from_scanner[0],
49          LL_SCAN_DEPLOY_RS_LENGTH);
50
51    memcpy( concatenation_Ra1_RS_scanner, challeng_from_advertiser,
52                  LL_CHALLENGE_FROM_ADV_LENGTH );
53    memcpy( concatenation_Ra1_RS_scanner + LL_CHALLENGE_FROM_ADV_LENGTH,
54          deployed_counter_from_scanner, LL_SCAN_DEPLOY_RS_LENGTH );
55
56    m_revcpy( &block_config.key[0], &irk_in_scanner[0], SOC_ECB_KEY_LENGTH);
57    m_revcpy( &block_config.cleartext[0], challeng_from_advertiser,
58          LL_CHALLENGE_FROM_ADV_LENGTH);
59    soc_ecb_block_encrypt(&block_config);
60
61
62    ll_xor_encrypt_fast(&concatenation_Ra1_RS_scanner[0],
63          SOC_ECB_CIPHERTEXT_LENGTH,
64          (void*)&block_config.ciphertext[0] );
65    memcpy( &deployed_counter_from_scanner_cipher[0],
66                  &concatenation_Ra1_RS_scanner[0],
67                  LL_SCAN_DEPLOY_COUNTER_CIPHER_LENGTH);
68    return true;
69
70  }
71
72  void scanner_pdu_build_address_and_deployed_counter(ul_pdu_dd_pdu_t * to_pdu)
73  {
74      if(scanner_3way_handshake_scan_address_generate() == true)
75      {
76        memcpy(&(to_pdu->content)[UL_PDU_DD_SENDER_ADDRESS_OFFSET],
```

```
77          response_address_in_3way_handshake, LL_SCAN_RPA_LENGTH);
78      }
79      if( scanner_3way_handshake_Rs_deploy_cipher_generate() == true )
80      {
81        memcpy(&(to_pdu->content)[UL_PDU_DD_PAYLOAD_OFFSET],
82          deployed_counter_from_scanner_cipher,
83          LL_SCAN_DEPLOY_COUNTER_CIPHER_LENGTH);
84      }
85
86  }
87
88  static void scanner_rcv_challenge_and_send_response()
89  {
90    ..............
91    if ( scanner_3way_handshake_flag == true ) &&
92          ( scanner_3way_handshake_scan_address_generate() == true )
93          && (scanner_3way_handshake_Rs_deploy_cipher_generate() == true ))
94    {
95      scanner_pdu_build_address_and_deployed_counter(&m_tx_pdu);
96    }
97
98    ..............
99  }
100
101 static void m_build_response_packet(ul_pdu_dd_pdu_t *p_pdu,
102                         ul_pdu_dd_packet_type_t reply_packet)
103 {
104   ..............
105   if(UL_PDU_DD_3WAY_RESPONSE == reply_packet)
106   {
107     ul_pdu_dd_pdu_reset(p_pdu);
108     ul_pdu_dd_set_packet_type(p_pdu,
109           UL_PDU_DD_3WAY_RESPONSE);
110
111   }
112   ..............
113 }
114
115 btle_status_codes_t scanner_prepare(uint32_t pkt_timeout_us)
116 {
117       ..............
```

```
118       else if(m_scanner.params.scan_type == LL_SCAN_ACTIVE)
119       {
120
121         m_build_response_packet(&m_tx_pdu, UL_PDU_DD_3WAY_RESPONSE );
122
123       }
124       ..............
125 }
126
127 void hal_message_receiving_callback(bool crc_ok)
128 {
129      ..............
130      case UL_PDU_DD_3WAY_CHALLENGE:
131         ..............
132         if( m_received_packet_type == UL_PDU_DD_3WAY_CHALLENGE)
133         {
134         scanner_3way_handshake_flag = true;
135         memcpy(&adv_RPA_in_3way_handshake[0],
136         (uint8_t *)&(m_rx_pdu.content)[UL_PDU_DD_SENDER_ADDRESS_OFFSET],
137                         UL_PDU_DD_DEV_ADDR_SIZE);
138         memcpy(&challeng_from_advertiser[0],
139                 (void*)&adv_RPA_in_3way_handshake[0],
140                 LL_CHALLENGE_FROM_ADV_LENGTH);
141         address_accepted = true ;
142         }
143         ..............
144
145         ..............
146      case M_STATE_RECEIVE_SCAN_RSP:
147         if( 0 == memcmp(adv_RPA_in_3way_handshake,
148           (uint8_t *)&(m_rx_pdu.content)[UL_PDU_DD_SENDER_ADDRESS_OFFSET],
149                 LL_ADV_RPA_LENGTH) )
150         {
151             address_accepted_in_3way_handshake = true;
152             if( le_initilize_keystream_to_decrypt_advertiser() == true )
153             {
154                m_revcpy(&confirm_from_advertiser[0],
155                 (uint8_t *)&(m_rx_pdu.content)[UL_PDU_DD_PAYLOAD_OFFSET],
156                         LL_NONCE_CONFIRM);
157                le_xor_encrypt_fast(&Rs_confirm_from_advertiser[0],
158                         LL_ADVERTISER_KEYSTREAM_LENGTH,
```

```
159                         (void*)&key_stream_for_advertiser_decryption[0]);
160
161
162             }
163         }
164         ..............
165  }
```

A.2.2 Advertiser

```
1  bool le_initilize_keystream_to_decrypt_scanner()
2  {
3    static nrf_ecb_hal_data_t block_config;
4    m_revcpy( &block_config.key[0], &irk_peer_scanner[0],
5                  SOC_ECB_KEY_LENGTH);
6    m_revcpy( &block_config.cleartext[0], challeng_in_3way_handshake,
7                  LL_CHALLENGE_FROM_ADV_LENGTH);
8    soc_ecb_block_encrypt(&block_config);
9    memcpy(&key_stream_for_scanner_decryption[0],
10                 (void*)&block_config.ciphertext[0],
11                 LL_SCANNER_KEYSTREAM_LENGTH);
12   return true;
13 }
14
15 bool le_initialize_key_stream_to_encrypt_advertiser()
16 {
17   static nrf_ecb_hal_data_t block_config;
18   m_revcpy( &block_config.key[0], &irk_in_advertiser[0],
19                 SOC_ECB_KEY_LENGTH);
20   m_revcpy( &block_config.cleartext[0], challeng_in_3way_handshake,
21                 LL_CHALLENGE_FROM_ADV_LENGTH);
22   soc_ecb_block_encrypt(&block_config);
23   memcpy(&key_stream_for_advertiser_encryption[0],
24                 (void*)&block_config.ciphertext[0],
25                 LL_ADVERTISER_KEYSTREAM_LENGTH);
26   return true;
27 }
28
29 bool advertiser_3way_handshake_challenge_generate()
30 {
31
32   static nrf_ecb_hal_data_t block_config;
```

```
33
34    (void)soc_rand_lowerstack_vector_get(&random_number_PRA[0],
35                  LL_ADV_RANDOM_LENGTH_PRA_ENHANCE);
36
37    m_revcpy( &block_config.key[0], &irk_in_advertiser[0],
38                  SOC_ECB_KEY_LENGTH);
39    m_revcpy( &block_config.cleartext[0], random_number_PRA,
40                  SOC_ECB_CLEARTEXT_LENGTH);
41    soc_ecb_block_encrypt(&block_config);
42
43    memcpy(challeng_in_3way_handshake, random_number_PRA,
44                  LL_ADV_RANDOM_LENGTH_PRA_ENHANCE );
45    memcpy(challeng_in_3way_handshake+LL_ADV_CIPHER_LENGTH_PRA_ENHANCE,
46                  &block_config.ciphertext[0],
47                  LL_ADV_CIPHER_LENGTH_PRA_ENHANCE);
48
49    return true;
50  }
51
52  void le_cmd_set_advertising_parameters()
53  {
54    ..............
55    if( parameters->own_address_type == BTLE_ADDR_TYPE_RANDOM )
56    {
57
58       if( parameters->own_address_type == BTLE_ADDR_TYPE_RANDOM &&
59                  parameters->type == BTLE_ADV_TYPE_3WAY_HANDSHAKE )
60      {
61         memcpy( &m_ll_ctrl.dev_cfg.random_bd_addr.value[0],
62                  &challeng_in_3way_handshake[0], LL_ADV_RPA_LENGTH );
63         le_initilize_keystream_to_decrypt_scanner();
64         le_initialize_key_stream_to_encrypt_advertiser();
65         m_ll_ctrl.dev_cfg.random_bd_addr.is_set = true;
66      }
67     ..............
68  }
69
70  void le_message_receiving_callback( bool crc_ok )
71  {
72    ..............
73    if(crc_ok && UL_PDU_DD_3WAY_RESPONSE == received_packet_type)
```

```
74    {
75
76      received_packet_type = ul_pdu_dd_get_packet_type(&m_received_pdu);
77      memcpy(&deployment_from_scanner[0], &m_received_pdu.content[15],
78                  LL_DEPLOYMENT_FROM_SCANNER_LENGTH);
79      le_xor_encrypt_fast(&deployment_from_scanner[0] ,
80                  LL_SCANNER_KEYSTREAM_LENGTH,
81                  &key_stream_for_scanner_decryption[0]);
82      if(memcmp(&deployment_from_scanner[0], &challeng_in_3way_handshake[0],
83                  LL_CHALLENGE_FROM_ADV_LENGTH)==0)
84      {
85
86          memcpy(&confirm_Rs_in_advertiser[0], &deployment_from_scanner[3],
87                  LL_SCANNER_DEPLOYED_Rs_LENGTH);
88          le_xor_encrypt_fast(&confirm_Rs_in_advertiser[0] ,
89                  LL_SCANNER_KEYSTREAM_LENGTH,
90                  &key_stream_for_advertiser_encryption[0]);
91
92      }
93      memcpy(&m_scan_resp_pdu.content[UL_PDU_DD_PAYLOAD_OFFSET],
94                  &confirm_Rs_in_advertiser[0], LL_RS_LENGTH);
95      memcpy(&m_scan_resp_pdu.content[UL_PDU_DD_PAYLOAD_OFFSET],
96          &m_scan_resp_pdu[0], m_advertiser.shadow_srsp_data_length);
97      le_xor_encrypt_fast(&m_scan_resp_pdu.content[UL_PDU_DD_PAYLOAD_OFFSET],
98          LL_ADVERTISER_KEYSTREAM_LENGTH,
99          &key_stream_for_advertiser_encryption[0]);
100
101     ..............
102 }
```