Institute of Architecture of Application Systems

University of Stuttgart
Universitätsstraße 38
D–70569 Stuttgart

Master Thesis

Easy to use methods for securing the channel between mobile apps and connected consumer devices

Saranyan Palaniappan

Course of Study:	M.Sc. Information Technology
Examiner:	Dr. Ilche Georgievski
Supervisor:	Dr. Martin Pohlmann, Robert Bosch Power Tools GmbH, Dipl. Heinz Haeberle, Robert Bosch Power Tools GmbH
Commenced:	August 01, 2023
Completed:	February 01, 2024

Abstract

Bluetooth is one of the primary short range wireless communication technology available in all consumer devices. The Bluetooth Low Energy specification made this technology even suitable for battery-operated devices. Also, the features and functionality of battery-operated consumer devices are increased significantly than before. These features can be accessed over Bluetooth Low Energy. Some devices are equipped with actuators which can be operated from a remote place. These actuators may harm users if they are controlled by an malicious agent. Also, it affects user experience if the malicious agent compromises the communication link while the device is in use. Moreover, these battery-operated devices are mostly headless i.e., human machine interfaces are very limited. But Bluetooth Low Energy authentication mechanisms are highly dependent on sophisticated human machine interfaces.

Initially, we evaluated state of the art solutions for encryption and authentication in Bluetooth Low Energy. It shows that Bluetooth Low Energy security concepts are defined at three different places of Bluetooth host stack. They are paring feature exchange parameters, Low Energy security modes and attribute permissions. A deep investigation were carried out on such security concepts defined in the Bluetooth core specification. As a result, we identified bottlenecks and flaws in the Bluetooth standard. The impact of such shortcomings on authentication and encryption mechanisms were clearly described in this research work.

In addition to Bluetooth Low Energy security mechanisms, we invented new encryption and authentication methods suitable for headless devices. These techniques can be combined with existing Bluetooth Low Energy security concepts. In the end, a suitable security mechanism is selected based on device capability and security regulations. The same technique is implemented and tested on the device. Furthermore, an alternative solution is suggested to overcome interoperability issues found between iOS and Android smart phones. Finally, we explored threat modelling frameworks for Bluetooth Low Energy, identified security issues and provided mitigation's for all the threats found in the system.

Acknowledgment

I would like to express my sincere and immense gratitude to Dr. Martin Pohlmann, Dipl. Heinz Haeberle and other team members for giving me the opportunity to work on this exciting thesis topic, incredible support and encouragement in every moment of this thesis.

I would like to convey my deepest gratitude to Dr. Ilche Georgievski for his valuable feedback, examining my thesis work, and reviewing all my steps during this thesis.

I extend my heartfelt thanks to Dr. Andreas Frischen, Dr. Heiko Sgarz and Dr. Krapf Reiner for their assistance in finding this research topic.

Contents

List of Figures

List of Tables

Acronyms

AES Advanced Encryption Standard. 53

API Application Programming Interface. 32

ATT Attribute Protocol. 27

BLE Bluetooth Low Energy. 7

BR Base Rate. 24

CRC Cyclic Redundancy Check. 29

CSRK Connection Signature Resolving Key. 29

DH Diffie-Hellman. 22

DoS Denial of Service. 45

DREAD Damage, Reproducibility, Exploitability, Affected users, and Discoverability. 18

ECDH Elliptic curve Diffie-Hellman. 23

EDIV Encrypted Diversifier. 33

EDR Enhanced Data Rate. 24

FIPS Federal Information Processing Standards. 40

GAP Generic Access Profile. 26

GATT Generic Attribute Profile. 26

HCI Host Controller Interface. 26

HMI Human Machine Interaction. 16

HS High Speed. 24

IoT Internet of Things. 15

IRK Identity Resolving Key. 29

LE Low Energy. 8

LTK Long Term Key. 29

MAC Media Access Control. 29

MITM Man-in-the-middle. 10

NFC Near Field Communication. 32

OOB Out-of-Band. 20

OS Operating System. 17

OSI Open Systems Interconnection. 78

QR Quick Response. 9

RNG Random Number Generator. 22

RTOS Real-time operating system. 23

SDK Software Development Kit. 28

SIG Special Interest Group. 16

SLC Security Levels Characteristic. 28

SoC System on a chip. 26

SPI Serial Peripheral Interface. 26

SSP Secure Simple Pairing. 25

STRIDE Spoofing, Tampering, Repudiation, Information Disclosure, Denial of Service and Elevation of Privilege. 18

USB Universal Serial Bus. 26

1 Introduction

The world becomes more data-driven, high volume of data is generated and consumed every day. Data is collected in tremendous amount by Internet of Things (IoT) edge devices deployed all over the world. From the last decade, the number of IoT connected devices deployed worldwide is increasing and it will reach 30 billion by 2030 according to Statista. This includes devices which run on a button cell for example, Apple AirTags to more sophisticated devices like Smart phones. These devices can collect data from sensors, receive software updates, and execute actions initiated by a remote device. Moreover, the devices which are not connected to Internet before, needs some means of connectivity.

As the number of connected devices increases sharply, this brings new challenges in connectivity, security, and interoperability between devices. A network is a heterogeneous system of components. In the past, security techniques and protocols are developed to ensure security and interoperability between devices which are connected to a network. These security techniques and protocols are not completely suitable for IoT devices exist in the modern network since IoT devices are highly customized for an application. It is necessary to invent new security methods based on IoT devices computation capability, means of connectivity and context of usage.

1.1 Context and Motivation

In the early days, consumer device functionalities are hard-coded which cannot be modified. But these days, consumer devices are equipped with processors and controllers which can be re-programmable at any point in time during the whole lifetime of the product. The consumer device functionalities are realized by means of a software. These software's are prone to errors. It should be updated periodically. Also, it is possible to add new features into existing products by doing software update and send feedback to software development team which will help them to solve issues in the firmware. To support these functionalities, consumer devices need some means of connectivity to update its software. It can be a wired or wireless technology. Usually, the communication method is selected based on the device capability and its usage.

Bluetooth is an ubiquitous wireless communication technology available in all consumer devices. It is simple, less computation intensive and consumes low power than other available communication technologies. It is the most preferred communication method for resource constrained devices. But Bluetooth protocol is not directly interoperable with internet protocol. So, they depend on another device which has support for both Bluetooth and internet connectivity, for example, a smart phone. First, the consumer device needs to be connected to the user smart phone over Bluetooth. After the connection is successfully established, users can update the device firmware or control the device remotely by using the smart phone application provided by device manufacture.

The measurement tools used in the construction industry are also battery-operated, resource constrained headless devices. They can take measurements with or without user intervention and send those measured data to a connected smart phone. These devices are equipped with laser which can be turned on or off remotely via Bluetooth. Even though these consumer devices are not directly connected to internet, someone can manipulate the device to cause minor injuries, frustrations and affect user experience. A malicious actor can impersonate as measurement tool and inject erroneous data into the system. One can eavesdrop on a connection to steal the measurement data. A malicious agent can connect with the measurement tool and control the laser remotely like devices mentioned in these articles [9], [22] and [8]. To protect the device from these threats, the channel established between the measurement tool and the smart phone should be encrypted and authenticated.

In the smart phone, the Bluetooth host stack is implemented as a middleware component. Even though Bluetooth Special Interest Group (SIG) imposed strict rules in the core specification for implementing the protocol, the real behavior of middleware component may vary from one smart phone to another with respect to operating system, Bluetooth hardware and software versions. The difference exists between middleware components causes issues during connection establishment, pairing and bonding. Since there are no ways for consumer device developer to impose modifications on the smart phone Bluetooth middleware, the measurement device firmware should be designed in such a way to handle all the compatibility issues.

1.2 Problem Statement and Objective

Bluetooth technology is available for more than two decades. The Bluetooth core specification has undergone several changes. The connection, pairing, bonding, encryption, and authentication methods are improved over the time. The Bluetooth key negotiation protocol, authentication process, encryption process and level of security are very limited and differs significantly when we compared it with state of the art web security standards. Even though the most recent Bluetooth standard supports advanced encryption and authentication methods, the old devices are still compatible with modern devices. If an old Bluetooth device connect with a most recent Bluetooth device which supports advanced security standards, then the security standards imposed for the channel established between the devices are downgraded to outdated encryption and authentication methods supported by the old device. Also, the authentication and encryption methods are automatically selected by pairing algorithm based on capabilities of both devices. Usually, the device capability information is exchanged over a non-encrypted channel. Adversaries eavesdropping the connection may influence on paring method selection by modifying the packet containing device capability information [21]. The encryption and authentication procedure varies from one pairing method to another. Also, the level of security offered by each pairing method differ from others.

In resource constrained headless devices, the implementation of highest level of security standard supported by Bluetooth itself is hard. The modern encryption and authentication methods are highly dependent on computational resources. The authentication methods defined by Bluetooth core specification are not based on asymmetric keys used in web security standard. It is completely relied on Human Machine Interaction (HMI) capabilities of the device. Since Bluetooth standard failed to address authentication methods for headless devices, the devices which does not have any additional authentication technique are vulnerable to MITM attack. Due to these limitations, some product manufactures implemented homegrown encryption and authentication techniques. The design and implementation of homegrown security protocol is complex and tedious. It should follow standard security principles otherwise it is vulnerable to serious attacks [9].

This research will analysis state of the art security techniques supported by Bluetooth standard and explore limitations in implementing desired level of security, compatibility concerns, speed, other requirements, and specifications for authentication and encrypting Bluetooth channel established between a headless resource constrained consumer device and a smart phone. A coalition of practical use cases, testing, and security assessment will be used to determine flaws in the proposed solution. The new security techniques presented here will improve security level of authentication and encryption methods used in headless devices by pointing out the gaps in the Bluetooth core specification. The main goal of this research is to design and implement a solution to secure the Bluetooth channel that will leverage security of connected consumer devices.

1.3 Research Questions

- **Research Question 1** – How to implement a simple, secure, reliable paring and re-connection scheme in Bluetooth? What are the shortcomings arising when using Bluetooth pairing and association methods?

  The research must be started from exploring simple pairing and re-connection techniques which are currently used in Bluetooth connected devices. The security vulnerabilities exist in these pairing/re-connection schemes should be identified. During the research work, student must look for other possible authentication forms than the standard ones. In the end, a suitable paring method should be suggested and justify the simplicity in connection establishment, security for this application context.

- **Research Question 2** – Identify challenges in interoperability between devices. Research on methods for solving these challenges.

  In Bluetooth, the services provided by host stack varies from device to device. It is not required to have all the hardware and software components which means, device manufacture can decide the components needed for the target application. If we take the case of smart phone, Android and iOS have completely different software stacks, architectures, interfaces, hardware, and software components. The Bluetooth chip used in mobile devices varies in terms of features, functionality, and capability. Also, there are several Operating System (OS) versions are currently in-use which impose different security standards. Since the initial release of BLE 4.0, there are seven different versions exists till now. Even though, Bluetooth SIG standardized interfaces between layers, the interoperability between devices is still a bottleneck.

- **Research Question 3** – What are the different factors that affects Bluetooth communication throughput? Suggest a suitable configuration for ensuring reliable speed of data exchange between the connected devices.

The data transfer rate of a Bluetooth connection varies based on several factors. Usually, unsecured channel is faster than secured channel because of encryption overhead. There are several pairing methods available in Bluetooth. The encryption key length can be selected upon design. The research will be carried out to determine to what extend data transfer speed will vary based on chosen pairing methods and encryption key length.

1.4 Organization of Thesis

This section describes the thesis organization and gives an overview of each chapter presented in this thesis:

Chapter 2 we will provide important fundamentals, such as security concepts, challenges, terminologies, system architecture and Bluetooth Low Energy. We will also make users familiar with the Bluetooth Low Energy security specification, difference between BLE security and web security, BLE software components, Bluetooth sniffer and packet analyzer.

Chapter 3 presents a overview on previous research work carried out in the field of BLE security. The potential gaps in the previous research will be identified and point out the goal of this thesis work.

Chapter 4 will describe several encryption and authentication solutions which are not specified in the Bluetooth standard.

Chapter 5 presents security topology selection process based on device input-output capability and IoT device cyber security regulation. It will explain the realization of this security topology and how we have implemented. The solution will be tested and results are provided.

Chapter 6 we do threat modelling for the implemented solution. The threat modelling approaches and frameworks are discussed in this chapter. All the threats are evaluated based on Spoofing, Tampering, Repudiation, Information Disclosure, Denial of Service and Elevation of Privilege (STRIDE) and Damage, Reproducibility, Exploitability, Affected users, and Discoverability (DREAD) methodologies. Finally, mitigations are provided for all the threats based on their severeness.

Chapter 7 will describe improvements for Bluetooth Low Energy. BLE security concept drawbacks will be addressed. These drawbacks are explained with appropriate solutions.

Chapter 8 summarise the complete research, it will point out new research findings and give attention to future scope.

2 Background Information

This chapter starts by defining the major concepts and terms associated with cyber security. The system architecture, limitations, constraints will be explained in a detailed manner. We provide an overview of how Bluetooth technology changed over the time. The Bluetooth stack, architecture, hardware, and software components will be explained. We will provide a deep insight on how Bluetooth authentication and encryption techniques varies from state of the art web security standard. Also, we elaborate on current state of Bluetooth eco-system for Android, Apple and resource constrained headless devices.

2.1 Security

According to the research paper [15], the topic of security is more than authentication and encryption of Bluetooth communication channel. Security capability of a product cannot be defined in two discrete levels "secure" and "not secure". It is a point on a scale which starts from no security to very high security. To choose an optimum point on the scale, the security requirements for the product should be carefully assessed. The product security requirements will depend on following factors,

- Functionality of the product
- Data stored, processed and exchanged between devices
- Cyber security regulations

In this thesis, the above three factors will be considered to frame security requirements for the product in fifth and sixth chapter.

2.2 Security Challenges

In this section, we will describe the fundamental challenges should be considered when data exchanged between two or more devices. It includes confidentiality, integrity, authentication, authorization, and privacy.

Confidentiality The act of reading or sniffing the data transmitted over a wired or wireless communication channel is called eavesdropping. It is carried out with a piece of hardware easily available in the market or with a dedicated hardware. If the transmitted and received data packets are not encrypted, then whoever sniffing the packet can easily understand the conversation. It is impossible to ensure the connection between sender and receiver is not sniffed by an adversary. To avoid losing confidentiality, the data exchanged between the intended parties should always be encrypted.

Authentication Adversaries can compromise the device by impersonating as a legitimate device. Even though the channel is encrypted, adversaries can still access and understand the conversation if devices are fails to authenticate with each other. To communicate securely, the intended parties should prove their identity to other devices who they say are they.

Integrity The data transmitted over a non-secure channel can be corrupted due to noise or adversaries can intentionally alter the message. To overcome this issue, the received data packets should have a mechanism to verity the sender and integrity of the received message.

Authorization In a network, each device holds multiple data resources. The access permission to each data resource is solely decided by the device itself. The other devices desire to read or modify protected resources must prove its identity on beforehand.

Privacy There are information which will be disclosed during setting up the connection or while transmitting or receiving data packets. It includes device identity, manufacture information, protocol headers, location, state, encryption, and authentication mechanism agreed between devices. A hacker can collect these data by simply eavesdropping on a connection. This information will leverage attacks executed on a target. To overcome this challenge, the product developer should confirm that the leaked information will not affect user privacy in any circumstances.

The communication link which failed to address above challenges may prone to security attacks. In this research work, a formal process will be carried out in the sixth chapter to identify threats and vulnerabilities exist in the system.

2.3 Terminology in Security Topic

2.3.1 Cryptographic Keys

Cryptographic keys are simply numbers exchanged between devices or derived within the device. It is used for encryption, authentication and signing data packets. There are two types of cryptographic keys,

Symmetric Keys – It is derived between devices or exchanged over Out-of-Band (OOB) technology. The key derived between sender and render result in same secret value shared between devices. They are mostly used for encryption and signing data packets.

Asymmetric Keys – Asymmetric key exist as public-private key pairs. Public key can be shared with others. It is used to generate certificates which will bind a human, an organization, or an entity to a device temporarily or long term. They are mostly used for authentication purpose and signing information.

It is possible to do encryption with asymmetric keys, but the encryption process is slower than encrypting data packets with symmetric keys. Since the symmetric keys are efficient, they are widely adopted for power constrained devices. Also, symmetric key is considered and used in many places of this research work.

2.3.2 Encryption and Decryption

The encryption algorithm takes plaintext and cryptographic keys as input and delivers the ciphertext as shown in Figure 2.1, such process is called encryption. The ciphertext can be transmitted to receiver even in presence of eavesdroppers. Since the data is encrypted, it will not lose its confidentiality.

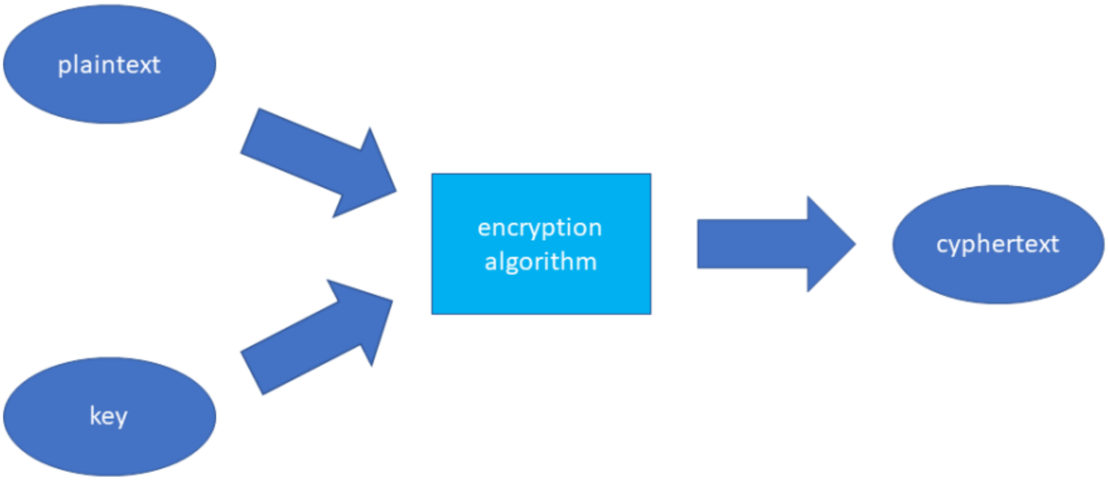


Figure 2.1: Encryption [15]

Decryption algorithms reverse the process of encryption. It delivers plaintext by taking cryptographic key and ciphertext as input.

2.3.3 Message Authentication Code

In some cases, encryption is not a security requirement. The receiver willing to get unencrypted data. Since the data is not encrypted, someone can tamper the message on the way to the receiver. To avoid this issue, messages which are sent in plaintext are attached with authentication code. The authentication code generation algorithms are based on hashing function. It will take symmetric key, plaintext and deliver message authentication code as shown in Figure 2.2. On the receiver side, the authentication code will be verified with the same secret symmetric key shared between devices.

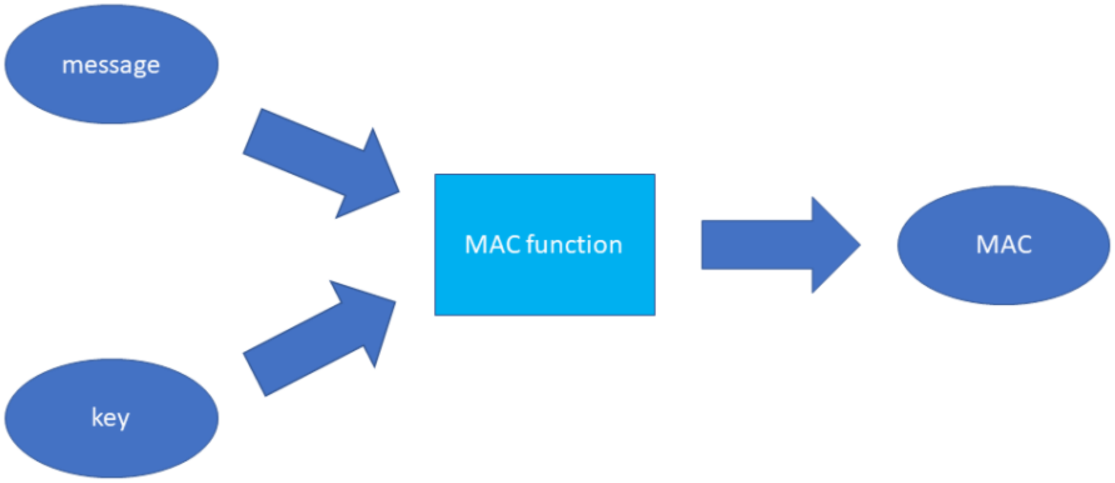


Figure 2.2: Generation of message authentication code [15]

2.3.4 Random Number Generators

Random number generator is one of the crucial components in symmetric key generation process. Because the foremost step in key derivation process is generation of private keys with Random Number Generator (RNG). If the number is less random then it can be easily predictable. There are many technologies exists to generate random numbers. There are certification regulations available to certify the operation of crypto hardware and software components, of which FIPS PUB 140-3 is the most recent one.

2.3.5 Nonce and Replay Attacks

An attack in which the attacker is able to replay previously captured messages (between a legitimate Claimant and a Verifier) to masquerade as that Claimant to the Verifier or vice versa [20]. To protect receivers from replay attacks, a nonce value will be added to each transmitted packet by the sender. The nonce value will be incremented for each data packet sent. Receivers reject packets that contain nonce value repeated more than once.

2.3.6 Diffie-Hellman and Elliptic Curve Cryptography

It is a key agreement protocol used to generate symmetric keys between two devices over an unsecured channel as shown in the Figure 2.3. The shared secret generated with Diffie-Hellman (DH) protocol is safe against eavesdropping but not against MITM attack.

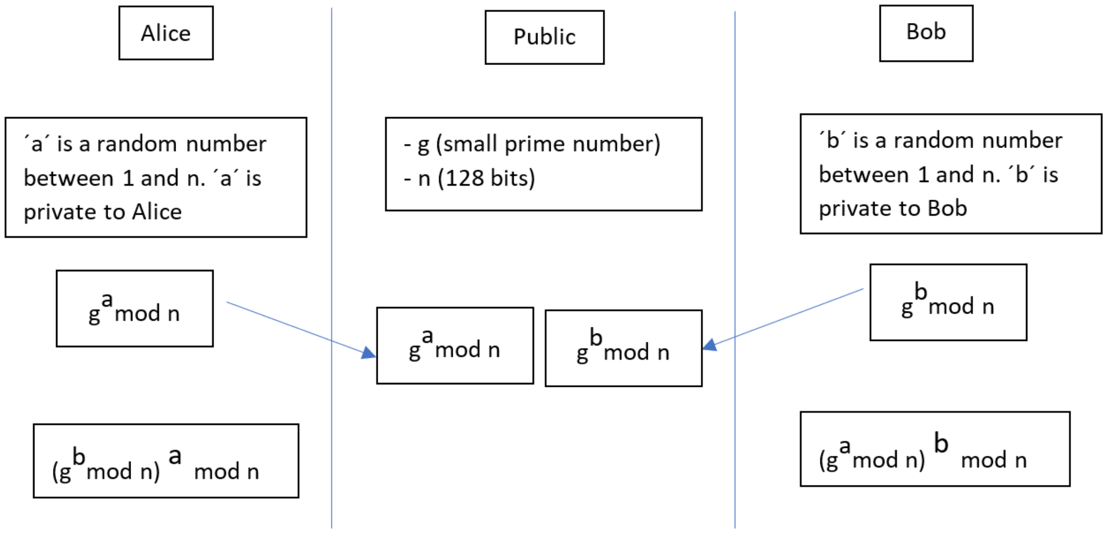


Figure 2.3: Diffie-Hellman key agreement protocol

$$\left[(g^{b} \bmod n)^{a} \bmod n\right] = \left[(g^{a} \bmod n)^{b} \bmod n\right] = \text{Shared secret}$$

The shared secret is the symmetric key derived between Alice and Bob. In Elliptic curve Diffie-Hellman (ECDH) protocol, elliptic curves are used instead of modulus operation. The number of computations in ECDH protocol is lower than DH algorithm. Also, the key generated from ECDH protocol is much stronger than DH algorithm.

2.4 System Architecture

The target of the project is to plan, research on simple secure pairing schemes and methods for ensuring interoperability between BLE connected devices as shown in the Figure 2.4. One of those devices is a measuring tool, which has an application processor and a Bluetooth chip. It is a battery-operated headless device. The embedded software stack on application processor is either bare metal or embedded Linux. The measuring tool application stack has a layer called Bluetooth manager which provides interface to Bluetooth application. The BLE application software is based on Real-time operating system (RTOS). Another device is a Bluetooth enabled smart phone. The smartphone could be an Android/iOS device.

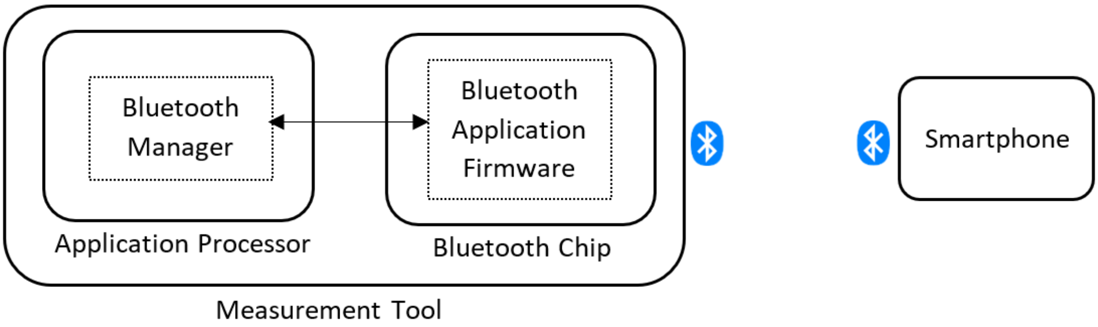
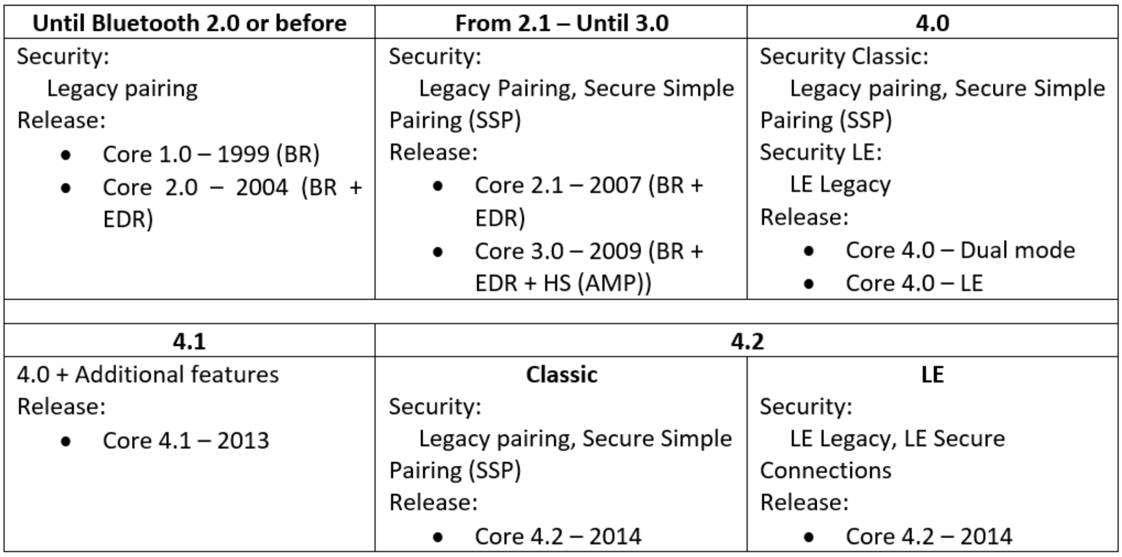


Figure 2.4: System architecture

2.5 Bluetooth

There are many Bluetooth standards, specifications, and protocols currently in use. The difference between specifications are shown in the Figure 2.5. The specification and standards are documents which contains information to regulate entities who are adhered to follow such specifications and standards for making services and products which are interoperable, ubiquitous for users without compromising their security. In general, these standards and specifications will change over the time. Bluetooth mesh protocol is not considered in this research project. The specifications and standards released before Bluetooth 4.0 has Base Rate (BR), Enhanced Data Rate (EDR) and High Speed (HS) controllers. Now they are called as Bluetooth classic. To support wireless communication for low power devices, Bluetooth introduced LE mode from core specification 4.0. The power consumption in BLE is lower than Bluetooth classic. A Bluetooth chip can be classic, LE or dual mode. In dual mode, the chip software stack supports both classic and LE. Usually, smart phone supports dual mode Bluetooth.



Until Bluetooth 2.0 or before	From 2.1 – Until 3.0	4.0
Security: Legacy pairing Release: • Core 1.0 – 1999 (BR) • Core 2.0 – 2004 (BR + EDR)	Security: Legacy Pairing, Secure Simple Pairing (SSP) Release: • Core 2.1 – 2007 (BR + EDR) • Core 3.0 – 2009 (BR + EDR + HS (AMP))	Security Classic: Legacy pairing, Secure Simple Pairing (SSP) Security LE: LE Legacy Release: • Core 4.0 – Dual mode • Core 4.0 – LE

4.1	4.2	
4.0 + Additional features Release: • Core 4.1 – 2013	**Classic** Security: Legacy pairing, Secure Simple Pairing (SSP) Release: • Core 4.2 – 2014	**LE** Security: LE Legacy, LE Secure Connections Release: • Core 4.2 – 2014

Figure 2.5: Bluetooth history

In Bluetooth classic, there are two pairing methods available, of which Legacy pairing is not safe against passive eavesdropping. On the other hand, Secure Simple Pairing (SSP) provides protection against passive eavesdropping but prone to MITM attacks. There are some association models exist in Bluetooth classic which provides limited protection against MITM attacks. Since the measuring tool supports only LE mode, Bluetooth classic is not considered for this research work. Like Bluetooth classic, LE legacy pairing in BLE is not safe against eavesdropping on the channel. LE secure connections pairing in BLE is similar to SSP in Bluetooth classic.

2.5.1 Bluetooth Low Energy

In Bluetooth, the underlying hardware can be shared by both Bluetooth classic and BLE stack. But the software stack and protocol are different for classic and LE. The BLE hardware capability and supported features will vary between different chips and vendors. The software layers supported by a chip also varies. Since the Bluetooth specification is not strictly adopted by chip manufactures, product developers can check hardware capability, functionalities, supported software layers and features of a chip in this site [5]. This site is maintained by Bluetooth. A typical BLE stack and its components are shown in the Figure 2.6.

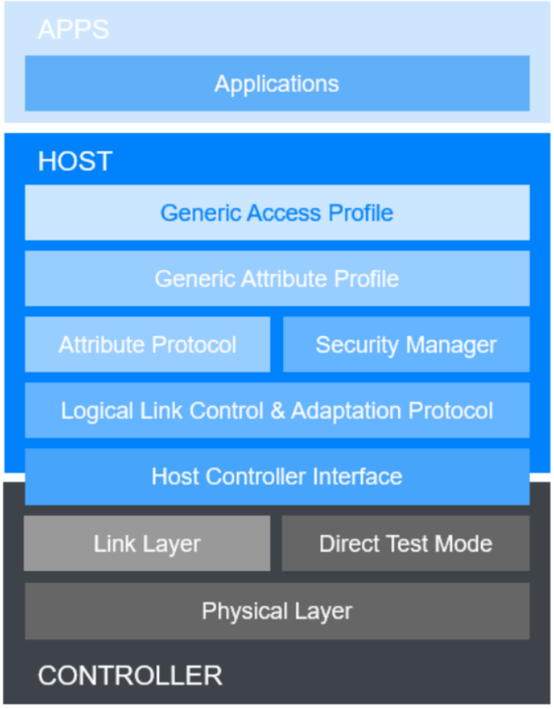


Figure 2.6: Bluetooth LE stack [15]

The software run on the controller is a proprietary component. It is provided by chip vendor. The link layer in the controller is responsible for generation of cryptographic keys, encryption, and decryption of data packets. The generated keys are not stored in the controller. It is stored on the host layer, if needed, link layer can request the host layer to provide a copy of cryptographic keys for a bonded device. The link layer can do tasks such as sending and receiving advertisement packets, filtering incoming devices autonomously without any support from host layer to reduce power consumption of the Bluetooth product. The host and controller exchange commands and data packets over the standard Host Controller Interface (HCI). The implementation of HCI is dependent on architecture of the product. This interface can be a virtual interface if both host and controller available on a single System on a chip (SoC). Otherwise, HCI will be realized as a standard physical bus such as Universal Serial Bus (USB) and Serial Peripheral Interface (SPI).

The logical link control and adaptation protocol in host layer is responsible for packet segmentation, congestion, and flow control. The security manager component manages pairing and bonding process. The attribute protocol manages sending and receiving objects present in the attribute table. In BLE, the collection of related services is called profile as shown in the Figure 2.7. The collection of related characteristics is called service. The characteristics are resource containers. In BLE, any number of profiles can be newly created to meet product functionality. The Generic Attribute Profile (GATT) is the generic definition of the profile. It is used by the application to exchange data between two connected devices. The Generic Access Profile (GAP) handles advertisement and connection related procedures in BLE.

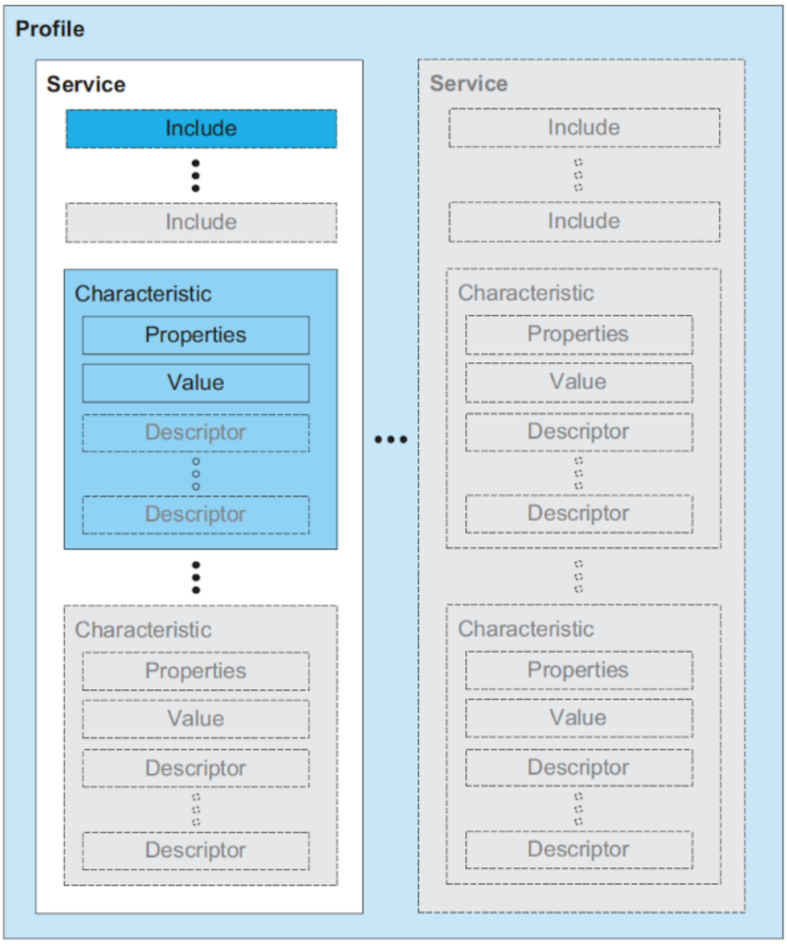


Figure 2.7: Profiles, services, and characteristics [17]

In BLE, there are four device roles available. They are,

1. Central
2. Peripheral
3. Broadcaster
4. Observer

In central role, a device can connect to multiple peripheral devices which implements GATT server. Central devices are GATT clients. The connection between a central device and a peripheral is called connection-oriented communication. In broadcaster role, the device can send authenticated or unauthenticated messages to other devices. Devices which are in observer role can receive messages from all the broadcasters. The communication between a broadcaster and an observer is called connection less communication. In general, a device can support more than one role at a time. Based on available different device roles, there are three different communication typology available as shown in the Figure 2.8. The point-to-point communication topology is considered in this research project. So, the smart phone performs central role, and the measurement tool is implemented as a peripheral device.

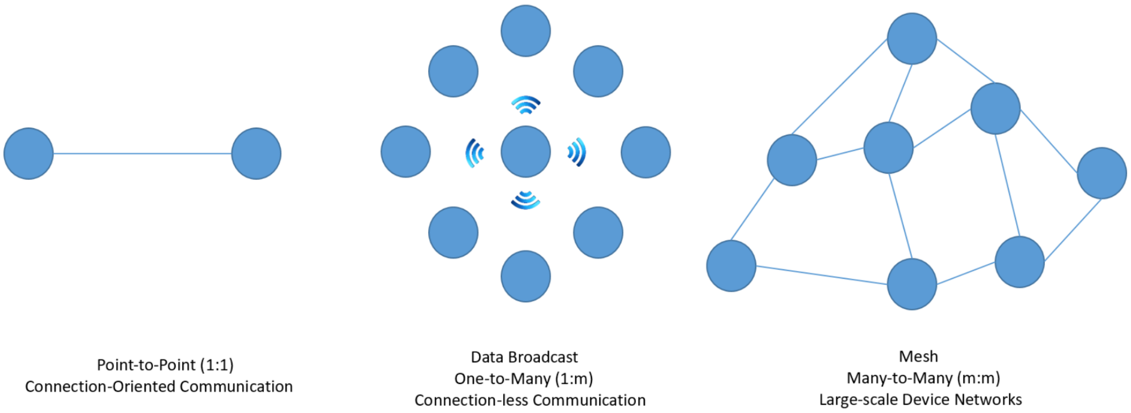

Point-to-Point (1:1)
Connection-Oriented Communication
Data Broadcast
One-to-Many (1:m)
Connection-less Communication
Mesh
Many-to-Many (m:m)
Large-scale Device Networks

Figure 2.8: BLE communication typology [10]

2.5.2 Bluetooth Low Energy Security

In point-to-point communication, the device which is in peripheral role implements GATT server. The central device will be GATT client. Attributes are the fundamental building blocks of GATT server. The GATT layer is built on top of Attribute Protocol (ATT). Each attribute has four properties. They are attribute handle, permission, type, and value. A product developer can define access permission to each attribute's data based on the product security requirements. There are five different attribute permissions available in BLE. They are,

1. Readable or non-readable
2. Writable or non-writable
3. Encrypted or non-encrypted
4. Authenticated or non-authenticated

5. Authorized or non-authorized

The attribute can be accessed and read by GATT client only if permission is set readable. If attribute permission is set as non-writable then the GATT clients have no rights to modify content of the attribute resource. The encryption and authentication properties define attribute security. To access an attribute whose permission is set as encryption and authentication, then the GATT client should already be successfully paired and bonded to access the resource instantly. Otherwise, insufficient encryption or authentication event will be generated that will eventually trigger security manager to initiate pairing process. The act of defining security permissions on data level is very convenient for product developers and less prone to errors. But Bluetooth specification does not allow an attribute to meet complete security requirements on data/resource level. For example, the authentication process in 'Just Works' pairing method is simply bypassed even though attribute has authentication permission. Since security is one of the main concerns of this research, this topic will be discussed in detail in the upcoming chapters.

The Bluetooth specification defined four security modes. In each mode, there are many security levels. Few embedded Software Development Kit (SDK) allows product developers to specify mode and security level. In BLE, GATT server can provide security mode and level information to GATT client via LE GATT Security Levels Characteristic (SLC). Using this characteristic, GATT clients can initiate pairing process in advance before accessing attributes. This characteristic is defined in the Bluetooth specification, but the implementation is optional. The attribute permission for SLC is set to no encryption and no authentication, so that central devices can access this information before initiating pairing flow. Since the security definition is spread over different stack layers, the selection of configuration is completely dependent on vendor implementation. The available modes and security levels are,

LE Security Mode 1

1. No security (No authentication and no encryption)
2. Unauthenticated pairing with encryption
3. Authenticated pairing with encryption
4. Authenticated LE Secure Connections pairing with encryption using a 128- bit strength encryption key

LE Security Mode 2

1. Unauthenticated pairing with data signing
2. Authenticated pairing with data signing

LE Security Mode 3

1. No security (no authentication and no encryption)
2. Use of unauthenticated Broadcast_Code
3. Use of authenticated Broadcast_Code

LE Security Mode 4

1. Secure Connections Only mode i.e., security mode 1 level 4

In BLE, the cryptographic key materials are generated during pairing process and stored for subsequent re-connection. The two devices are said to be bonded only if they made trusted relationship and reuse their stored keys for setting up encrypted channel whenever needed. In BLE, three types of keys are required to feature encryption, authentication, and privacy. They are,

1. Long Term Key (LTK)
2. Connection Signature Resolving Key (CSRK)
3. Identity Resolving Key (IRK)

Of which, the LTK is used for encrypting the channel established between two devices. In BLE, AES-CCM algorithm is used for encryption, decryption and data signing. The CSRK is used to sign data. The data transferred over the encrypted channel are not signed because Cyclic Redundancy Check (CRC) mechanism is sufficient to ensure message integrity. In some cases, data is transmitted over unencrypted channel are signed with CSRK. It is possible to sign the attribute value alone to avoid packet header overhead.

There are four types of Media Access Control (MAC) address available in BLE as shown in the Figure 2.9. A Bluetooth device can use either public address or random static address at a time. The public MAC address and random static address of a device is fixed for the whole lifetime of the product. A hacker may use device MAC address to track, record device activity and proximity information. To avoid such problems, IRK are used to stop unknown devices from using other devices MAC address without its permission. The resolvable private address changes periodically and visible to all known and unknown devices. It can be resolved back to public or random static address with an IRK. The devices which are already authenticated and exchanged their IRK during key distribution phase can track progress of the device. A Bluetooth device can use either resolvable private address or non-resolvable private address at a time. Like resolvable private address, the non-resolvable private address also changes periodically. The device which has only non-resolvable private address will not be tracked at all.

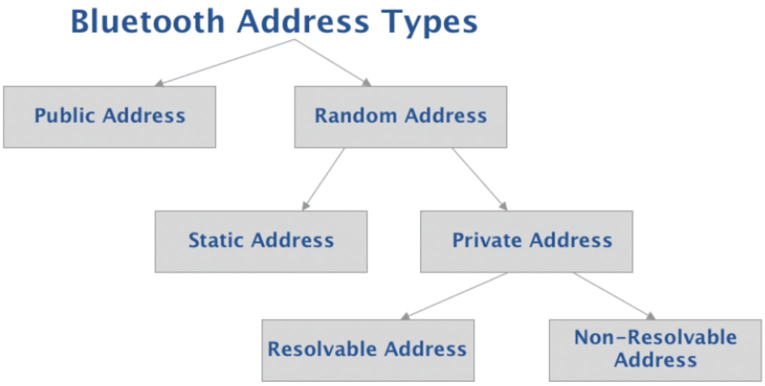


Figure 2.9: Bluetooth device address types [16]

There are two distinct pairing methods available in BLE. They are,

1. LE Legacy Pairing

2. LE Secure Connections

Each pairing method offer multiple association models. Every association model provides a unique methodology for encryption and authentication of Bluetooth link. There are three association models available in the LE legacy pairing. They are,

1. Just Works
2. Passkey Entry
3. Out of Band

The four association models available in the LE secure connections are listed below,

1. Just Works
2. Passkey Entry
3. Numeric Comparison
4. Out of Band

Even though the first two association model names are same in both pairing methods, the actual process of cryptographic key generation differs in every association model. According to Bluetooth specification, the pairing method and association model are selected automatically based on a predefined algorithm. This algorithm takes a list of parameters as input. This parameter list is called pairing feature exchange. In BLE, the process of pairing is carried out in three phases as shown in the Figure 2.10. The phase one will be started if any one of the device initiates pairing process. Before pairing, both devices should establish a successful connection. The device which initiates connection process is called initiator. Another device is called responder. In BLE, there are four methods available to send connection request from initiator. They are,

1. Connect
2. autoConnect
3. Connect with preferred PHY
4. Connect and Bond immediately

With connect method, the central device can send a single connection request to the peripheral. The connection requests are sent continuously for a specific period in autoConnect method. In Connect with preferred PHY, the central can request the peripheral to start connection with a specific physical layer. In the last method, the central device initiates pairing process immediately after a successful connection. All these methods are available in Android devices. In iOS, only the second method is available.

Once the connection is successfully established, the initiator device is called master and responder device becomes slave. The pairing process can be initiated by master in four different ways. They are,

1. Master will start pairing process upon a new connection
2. Master starts pairing flow once Out of band authentication data is available

3. Master initiate pairing after insufficient encryption or authentication error received from the slave

4. Slave device (peripheral device) may request central device to start the pairing process

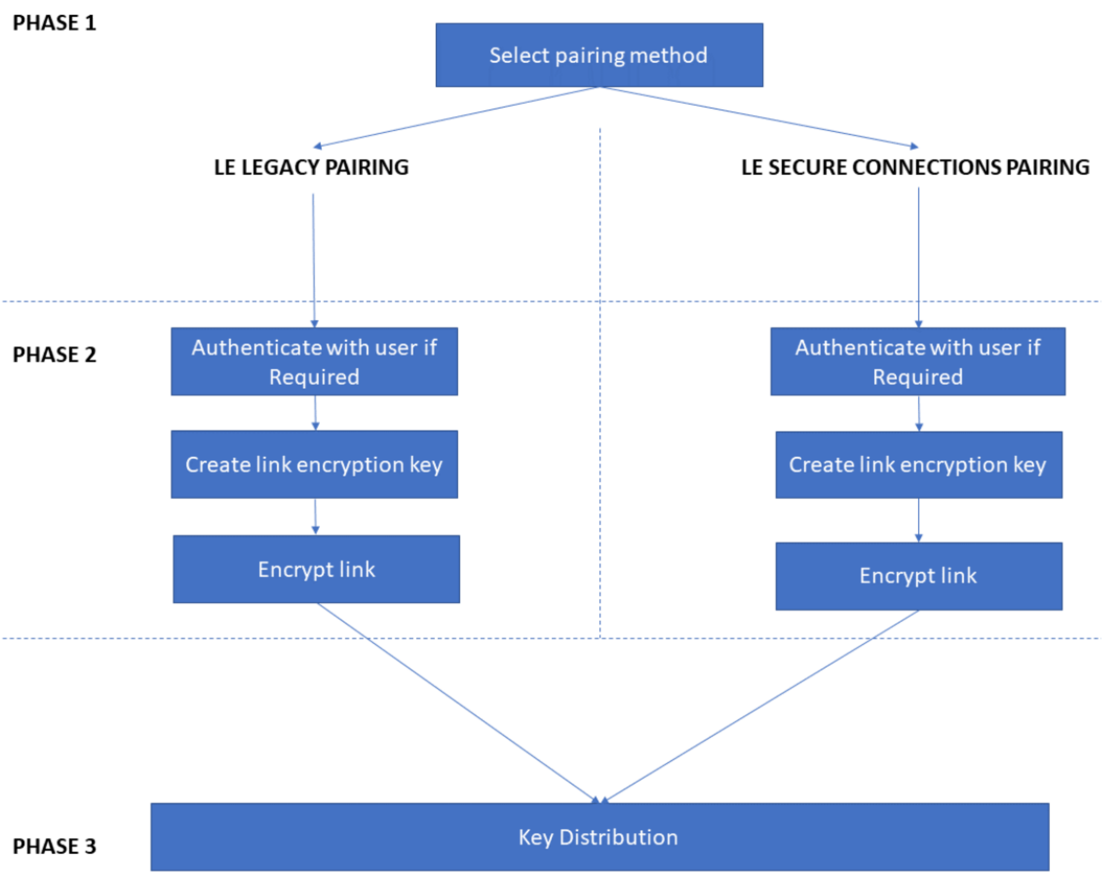
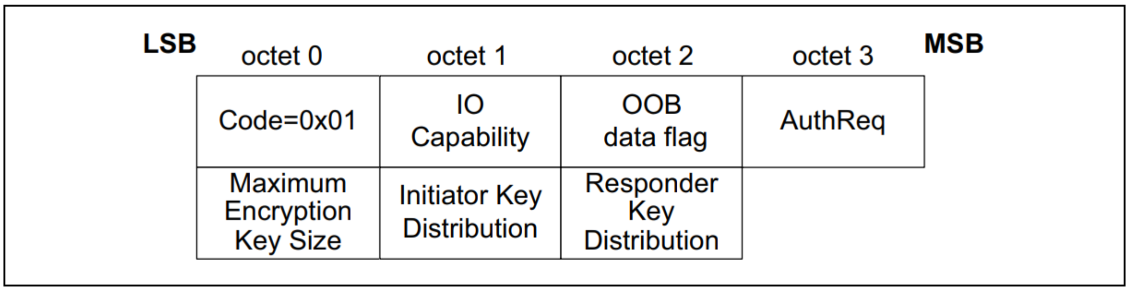

PHASE 1
Select pairing method
LE LEGACY PAIRING
LE SECURE CONNECTIONS PAIRING
PHASE 2
Authenticate with user if Required
Create link encryption key
Encrypt link
Authenticate with user if Required
Create link encryption key
Encrypt link
PHASE 3
Key Distribution

Figure 2.10: BLE pairing flow [15]

The first step in phase one is paring feature exchange. The packet which has pairing feature exchange information is called pairing request packet as shown in the Figure 2.11.



LSB	octet 0	octet 1	octet 2	octet 3	MSB
	Code=0x01	IO Capability	OOB data flag	AuthReq	
	Maximum Encryption Key Size	Initiator Key Distribution	Responder Key Distribution		

Figure 2.11: Pairing request packet [4]

In peripheral devices, the pairing feature exchange information can be configured by product designer. During pairing feature exchange, the following parameters and their corresponding values are exchanged between two devices.

1. Input-output capability
2. Out of Band availability flag
3. AuthReq
4. Maximum encryption key size
5. Initiator key distribution field
6. Responder key distribution field

Input-output capability This parameter provides information about HMI capability of a device. This parameter can take five different values. Those values are,

1. DisplayOnly
2. KeyboardOnly
3. DisplayYesNo
4. NoInputNoOutput
5. KeyboardDisplay

In DisplayOnly, the device can only be able to show a piece of information on the display. The information may disappear after a predefined period. User can give numeric or text input in KeyboardOnly mode. User will be allowed confirm an event or compare a piece of information shown on the display in DisplayYesNo mode. NoInputNoOutput means both display and keyboard are not available, and this type of device is called headless device. In KeyboardDisplay, user can read the information provided by the device and give inputs with a numeric keypad or a standard keyboard.

Out of Band availability flag The Out of Band association model can be implemented in several ways. Some examples of Out of Band technology are Near Field Communication (NFC) and QR code. The product developers can use host stack Application Programming Interface (API) to generate or set Out of Band authentication data. This flag takes two distinct values which describes whether the device have Out of band authentication data or not.

AuthReq AuthReq is an entity which comprises of multiple parameters. The five parameters combined in AuthReq are listed below,

1. Bonding flags
2. MITM field
3. Secure Connections field
4. Keypress field
5. CT2 field

The bonding flag describes whether the device will store cryptographic materials or not. In the peripheral device, the product developer has access only for bonding flags and MITM field. The other fields are directly controlled by the host stack. The MITM field conveys device willingness for protection against MITM attack. The device will not be protected against MITM attack by only setting this field to true. The Secure Connections field provides information about the device whether it supports LE Secure Connections or not. The Keypress field is used when passkey entry association model is selected for authentication and encryption. If this field is set to one by both devices, then the device which receives passkey from user will send notifications for each keypress to its peer. The CT2 field describes whether the device supports link key conversion function (h7) or not. The h7 function is based on AES-CMAC used to convert link key from LE transport to BR/EDR transport.

Maximum encryption key size This parameter conveys maximum encryption key size supported by the device. The parameter value ranges from seven to sixteen octets. In Bluetooth, the maximum size of symmetric key is sixteen octets. In the peripheral device, product developers can choose a value which will be send during pairing feature exchange.

Initiator key distribution field This field comprises multiple information which is used during phase three of the pairing flow. The phase three is a key distribution phase. In LE legacy paring, the following keys and other resources are exchanged between initiator and responder,

1. LTK
2. Encrypted Diversifier (EDIV) and Rand
3. IRK
4. Public device or static random address
5. CSRK

In LE secure connections, the following key materials are exchanged,

1. IRK
2. Public device or static random address
3. CSRK

There are four fields available in the key distribution format as shown in the Figure 2.12. The IdKey and SignKey purpose are same for LE legacy pairing and LE secure connections. The IdKey indicates that the device will distribute both IRK and public device or static random address. The SignKey indicates that the device shall distribute CSRK. The purpose of EncKey and LinkKey are different for LE legacy pairing and LE secure connections. In LE legacy pairing, setting the field EncKey to true means that the device will distribute LTK, EDIV and Rand to its peer. The LinkKey field is ignored for LE legacy pairing. In LE secure connections, the field EncKey is ignored. If LinkKey field is set to one, then the link key for BR/EDR will be generated from LE LTK.

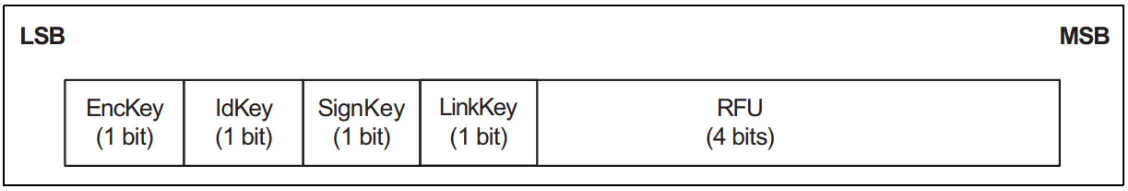


Figure 2.12: LE key distribution format [4]

Both initiator and responder send eight bits of information. The first four bits indicates the key materials shall be distributed by the initiator. The next four bits also send by initiator which describes the list of keys initiator expects from the responder during key distribution phase. The product developer has direct access to modify these eight bits in the peripheral device.

Responder key distribution field This field is same as Initiator key distribution field, but this is sent by the responder device during pairing feature exchange.

Once the paring feature exchange is completed then an algorithm decides the pairing method and the association model based on the information exchanged during pairing feature exchange step. In phase two, the process of generating encryption key and device authentication will be carried out. The authentication process may require user interaction on the device. In some association models, the cryptographic key generation step depends on device authentication i.e., the device authentication should be completed before proceeding with key generation for encryption. In some scenarios, the authentication step may fail which led to immediate termination of the pairing process. If the authentication step and the cryptographic key generation step are successful, then the link established between two devices are encrypted at the end of phase two. The phase three is a key distribution phase. In key distribution phase, both devices will exchange keys and other resources based on the initiator and responder key distribution fields in the pairing feature exchange.

2.6 BLE Security versus Web Security

In this section, we compared security mechanisms and protocol supported by BLE against web security technologies. The fundamental difference between devices connected over internet and Bluetooth devices are,

1. The two devices communicating over internet may exist anywhere in this world. Since Bluetooth is a short-range communication technology, the devices must exist close proximity to one another to establish connection.

2. The devices such as servers, personal computers have good computing power, HMI capability and no constraints over energy consumption. We can classify Bluetooth devices into two categories. The first category of devices does not have limitation over energy consumption for Bluetooth and HMI capability, for example, smart phones. The second category is highly resource constraint. It does not have HMI.

The two differences mentioned above have impact on encryption and authentication methods supported by both technologies. Also, the threats involved in these categories of devices differs in some cases.

2.6.1 Selection of Security Mechanism

In the IT systems, the communication begins by negotiating available key generation, encryption, and authentication methods. In the end, both devices should agree on a common standard security practice. In this system, at least anyone of the device should be updated with most recent security standards. Otherwise, devices may agree to use an outdated security mechanism so a malicious agent may intercept the connection. If the devices are unable to find a common security practice supported on either side, then the connection will be terminated immediately.

In BLE, both central and peripheral exchange their capabilities during pairing flow phase one. This is called pairing feature exchange. Since Bluetooth standard gives significant importance to backward compatibility, the encryption and authentication methods selected based on paring feature exchange shall not meet product security requirements. Unlike web security, the chosen security mechanism will be used in phase two of the pairing process if product developer does not take any further measures. At least the peripheral embedded stack provides mechanisms such as LE secure connections only mode and minimum acceptable encryption key size protects Bluetooth devices from threats to a limited extent.

2.6.2 Authentication

The devices connected over internet uses certificates to prove their identity. The certificates are generated from asymmetric keys such as public and private keys. The asymmetric keys are explained in section 2.3.1. Usually, public key from a pair can be signed by a private key from another pair. This can be scaled to any extent that result in chain of trust. In server-client model, server shall distribute its certificate to client. Client verifies the certificate chain. If the sever certificate is valid then the signature must be traceable back to a root certificate signed by a legitimate organization. If the verification fails, then the connection will be terminated immediately. Usually, the root certificate is available in the device local storage. The certification verification results are valid only if the root certificates exist in the local storage are legitimate. Otherwise, a hacker can easily carry out MITM attack using invalid certificates. Even though clients can be authenticated with login information, still they should distribute its public key to server. Because the authentication process is not completed yet.

It is very clear that initially devices distribute their certificates over a non-encrypted channel. Since the server certificates are available to public, an attacker can provide a copy of certificate from a legitimate organization to carry out device impersonation attack. To avoid such issue, the communication between the server and the client are encrypted with asymmetric keys until symmetric key is available. After the certificate verification, the connection is encrypted with public key. The packets will be decrypted by an entity who possess private key. The adversaries can do passive eavesdropping on the channel, but they cannot decrypt the packet since the private key is not available to them.

The process of generating asymmetric keys, verifying certificate chain, access to certificate authorities, encrypt and decrypt packets using asymmetric keys are computation intensive operations. This type of authentication process is not suitable for BLE. By considering the first fundamental difference mentioned above, it is possible to invent novel authentication methods suitable for any short range wired or wireless communication. The passkey entry and numeric comparison are such association models i.e., authentication methods defined in Bluetooth specification. In this research work, the BLE authentication methods will be examined, and further possible authentication methods will be explored.

2.6.3 Encryption

As described in section 2.6.2, initially the connection with the sever will be encrypted using asymmetric keys later it is switched to symmetric keys. There are several symmetric key generation methods available. Of which, ECDH is a symmetric key generation method available in web security and BLE security. In web security, the typical length of encryption key is 128 bits, nowadays 256 bits is mostly preferred. The connection is considered insecure if the key size is less than 80 bits. In BLE, the encryption key size can range from 56 bits to 128 bits. In the peripheral device, the product developer can specify the encryption key size, but the pairing algorithm selects a minimum value from maximum encryption key size specified by both devices during pairing feature exchange. So, the smallest encryption key length in web security is larger than the minimum possible key length in BLE security.

2.6.4 Privacy

The device IP address is visible to all other devices in the network. The IP address can be protected by using virtual private network, but the address is still visible to internet service provider. In web application, websites usually leave their footprint on clients which is used to track user activity and shared with other websites for marketing purposes. In BLE, the device MAC address can be protected with IRK and resolvable private address. Apart from the device address, a Bluetooth device may lose its privacy over transmitter signal strength.

2.6.5 Losing Cryptographic Keys

In web security, the process of authentication and encryption is ubiquitous. The certificate used for authentication can be regenerated for any number of times if private key is lost or certificate validity is completed. There are several ways exists to recover login information for clients. So, the devices on the internet does not encounter serious user experience issues because of missing cryptographic keys and login information. In BLE, the process of pairing need to be carried out from the beginning that may require user interaction on the device. This may affect user experience severely if the bonding information is not properly stored and maintained for a long time.

2.7 Low Power BLE Devices

The low power BLE devices are often embedded devices tailored for a particular application. The components exist in the controller and host stack are highly optimized to reduce power consumption. Unlike smart phones, the peripheral SDK offers more flexibility for the implementation of target application. The main advantage is that the application can be implemented in such a way that the security configuration of peripheral device impose significant impact on selecting encryption and authentication methods. BLE devices can filter new connections at the link layer. This reduces number of events in host stack thereby reduces power consumption.

2.8 BLE in Smart Phones

The Bluetooth chipset used in smart phones are often supports dual mode. The Bluetooth classic software components overlap BLE stack on some areas. For example, the link key for Bluetooth classic can be derived from LE link key. This may have impact on product security so this should be considered while accessing the product threat and vulnerabilities. The Bluetooth functionality and supported features of every smart phone differs from others. Even though the Bluetooth system is very large, the supported features and Bluetooth security are confined to smart phone operating system. In both Android and iOS, the operating system does not allow product developers to access and configure Bluetooth security parameters. So, the BLE security configuration are selected by the operating system based on Bluetooth chip capability, available hardware resources and meet overall security requirements of the smart phone. This has significant impact on pairing feature exchange parameters value and limits access to some security features defined in the Bluetooth specification. In this research, Android and iOS operating systems are considered for designing, implementation and testing of peripheral devices. The stock Bluetooth user interface that comes with the smart phone are very limited for BLE. They are,

1. Scanning and pairing
2. Deletion of Bonding information

Also, the above listed functions are supported only for specific device appearance mentioned in this document [3]. For peripheral devices which does not support any appearances listed in the document, a generic or specific BLE application should be installed on the smart phone to connect and pair with the measurement tool. There are several generic BLE applications available for both operating systems. These applications are not designed for a specific purpose so it can be used to test and debug the functionality of a peripheral device. In this research, the nRF Connect application from Nordic Semiconductor will be installed on both operating systems.

Since the Bluetooth functionality is confined to operating system, the supported API for application development is different for both operating systems. The available API in iOS is a subset of API available in the Android. For example, the access to smart phone bonding list and deleting bond information directly from smart phone application are not possible in iOS devices.

2.9 Sniffer and Packet analyzer

Sniffer is a piece of hardware used to capture packets sent over the air. There are several types of Bluetooth sniffers available in the market. In this research, nRF sniffer as shown in the Figure 2.13 from Nordic Semiconductor is used to test and debug Bluetooth connection. To analyze the packets captured by sniffer, Wireshark application is used in this project.

Figure 2.13: nRF sniffer hardware

2.10 Debugging Encrypted Connection

It is not possible to capture Bluetooth traffic after the connection is encrypted. An encrypted Bluetooth channel can be captured and analyzed in two different ways. The first method is already defined in the Bluetooth specification. In first method, debug keys mentioned in the core specification is used on anyone of the device instead of public and private keys generated by ECDH algorithm. Another device should allow the usage of debug keys otherwise the pairing process will fail. Finally, the private key given in the core specification should be provided to the packet analyzer to decrypt packets. The second method is only possible if the device SDK allows to read LTK stored in the host stack. In this method, the LTK should be distributed to the packet analyzer for the decryption. The difference between two methods is, the LTK should be provided to the packet analyzer before the connection starts in the second method. Since the LTK will not be available until pairing is completed, so the packets involved in the pairing flow will not be decrypted at all in second method. In first method, all the packets including pairing process can be decrypted without any limitations.

3 State of the Art

In the previous research work [2] and [7], Bluetooth Low Energy security mechanisms are analyzed based on threats and vulnerabilities exist in the Bluetooth products. But they failed to review Bluetooth specification and identify actual cause for these threats. In this research work, we are going to examine security mechanism agreement, connection and pairing protocols defined in the core specification. The BLE security concepts such paring feature exchange, LE security modes and attribute permissions are also considered for the research to identify flaws in the actual implementation. The research paper [24] claims that only passkey and numeric comparison association models are feasible in secure connection only mode. In this thesis, we are going to configure and test secure connection only mode for just works association model.

A new security framework for wearable devices is proposed in the research paper [23]. But this paper failed to address authentication mechanisms for the wearable devices. Without a proper authentication procedure, the encrypted link established between two devices are prone to MITM attacks. A unique authentication mechanism for headless devices is proposed in the research work [19]. In this approach, at least two paired devices are needed in the network to successfully authenticate a third device. This paper failed to address the pairing procedure for the devices initially formed the network. This authentication mechanism is not considered for this thesis because point-to-point connection topology is chosen in this project. A new set of authentication methods for headless devices were discussed in the research articles [14] and [13]. It is based on spatiotemporally properties exist between devices that are closely located to each other. They are vibration, electromagnetic radiation, capacitive and inductive measurements. For these authentication methods, the device should be equipped with appropriate sensor to measure spatiotemporally properties. The challenges are device context, distance between devices and sensor accuracy. According to the research article [14], a complete 128-bit key is generated within twenty-four seconds. This process needs to be repeated until a common secret key generated between devices. The antenna orientation has significant impact on key accuracy. An asymmetric key based authentication technique is proposed in the research work [12]. The asymmetric key length is reduced drastically. To authenticate new devices, internet connection is required for downloading certificate chains. This method also failed to address encryption procedure. In this project, a set of new authentication methods will be proposed based on limited input-output capabilities of the headless device. Also, a suitable authentication method will be selected and combined with a best possible encryption mechanism to meet product security requirements.

4 Methodology

In this chapter, we propose new solutions for securing the channel established between BLE devices. An ideal solution will be presented to address the problem. In some cases, it is hard to implement proposed solution with respect to current state of the Bluetooth specification. Such difficulties will be explained. Apart from the standard Out of Band technologies, some new ideas are suggested for the device authentication.

4.1 Link Encryption

There are several channel encryption methods available in BLE. Even though these methods differ from one to another, the speed of data transmission is same. Because the key generation process is different in some cases, but the ultimate key size is same in all the methods. The entropy of key generated in LE legacy pairing passkey entry association model is the lowest value among all other association models in . The 128-bit key generated based on public key cryptography is the strongest of all. Also, we confirmed that the crypto modules exist in the project hardware are certified according to Federal Information Processing Standards (FIPS). So, the public key cryptography based encryption key generation method is selected for the implementation.

4.2 Link Authentication

The authentication methods defined in the Bluetooth specification demands HMI capabilities heavily. In the second chapter section 2.5.2, there are five different types of input-output capabilities are mentioned. Of which, the authentication methods based on DisplayOnly, KeyboardOnly, DisplayYesNo and KeyboardDisplay provides mechanism to detect MITM attacks. The connection will be aborted immediately if there is a MITM attack. These kind of solutions affects user experience because the pairing process need to be carried out once again from the beginning if there is a MITM attack. Even though these authentication methods are comparable with respect to input-output capabilities, the level of authentication varies in each association model for the same input-output capabilities. The NoInputNoOutput capability does not provides any means of authentication. In other words, it is impossible to detect MITM attacks in such input-output capability.

The input-output capabilities such as DisplayOnly, KeyboardOnly, DisplayYesNo and Keyboard-Display are not suitable for headless device at all even if anyone of the device belongs to these input-output capabilities. Since the measurement tool is a headless device, it belongs to NoInput-NoOutput category. The Bluetooth connection established between the measurement tool and smart phone is prone to MITM attack.

4.3 Ideal Solution

The encryption and authentication challenges are not specific to Bluetooth. It is common to all kind of wired and wireless communication. In this section, we addressed the problem of authentication and encryption in general as shown in the Figure 4.1.

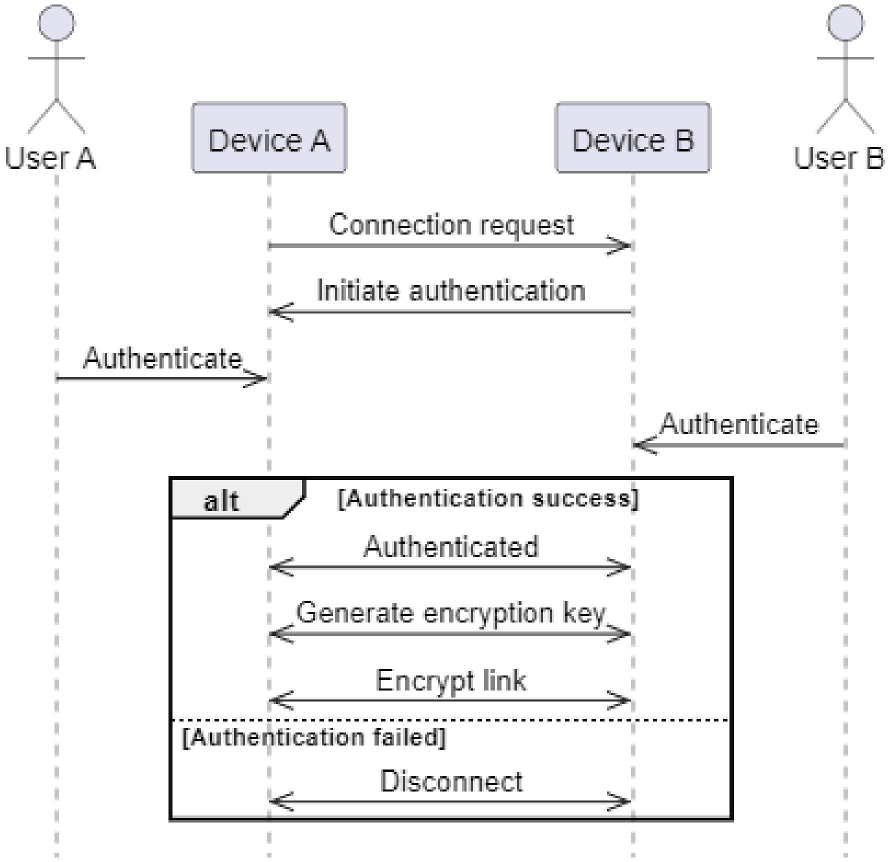


Figure 4.1: Authentication and encryption sequence diagram

Anyone of the devices can initiate connection request. In BLE, both devices can access data resources based on attribute permission without encrypting and authenticating the link. In web services, the connection is encrypted and authentication before any transaction. The authentication mechanism depicted in the Figure 4.1 may expect user interaction in some scenarios. But the asymmetric key based authentication methods are ubiquitous, and it does not need user interaction. The encryption and authentication processes can be carried out together or separately which is completely dependent on security mechanism in use. The encryption and authentication process in LE legacy pairing passkey entry method cannot be separated at all. Also, the order of processes cannot be changed. On the other hand, the encryption and authentication process in LE secure connections numeric comparison method can be executed separately i.e., the encryption and authentication processes are independent to one another. In asymmetric key cryptographic system, these processes cannot be separated at all.

4.4 Bit Flip Authentication

Once Bluetooth is enabled, the measurement tool sends advertisement packets to all nearby devices. Now, user must complete the authentication step. There are three authentication methods as follows,

1. Press authentication button multiple times
2. Press and hold the authentication button
3. Display a random number on smart phone

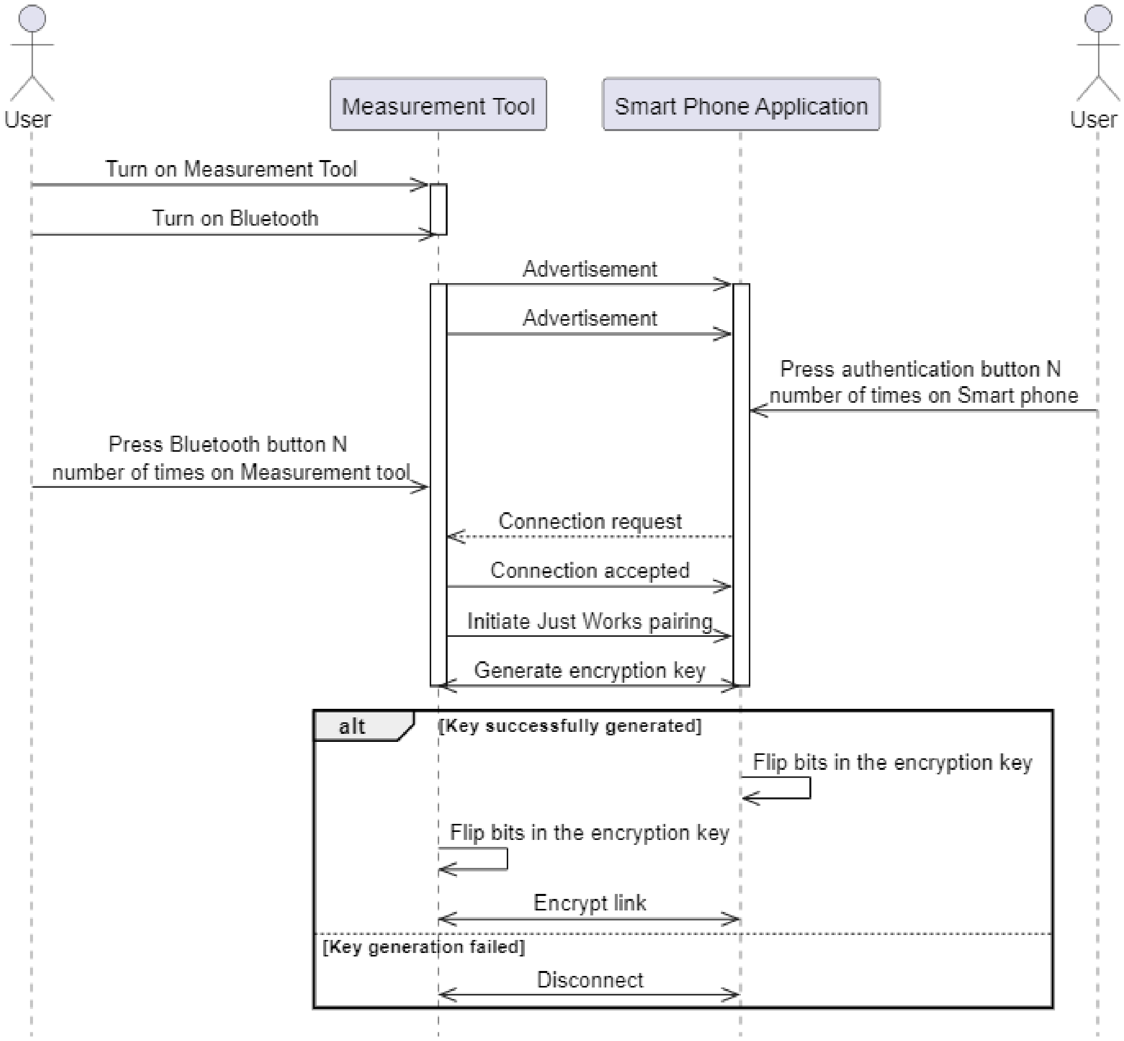


Figure 4.2: Bit flip authentication

The Figure 4.2 illustrates the first method. In the first method, user need to press the authentication button multiple times on both devices. The button press count on both sides should be equal. This value is stored on the devices for the later use. The connection request will be sent by the smart phone immediately after the authentication step. Once the connection is established, the measurement tool initiates pairing process. The LE secure connection Just Works association model should be implemented on the devices. It will generate LTK which will be used for link encryption. Before encrypting the link, the least significant bits in the LTK are toggled based on the authentication button press count value.

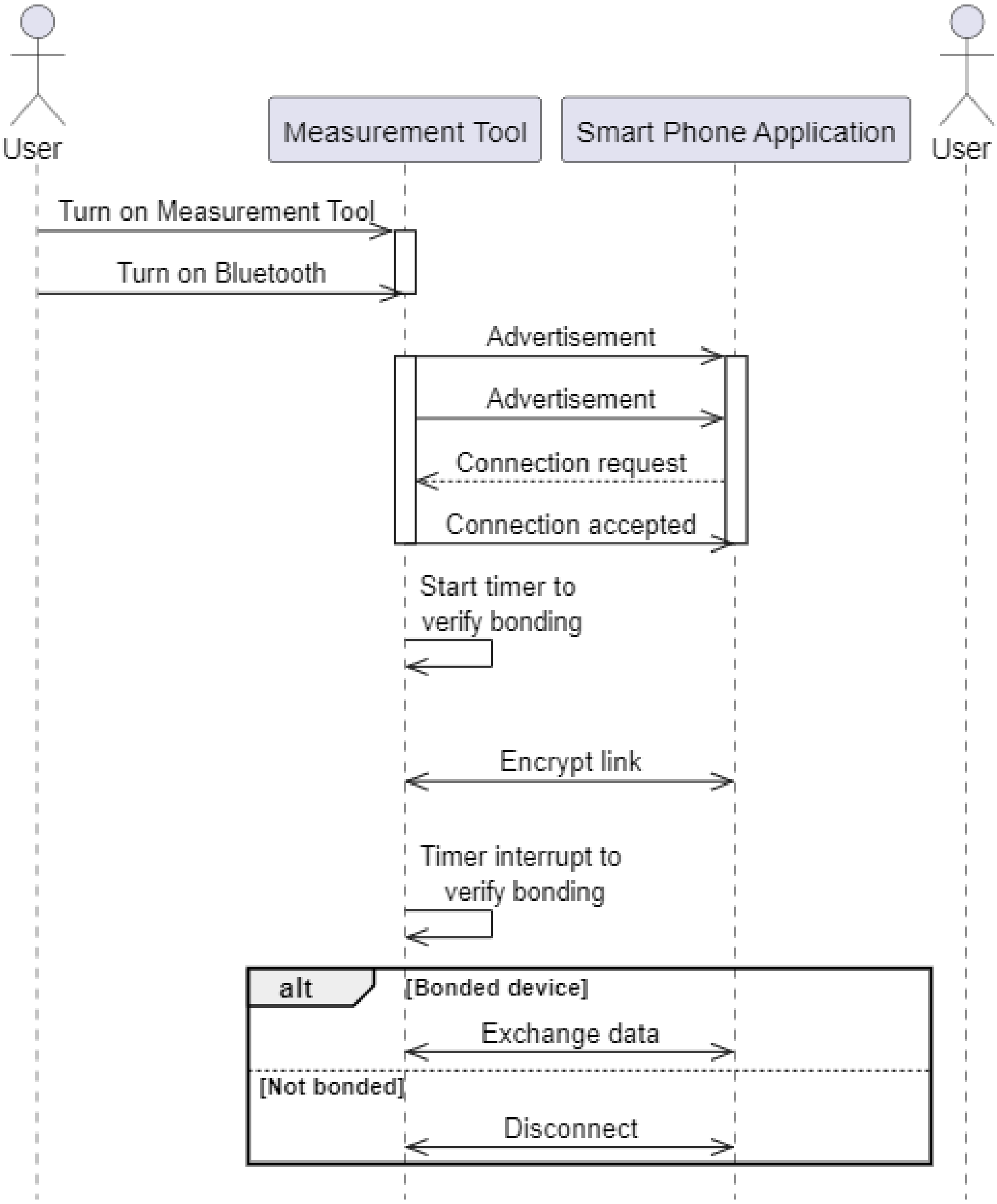


Figure 4.3: Reconnection

In the second method, user need to press and hold authentication button on both devices simultaneously for some time. The encryption key bits are toggled based on authentication button holding time. In the last method, smart phone application need to generate a random number between two and five. This value is displayed on the screen. Users need to press authentication button multiple times based on the random number. These kind of authentication methods are not available in the

Bluetooth standard. It must be realized on the application layer. It is very convenient to use these authentication methods if they are implemented on the host stack or controller stack. The MITM attacks can be identified in Bit flip authentication. The Bit flip authentication security level is equal to Numeric comparison association model in LE secure connections.

The Figure 4.2 depicts connection process for the first time where devices are new. The reconnection sequence is shown in the Figure 4.3. In general, the link is encrypted immediately if the devices are bonded already, and the encryption key is valid. Otherwise, the devices exist in the connection state until they are manually disconnected. A malicious agent can connect to the measurement tool and affect legitimate user from accessing the device. This is called Denial of Service. To solve this issue, a timer will be configured to automatically disconnect the unencrypted connection. The timer starts immediately after every successful connection.

4.5 Key Press Authentication

The key press authentication method is implemented on few products available in the market. The pairing confirmation requested by smart phone Bluetooth middleware in Just Works association model can be considered as key press authentication for headless devices. The pairing confirmation feature implementation is not defined in the specification, so it is dependent on vendor who provide the host stack. It is convenient to implement key press authentication on host stack and controller firmware then directly at application layer. The advantage is, authentication can be carried out at the connection step itself.

The Bluetooth chip manufacture, controller firmware and host stack providers have lot of options to deal with key press authentication mechanism for headless devices than an application developer. This feature can be implemented exclusively on controller firmware. Then, the product developer can configure controller with vendor specific HCI commands. Since the new connection requests are handled by host stack, this feature can be implemented merely on security manager protocol layer or host stack can expose key press authentication API for the application layer. It is feasible to implement key press authentication mechanism at application layer by abstracting the feature over available BLE application interfaces.

The two key press authentication methods are as follows,

1. Authentication during connection
2. Authentication during pairing

In the first method, user need to confirm the connection request on the measurement tool as shown in the Figure 4.4. On the other hand, user need to accept pairing on any one of the devices or both devices in the second method as shown in the Figure 4.5. The advantage of first method is, the connection will be rejected at very early stage if it is initiated by a malicious agent. The advantage of second method is, device can exchange information over unencrypted and unauthenticated attributes even before pairing.

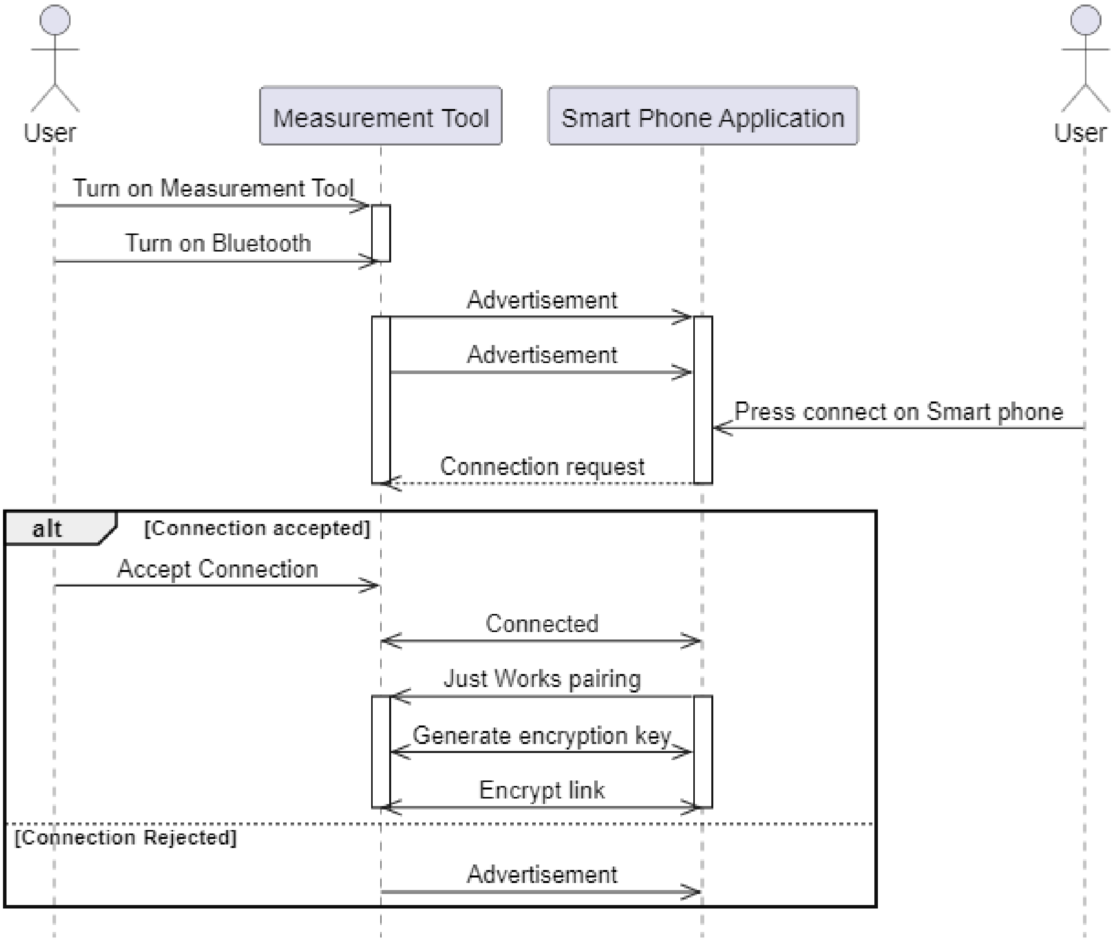


Figure 4.4: Key press authentication during connection

These authentication methods protect Bluetooth devices from following issues,

1. Denial of Service (DoS) attack
2. Unauthorized accessing, controlling, and resource usage

But the devices are vulnerable to MITM attack. The reconnection sequence is same as shown in the Figure 4.3.

4.6 Out of Band

The Out of Band mechanism can be achieved in several ways. In this context, it is limited to both smart phone and measurement tool capabilities. They are,

1. NFC
2. QR code
3. Random passkey entry

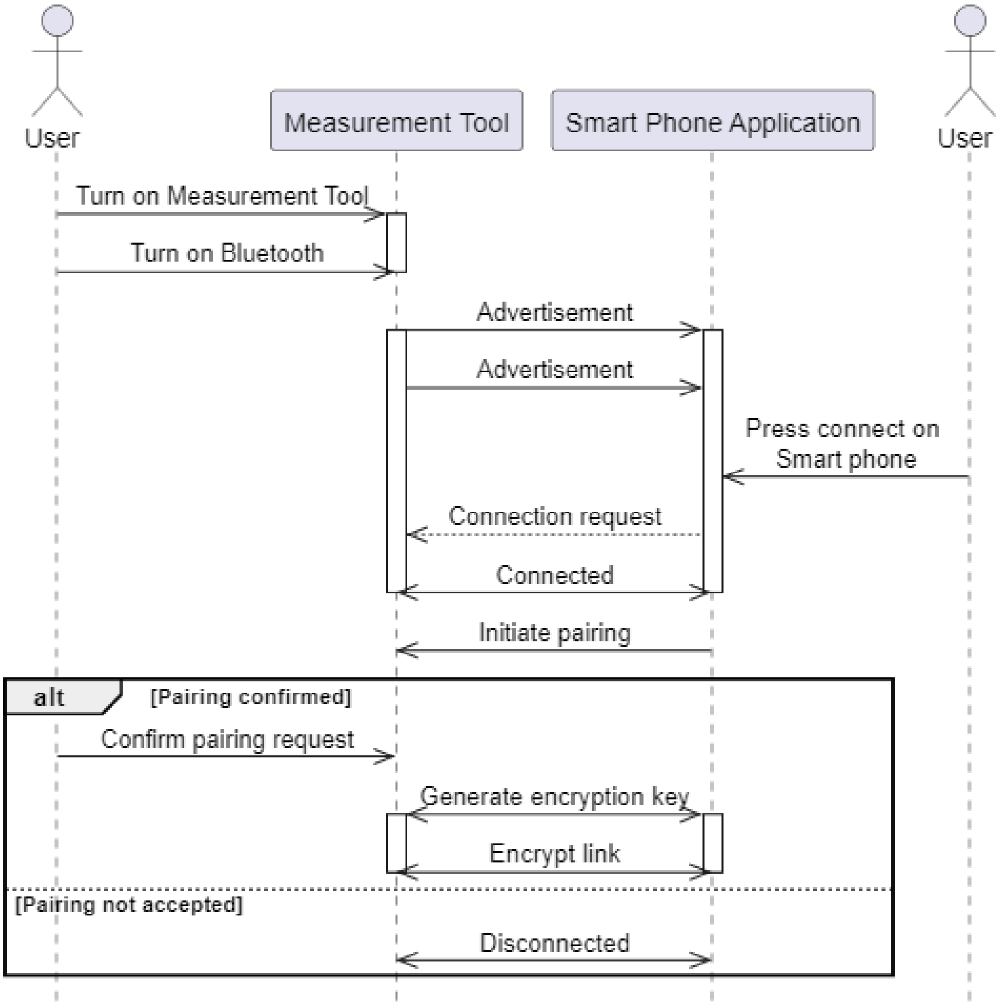


Figure 4.5: Key press authentication during pairing

The reconnection sequence is same in all the methods as shown in the Figure 4.3. The NFC and QR code based authentication methods are already defined in the specification. In these methods, a 128-bit confirm, and a random value is transferred during authentication. For NFC, the Out of Band data can be generated at the runtime. In QR code method, the Out of Band data can be generated at runtime if device have capability to display the QR code otherwise a static QR code will be generated and printed on the device.

The QR code based static passkey entry authentication is shown in the Figure 4.6. This method is not available in the specification. In passkey entry association model, either of device can define a static passkey. The static passkey is converted to QR code and printed on the device. Users need to scan the QR code and insert it manually. Because the passkey entry pop-up is initiated by the Bluetooth middleware. This authentication technique is feasible on both LE legacy pairing and LE secure connections. The passkey size is twenty bits. The protection against MITM attacks depend on access to QR code and the security level of respective pairing methods.

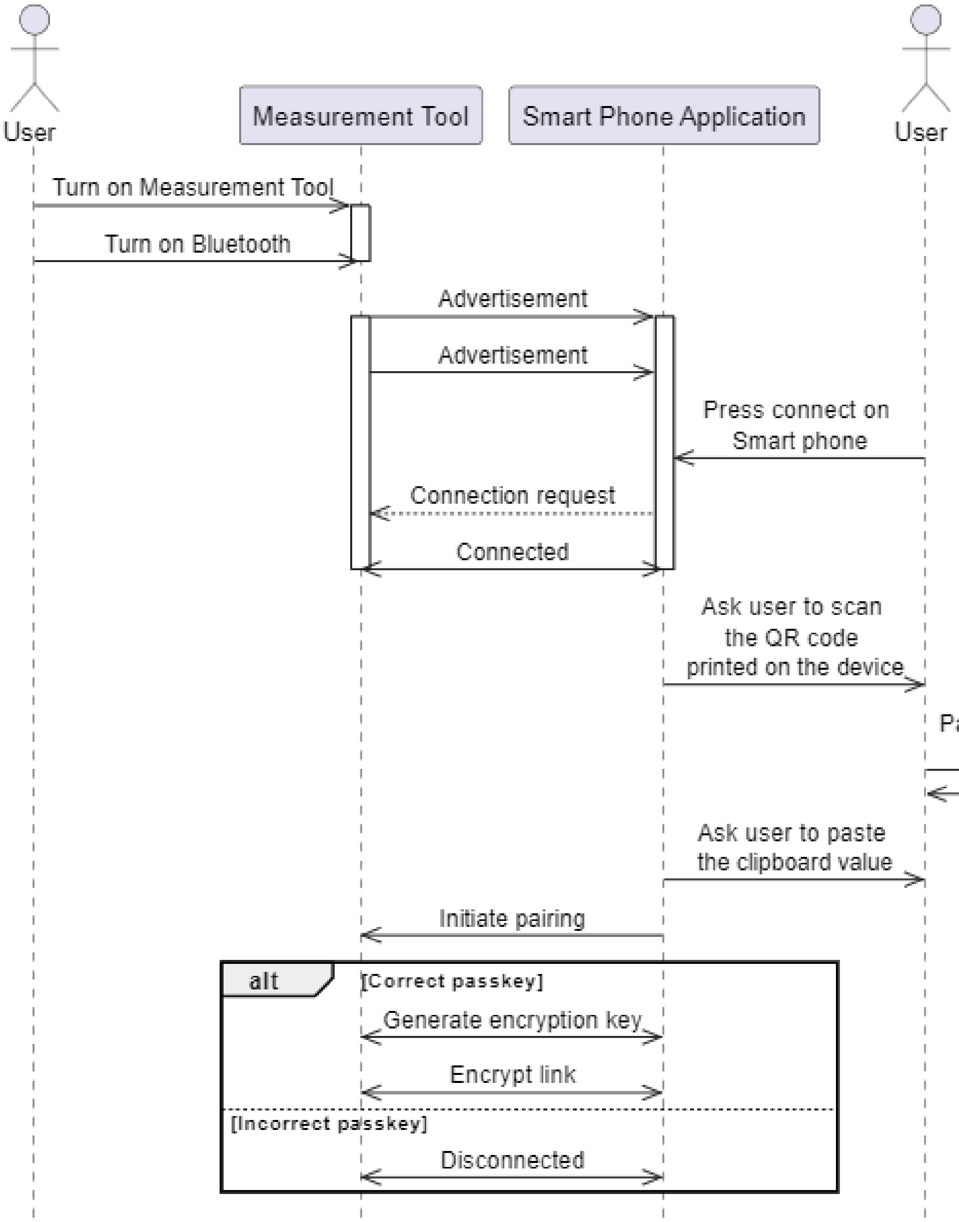


Figure 4.6: QR code based passkey authentication

The Random passkey entry method is also not defined in the specification. Unlike QR code based static passkey entry authentication; the passkey can be generated at runtime on the smart phone. This value will be set as constant value for passkey entry while initiating pairing. The same passkey value is transferred digitally from the smart phone to the measurement tool. The measurement tool should be equipped with a phototransistor. The passkey value is flashed on a predefined area on the smart phone display for a short period. The measurement tool should contact with the smart phone display in such way that the phototransistor is able to receive the flashing light from the smart phone display. The remaining authentication and encryption process is same as the passkey entry association model in LE legacy pairing and LE secure connections. Like QR code based static passkey entry authentication, the protection against MITM attacks depend on Out of Band mechanism strength and the security level of respective pairing methods.

4.7 Security on Application Layer

The encryption and authentication mechanism can be implemented directly on the application layer. The standard Bluetooth stack used only for transporting data packets. The existing crypto hardware can be reused. It is assumed that the Bluetooth chip have necessary hardware for generating shared secret using ECDH algorithm, encrypting and decrypting packets. The Bit flip authentication method explained in section 4.4 is considered here. The process of creating shared secret and authentication between devices are shown as UML sequence diagram in the Figure 4.7,
where,
SPpri – Smart phone private key
SPpub – Smart phone public key
MTpri – Measurement tool private key
MTpub – Measurement tool public key
SHsec – Shared secret

The complete process is divided in to three phases. In the first phase, the Bit flip authentication steps are executed to generate the authentication data. The authentication data is stored in a safe place. The phase one is broken into several individual steps that are explained below,
Steps 1 to 8: Once the measurement tool advertisement is received on smart phone, user can initiate connection request. The measurement tool is designed to allow connection establishment without any restriction.
Steps 9 and 10: A random number is generated on the Smart phone. The number should exist between two and five. This number is displayed on the Smart phone and requested user to press the Bluetooth button multiple times based on the number shown on the display.
Step 11: The button press count is stored on the measurement tool.

The ECDH algorithm is executed to generate shared secret in the second phase. The individual steps are,

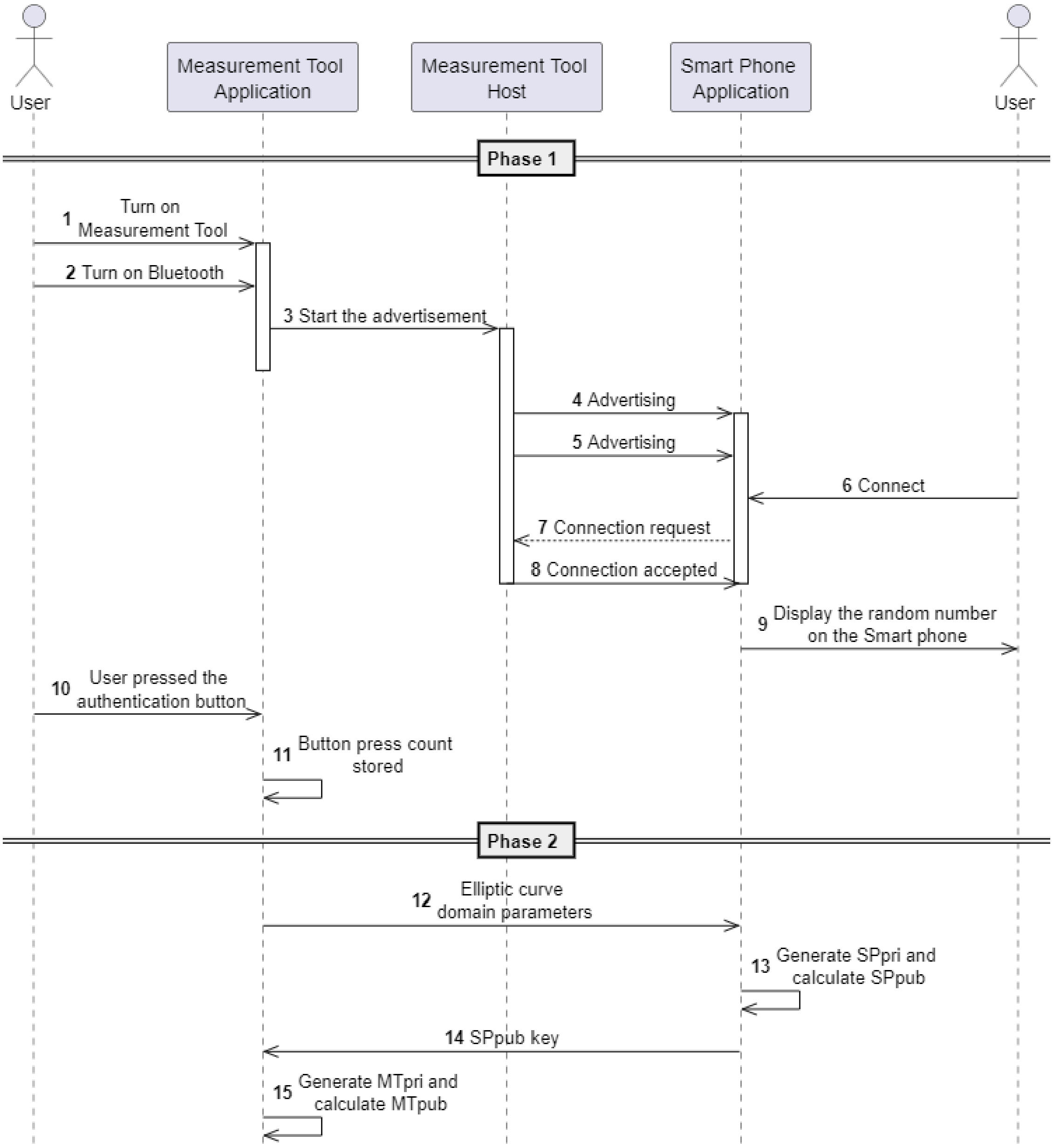
Step 12: In this step, the domain parameters are initialized for elliptic curve function. There are six domain parameters. They are,
N – Modulo
X – Elliptic curve first coefficient
Y – Elliptic curve second coefficient
G – Subgroup base point
P – Subgroup order
H – Subgroup cofactor

Step 13: The private key SPpri is generated using true random number generator on the Smart phone. Then, the Smart phone public key SPpub is generated by using following formula,

$$SPpub = G^{SPpri} \text{ modulo } N$$

Step 14: The public key SPpub is transferred to the measurement tool.
Step 15: The measurement tool private key MTpri is generated using true random number generator. The measurement tool public key MTpub is calculated by,

$$MTpub = G^{MTpri} \text{ modulo } N$$


User
Measurement Tool Application
Measurement Tool Host
Smart Phone Application
User
Phase 1
1 Turn on Measurement Tool
2 Turn on Bluetooth
3 Start the advertisement
4 Advertising
5 Advertising
6 Connect
7 Connection request
8 Connection accepted
9 Display the random number on the Smart phone
10 User pressed the authentication button
11 Button press count stored
Phase 2
12 Elliptic curve domain parameters
13 Generate SPpri and calculate SPpub
14 SPpub key
15 Generate MTpri and calculate MTpub

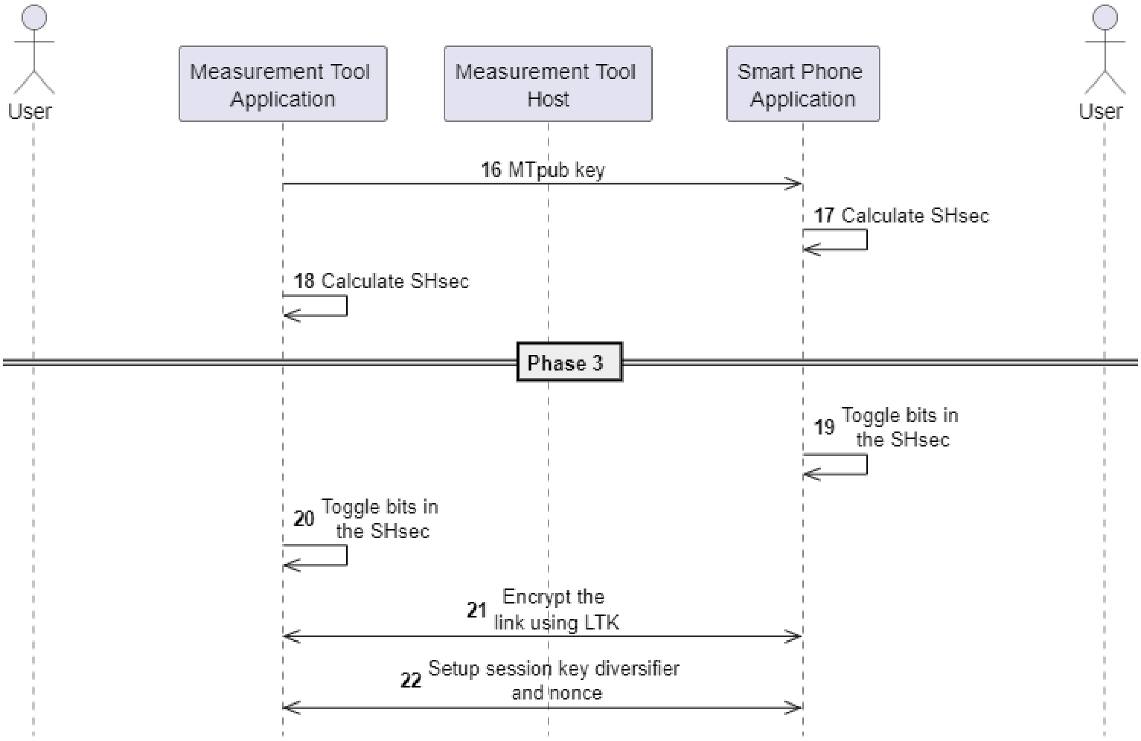

User
Measurement Tool Application
Measurement Tool Host
Smart Phone Application
User
16 MTpub key
17 Calculate SHsec
18 Calculate SHsec
Phase 3
19 Toggle bits in the SHsec
20 Toggle bits in the SHsec
21 Encrypt the link using LTK
22 Setup session key diversifier and nonce

Figure 4.7: Security implementation on application layer

Step 16: The public key MTpub is transferred to Smart phone.
Step 17 and 18: Both device calculates the shared secret SHsec i.e., LTK.

The following steps are accomplished in the third phase,
Step 19 and 20: The LTK least significant bits are toggled based on the authentication count value.
Step 21: The link is encrypted using LTK.
Step 22: Once the link is encrypted, the session key diversifier and nonce value are configured.

The LTK is used only once. In future, a session key is used to encrypt the link. The session key is generated by hashing LTK and session key diversifier.

5 Implementation and Testing

In this chapter, all the encryption and authentication methods including techniques defined in the specification and the newly invented methods documented in the previous chapter are compared. They are compared based on HMI capability, available resources on the smart phones including both android and iOS, level of security etc. Finally, a suitable security mechanism will be selected and implemented on the measurement tool. The results from the tests carried out between the measurement tool and smart phones are present in this chapter.

5.1 Device Capability

The selected solution needs to be implemented on a wide range of measurement tools. The measurement tool user interfaces are different for every device. A single push button without display case is the common among all the measurement tools. The measurement tool should be compatible with both android and iOS phones. The available application development interfaces in iOS phone are a subset of android OS. It has significant impact on implementation. So, the solution needs to be selected by considering application development interfaces available in the iOS.

The Out of Band association require special hardware resources. Since these resources are not suitable for very small devices, the Out of Band methods which needs additional hardware resources are not considered for the implementation. It is very important to note that the Apple Bluetooth middleware does not offer any Out of Band development interface to the application layer. The following association models are short listed based on the device capability mentioned in this section,

1. LE legacy pairing Just Works
2. LE legacy pairing Out of Band static passkey entry
3. LE secure connection Just Works
4. Bit flip authentication with LE secure connection Just Works
5. Key press authentication with LE secure connection Just Works
6. LE secure connection Out of Band static passkey entry
7. Bit flip authentication with LE legacy pairing Just Works
8. Key press authentication with LE legacy pairing Just Works

5.2 Regulation

Some measurement tools are equipped with laser to measure distance. The laser can be turned on and off remotely over BLE. The intensity of laser light emitted from these devices can cause eye injuries. On some devices, the laser light is installed on a rotating mount. So, the device may rotate and record measurements in some cases. If the communication link established between two devices are not authenticated, then a malicious agent can connect and control the measurement tool from a remote place. It is clearly mentioned in the regulation [11] provision 5.5-4 that the connection should be authenticated if device actuators are controlled over a network interface. Also, the cryptographic key materials should not be transferred over an unencrypted channel. The association models confine to above guidelines are,

1. LE legacy pairing Passkey Entry
2. LE legacy pairing Out of Band
3. LE legacy pairing Out of Band static passkey entry
4. LE legacy pairing Out of Band dynamic passkey entry
5. LE secure connection Passkey Entry
6. LE secure connection Numeric Comparison
7. LE secure connection Out of Band static passkey entry
8. LE secure connection Out of Band dynamic passkey entry
9. Bit flip authentication with LE secure connection Just Works pairing
10. Key press authentication with LE secure connection Just Works pairing

5.3 Selection of Authentication and Encryption method

The association models which exist under both 5.2 and 5.3 sections are,

1. LE legacy pairing Out of Band static passkey entry
2. Key press authentication with LE secure connection Just Works pairing
3. Bit flip authentication with LE secure connection Just Works
4. LE secure connection Out of Band static passkey entry

Finally, there are four association models suitable for headless BLE devices. The encryption method is based on ECDH algorithm for all association models expect first method. In the first method, the same authentication value is used for encrypting and decrypting data packets. It is printed as QR code on the device. The size of authentication value is twenty bits as well as the LTK entropy.

The key press authentication method can be implemented in two ways as explained in fourth chapter section 4.5. Of which, the first technique is not feasible on smart phones in accord with the core specification. But it is possible to realize on low power embedded BLE devices based on the API available to the application. The Measurement tool Bluetooth chip application interfaces are very limited to implement such functionality, but it is sufficient for the second technique. According to the document [4] volume 3 part H section 2.3.2, the device ‘Yes/no’ capability is not considered as input. In the same document volume 1 part A section 5.2.4.2, the specification permits application developer to implement ‘Yes/no’ confirmation for Just Works pairing so the Bluetooth specification is vague on this context.

The Bit flip authentication method is better than key press authentication. To implement Bit flip authentication method, both central and peripheral applications should have access to LTK generated in the Just Works association model. In the Measurement tool, the chip vendor had defined interfaces to access and manipulate LTK stored in the host stack. On smart phones, it is impossible to access such key resources. So, the Bit flip authentication method is not considered for the implementation.

In the last method, the authentication value is used only for the device authentication but not for the link encryption. The encryption key is generated based on ECDH algorithm and the entropy is 128-bit.

It is possible to implement whole authentication and encryption operations on the application layer for all association models. But the public key cryptography algorithm is a standard, and it is already mentioned in the Bluetooth specification. The implementation of ECDH algorithm once again in the application layer is redundant and complex. So, this concept is not considered for the implementation.

Since the Bluetooth chip has crypto hardware for executing ECDH and Advanced Encryption Standard (AES) algorithms, the first method is obsolete. The key press authentication method is faster than fourth one. The third method is not suitable for smart phones. So, the key press authentication method is selected for the project implementation.

5.4 Implementation

The key press authentication method explained in the fourth chapter section 4.5 provides an abstract notion. It can be implemented in several ways. In this project, a scheme is selected in conformity with BLE application interface and product requirements.

5.4.1 Pairing Feature Exchange

In BLE pairing, the foremost step is pairing feature exchange. The measurement tool embedded stack allows product developers to select pairing feature exchange parameters value based on the product requirements. On the other hand, it is impractical to modify values of such parameters on smart phones i.e., the parameter values are solely dependent on the Bluetooth middleware and overall smart phone security requirements. So, the parameters values may differ from one smart phone to other. However, the peripheral device pairing feature exchange values can alone influence the security mechanism selection process.

The pairing feature exchange parameters value varies for every association model. In this project, the pairing feature exchange parameters such as input-output capability, MITM field and Out of Band flag values are carefully selected to enforce LE secure connection Just Works association model.

The association model selection algorithm is mentioned in Appendix A.1 should be considered here. The measurement tool pairing feature exchange parameters values are given in the Table 5.1, Table 5.2 and Table 5.3.

Table 5.1: Pairing feature exchange

S.No.	Parameters	Values
1	Input-output capability	NoInputNoOutput
2	Out of Band data flag	False
3	Bonding flag	True
4	MITM field	False
5	Secure Connection	Not applicable
6	Keypress field	Not applicable
7	CT2 field	Not applicable
8	Maximum encryption key size	sixteen octets

Table 5.2: Initiator key distribution

Key Type	Configuration	Description
Encryption Key	Not set	We do not want to derive LTK from BR/EDR controller link key since we don't know whether BR/EDR controller link key is legitimate or not.
IRK and Identity	Set	We need Master IRK and ID information because master devices prefer to use private resolvable address.
CSRK	Not Set	There is no such characteristic mentioned in the product requirement which will send unencrypted data from master to slave but requires signing of data.
Link key	Not Set	We do not want to share LE LTK key with BR/EDR controller.

Table 5.3: Responder key distribution

Key Type	Configuration	Description
Encryption Key	Not set	Peripheral does not have BR/EDR controller.
IRK and Identity	Not set	Peripheral uses public address.

Continued on next page

Table 5.3 – continued from previous page

Key Type	Configuration	Description
CSRK	Not Set	Data signing is not required because application exchange data over an encrypted and authenticated channel.
Link key	Not Set	Not applicable.

The authentication procedure is implemented on the application layer. But the key generation, data encryption and decryption process are executed on host stack and controller. To meet such requirements, Just Works association model must be selected for the pairing process. Since the peripheral input-output capability is set to NoInputNoOutput, the association model selection algorithm result is always Just Work method according to Appendix A.1. Because the Out of Band flag is set to false and the MITM flag is obsolete. We examined the association model selection algorithm implemented on the embedded device host stack. From those examination, we found that MITM field is automatically set to false without concerning user set value for NoInputNoOutput capability.

The product developer does not have access to secure connection, keypress and CT2 fields. The keypress and CT2 fields are irrelevant for this research. The secure connection field value is automatically selected by the host stack. This field is set to true for BLE devices whose version is 4.2 and above. The secure connection field value does not dependent on secure connections only mode parameter. From core specification version 5.0, the encryption key size is obsolete if secure connections only mode is enabled.

The initiator and responder field intention are different for LE legacy pairing and LE secure connection. Since the secure connection only mode is enabled, the initiator and responder distribution field values meet security requirements only for LE secure connection.

5.4.2 Peripheral Device Configuration

The secure connections only mode is enabled on the peripheral device. According to specification, the device mode is LE security mode one level four. According to vendor documentation, the following security mechanisms are enforced in secure connections only mode,

1. Link authentication against MITM attacks
2. P-256 ECDH algorithm
3. AES-CCM encryption

Since, the encryption key length guideline is missing in the vendor documentation, we confirmed the encryption key size with the Bluetooth chip vendor. The public address is selected for peripheral device because device address is not significant for privacy.

Even though several smart phones can connect and pair with measurement tool, only one connection is allowed at a time. The total number of bonding information need to be stored on the peripheral device can be selected based on the product requirements. The LTK regeneration policy is set to zero, so that, the pairing process will be initiate from the beginning if there is a single failure during reconnection. The connection parameters are set according to Apple accessory design guideline [1].

5.4.3 Application

The authentication mechanism is implemented on the application layer. Since the embedded stack does not have any direct interface for pairing confirmation, the authentication technique needs to be implemented manually. A push button is connected to the peripheral device. It is used for device authentication. To implement key press authentication, two advertisement sets are needed. They are,

1. Advertisement set A
2. Advertisement set B

The advertisement set A can be used to pair new devices. This can be achieved by setting advertisement filter policy to process requests from all devices. The advertisement set B is used to connect with devices which are already bonded with peripheral device. The advertisement set B filter policy is set to process scan request from all devices and allow connection request for devices exist in the whitelist. The scan request is performed to access additional advertisement data provided by the peripheral device. The peripheral device process connection requests from central device whose address is available in the device whitelist. The whitelist is maintained at the link layer. The whitelist has a list of address synchronized with existing bonding information. This mechanism is not meant for security. Usually, the controller needs to interrupt host stack for every new connection requests. To reduce number of connection request processed by the host stack, the controller rejects connection request from devices whose address is not available in the whitelist. This helps to reduce energy consumption for battery powered BLE devices.

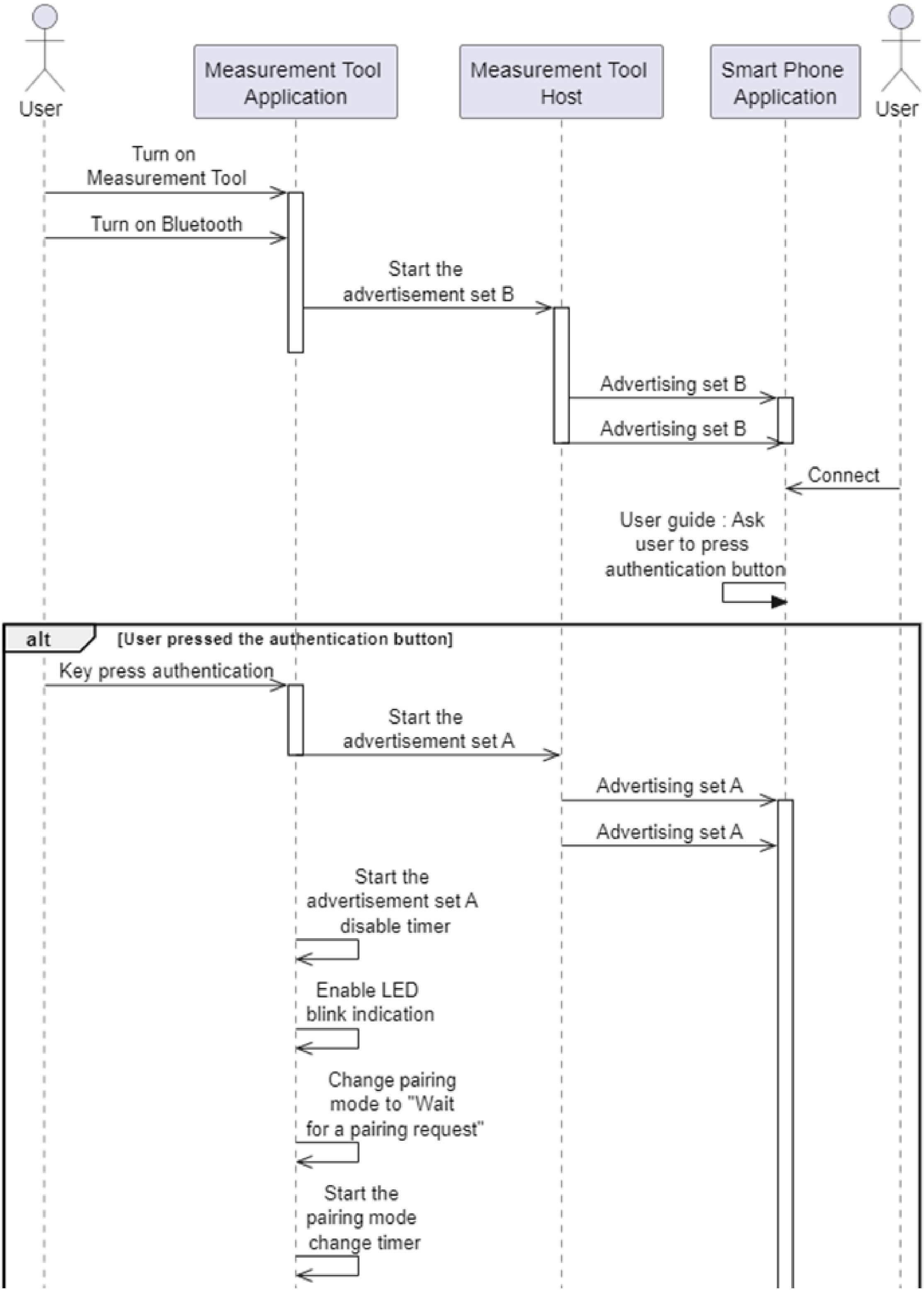
There are three paring modes available for peripheral device. They are,

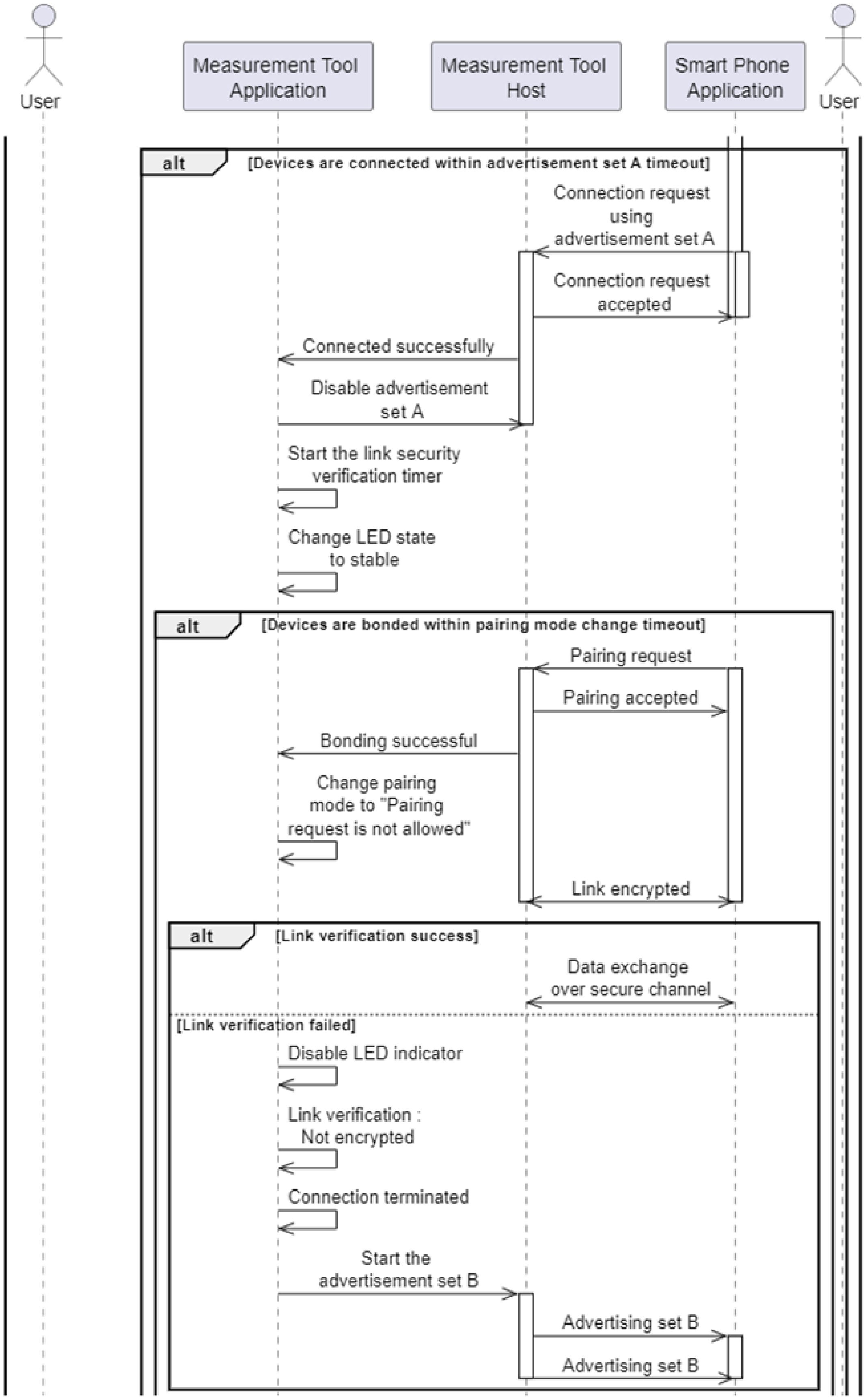
1. Initiate a pairing request
2. Wait for a pairing request
3. Pairing request is not allowed

In the first mode, the peripheral device requests central device to start pairing process. If the peripheral device is set to Wait for a pairing request mode, then the central can initiate pairing process whenever required. In the third mode, the pairing request from all central devices are rejected.

The entire authentication and encryption process between two new devices is illustrated as sequence diagram in the Figure 5.1. There are three timers used in this application. They are,

1. Advertisement set A disable timer – Timer A
2. Pairing mode change timer – Timer B
3. Link security verification timer – Timer C


User
Measurement Tool Application
Measurement Tool Host
Smart Phone Application
User
Turn on Measurement Tool
Turn on Bluetooth
Start the advertisement set B
Advertising set B
Advertising set B
Connect
User guide : Ask user to press authentication button
alt
[User pressed the authentication button]
Key press authentication
Start the advertisement set A
Advertising set A
Advertising set A
Start the advertisement set A disable timer
Enable LED blink indication
Change pairing mode to "Wait for a pairing request"
Start the pairing mode change timer


User
Measurement Tool Application
Measurement Tool Host
Smart Phone Application
User
alt
[Devices are connected within advertisement set A timeout]
Connection request using advertisement set A
Connection request accepted
Connected successfully
Disable advertisement set A
Start the link security verification timer
Change LED state to stable
alt
[Devices are bonded within pairing mode change timeout]
Pairing request
Pairing accepted
Bonding successful
Change pairing mode to "Pairing request is not allowed"
Link encrypted
alt
[Link verification success]
Data exchange over secure channel
[Link verification failed]
Disable LED indicator
Link verification : Not encrypted
Connection terminated
Start the advertisement set B
Advertising set B
Advertising set B

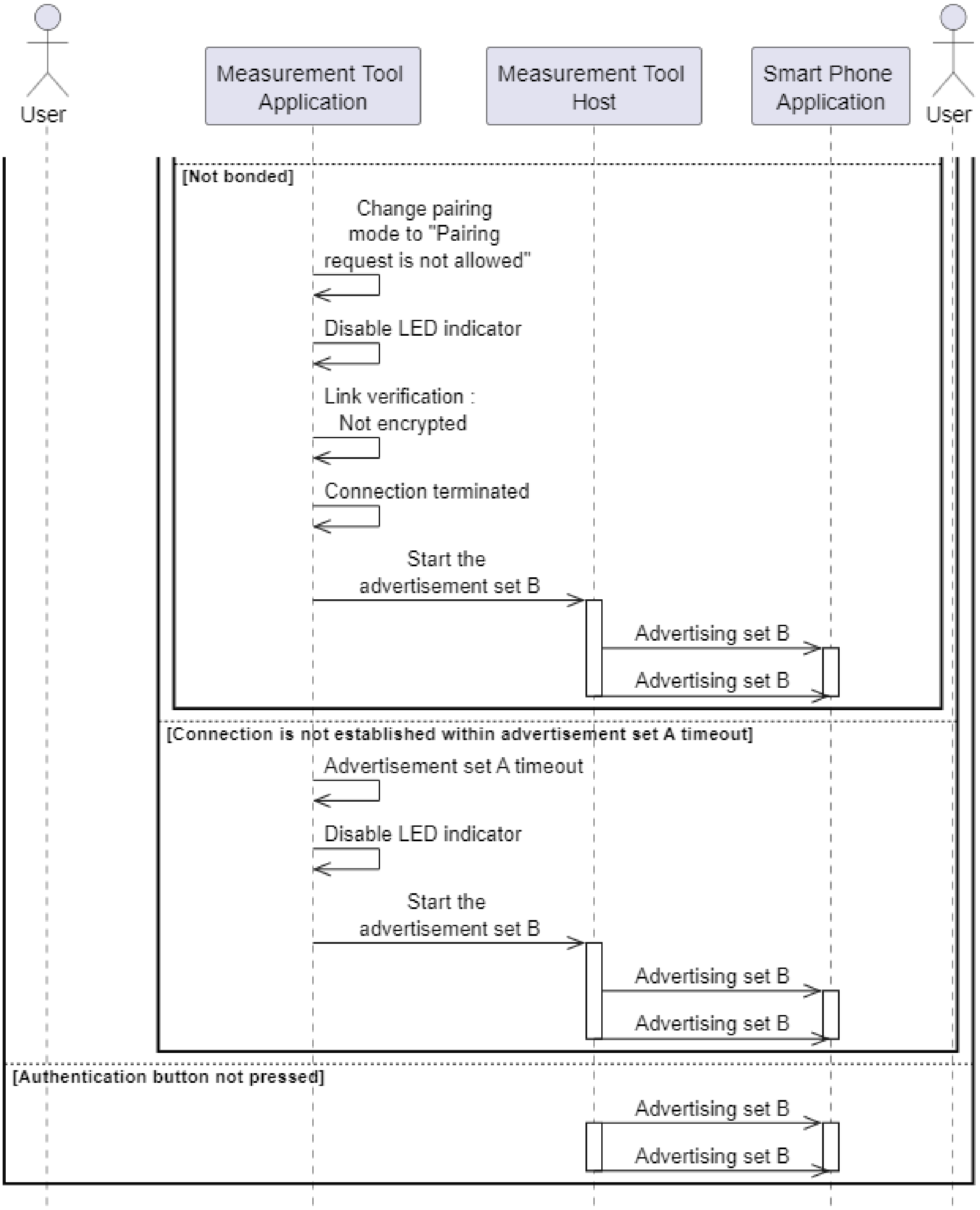
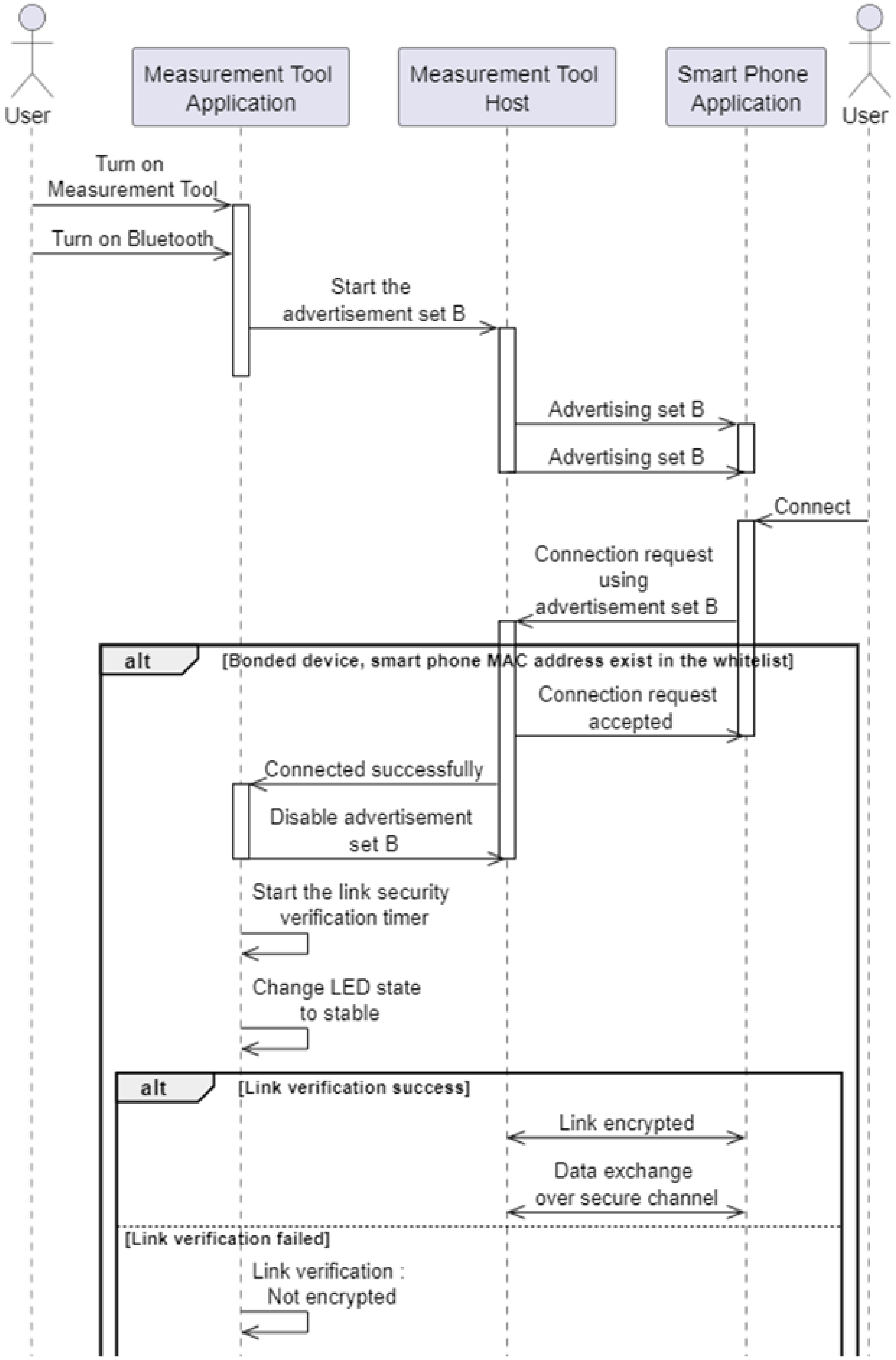

User
Measurement Tool Application
Measurement Tool Host
Smart Phone Application
User
[Not bonded]
Change pairing mode to "Pairing request is not allowed"
Disable LED indicator
Link verification : Not encrypted
Connection terminated
Start the advertisement set B
Advertising set B
Advertising set B
[Connection is not established within advertisement set A timeout]
Advertisement set A timeout
Disable LED indicator
Start the advertisement set B
Advertising set B
Advertising set B
[Authentication button not pressed]
Advertising set B
Advertising set B

Figure 5.1: Key press authentication - Device whitelisting

The timeout value is selected based on product requirement. The timer B timeout value should be greater than timer A because the time taken for pairing process is greater than connection establishment. Also, the timer C timeout value should be greater than timer B. A light-emitting diode is connected to the measurement tool. It updates peripheral device state to user in real time.


User
Measurement Tool Application
Measurement Tool Host
Smart Phone Application
User
Turn on Measurement Tool
Turn on Bluetooth
Start the advertisement set B
Advertising set B
Advertising set B
Connect
Connection request using advertisement set B
alt
[Bonded device, smart phone MAC address exist in the whitelist]
Connection request accepted
Connected successfully
Disable advertisement set B
Start the link security verification timer
Change LED state to stable
alt
[Link verification success]
Link encrypted
Data exchange over secure channel
[Link verification failed]
Link verification : Not encrypted

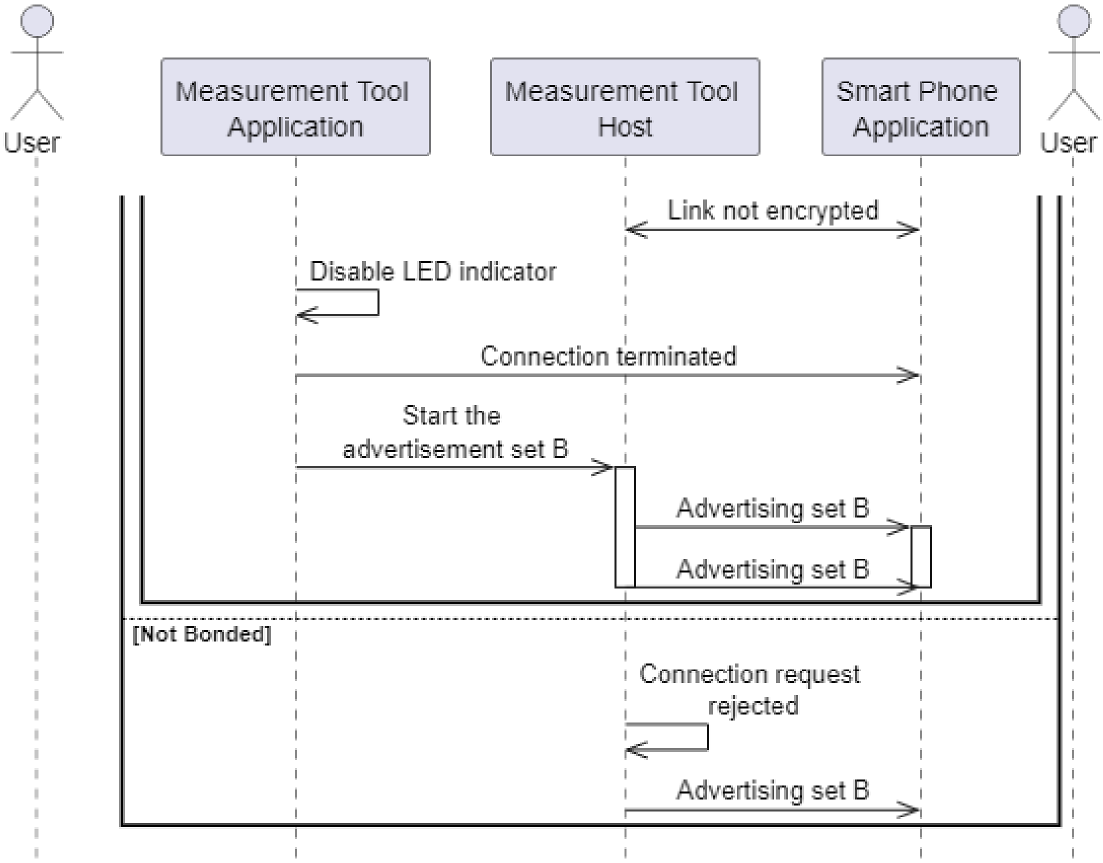


Figure 5.2: Device reconnection in key press authentication

Initially, the pairing mode is set to pairing request is not allowed. The measurement tool starts advertising immediately by enabling Bluetooth. In standby mode, the advertisement set B is used. To connect and pair a new smart phone, user should press the authentication button on the measurement tool. After the button press, the advertisement set B is disabled and the advertisement set A is enabled. The advertisement set A lasts until timer A timeout.

The smart phone should establish connection within this short interval. Once the connection is established, the link security verification timer will be started automatically, and the smart phone can initiate pairing process. The pairing should be completed before timer B timeout. The LE secure connections Just Works paring process is explained in the Appendix A.2. Finally, the timer C calls link security verification function to ensure connection encryption state. If the connection is not encrypted, the link security verification function will terminate the connection.

Once a smart phone is paired successfully for the first time, its address will be added to the peripheral device whitelist. From next time onwards, the smart phone can connect and encrypt connection without any user interaction as shown in the Figure 5.2.

5.5 Testing

In this section, the measurement tool is tested against nRF Connect application on both Android and iOS devices. On the measurement tool, the debug keys are enabled for debugging purpose only. The BLE packets are captured with nRF sniffer and analysed using Wireshark application. The smart phone specifications are given below,

Android:

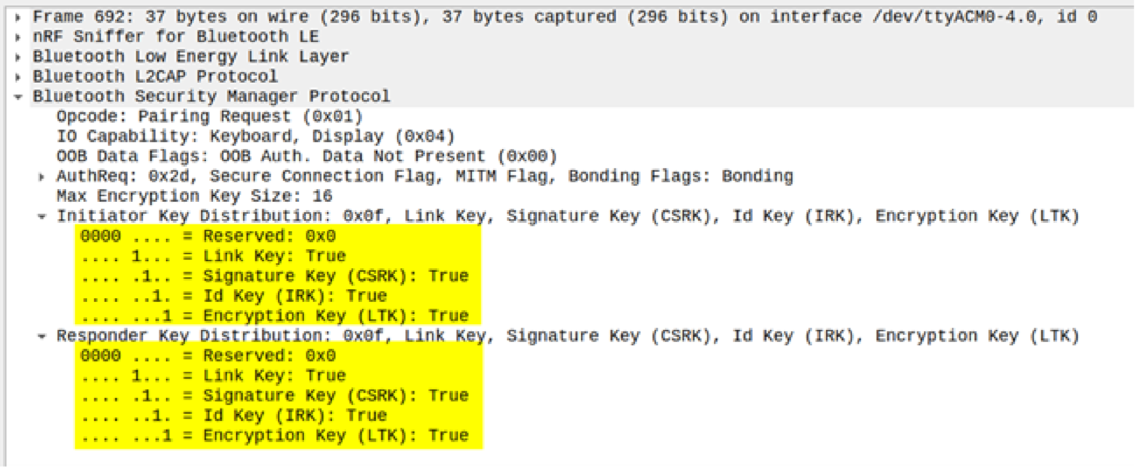
- Model: Moto G
- Bluetooth version: 4.2
- nRF Connect version: 4.27.0

iOS:

- Model: iPhone 6 plus
- Bluetooth version: 4.2
- nRF Connect version: 2.5.3

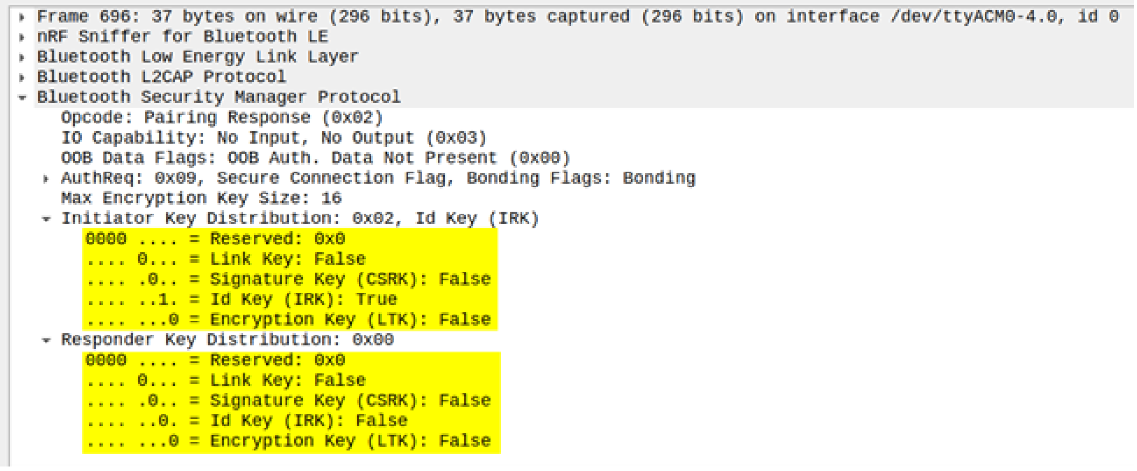
5.5.1 Pairing Feature Exchange

The encryption key generation algorithm and key length are enforced with respect to secure connections only mode. The association model selection algorithm sort out Just Works method for pairing. The initiator and responder key distribution lists are shown in the Figure 5.3 and Figure 5.4. All fields in the initiator key distribution list are set to true. In the responder key distribution packet, only identity key bit is enabled. During the key distribution phase, we confirmed that only central identity address and IRK are transferred to the peripheral device. So, the peripheral device key distribution field value dominates over the central device.



```
▸ Frame 692: 37 bytes on wire (296 bits), 37 bytes captured (296 bits) on interface /dev/ttyACM0-4.0, id 0
▸ nRF Sniffer for Bluetooth LE
▸ Bluetooth Low Energy Link Layer
▸ Bluetooth L2CAP Protocol
▾ Bluetooth Security Manager Protocol
    Opcode: Pairing Request (0x01)
    IO Capability: Keyboard, Display (0x04)
    OOB Data Flags: OOB Auth. Data Not Present (0x00)
  ▸ AuthReq: 0x2d, Secure Connection Flag, MITM Flag, Bonding Flags: Bonding
    Max Encryption Key Size: 16
  ▾ Initiator Key Distribution: 0x0f, Link Key, Signature Key (CSRK), Id Key (IRK), Encryption Key (LTK)
      0000 .... = Reserved: 0x0
      .... 1... = Link Key: True
      .... .1.. = Signature Key (CSRK): True
      .... ..1. = Id Key (IRK): True
      .... ...1 = Encryption Key (LTK): True
  ▾ Responder Key Distribution: 0x0f, Link Key, Signature Key (CSRK), Id Key (IRK), Encryption Key (LTK)
      0000 .... = Reserved: 0x0
      .... 1... = Link Key: True
      .... .1.. = Signature Key (CSRK): True
      .... ..1. = Id Key (IRK): True
      .... ...1 = Encryption Key (LTK): True
```

Figure 5.3: Initiator key distribution



```
▸ Frame 696: 37 bytes on wire (296 bits), 37 bytes captured (296 bits) on interface /dev/ttyACM0-4.0, id 0
▸ nRF Sniffer for Bluetooth LE
▸ Bluetooth Low Energy Link Layer
▸ Bluetooth L2CAP Protocol
▾ Bluetooth Security Manager Protocol
    Opcode: Pairing Response (0x02)
    IO Capability: No Input, No Output (0x03)
    OOB Data Flags: OOB Auth. Data Not Present (0x00)
  ▸ AuthReq: 0x09, Secure Connection Flag, Bonding Flags: Bonding
    Max Encryption Key Size: 16
  ▾ Initiator Key Distribution: 0x02, Id Key (IRK)
        0000 .... = Reserved: 0x0
        .... 0... = Link Key: False
        .... .0.. = Signature Key (CSRK): False
        .... ..1. = Id Key (IRK): True
        .... ...0 = Encryption Key (LTK): False
  ▾ Responder Key Distribution: 0x00
        0000 .... = Reserved: 0x0
        .... 0... = Link Key: False
        .... .0.. = Signature Key (CSRK): False
        .... ..0. = Id Key (IRK): False
        .... ...0 = Encryption Key (LTK): False
```

Figure 5.4: Responder key distribution

5.5.2 Bonding Information

The devices may lose their bonding information on some circumstances, for example, factory reset. If either of the device lost its bonding information, it is impossible to encrypt the link unless pairing process is initiated once again from the beginning. Since the peripheral device whitelist is synchronized with the bonding list, the device address is removed from the whitelist for devices whose bonding information is lost. There are three cases possible in this context. They are,

1. Both peripheral and central lost their bonding information
2. Bonding information lost on central but available on peripheral device
3. Bonding information lost on peripheral but available on central device

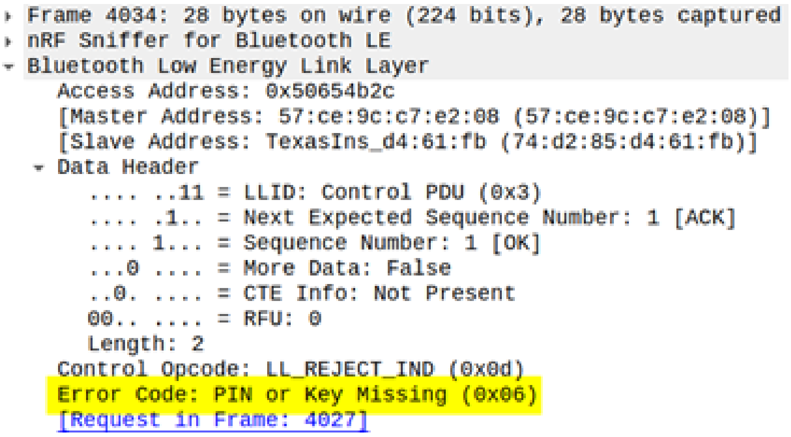
The first case is same as the condition as shown in the Figure 5.1 where both devices are new. During testing, we did not find any issues with the first case. In the second case, the paring need to be conducted once again from the beginning. The peripheral residual bonding information does not cause any issues. In the third case, the bonding information is missing on the measurement tool, but it exists on the smart phone. Even though the smart phone has bonding information, it is impossible to establish connection with the measurement tool. Because the smart phone device address will be removed from the device whitelist.

The Android and iOS devices are behaving differently in the third case. The iOS BLE application does not have access to bonded list. It is impossible to connect with the measurement tool unless the bonding information is removed manually from iOS Bluetooth settings. But it is difficult to distinguish iOS middleware callbacks for the first case and the third case. Since the BLE application state is unknown, it is hard to determine the condition where user guide for manual deletion of bonding information from iOS Bluetooth settings should be available to the user. Since there are three pairing modes as mentioned in the section 5.4.3, we conducted experiment on the following two cases,

1. Initiate a pairing request

2. Wait for a pairing request

iOS:
Case – Initiate a pairing request:
The bonding information is removed on peripheral device, but it still exists on the central device. It is very clear that connection establishment is not possible at all because central device address does not exist in the whitelist. if peripheral is in connectable and pairable mode, then the connection is established but pairing is not successful. If we try to read encrypted characteristics, the iOS middleware went to an unknown state where user interaction is not possible anymore. The encryption key missing indication packet sent by the measurement tool is shown in the Figure 5.5. According to core specification [4] third volume, part c section 10.3.2, the pairing process should be started automatically if either of the device lost the bond.



```
▸ Frame 4034: 28 bytes on wire (224 bits), 28 bytes captured
▸ nRF Sniffer for Bluetooth LE
▾ Bluetooth Low Energy Link Layer
    Access Address: 0x50654b2c
    [Master Address: 57:ce:9c:c7:e2:08 (57:ce:9c:c7:e2:08)]
    [Slave Address: TexasIns_d4:61:fb (74:d2:85:d4:61:fb)]
  ▾ Data Header
      .... ..11 = LLID: Control PDU (0x3)
      .... .1.. = Next Expected Sequence Number: 1 [ACK]
      .... 1... = Sequence Number: 1 [OK]
      ...0 .... = More Data: False
      ..0. .... = CTE Info: Not Present
      00.. .... = RFU: 0
      Length: 2
    Control Opcode: LL_REJECT_IND (0x0d)
    Error Code: PIN or Key Missing (0x06)
    [Request in Frame: 4027]
```

Figure 5.5: Pairing rejection packet

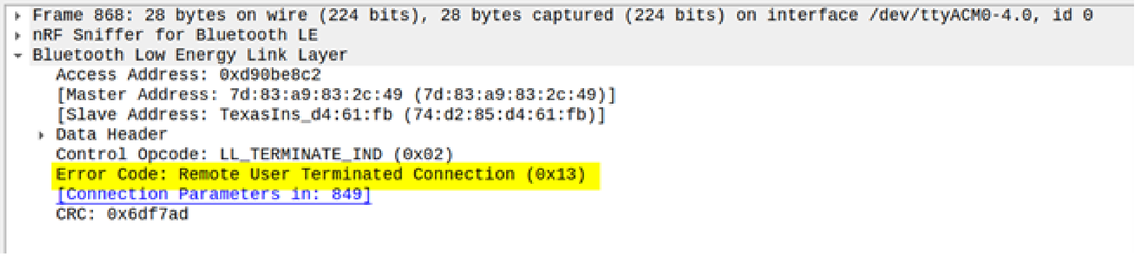
Case – Wait for a pairing request The iOS device response is same as the previous case.

The BLE application can access bonding list in the Android devices. Unlike iOS, the Android device behavior is different for Initiate a pairing request and Wait for a pairing request case.

Android:
Case – Initiate a pairing request:
The peripheral device is set to pairable and connectable mode. The peripheral device sent pairing request to the central device, but the connection is terminated by central as shown in the Figure 5.6. On the smart phone side, the residual bonding information is deleted automatically as shown in the Figure 5.7.

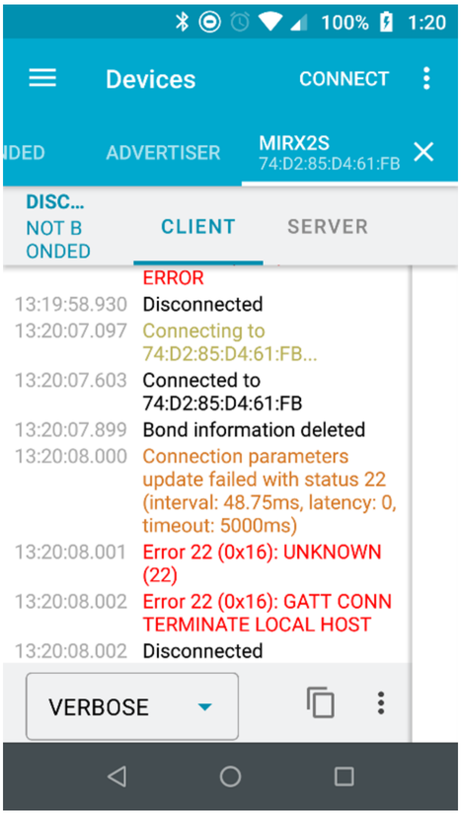
Again, the measurement tool is set to connectable and pairable state by pressing the authentication button. In the second time, both devices can connect, pair, and generate a new bonding information.

Case – Wait for a pairing request: In wait for a pairing request mode, the smart phone behavior is same as iOS device. The connection is terminated by central device as shown in the Figure 5.8.



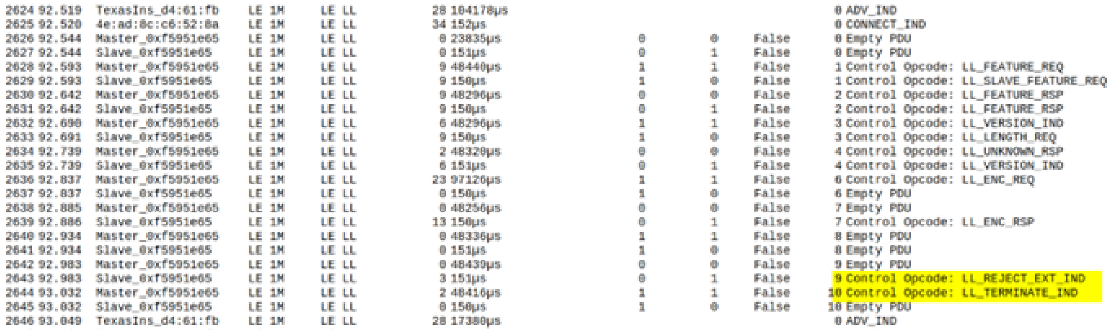
```
▸ Frame 868: 28 bytes on wire (224 bits), 28 bytes captured (224 bits) on interface /dev/ttyACM0-4.0, id 0
▸ nRF Sniffer for Bluetooth LE
▾ Bluetooth Low Energy Link Layer
     Access Address: 0xd90be8c2
     [Master Address: 7d:83:a9:83:2c:49 (7d:83:a9:83:2c:49)]
     [Slave Address: TexasIns_d4:61:fb (74:d2:85:d4:61:fb)]
   ▸ Data Header
     Control Opcode: LL_TERMINATE_IND (0x02)
     Error Code: Remote User Terminated Connection (0x13)
     [Connection Parameters in: 849]
     CRC: 0x6df7ad
```

Figure 5.6: Connection terminated by central device



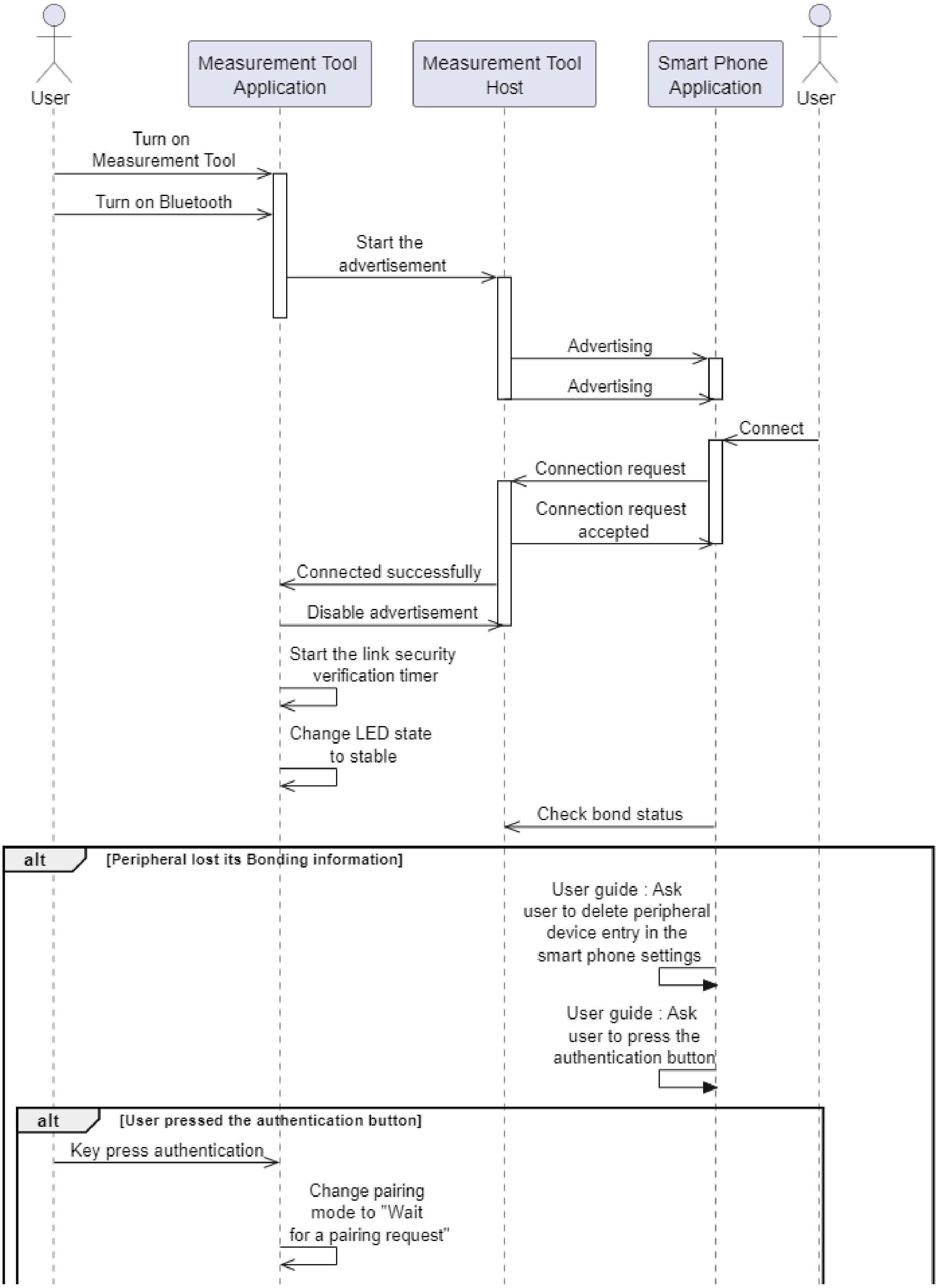
```
              ERROR
13:19:58.930  Disconnected
13:20:07.097  Connecting to
              74:D2:85:D4:61:FB...
13:20:07.603  Connected to
              74:D2:85:D4:61:FB
13:20:07.899  Bond information deleted
13:20:08.000  Connection parameters
              update failed with status 22
              (interval: 48.75ms, latency: 0,
              timeout: 5000ms)
13:20:08.001  Error 22 (0x16): UNKNOWN
              (22)
13:20:08.002  Error 22 (0x16): GATT CONN
              TERMINATE LOCAL HOST
13:20:08.002  Disconnected
```

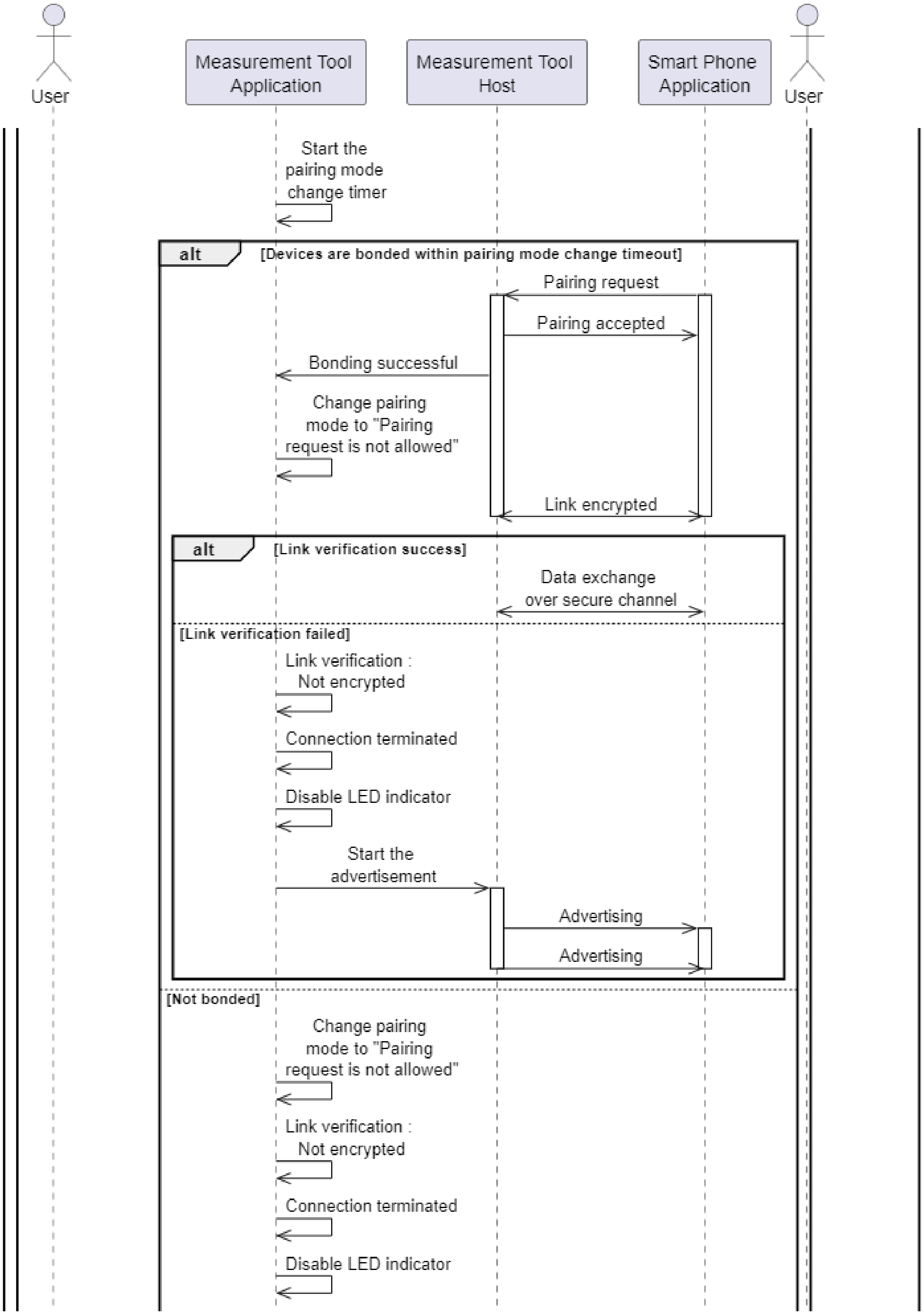
Figure 5.7: Bonding information deleted



```
2624 92.519  TexasIns_d4:61:fb   LE 1M   LE LL        28 104178µs                                   0 ADV_IND
2625 92.520  4e:ad:8c:c6:52:8a   LE 1M   LE LL        34 152µs                                      0 CONNECT_IND
2626 92.544  Master_0xf5951e65   LE 1M   LE LL         0 23835µs          0         0      False    0 Empty PDU
2627 92.544  Slave_0xf5951e65    LE 1M   LE LL         0 151µs            0         1      False    0 Empty PDU
2628 92.593  Master_0xf5951e65   LE 1M   LE LL         9 48440µs          1         1      False    1 Control Opcode: LL_FEATURE_REQ
2629 92.593  Slave_0xf5951e65    LE 1M   LE LL         9 150µs            1         0      False    1 Control Opcode: LL_SLAVE_FEATURE_REQ
2630 92.642  Master_0xf5951e65   LE 1M   LE LL         9 48296µs          0         0      False    2 Control Opcode: LL_FEATURE_RSP
2631 92.642  Slave_0xf5951e65    LE 1M   LE LL         9 150µs            0         1      False    2 Control Opcode: LL_FEATURE_RSP
2632 92.690  Master_0xf5951e65   LE 1M   LE LL         6 48296µs          1         1      False    3 Control Opcode: LL_VERSION_IND
2633 92.691  Slave_0xf5951e65    LE 1M   LE LL         9 150µs            1         0      False    3 Control Opcode: LL_LENGTH_REQ
2634 92.739  Master_0xf5951e65   LE 1M   LE LL         2 48320µs          0         0      False    4 Control Opcode: LL_UNKNOWN_RSP
2635 92.739  Slave_0xf5951e65    LE 1M   LE LL         6 151µs            0         1      False    4 Control Opcode: LL_VERSION_IND
2636 92.837  Master_0xf5951e65   LE 1M   LE LL        23 97120µs          1         1      False    6 Control Opcode: LL_ENC_REQ
2637 92.837  Slave_0xf5951e65    LE 1M   LE LL         0 150µs            1         0      False    6 Empty PDU
2638 92.885  Master_0xf5951e65   LE 1M   LE LL         0 48256µs          0         0      False    7 Empty PDU
2639 92.886  Slave_0xf5951e65    LE 1M   LE LL        13 150µs            0         1      False    7 Control Opcode: LL_ENC_RSP
2640 92.934  Master_0xf5951e65   LE 1M   LE LL         0 48336µs          1         1      False    8 Empty PDU
2641 92.934  Slave_0xf5951e65    LE 1M   LE LL         0 151µs            1         0      False    8 Empty PDU
2642 92.983  Master_0xf5951e65   LE 1M   LE LL         0 48439µs          0         0      False    9 Empty PDU
2643 92.983  Slave_0xf5951e65    LE 1M   LE LL         3 151µs            0         1      False    9 Control Opcode: LL_REJECT_EXT_IND
2644 93.032  Master_0xf5951e65   LE 1M   LE LL         2 48416µs          1         1      False   10 Control Opcode: LL_TERMINATE_IND
2645 93.032  Slave_0xf5951e65    LE 1M   LE LL         0 150µs            1         0      False   10 Empty PDU
2646 93.049  TexasIns_d4:61:fb   LE 1M   LE LL        28 17380µs                                    0 ADV_IND
```

Figure 5.8: Connection terminated by central device


User
Measurement Tool Application
Measurement Tool Host
Smart Phone Application
User
Turn on Measurement Tool
Turn on Bluetooth
Start the advertisement
Advertising
Advertising
Connect
Connection request
Connection request accepted
Connected successfully
Disable advertisement
Start the link security verification timer
Change LED state to stable
Check bond status
alt
[Peripheral lost its Bonding information]
User guide : Ask user to delete peripheral device entry in the smart phone settings
User guide : Ask user to press the authentication button
alt
[User pressed the authentication button]
Key press authentication
Change pairing mode to "Wait for a pairing request"


User
Measurement Tool Application
Measurement Tool Host
Smart Phone Application
User
Start the pairing mode change timer
alt
[Devices are bonded within pairing mode change timeout]
Pairing request
Pairing accepted
Bonding successful
Change pairing mode to "Pairing request is not allowed"
Link encrypted
alt
[Link verification success]
Data exchange over secure channel
[Link verification failed]
Link verification : Not encrypted
Connection terminated
Disable LED indicator
Start the advertisement
Advertising
Advertising
[Not bonded]
Change pairing mode to "Pairing request is not allowed"
Link verification : Not encrypted
Connection terminated
Disable LED indicator

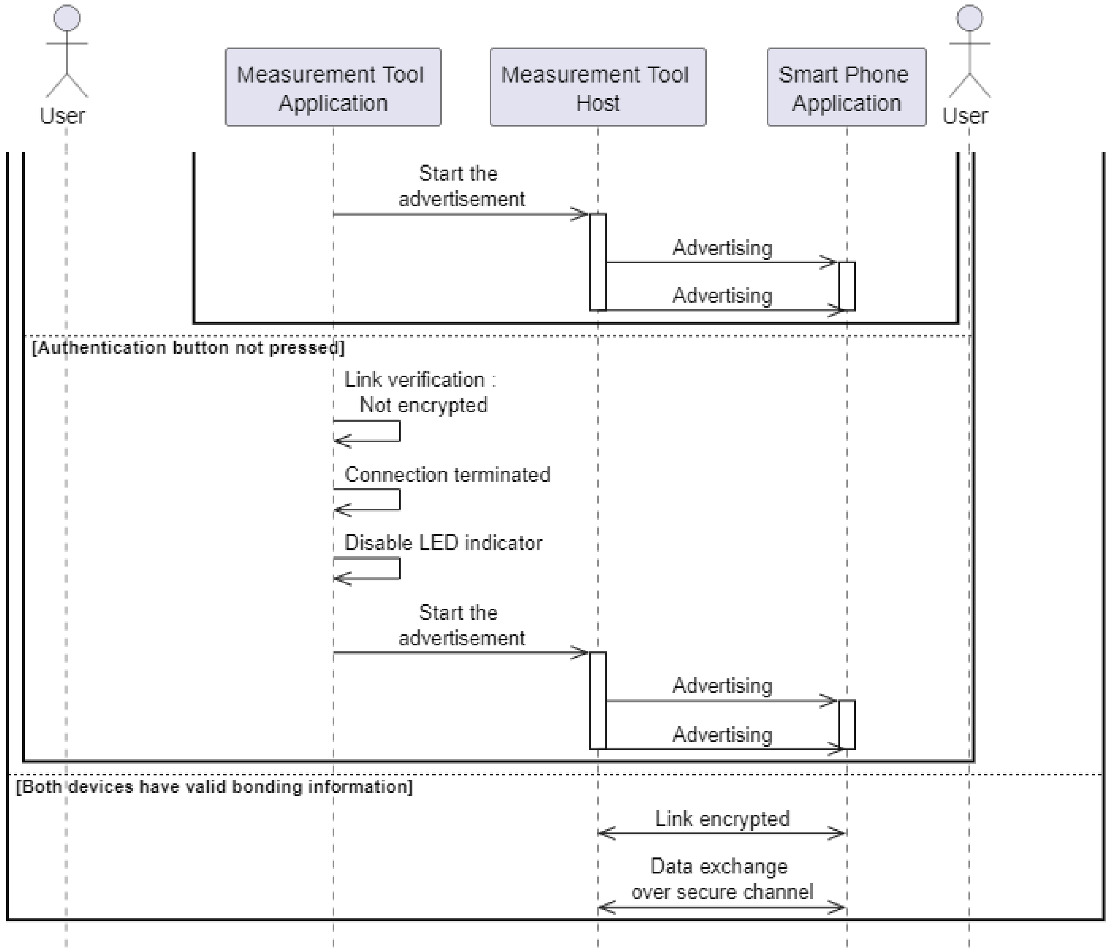
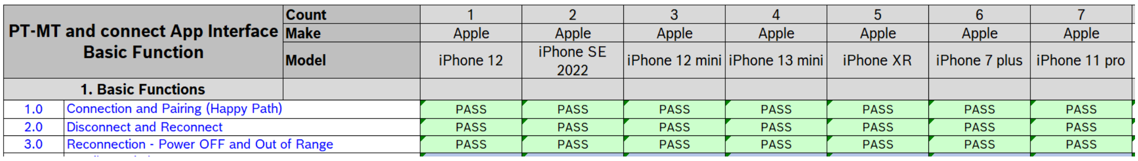


Figure 5.9: Key press authentication alternative method

These issues can be solved by removing device whitelisting feature in the peripheral device as shown in the Figure 5.9. To achieve a common workflow on both Android and iOS, the Wait for a pairing request mode should be implemented on the peripheral device. Only one advertisement set is sufficient. A new characteristic is added to the existing application called bonding status attribute. It is defined with unencrypted and unauthenticated permission. This attribute is used to determine whether bonding information is available on the peripheral. If a smart phone connected to measurement tool, the peripheral application inspects the bonding list. The attribute value is set to “Bonded” only if a valid bonding information for the newly connected smart phone is available otherwise it is set to “Not bonded”. On the smart phone side, the BLE application can read bonding status attribute and decide whether the user guide for the manual deletion of measurement tool in Bluetooth setting is required or not.

5.5.3 Interoperability

Though the core specification is the common standard for everyone, the hardware and software provided by Bluetooth chip manufacturers differs from one to another. This could cause problems if the peripheral device protocol is compatible with the smart phones. To avoid such issues, the peripheral device should be evaluated against wide range of smart phones available in the market. In this research work, the measurement tool is assessed against seven different iOS smart phones and the results are given in the Figure 5.10.



PT-MT and connect App Interface Basic Function		Count	1	2	3	4	5	6	7
		Make	Apple	Apple	Apple	Apple	Apple	Apple	Apple
		Model	iPhone 12	iPhone SE 2022	iPhone 12 mini	iPhone 13 mini	iPhone XR	iPhone 7 plus	iPhone 11 pro
1. Basic Functions									
1.0	Connection and Pairing (Happy Path)		PASS	PASS	PASS	PASS	PASS	PASS	PASS
2.0	Disconnect and Reconnect		PASS	PASS	PASS	PASS	PASS	PASS	PASS
3.0	Reconnection - Power OFF and Out of Range		PASS	PASS	PASS	PASS	PASS	PASS	PASS

Figure 5.10: Interoperability test results

6 Threat Modelling

Threat modelling is a process in security engineering to understand complexity of the system and identifying all the threats and vulnerabilities exist in the system. Threats are categorized based on their severeness and countermeasures should be provided to protect the system from malicious agents. This is a well-researched topic, and the research outcome is already available. Before doing threat modelling, Bluetooth product designers and security advisors should understand functionality of the product, their use cases and under what circumstances user using the product. Also, the complete scope of data involved in the system. It includes user data, cryptographic keys, firmware, and data generated by the device. Developers should analyze the way in which device will store, access, modify, process and exchange above data elements between devices. Also, the consequences arise when an unauthorized user accessing data categorized to high privilege and the result if this going undetected.

6.1 Threat Modelling for BLE

Threat modelling can be done in multiple ways. There is no single way of doing threat modelling for a particular product. According to the research paper [6], threat modelling can be carried out in three different ways,

1. Threat-Centric
2. System-Centric
3. Asset-Centric

Product designer can choose one or combine two or more methods based on the product security requirements and the project context.

6.1.1 Threat-Centric

In this approach, all the possible threats exist in the system including the vulnerabilities which are specific to a particular product are identified, and then each threat is applied to the product of interest. This method is very important because it provides overall information to security advisor about complete list of threats. Since there is no boundary, the list may grow indefinitely. The process of analyzing vulnerabilities and providing a mitigation to each problem become a tedious process. The main advantage of this technique, security researcher can start exploring threats exist in the system even before product design phase, but this is only applicable for less complex system like BLE. The identified threats are quantified and categorized according their severeness. Based on product security requirements, product designer can work with security advisor on severe threats to provide countermeasures as part of the application design. This will avoid major changes in application architecture in the later stages.

6.1.2 System-Centric

The system-centric approach is possible only if the system model is available to the security advisor. The data resources involved, and boundaries of the product should be defined in the system model. With the system model, the threats relevant to system are identified. This method highly suitable for heterogenous systems like BLE. Even though Bluetooth core specification provides a standard guide to Bluetooth chip manufactures and firmware developers, there exist major difference in Bluetooth security features between different vendors and versions. It may cause problems in the functionality of the device. So, the system-centric threat assessment process become hard. For instance, according to the specification, passive eavesdropping protection in the BLE is available from core specification 4.2. If system boundary is defined to core specification 4.2, then the number of vulnerabilities addressed during threat modelling will be reduced. Since BLE is highly backward compatible, the restriction to specification 4.2 will be overruled by pairing feature selection algorithm.

6.1.3 Asset-Centric

The asset-centric approach is suitable for applications where an asset such as health data, personally identifiable information and monetary funds should be protected from hackers. The measurement tool data resources, software and hardware components are not belonged to above categories. There is a risk of user experience degradation if the measurement tool is compromised. But this issue will be addressed in other threat modelling approaches.

6.2 Threat Modelling

6.2.1 Modelling Frameworks

There are several threat modelling frameworks available for risk assessment to identify, estimate, and prioritize risk to assets. These tools are developed for modelling threats exist in the IT Infrastructures. These tools are not suitable for system like BLE where key negotiation, authentication and encryption methods are different from methods and technologies used in modern IT Infrastructures. Also, there exist only two devices in the system of interest. The data flow occurs only between these two devices. Initially, STRIDE methodology is used to classify threats. Later, the threat modelling approach proposed in the research paper [2] is considered for listing potential threats. Apart from the threats and vulnerabilities mentioned in this research paper, an in-depth analysis will be carried out to identify new threats which are not listed already. Finally, we use DREAD model to prioritize all risks based on threat severeness.

6.2.2 STRIDE Methodology

The solution discussed in the previous chapter section 5.4.3 is considered for the threat modelling. The threat centric approach is not used here because the BLE threat space is very large. It is inefficient to identify all the threats by threat centric approach since the overall system is simple. The system architecture and the data flow between system components are illustrated with Microsoft Threat Modelling Tool as shown in the Figure 6.1.

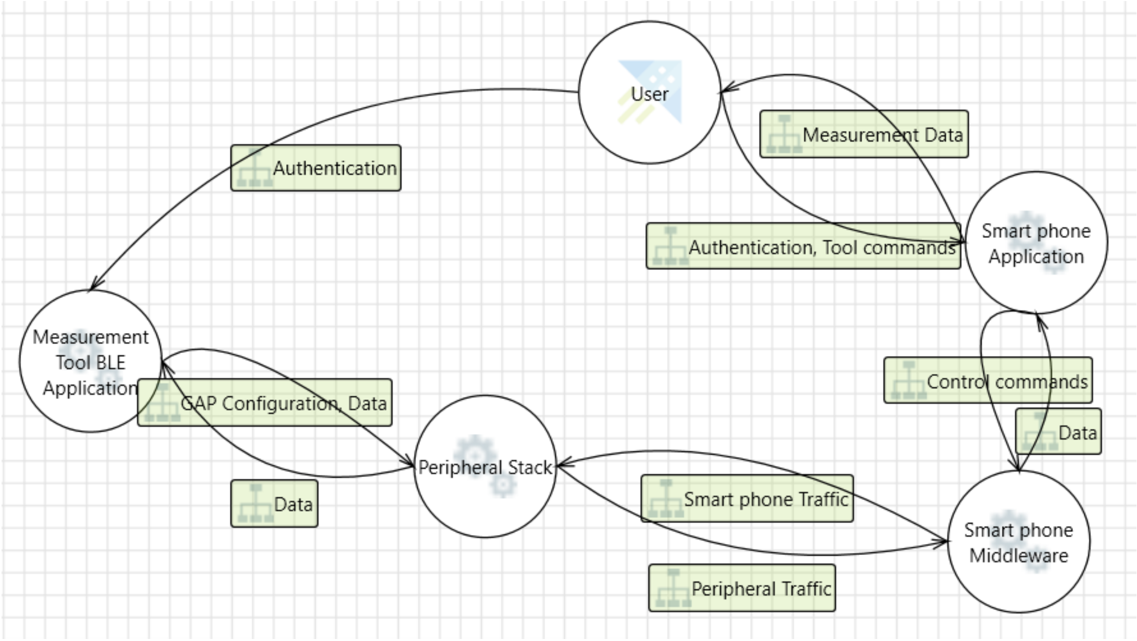


Figure 6.1: Key press authentication data flow graph

The threats present in the system are classified according to STRIDE. They are,

Spoofing Threats

- Illegal use of central device address
- Forced repairing
- Replay attack
- Active MITM attack

Tampering Threats

- Modify pairing feature exchange parameters

Information Disclosure

- Device fingerprinting
- Default profiles

Denial of Service

- Intense connection request
- Unauthenticated connection
- Jamming attack

Elevation of privilege

- Unintended authentication

6.2.3 Other Threats

As mentioned in section 6.2.1, the threat modelling tools such as STRIDE is limited for BLE. Here, the threat centric approach is considered to identify threats as shown in the Figure 6.2. They are,

1. Passive sniffing attack
2. PIN cracking attack
3. Fuzzing attack
4. Blue-Smack attack
5. Blue-Printing and stumbling
6. Co-located attack

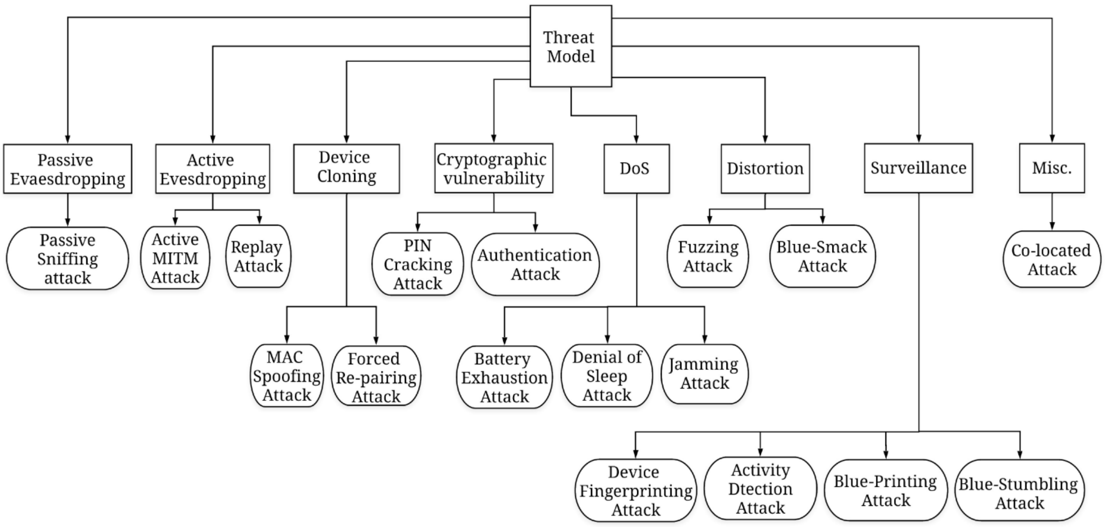


Figure 6.2: BLE threat model based on the attack domain [2]

6.2.4 DREAD Methodology

The threats listed in the section 6.2 need to be analyzed and prioritized according to their severeness. To determine threat risk, DREAD methodology is applied to each threat. The threat risks are quantified based on following five parameters,

1. Damage
2. Reproducibility
3. Exploitability
4. Affected-users
5. Discoverability

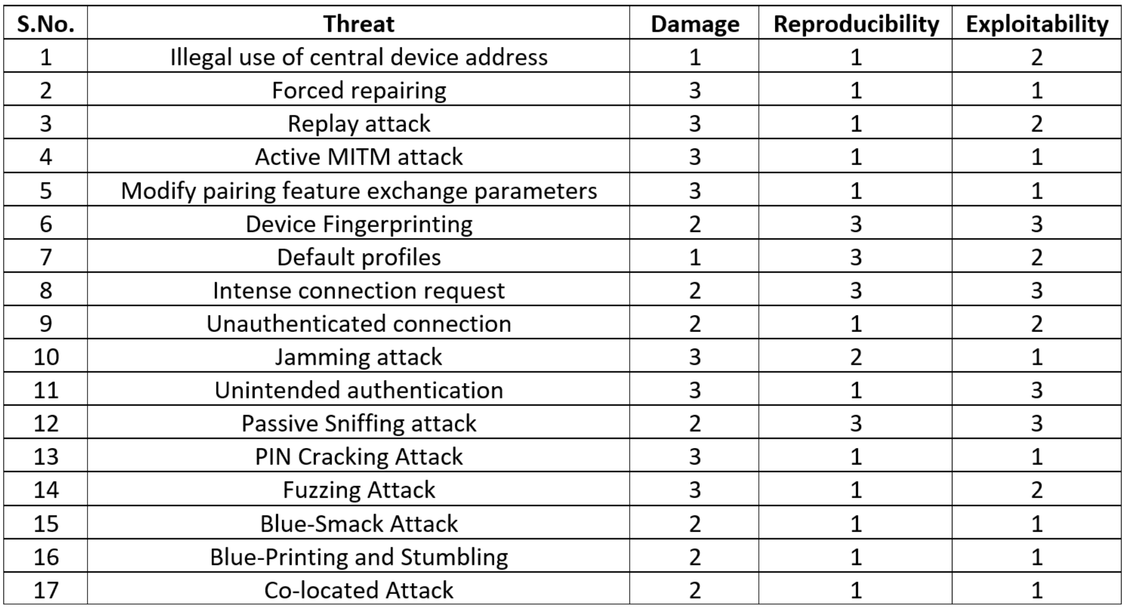
A value is chosen between one and three and assigned to each parameters listed above. Finally, the following formula is used to calculate the risk value,

$$RiskValue = Damage+Affected\_users+Reproducibility+Exploitability+Discoverability$$

The risk value must be calculated for all threats. It ranges from five to fifteen. After calculating the risk value, the threats need to be ranked based on following criteria,

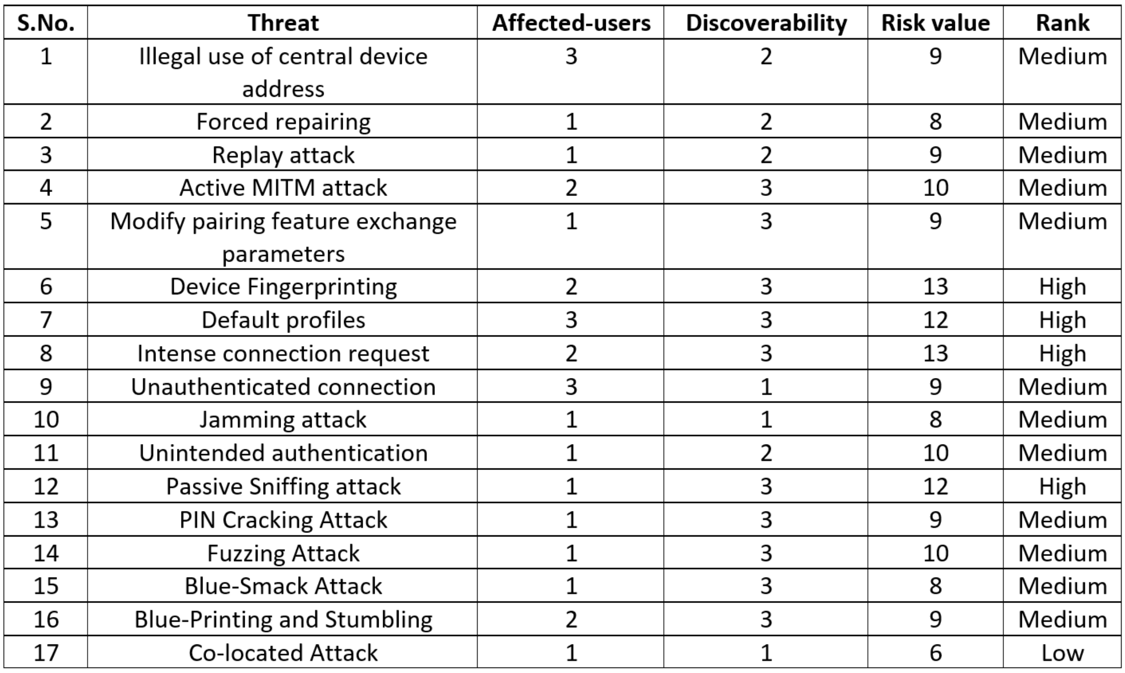
1. Overall rating exists between twelve and fifteen is considered as high risk
2. Overall rating exists between eight and eleven is considered as moderate risk
3. Overall rating exists between five and seven is considered as low risk

The high risk threats are critical, should be solved immediately. Once all high risk threats are fixed, then medium risk and go on. The risk value and rank are determined for all the threats and listed in the Figure 6.3 and Figure 6.4.



S.No.	Threat	Damage	Reproducibility	Exploitability
1	Illegal use of central device address	1	1	2
2	Forced repairing	3	1	1
3	Replay attack	3	1	2
4	Active MITM attack	3	1	1
5	Modify pairing feature exchange parameters	3	1	1
6	Device Fingerprinting	2	3	3
7	Default profiles	1	3	2
8	Intense connection request	2	3	3
9	Unauthenticated connection	2	1	2
10	Jamming attack	3	2	1
11	Unintended authentication	3	1	3
12	Passive Sniffing attack	2	3	3
13	PIN Cracking Attack	3	1	1
14	Fuzzing Attack	3	1	2
15	Blue-Smack Attack	2	1	1
16	Blue-Printing and Stumbling	2	1	1
17	Co-located Attack	2	1	1

Figure 6.3: DREAD parameters value for threats



S.No.	Threat	Affected-users	Discoverability	Risk value	Rank
1	Illegal use of central device address	3	2	9	Medium
2	Forced repairing	1	2	8	Medium
3	Replay attack	1	2	9	Medium
4	Active MITM attack	2	3	10	Medium
5	Modify pairing feature exchange parameters	1	3	9	Medium
6	Device Fingerprinting	2	3	13	High
7	Default profiles	3	3	12	High
8	Intense connection request	2	3	13	High
9	Unauthenticated connection	3	1	9	Medium
10	Jamming attack	1	1	8	Medium
11	Unintended authentication	1	2	10	Medium
12	Passive Sniffing attack	1	3	12	High
13	PIN Cracking Attack	1	3	9	Medium
14	Fuzzing Attack	1	3	10	Medium
15	Blue-Smack Attack	1	3	8	Medium
16	Blue-Printing and Stumbling	2	3	9	Medium
17	Co-located Attack	1	1	6	Low

Figure 6.4: Threat risk and rank calculation

6.3 Mitigations

Device fingerprinting BLE devices advertise services. It is visible to all nearby devices. A malicious actor can record advertisement data, GATT structure, device behavior, response, and transmitter signal strength over a period. This information can be used to find device specific information and track user activity.
Mitigation: Advertisement data resources should not contain any sensitive information. The ATT should be protected with appropriate permissions. The advertisement frequency can be reduced gradually over the period and stopped after a certain time. It can be restarted automatically with the device wake up sensor. Also, the transmitter signal strength can be reduced for advertisements to a level so that the advertisement packets are visible only to near devices.

Default profiles The Bluetooth specification offers several default services which are implemented in the host stack.
Mitigation: During implementation, these unnecessary services should be disabled.

Intense connection request While advertising, any smart phone can initiate connection request to the measurement tool. For every connection request, the measurement tool need to interrupt the host stack. A hacker can send connection request continuously to drain the device battery quickly.
Mitigation: The device whitelisting feature should be enabled. This solution is already defined in the core specification. The advertisement frequency can be reduced gradually over the period and stopped after a certain time.

Passive sniffing attack The encrypted link established between two devices can be eavesdropped if the key entropy and key generation algorithm are not according to the standard. The pairing techniques can be altered by a malicious agent during pairing feature exchange.
Mitigation: To overcome such problems, the secure connection only mode should be enabled to enforce the encryption key length as 128-bit and ECDH algorithm for the shared secret generated between devices.

Illegal use of central device address Usually, the resolvable private address is used in the smart phones. This address change periodically. Only those devices which are already bonded with smart phone can track. A malicious agent can record resolvable private address from nearby devices over a period. Later, the adversary can use these addresses for the connection request with the measurement tool.
Mitigation: The pairable mode should be enabled only if measurement tool is authenticated. The ATT permissions should be properly configured to restrict data access. The measurement tool needs to ensure the link state for every connection. If it is not encrypted within a certain time, the connection should be terminated.

Forced repairing A malicious actor can clone the measurement tool. The bonding information will be deleted on the smart phone automatically if it connects with the clone. In this context, user forced to initiate pairing procedure once again.
Mitigation: Only Android devices are affected in the Forced repairing attack. The smart phone manufacture needs to solve this issue exist in the Bluetooth middleware.

Replay attack An attacker can record encrypted data packets and reuse them to access data resources.
Mitigation: Since, the BLE protocol data units generated by AES crypto module are authenticated with an incremental nonce value, the repeated packets are rejected on both sides.

Active MITM attack The key press authentication mechanism does not provide protection against MITM attacks during pairing.
Mitigation: To mitigate the attack, the transmitter signal strength can be reduced during paring to a level such that only nearby devices can pair with the measurement tool. Since the pairing window is very short and opened occasionally, the chance of MITM attack is very less.

Modify pairing feature exchange parameters The pairing feature parameters are exchanged over an unencrypted channel. An adversary can listen, jam, modify and inject packets. He can alter the parameter values for out of band, bonding, and key distribution flags.
Mitigation: To overcome such problems, the secure connection only mode should be enabled to enforce the encryption key length as 128-bit and ECDH algorithm. Still, the key distribution flags can be altered by a malicious agent. The device may crash. Even though, both devices exchange cryptographic keys unnecessarily during key distribution phase, the hacker cannot access the key materials because they are exchanged over an encrypted channel. Also, the transmitter signal strength can be reduced during paring to a level such that only nearby devices can pair with the measurement tool.

Unauthenticated connection An attacker can brute force central device resolvable private

address. The effort needed for determining a valid central device address decreases with respect to increase in elapsed time from bonding.
Mitigation: The measurement tool needs to ensure the link state for every connection establishment. If it is not encrypted within a certain time, the connection should be terminated. The advertising period and transmitter signal strength can be reduced based on total number of link verification failures.

Jamming attack A hacker can use Bluetooth jammer to block all the traffic between two devices. The connection will be broken due to response timeout. It is hard to mitigate this attack.

Unintended authentication The measurement tool is target by a custom hacking tool which can send connection requests continuously. In general, user need to press the authentication button for bonding new devices. Since the hacking tool sending connection request continuously, the measurement tool may connect and pair with the hacker tool.
Mitigation: The transmitter signal strength can be reduced during paring to a level such that only nearby devices can pair with the measurement tool.

PIN cracking attack It depend on the pairing method and the association model selected for the bonding procedure.
Mitigation: To overcome such problems, the secure connection only mode should be enabled to enforce the encryption key length as 128-bit and public key cryptography.

Fuzzing attack A malicious agent can inject arbitrary packets into the encrypted connection and record the response if available. The Bluetooth device may crash in some scenarios.
Mitigation: The Bluetooth chip vendor should ensure the device hardware and software components are resilient against fuzzing attack.

Blue-Smack attack This attack is possible only on devices where the device software is not designed to manage long ping request packets. The device will crash if a ping packet size larger than standard size is received.
Mitigation: The Bluetooth chip vendor should ensure the device software components is able to handle non-standard ping packet size safely.

Blue-Printing and stumbling In this attack, the adversary scan for nearby devices and read the advertisement packets. If possible, he tries to establish connection with the target. He collects device information such as Bluetooth version, manufacturer, device part number etc. Later, He use this information to search vulnerabilities registered in databases such as [18]. Finally, the BLE device is targeted against the vulnerabilities found on the database.
Mitigation: The device software should be updated whenever a new version is available.

Co-located attack On some smart phones, the BLE application or service run in the background so that nearby peripheral devices can connect with the smart phone automatically. This feature can be leveraged to carry out forced repairing attack.
Mitigation: The advanced features should be enabled only to limited accessories.

7 Further Improvements for Bluetooth Architecture

In this chapter, the Bluetooth security concepts are improved to a next level. The BLE can offer heterogenous services. Instead of defining security on link level, assigning security requirements for each data resources will be discussed along with its advantages. An alternative authentication procedure for LE secure connections passkey entry association model is presented here. The advantages of “Yes/no” Input-output capability is discussed. Finally, the shortcomings of secure connection only mode is listed.

7.1 Modified Bluetooth Architecture

The Bluetooth security concepts are defined at multiple layers. The generation of shared secret, packet encryption and decryption are taking place at the controller link layer. The authentication techniques are specified at host security manager protocol layer. A new protocol needs to be designed and implemented if a product developer desired on homegrown security mechanisms. The Bluetooth security methodology differs from the Open Systems Interconnection (OSI) model. In OSI model, all security related concepts are embedded very close to the application layer. The user data is encrypted as soon as it leaves from the application. In BLE, the encryption takes place at the link layer. Since the controller firmware is mostly proprietary, the security concepts implemented on controller are completely vague to product developer. The encryption and decryption data packet consists of application data, headers, nonce and protocol metadata. The data transfer rate increases if we move the security implementation from link layer and security manager protocol to application layer as shown in the Figure 7.1.

Controller The controller software stack is almost same as the standard except the encryption key generation algorithms, crypto accelerators, data packet encryption and decryption modules are moved to a new layer called security manager. The controller software can be proprietary too.

Host There are no changes in HCI. The Security Manager Protocol should be removed from the host. The authentication procedure moved to the security manager. The attribute permission defined in the ATT layer is not required anymore. Because product developer can define security permissions such as authentication level, encryption level, encryption key size and authorization permissions directly on the security manager. The host stack should provide two standard data interfaces to security manager called Real-time and Non-Real-time data interfaces. The Real-time data interface can be used for services such as audio streaming. In the above proposed architecture, the protocol metadata such as packet headers, fragmentation, segmentation, flow control and congestion information added to every packet is available to anyone by sniffing the Bluetooth link. So, the host stack should be designed and tested to ensure the component is resilient to the fuzzing attack mentioned in the previous chapter section 6.3.

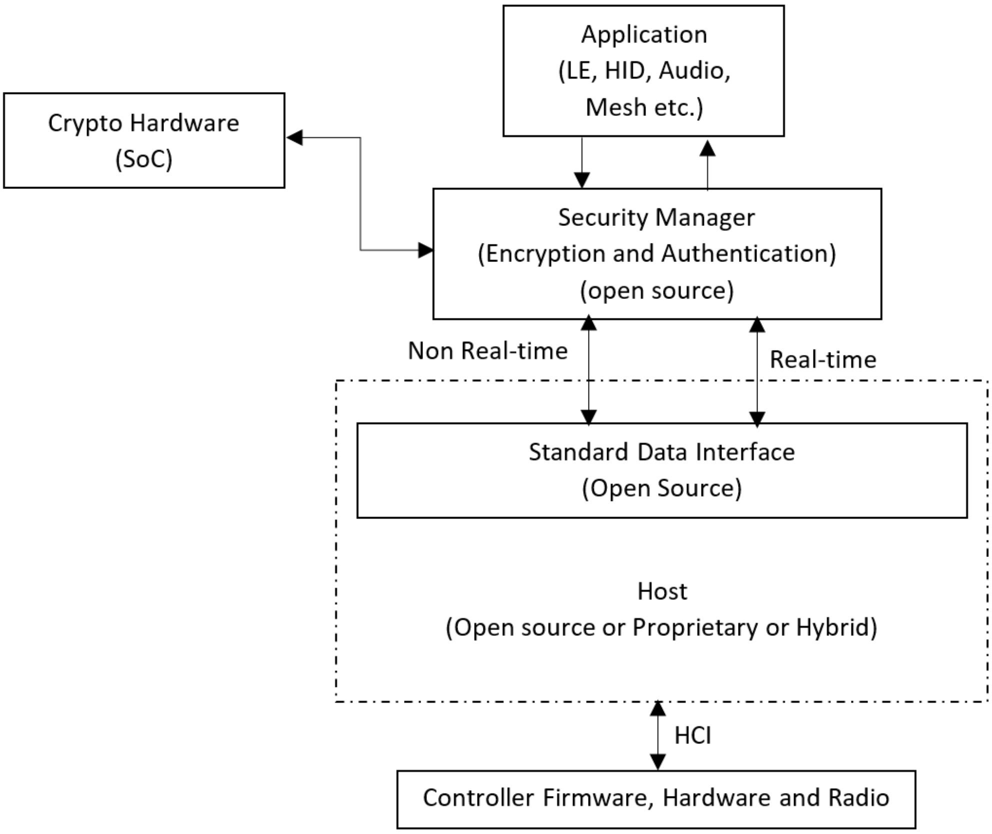


Figure 7.1: Modified Bluetooth Architecture

Security Manager The security manager incorporates authentication and encryption procedure. Since it is open sourced, new homegrown security techniques can be easily integrated into the security manager. User has lot of options than existing BLE security mechanisms in the current architecture. The process of shared secret generation, encryption, decryption, and authentication takes place at security manager. The security manger can be configured from the application layer based on product security requirements. The application source code may not be open sourced. In the current architecture, the controller is equipped with necessary crypto hardware separately form the main systems. In the newly proposed architecture, the Bluetooth subsystem can share the crypto hardware exist in the main system.

Application The current Bluetooth host layer provides several services by default. The service type and number of services offered are completely dependent on device manufacture. Some services may require special hardware. Since the host stack definition is highly rigid, it is inappropriate to add manufacture specific features and services to the host stack. This problem is solved in the modified architecture as shown in the Figure 7.1. All Bluetooth services, features and its functionality should be implemented in the application layer.

7.2 Data Level Security and Secure Connections Only mode

In the current architecture, the attribute permission is limited to readable, writable, encryption, authentication, and authorization. Of which except authorization, all other permissions are highly intact. The encryption and authentication permissions ensure only whether the link is encrypted or not and authenticated or not. It does not guarantee the encryption and the authentication method selected for the pairing process. The BLE supports several encryption and authentication mechanisms at all the time because of backward compatibility. It is a mix of both less and highly secure mechanisms. Since the selection of authentication and encryption methods are highly flexible and dynamic, the encryption and authentication attribute permissions alone are not enough to ensure the security.

In BLE, the LE security modes can be used to enforce a security standard. This functionality is available only in some devices. The security levels in each mode are very limited. For example, the level four in mode one enforces 128-bit encryption key size, public key cryptography and authentication together. Practically, there are applications which requires more security levels with different combinations. For instance, the device authentication is impossible in Just Works association model. But one can enforce 128-bit encryption key size and public key cryptography. Currently, there is no such level exist in the BLE.

Also, the pairing feature exchange parameters influence security mechanism selection process but it less dominant than LE security modes and attribute permissions. Though these three concepts ensure Bluetooth security, they are implemented independently at three different layers of host stack. This is a big drawback in Bluetooth. This issue is solved in the proposed architecture by assigning all security requirements directly to data resources. The pairing feature exchange parameters can be extracted directly from the data resource access permissions.

7.3 Passkey Authentication Procedure

In LE secure connections, the passkey value is transferred over an unencrypted channel for authentication in the passkey entry association model. The passkey is transferred bit by bit to another device. This is a computation intensive complex process. The passkey entry authentication mechanism can be implemented in much simpler way as follows. Initially, anyone of the device need to generate a 20-bit random number and displayed on the device screen. User will transfer the passkey value to another device. At this point, the passkey value is available on both devices. Since public key cryptography algorithm is used in LE secure connections, the shared secret i.e., LTK can be generated without user intervention. Once the LTK is available, both devices need to compute XOR value for the passkey and the LTK. The XOR operation result is the new LTK. Finally, anyone of the device need to create link encryption challenge as defined in the core specification. This link encryption challenge will succeed only if there are no MITM attacks. This mechanism does not suitable for LE legacy pairing passkey entry association model.

7.4 Numeric Comparison for KeyboardOnly devices

In BLE, the numeric comparison method is made suitable only if both devices have display as shown in the Figure A.5. But this association model can also used if the input-output capability of a device is KeyboardOnly and another device has a display. According to the core specification, the numeric comparison value is generated on both devices. The value will be displayed on the device which has a display. User need to enter the same numeric value on the KeyboardOnly device. On the KeyboardOnly device, the entered value is compared with the generated numeric comparison value. If they are equal, the connection will be proceeded to the next step otherwise terminated immediately.

8 Conclusion

This study has evaluated security strength of encryption and authentication methods defined in the BLE core specification. The implementation drawbacks in such security techniques are highlighted. Three different BLE security concepts were found during the research. They are, pairing feature exchange parameters, LE security modes and attribute permission. The advantage and disadvantages of each security concepts were discussed. We found that the authentication methods defined in the core specification are not suitable for headless BLE devices. To solve this issue, there are four novel authentication methods were presented in this research work. These authentication methods can be combined with standard BLE encryption methods. During research, we compiled several combinations of authentication and encryption typologies. From these typologies, we selected key press authentication technique based on device input-output capability and ETSI EN 303 645 regulation.

The implementation difficulties were discussed. During testing, the pairing feature exchange values are verified against design parameters. The impact of encryption key length and connection parameter on link speed were explained clearly. Apart from this, the key press authentication method is tested for the condition where bonding information is lost on anyone of the device. In this context, we determined that the re-pairing flow is completely different for Android and iOS smart phones. So, the key press authentication method is revised, and we provided an alternative solution suitable for all smart phones. The advantages and disadvantages of alternative key press authentication method were described. Finally, threat modelling approaches such as threat-centric, system-centric were applied to determine threats and vulnerabilities exist in the system. These vulnerabilities are evaluated based on STRIDE and DREAD methodology.

There are two fundamental security requirements must be ensured to protect the device from PIN cracking vulnerability. They are, encryption key length and shared secret generation algorithm. We examined LE security modes and determined the fourth level in mode one partially meets those two fundamental requirements. We elaborated this issue and suggested new solutions which may be considered for the upcoming Bluetooth core specification. At present, only the state of bonding is provided over the unencrypted channel. In future, the same channel can be used for sending pairing state and key press authentication notification. Since the BLE authentication mechanisms are different from web security, there are no methodologies available to estimate security strength. So, further research can be carried out to find a quantitative evaluation method based on effort needed for the MITM attack. Moreover, the authentication and interoperability problems can be solved in a better manner by implementing a custom security protocol on top of GATT interface.

Bibliography

[1] Apple. *Accessory Design Guidelines for Apple Devices*. 2023. URL: https://developer.apple.com/accessories/Accessory-Design-Guidelines.pdf (cit. on p. 56).

[2] A. Barua, M. A. Al Alamin, M. S. Hossain, E. Hossain. "Security and privacy threats for bluetooth low energy in iot and wearable devices: A comprehensive survey". In: *IEEE Open Journal of the Communications Society* 3 (2022), pp. 251–281 (cit. on pp. 39, 71, 73).

[3] Bluetooth SIG. *Assigned Numbers*. Dec. 2023. URL: https://www.bluetooth.com/wp-content/uploads/Files/Specification/Assigned_Numbers.pdf?v=1703431207015 (cit. on p. 37).

[4] Bluetooth SIG. *Bluetooth Core Specification 5.2*. Dec. 2019. URL: https://www.bluetooth.com/specifications/specs/core-specification-5-2/ (cit. on pp. 31, 34, 53, 64, 85–88).

[5] Bluetooth SIG. *Bluetooth Launch Studio*. URL: https://www.bluetooth.com/ (cit. on p. 25).

[6] D. J. Bodeau, C. D. McCollum, D. B. Fox. "Cyber threat modeling: Survey, assessment, and representative framework". In: *Mitre Corp, Mclean* (2018) (cit. on p. 70).

[7] M. Cäsar, T. Pawelke, J. Steffan, G. Terhorst. "A survey on Bluetooth Low Energy security and privacy". In: *Computer Networks* 205 (2022), p. 108712 (cit. on p. 39).

[8] R. Cayre, F. Galtier, G. Auriol, V. Nicomette, M. Kaâniche, G. Marconato. "InjectaBLE: Injecting malicious traffic into established Bluetooth Low Energy connections". In: *2021 51st Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*. IEEE. 2021, pp. 388–399 (cit. on p. 16).

[9] Christiaan Brand. *Security Issue with Bluetooth Low Energy (BLE) Titan Security Keys*. May 2019. URL: https://security.googleblog.com/2019/05/titan-keys-update.html (cit. on pp. 16, 17).

[10] Embeddedcentric. *BLE profiles services characteristics device roles and network topology*. URL: https://embeddedcentric.com/lesson-2-ble-profiles-services-characteristics-device-roles-and-network-topology/ (cit. on p. 27).

[11] European Telecommunications Standards Institute. *ETSI EN 303 645 V2.1.1 - Cyber Security for Consumer Internet of Things*. 2020. URL: https://www.etsi.org/ (cit. on p. 52).

[12] C. Gupta, G. Varshney. "An improved authentication scheme for BLE devices with no I/O capabilities". In: *Computer Communications* 200 (2023), pp. 42–53 (cit. on p. 39).

[13] K. Lee. *Secure, Usable and Practical Authentication for the Internet of Things*. The University of Wisconsin-Madison, 2022 (cit. on p. 39).

[14] K. Lee, Y. Yang, O. Prabhune, A. L. Chithra, J. West, K. Fawaz, N. Klingensmith, S. Banerjee, Y. Kim. "AEROKEY: Using ambient electromagnetic radiation for secure and usable wireless device authentication". In: *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 6.1 (2022), pp. 1–29 (cit. on p. 39).

[15] Martin Woolley. *Developer Study Guide: Bluetooth Low Energy Security 1.2.1*. Mar. 2023. URL: https://www.bluetooth.com/bluetooth-resources/le-security-study-guide/ (cit. on pp. 19, 21, 22, 25, 31, 90).

[16] Mohammad Afaneh. *Bluetooth Addresses Privacy in Bluetooth Low Energy*. Apr. 2020. URL: https://novelbits.io/bluetooth-address-privacy-ble/ (cit. on p. 29).

[17] Mohammad Afaneh. *Bluetooth GATT*. Apr. 2023. URL: https://novelbits.io/bluetooth-gatt-services-characteristics/ (cit. on p. 26).

[18] National Institute of Standards and Technology. *National Vulnerability Database*. URL: https://nvd.nist.gov/ (cit. on p. 77).

[19] T. L. Nguyen, T. K. Dang, T. T. Dang, A. T. Nguyen Thi. "A three-way energy efficient authentication protocol using bluetooth low energy". In: *Future Data and Security Engineering: 7th International Conference, FDSE 2020, Quy Nhon, Vietnam, November 25–27, 2020, Proceedings 7*. Springer. 2020, pp. 273–289 (cit. on p. 39).

[20] NIST. *replay attack*. URL: https://csrc.nist.gov/glossary/term/replay_attack#:~:text=An%20attack%20in%20which%20the,NIST%20SP%20800%2D63%2D3 (cit. on p. 22).

[21] M. von Tschirschnitz, L. Peuckert, F. Franzen, J. Grossklags. "Method confusion attack on bluetooth pairing". In: *2021 IEEE symposium on security and privacy (SP)*. IEEE. 2021, pp. 1332–1347 (cit. on p. 16).

[22] Q. Zhang, Z. Liang. "Security analysis of bluetooth low energy based smart wristbands". In: *2017 2nd International Conference on Frontiers of Sensors Technologies (ICFST)*. IEEE. 2017, pp. 421–425 (cit. on p. 16).

[23] Q. Zhang, Z. Liang, Z. Cai. "Developing a New Security Framework for Bluetooth Low Energy Devices." In: *Computers, Materials & Continua* 59.2 (2019) (cit. on p. 39).

[24] Y. Zhang, J. Weng, R. Dey, Y. Jin, Z. Lin, X. Fu. "Breaking secure pairing of bluetooth low energy using downgrade attacks". In: *29th USENIX Security Symposium (USENIX Security 20)*. 2020, pp. 37–54 (cit. on p. 39).

All links were last followed on January 31, 2024.

A BLE Security Specification

A.1 Association Model Selection Algorithm

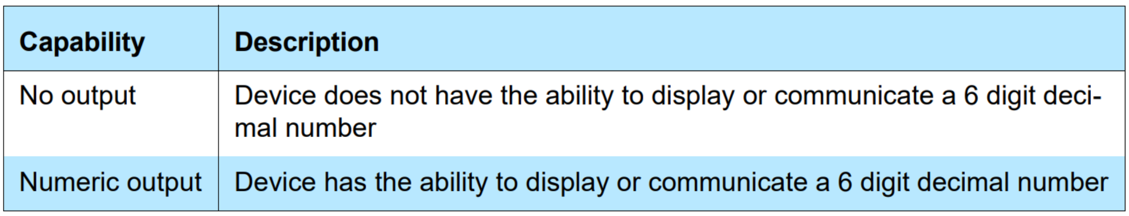
In this section, the influence of pairing feature exchange parameters on selection of association model, symmetric key size, encryption method, and authentication method will be discussed in depth.

A.1.1 Input-output Capability

According to Bluetooth specification [4], all possible means of input a device can receive from external actor are listed in the Table A.1. The output methods are shown in the Figure A.1

Table A.1: Device input capabilities [4]

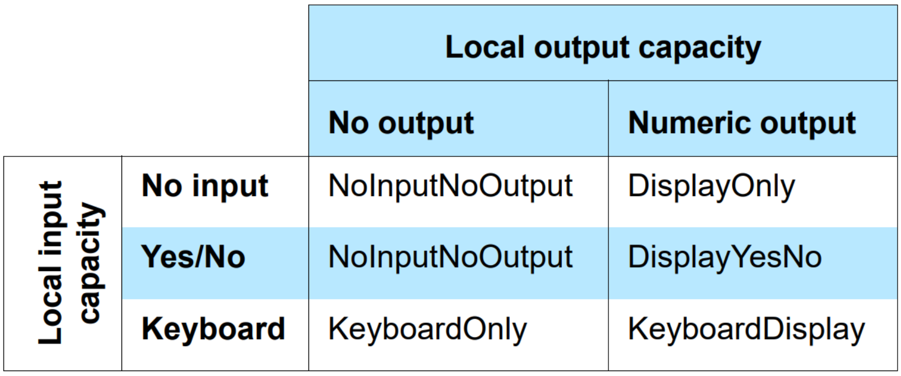
S.No.	Capability	Description
1	No input	Device does not have the ability to indicate 'yes' or 'no'
2	Yes/No	Device has at least two buttons that can be easily mapped to 'yes' and 'no' or the device has a mechanism whereby the user can indicate either 'yes' or 'no'. Note: 'yes' could be indicated by pressing a button within a certain time limit otherwise 'no' would be assumed.
3	Keyboard	Device has a numeric keyboard that can input the numbers '0' through '9' and a confirmation. Device also has at least two buttons that can be easily mapped to 'yes' and 'no' or the device has a mechanism whereby the user can indicate either 'yes' or 'no'. Note: 'yes' could be indicated by pressing a button within a certain time limit otherwise 'no' would be assumed.



Capability	Description
No output	Device does not have the ability to display or communicate a 6 digit decimal number
Numeric output	Device has the ability to display or communicate a 6 digit decimal number

Figure A.1: User output capabilities [4]

The Bluetooth SIG classified HMI capability of a device in to five categories as shown in the Figure A.2 based on Table A.1 and Figure A.1. It is important to note that the device ‘Yes/No’ capability is not considered as input if the device has no output interface, so the device is classified in to NoInputNoOutput category.



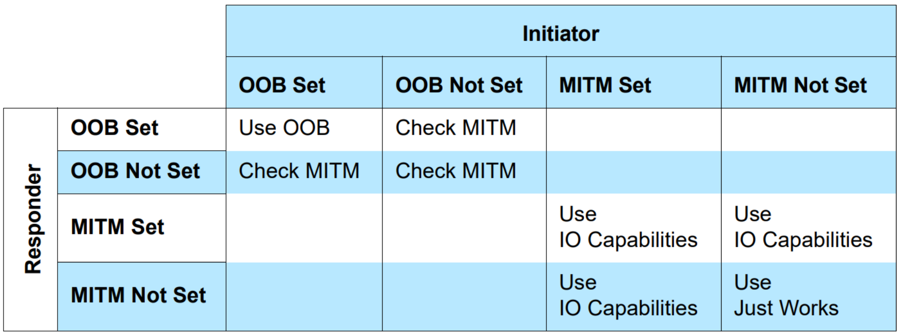
		Local output capacity	
		No output	Numeric output
Local input capacity	No input	NoInputNoOutput	DisplayOnly
	Yes/No	NoInputNoOutput	DisplayYesNo
	Keyboard	KeyboardOnly	KeyboardDisplay

Figure A.2: Input-output capabilities [4]

A.1.2 Out-of-Band

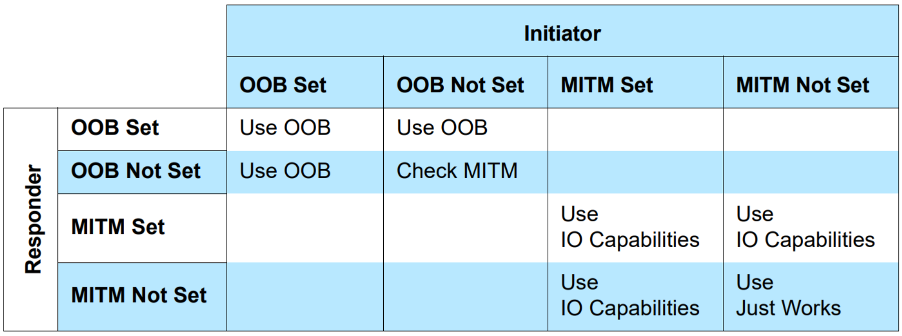
The range of Out-of-Band technology is not same on all technologies. So, product developers should ensure the available Out-of-Band is safe against MITM attack. The process of key generation and authentication in Out-of-Band association model is different for LE legacy pairing and LE secure connections. In LE legacy pairing, Out-of-Band association model is selected only if Out-of-Band authentication data available on both devices and confirmed by Out-of-Band flag in pairing feature exchange. The Out-of-Band association model is selected in LE secure connections if anyone or both devices possess Out-of-Band authentication data.

In both LE legacy pairing and LE secure connections, the selection process for encryption and authentication methods starts from checking availability of Out-of-Band authentication data as shown in the Figure A.3 and Figure A.4.



		Initiator			
		OOB Set	OOB Not Set	MITM Set	MITM Not Set
Responder	OOB Set	Use OOB	Check MITM		
	OOB Not Set	Check MITM	Check MITM		
	MITM Set			Use IO Capabilities	Use IO Capabilities
	MITM Not Set			Use IO Capabilities	Use Just Works

Figure A.3: Rules for using Out-of-Band and MITM flags in LE legacy pairing [4]



		Initiator			
		OOB Set	OOB Not Set	MITM Set	MITM Not Set
Responder	OOB Set	Use OOB	Use OOB		
	OOB Not Set	Use OOB	Check MITM		
	MITM Set			Use IO Capabilities	Use IO Capabilities
	MITM Not Set			Use IO Capabilities	Use Just Works

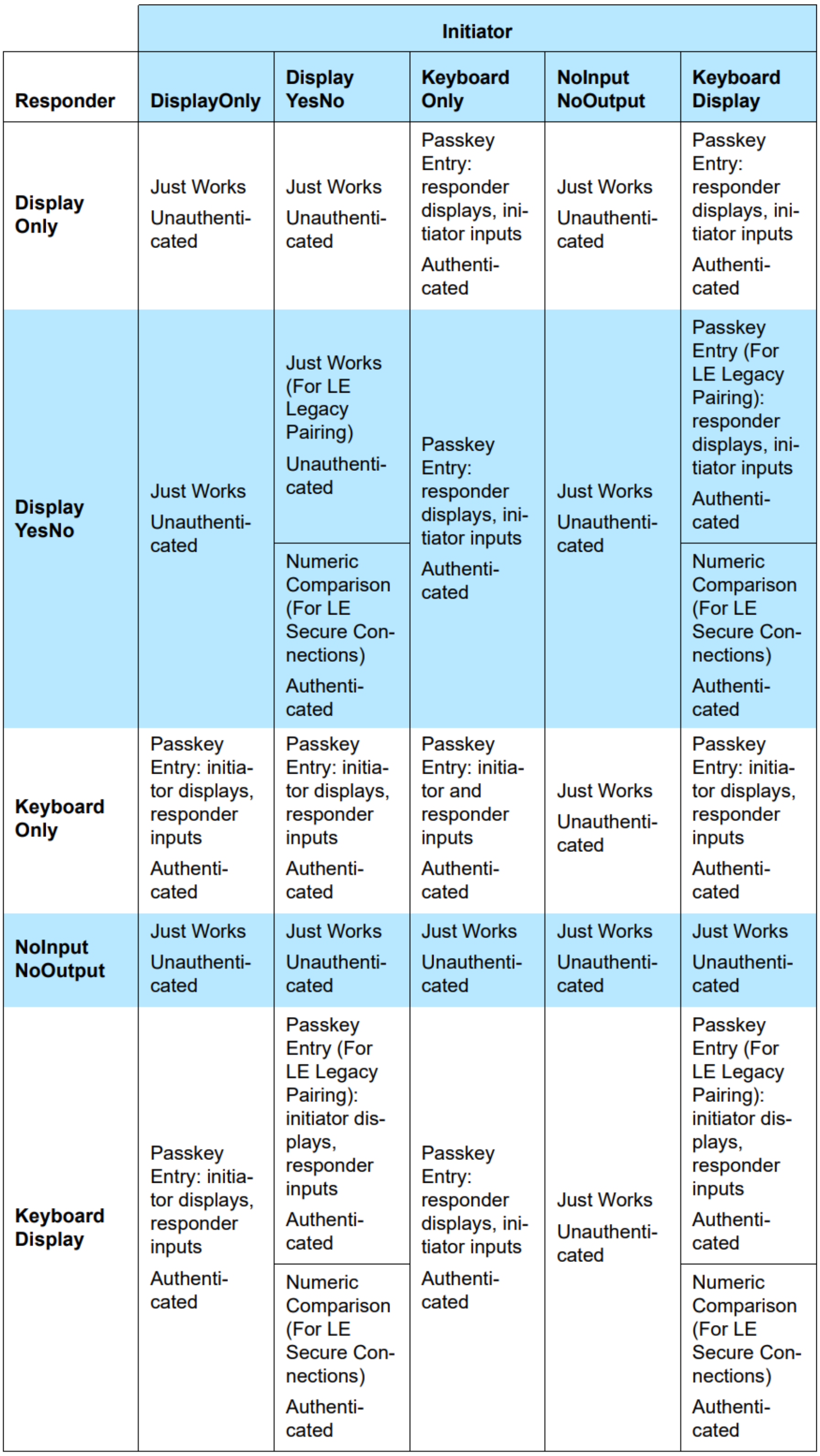
Figure A.4: Rules for using Out-of-Band and MITM flags in LE secure connections pairing [4]

A.1.3 MITM Protection

The MITM flags are examined if Out-of-Band association is not succeeded. As shown in the Figure A.3 and Figure A.4, the outcome of MITM flag comparison between initiator and responder are,

1. Use IO Capabilities
2. Use Just Works

The algorithm is same for both LE legacy pairing and LE secure connections. In IO Capabilities case, the table shown in the Figure A.5 will be used to select an association model for the pairing process. The setting of MITM flag alone will not protect the communication link against MITM attacks. Because the outcome of MITM flag comparison may select an association model which may not provide protection against MITM attacks.



	Initiator				
Responder	**DisplayOnly**	**Display YesNo**	**Keyboard Only**	**NoInput NoOutput**	**Keyboard Display**
Display Only	Just Works Unauthenticated	Just Works Unauthenticated	Passkey Entry: responder displays, initiator inputs Authenticated	Just Works Unauthenticated	Passkey Entry: responder displays, initiator inputs Authenticated
Display YesNo	Just Works Unauthenticated	Just Works (For LE Legacy Pairing) Unauthenticated Numeric Comparison (For LE Secure Connections) Authenticated	Passkey Entry: responder displays, initiator inputs Authenticated	Just Works Unauthenticated	Passkey Entry (For LE Legacy Pairing): responder displays, initiator inputs Authenticated Numeric Comparison (For LE Secure Connections) Authenticated
Keyboard Only	Passkey Entry: initiator displays, responder inputs Authenticated	Passkey Entry: initiator displays, responder inputs Authenticated	Passkey Entry: initiator and responder inputs Authenticated	Just Works Unauthenticated	Passkey Entry: initiator displays, responder inputs Authenticated
NoInput NoOutput	Just Works Unauthenticated	Just Works Unauthenticated	Just Works Unauthenticated	Just Works Unauthenticated	Just Works Unauthenticated
Keyboard Display	Passkey Entry: initiator displays, responder inputs Authenticated	Passkey Entry (For LE Legacy Pairing): initiator displays, responder inputs Authenticated Numeric Comparison (For LE Secure Connections) Authenticated	Passkey Entry: responder displays, initiator inputs Authenticated	Just Works Unauthenticated	Passkey Entry (For LE Legacy Pairing): initiator displays, responder inputs Authenticated Numeric Comparison (For LE Secure Connections) Authenticated

Figure A.5: Mapping of IO capabilities to key generation method [4]

A.1.4 Secure Connection

The LE Secure Connection pairing method will be selected only if both devices support secure connection and corresponding flags in pairing feature exchange should be set according to the specification. In all other cases, the LE legacy pairing method is selected.

A.1.5 Selection of Encryption Key Length

The encryption key length is a minimum value of encryption key length supported by initiator and responder. The initiator and responder supported key length information is available in pairing feature exchange. The lowest value could be fifty-six bits and the maximum value can raise up to 128 bits. The speed of encryption and decryption process will not increase by reducing encryption key size. Because the standard size of encryption key is 128 bits. If the key size is smaller than 128 bits, then the unknown bits are filled with zeros.

A.2 LE Secure Connections

The steps involved in LE secure connections pairing process is shown in the Figure A.6. In LE secure connections, initially a long-term key is generated to encrypt the link. Later, session key is used to encrypt the link which is generated from long term key. The long-term key is generated based on public key cryptography, ECDH in all association models. The authentication process is different for each association models.

In phase three, the CSRK and IRK are distributed based on initiator and responder key distribution list. Unlike LE legacy pairing, the distribution of shared secret is not required at all in LE secure connections.

A.2.1 Just Works

A hacker can do device impersonation attack to read all the data packets. To eavesdrop and carry out MITM attack, highly sophisticated hardware and precise attack timing are required. This type of attacks can be identified through device authentication. Since the Just Works does not support device authentication, the devices are prone to MITM attacks as shown in the Figure A.7.

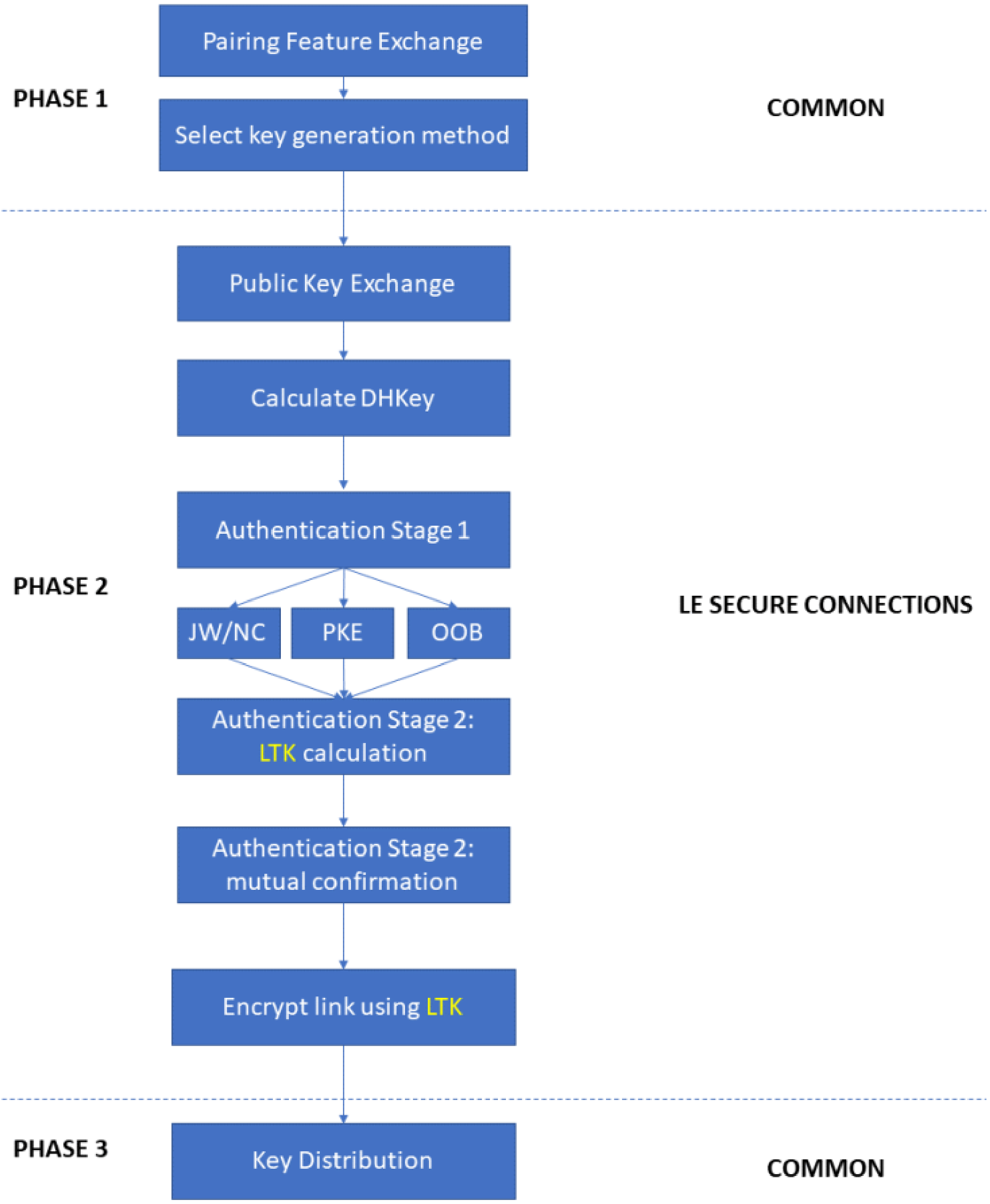


Figure A.6: LE secure connections pairing [15]

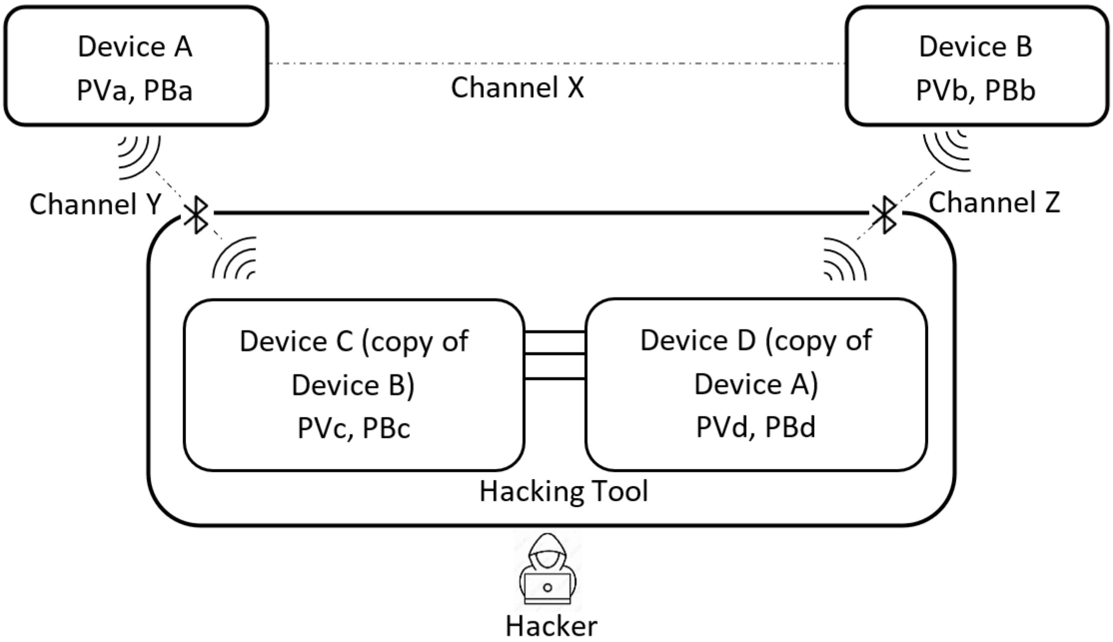


Figure A.7: MITM attack in Just Works method

Where,

- PVa – Private Key of Device A
- PBa – Public Key of Device A
- PVb – Private Key of Device B
- PBb – Public Key of Device B
- PVc – Private Key of Device C
- PBc – Public Key of Device C
- PVd – Private Key of Device D
- PBd – Public Key of Device D
- SHy – Shared secret key (symmetric key) of Channel Y
- SHz – Shared secret key (symmetric key) of Channel Z

The device A and B intended to establish the channel X, but it is failed due to MITM attack. The channel Y and Z are established by the malicious actor by impersonating as legitimate device. The following keys are available to hacker, PBa, PBb, PVc, PBc, PVd, PBd, SHy and SHz. Since it is very hard to derive private keys of Device A and B from SHy, SHz, PBa and PBb, the private keys PVa and PVb are not available to hacker. But the keys SHy and SHz are sufficient for carrying out MITM attack.

Declaration

I hereby declare that the work presented in this thesis is entirely my own and that I did not use any other sources and references than the listed ones. I have marked all direct or indirect statements from other sources contained therein as quotations. Neither this work nor significant parts of it were part of another examination procedure. I have not published this work in whole or in part before. The electronic copy is consistent with all submitted copies.

place, date, signature