NTNU

Norwegian University of
Science and Technology

Secure and User-Friendly Commissioning and Bootstrapping of Constrained Devices

Alessandro Coppa

Master of Science in Communication Technology
Submission date: December 2015
Supervisor: Frank Alexander Krämer, ITEM

Norwegian University of Science and Technology
Department of Telematics

Title: Secure and User-Friendly Commissioning and Bootstrapping of Constrained Devices

Student: Alessandro Coppa

Problem description:

The number of the low energy systems connected to the Internet is quickly raising and the experts foresee over 30 billions of devices by 2020. This is due to the recent low-cost and low-energy technologies like BLE and ZigBee. In addition specific protocols have been created to supply new scenarios. Particular efforts are given to 6LoWPAN, that introduce the advantages of IP-based technologies in the constrained devices which allow end-to-end connections between them.

At the current state there is not a precise architecture on how to build this novel network. Problems like security and privacy are creating some doubts about the promises of this computing revolution. In particular, one of the most critical issue is the key exchange mechanism used to establish a secure connection. In fact, a loophole during the setup procedure could compromise the entire communication.

This thesis will focus on the secure and user-friendly methods to initialize a connection between constrained devices for IoT applications. At first I will study some actual IoT use cases in the smart home product family and their pairing mechanism, focusing on how they exchange the cryptography keys. Then I will propose some solutions for commissioning and bootstrapping of these devices in a plausible future scenario.

Responsible professor: Frank Alexander Kraemer, ITEM

Abstract

The spread of Internet of Things (IoT) systems, based on the introduction of constrained devices into physical objects, has required particular efforts to improve security, privacy and simplicity of such systems. The goal of this work is to find secure and user-friendly ways for the commissioning and bootstrapping of constrained devices, working with Bluetooth Low Energy (BLE) as the main wireless communication technology. An important assumption for future constrained devices is the absence of input/output interfaces like keyboards and displays. That represents the real challenge which makes the traditional security mechanisms unfeasible.

At first, an analysis of some products from the current generation of IoT systems has shown a lack in term of security or simplicity. Starting from these results this work defines the security requirements to ensure authentication and confidentiality/integrity for the information exchanged. Then it presents some user-friendly solutions to initiate such devices based on the security requirements defined. These solutions require only few interactions and knowledge for the final users.

After a study of the security features offered by BLE and defined the user-friendliness level required, scenarios that combine BLE and NFC technologies seem to be the best solutions. NFC can be used as the Out-of-Band (OOB) channel for BLE pairing method, providing authentication and limiting the risk of Man In The Middle (MITM) attacks and passive eavesdropping. This combination also increase the simplicity and avoid typical authentication techniques like passkey insertion. The conjunction BLE-NFC can be easily managed to produce several solutions for different scenarios. Scenario with movable devices represents the easiest solution. Instead, fixed targets require support devices, like smartphone or tablet, to reach the goal. However, both are based on the same principles.

The presented solutions aim to be considered for the next generation of IoT systems, increasing security and user-friendly level. Current and near future chips combine BLE and NFC in a unique product which make the solutions low cost and easy to implement.

Preface

The work on this Thesis has been performed in collaboration with the Department of Telematics at NTNU and the Department of Information Engineering (DII) at UNIVPM. I would like to thank Professor Luca Spalazzi for giving me the opportunity and the support needed to produce this work. Further thanks goes to Associate Professor Frank Alexander Kraemer who provided big help and advices.

Last, but certainly not least, thanks to my brother Alessio who has helped me with my 'language problems' and my parents who have always supported me.

Ancona, October 2015
Alessandro Coppa

Contents

List of Figures

List of Tables

List of Acronyms

6LoWPAN IPv6 over Low power Wireless Personal Area Networks.

AES Advanced Encryption Standard.

BLE Bluetooth Low Energy.

ECC Elliptic Curve Cryptography.

ECDH Elliptic Curve Diffie–Hellman.

IETF Internet Engineering Task Force.

IO Input/Output.

IoT Internet of Things.

MITM Man In The Middle.

NFC Near Field Communication.

OOB Out-of-Band.

SIG Special Interest Group.

SoC system on a chip.

TTP Trusted Third Party.

WPAN Wireless Personal Area Network.

Chapter 1
Introduction

The idea behind the IoT, as exposed by [MC14], is to make the **Things** smart through the use of the **Internet** to send, receive and communicate information. The word things is not intended computer, smartphone or table, but objects of everyday life like dishwasher, door, alarm clock and so on. Figure 1.1 represents a good definition of IoT.

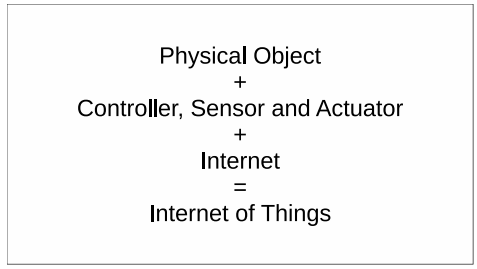

Figure 1.1: IoT definition [MC14]

To imagine this solution, the physical objects need to be unreached of tiny electronic devices wireless enabled. These devices have to fit two important constrains:

- **Low-cost** in order not to increase too much the price of the things.
- **Low-power** in order not to increase the power consumption and provide long durability of the things which are, usually, not plugged into any cabled power supplies.
- **Low-dimension** in order not to increase the volume of the things.

Currently this implies that the constrained devices are limited in computational capacity, memory and number of interfaces.

In this thesis we focus on devices with no input and no output interfaces. Interfaces are useful for interacting with the device. A typical example is the possibility to introduce, with a keyboard as an input interface, a key password in order to provide authentication and start the communication with a WiFi router. However, adding a keyboard or a display in everyday objects needs to redesign their structure which increase the volume. In addition it raises cost and power consumption. This is sufficient to understand the research on solutions with limited devices without Input/Output (IO) interfaces.

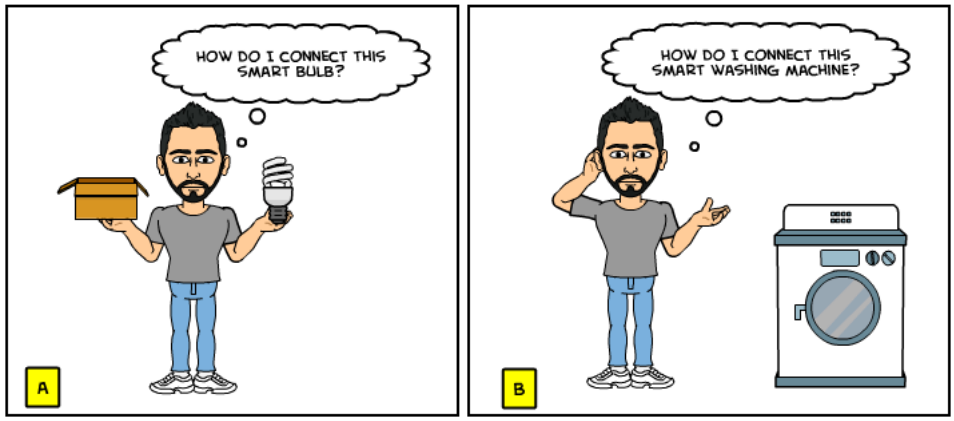

Figure 1.2: User scenario. [A] The user wants to setup a new smart bulb. [B] The user wants to setup a new smart washing machine.

Starting from the concepts exposed above, one of the thesis topic is **security**. Since the physical objects are part of our life, they often manage user's confidential information. For example, a smart watch could measure the heart rate and probably the user does not want to share it. It is clear that *confidentiality*, *integrity* and *authentication* are essential requirements in the IoT field. According to what has been reported in [SA00], authentication is the most critical point in a wireless communication. If we provide a secure authentication, we would avoid problems like MITM attacks. In simple terms, MITM is the situation where an attacker C, placed in the middle of a channel, can read, insert and modify messages in a communication between user A and user B, acting on behalf of them. MITM is also called active eavesdropping, in contrast with passive eavesdropping, where an attacker can read the content of the messages.

The second topic of the thesis is **user-friendliness**. We assume that every kind of users, including users not familiar with electronic devices, will use the IoT solutions in the near future. Lockitron, for example, is a smart solution that allowed to remotely control the house door. It is clear that a complex usage could increase

the risks or inconveniences. Therefore it is important to maintain the setup process of the objects as simple as possible.

1.1 Scope

The goal of the thesis is to find some secure and user-friendly solutions for the *commissioning* and *bootstrapping* of constrained devices without IO interfaces. Figure 1.2 shows the user scenario that this work wants to solve. Commissioning refers to the act of putting a system or a device into operation, giving to him a desired configuration. Bootstrapping, instead, indicates the process required to configure and connect nodes with no prior knowledge of each other in order to build a ready to use network [SOM+13]. Particular attention is given to the key management process which is in charge of the critical keys exchange during the setup of a connection.

As an example of a working scenario, during the setup phase of a new IoT device, the priority is to provide a secure way to exchange the cryptographic keys and offer authentication features. It has to avoid attacks from malicious users who can act to control the device or steal the confidential information. A well implemented commissioning and bootstrapping should prevent MITM attacks and passive eavesdropping of the initial keys exchanged. An eavesdrop of them can comprise the entire channel for the subsequent communication. In addition all these procedures have to make the system easy to use for the final users.

We chose the new BLE standard as the main wireless communication technology. Compared to the classical Bluetooth technology it is cheaper and it has less power consumption, but also reduced throughput. Compared to other low-power technologies it is more spread over the current generation of smartphones and laptops. Some commissioning and bootstrapping solutions will be presented after a study of the BLE pairing methods. Due to the no IO interfaces assumption, OOB channels will play a relevant role in the working scenario proposed. We will present some of them with focus on NFC technology, identified as one of the most secure and user-friendly wireless channel to exchange information.

This work could be used as starting point for next implementations of secure and user-friendly IoT products. BLE can be used for IoT products due to the introduction of the new IPv6 over Low power Wireless Personal Area Networks (6LoWPAN) protocol. It allows low-energy wireless communication technologies to take part of the Internet. A final analysis of the hardware commercially available will show the potentiality of the proposed solutions in the near future.

1.2 Structure of Thesis

First chapter is the current chapter and it includes introduction to the work. Chapter 2 and Chapter 3 contain the theoretical background. The former analyzes the communication technologies and protocols like Bluetooth Low Energy, NFC and 6LoWPAN protocol. The latter is an overview of security aspects with particular regards for the most commonly used cryptographic algorithms. Chapter 4 discusses some commercially available IoT products with a focus on the architecture, the technologies, the security and user-friendliness features. Chapter 5 defines a feasible working scenario for future implementations. Chapter 6 is the core of the work and it proposes some solutions to establish a secure and user-friendly connection with BLE devices. A discussion of the current hardware support for the proposed solutions is presented in Chapter 7. Finally a conclusion and the prospects for the further work are described in the Chapter 8.

Chapter 2
Background Technologies

This chapter presents an introduction to some communication technologies and protocols relevant for IoT systems. They form the basis for the following chapters of the thesis.

2.1 Bluetooth Low Energy

Bluetooth is a wireless communication technology, developed and managed by Bluetooth Special Interest Group (SIG). It was standardized as IEEE 802.15.1 but it is no longer maintained by IEEE. It allows to build Wireless Personal Area Network (WPAN) and exchange data between devices, working at a radio frequency of 2.4 GHz.

In Bluetooth specifications, *pairing* is the term describing the process that produce one or more shared secret keys between the parties involved in the communication. These keys are used to encrypt and protect the channel. Pairing is a mandatory step and it is generally followed by *bonding*, the act of storing the keys previously generated for using them in subsequent connection restart. In other terms, *bonding* creates a trusted link between the devices which prevent the repetition of the pairing process [PMoBS14].

Since Core Specification 4.0, Bluetooth has two forms of wireless technology, the Basic Rate / Enhanced Data Rate (BR/EDR), also called classic Bluetooth, and the new Low Energy (LE) [PMoBS14]. BR/EDR typically refers to Core Specification 2.1. Both forms operate at the same frequency but they are not interoperable.

BLE presents several differences from the classic version. The main feature is the low power consumption which allow new working scenario and increase the battery life of the electronic devices. In addition, low cost of production and low latency make this technology ideal for IoT systems, where the reduced throughput does not produce negative effects. Table 2.1 shows a comparison of the two forms.

Table 2.1: Classic Bluetooth and BLE Comparison

Technical Comparison	Classic Bluetooth	BLE
Radio Frequency	2.4Ghz	2.4Ghz
Max Range	10-100 m	50 m
Bit Rate	1-3 Mbps	1 Mbps
Throughput	0.7-2.1 Mbps	0.2 Mbps
Setup time	< 6 sec	< 0.003 sec
Power Consumption	Medium	Very Low
Battery Life	Days	Months to Years

2.1.1 BLE Core Specification 4.2

BLE Core Specification 4.2 was released on December 2014 [PMoBS14] and at the moment is the latest version available. Taken from Bluetooth BR/EDR, it introduces in BLE several mechanisms to increase security during the pairing phase. This extended version of the specification is also called *Secure Connection*. The previous version is still active and it is referred as *Legacy* or *Simple Pairing*.

Compared to the previous v4.0 and v4.1, first remarkable introduction of BLE 4.2 Secure Connection is the Elliptic Curve Diffie–Hellman (ECDH) key exchange algorithm during the pairing. After an initial generation of the Elliptic Curve Cryptography (ECC) asymmetric keys, the parties involved can exchange the public information and derive a shared secret between them. This procedure avoids the risk of passive eavesdropping. This was a lack of the previous BLE versions, since an eavesdrop of the initial secret keys by a malicious user can compromise the entire communication, mining the confidentiality and integrity requirements.

The ECDH protocol is implemented in all the pairing methods of BLE 4.2 Secure Connection. They consist in four methods of authentication. Three of them are presented in the previous versions, *Just Works*, *Passkey entry* and *OOB*. *Numeric Comparison* is the only new one. *Just Works* is the simpler to use but it is the only unauthenticated. It means that the connection is not protected from a MITM attack. *Passkey entry* requires a 6-digit passkey insertion to provide authentication. *Out Of Band OOB* uses an alternative channel to send the information required for authenticate the parties, leaving the security level up to the strength of the specific channel. Finally, *Numeric Comparison* requires to compare a number in both devices.

Table 2.2 presents a brief comparison between BLE 4.2 Secure Connection and BLE Legacy. Due to the significant improvements and for its recent release, we focus on BLE Specification Version 4.2 Secure Connection for the rest of this work.

Table 2.2: BLE security comparison

	Specs	Pairing Methods	Passive eavesdropping	MITM
BLE Legacy	4.0-4.2	Just Works Out Of Band Passkey	-	if Authenticated
BLE Secure Connection	4.2	Just Works Out Of Band Passkey Numeric Comparison	ECDH	if Authenticated

2.1.2 BLE 4.2 Secure Connection - Pairing Algorithm

In BLE Core Specification 4.2 Secure Connection a module called Security Manager (SM) is in charge to handle the pairing process. Pairing consists of three phases with the goal to distribute the cryptographic and provide an authentication method. [PMoBS14]

- in **Phase 1** where the devices exchange input/output capabilities, OOB data availability, authentication requirements, key sizes requirements and which transport specific keys to distribute. Some of them are essential to decide the pairing method to use in phase 2.
- in **Phase 2** where the devices use a particular pairing method to generate a Long Term Key (LTK), a 128-bit symmetric key used to encrypt the link using AES-CCM cryptography.
- **Phase 3 (Optional)** where the devices could exchange some transport specific keys performed in an encrypted link using LTK.

Figure 2.1 shows the pairing algorithm. It can be noted that Phase 1 and Phase 3 are independent from the method used in Phase 2. Phase 2 is the most relevant step of the entire process and it is described in the following.

Phase 2 - Public Key Exchange

The pairing algorithm starts with the generation of the ECC keys. This step is completed only once and it can be done before the pairing process. SKa and PKa are respectively the secret key and the public key of the user A.

Assuming of a completed Phase 1, where the devices decide the pairing method to use, the first step of Phase 2 is to exchange the public keys. When the keys are

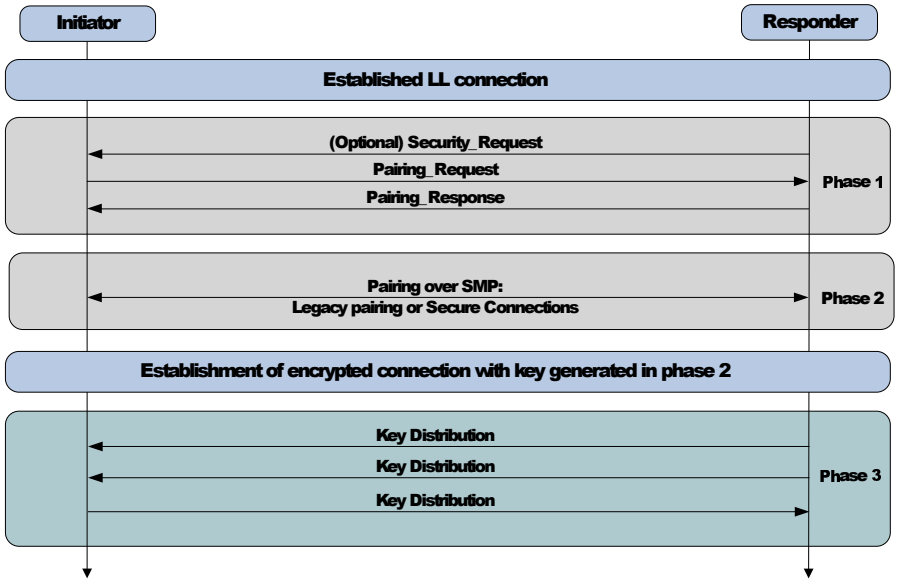

Figure 2.1: BLE Pairing Algorithm - taken from [PMoBS14]

exchanged both devices can compute the Diffie-Hellman shared secret. This step is the same for all the pairing methods. Figure 2.2 shows the process.

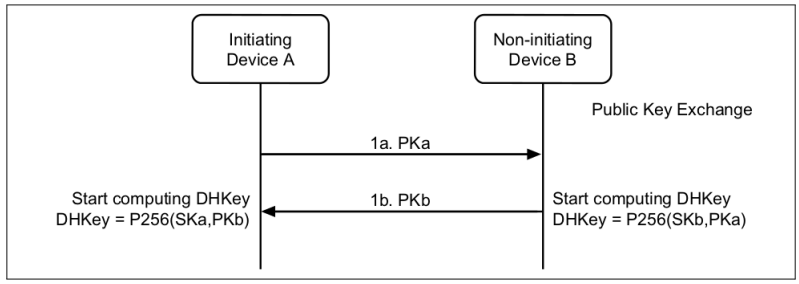

Figure 2.2: BLE Public Key Exchange - taken from [PMoBS14]

Phase 2 - Authentication Stage 1 - Out of Band

This step is different for each pairing method. For simplicity we report only the OOB method where every device involved in the communication is able to send data in an alternative channel different from BLE.

Firstly, each device choose a private number r and compute a confirmation value C. Secondly, the parties can start the OOB communication where r, G and the BLE *address* are exchanged. With these information and the public key obtained in the previous step the devices can compute again the confirmation value using the the partner's values and compare them. If the comparison is successful, each device selects and sends a number N for the next stage. Otherwise, the pairing is aborted.

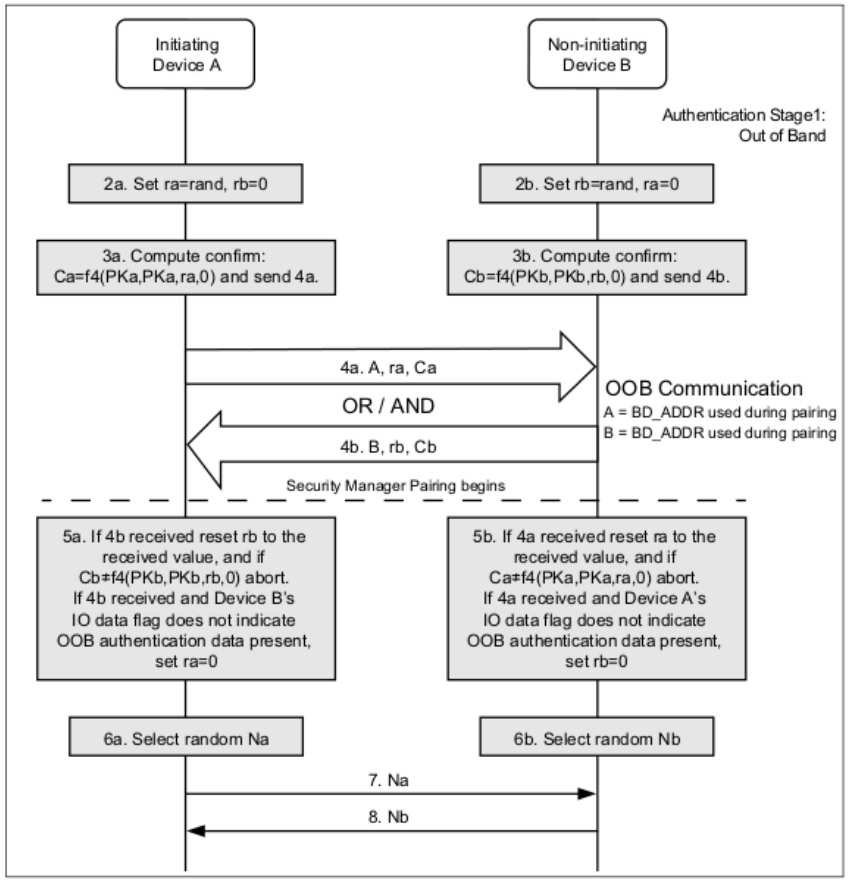

Figure 2.3: BLE Authentication Stage 1 with OOB - taken from [PMoBS14]

It is important to note that the OOB channel can be used to start the pairing algorithm. In this case, the in-band public keys exchange (step 1 of Phase 2) is placed between step 4 and step 5 after the OOB communication. In other terms, the OOB channel helps with the device discovery.

Phase 2 - Authentication Stage 2

In stage 2 the devices can compute the Long Term Key (LTK) and a MacKey using the information previously obtained. After that, they produce another confirmation value *Ei*. Like stage 1, they exchange their own confirmation value and proceed to compare them. If the comparison is successful, the devices are correctly paired. Otherwise, the pairing is aborted.

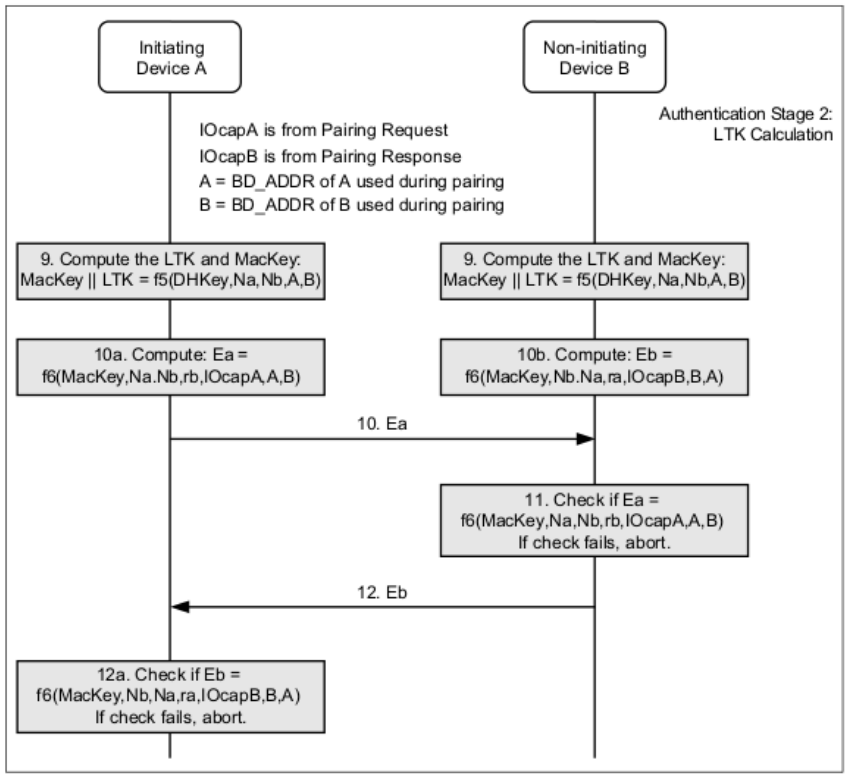

Figure 2.4: BLE Authentication Stage 2 - taken from [PMoBS14]

2.1.3 Star Network Topology

Bluetooth offers two kind of roles: *Master* and *Slave*. Master is the initiator of the connection and Slave is the responder. Anyway, BLE only support a star network topology. It means that in a BLE network there is only one Master and several Slaves connected to it. In addition, a device cannot be Master and Slave at the same time and a Slave can be paired with only one Master.

2.2 NFC

NFC is a set of technologies that enables electronic devices to establish a bidirectional interaction with each other by putting the devices in contact or bringing them close enough to a distance of typically 4 to 10 cm. NFC devices employ electromagnetic induction to exchange information within the unlicensed radio frequency ISM band of 13.56 MHz at rates ranging from 106 kbit/s to 424 kbit/s [For].

NFC Forum is the no-profit association responsible of the projects. Founded in 2004 by Nokia, NXP Semiconductor and Sony, it counts now more than 160 members. It provides the NFC specifications, as well as promote NFC over the world and certify the devices compliance. They released the fist standard in 2006 as ECMA-340 and ISO/IEC 18092 which are based on existing radio-frequency identification (RFID).

NFC devices can operate both in active and passive modes. If the device generates its own RF field than it is called active. It needs a power supply and it can initiate a communication with another NFC device. Passive devices instead does not produce a RF field but it retrieves the field from another active device [HB06].

Considering the usage, NFC devices can work in three different modes [For]:

- **Tag Reader/Writer**: enables NFC-enabled devices to read information stored on inexpensive NFC tags embedded in smart posters and displays, providing a great marketing tool for companies. NFC tags typically contain data between 96 and 4,096 bytes of memory and are read-only, but may be rewritable.
- **Peer-to-Peer**: enables two NFC-enabled devices to communicate with each other to exchange information and share files.
- **Card Emulation**: enables NFC-enabled devices to act like smart cards, allowing users to perform transactions such as purchases, ticketing, and transit access control with just a touch. The NFC-enabled device communicates with an external reader much like a traditional contactless smart card.

NFC has drawn particular attentions during latest years and it is spreading over the next generation of devices for IoT systems.

2.3 IPv6 and 6LoWPAN

Internet Protocol version 6 (IPv6) is the latest version of the Internet Protocol (IP) developed by Internet Engineering Task Force (IETF). If provides an identification and location service for devices connected on a network, allowing the routing of messages exchanged on it. IETF introduced IPv6 specification in 1998 in order to

overcome the limits of the previous IP protocol, in particular the problem of IPv4 address exhaustion.

IPv6 uses a 128-bit address, represented as eight groups of four hexadecimal digits, allowing $3{,}4x10^{38}$ addresses. IPv4, instead, uses 32-bit addresses and provides approximately 4.3 billion addresses. Hence, both solutions are interoperable with each other. The increment of IP address opens scenarios where even new devices can take part of the Internet, which is the case of IoT systems.

Recently, IETF defined 6LoWPAN protocol with the goal to extend IPv6 usage over WPAN. The concept idea behind 6LoWPAN is to connect more things to the Internet, allowing IPv6 packets to be carried over low-power communication technologies like BLE and ZigBee, different from the classic WiFi. Figure 2.5 shows the basic architecture where a 6LoWPAN network is connected to the Internet through an *edge router*.

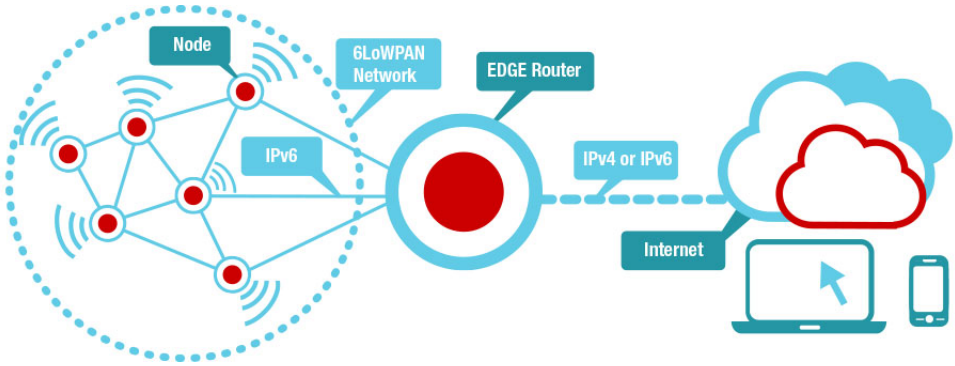

Figure 2.5: 6LoWPAN network - taken from [Ins]

The big advantage of this solution is the simple end-to-end connection between devices. Up until now a complex application gateway was needed to allow low-power devices contact the Cloud. 6LoWPAN solves the problem by introducing an adaptation layer on the classic TPC/IP stack that allow IPv6 packets to work with IEEE 802.15.4 wireless links. An edge router is needed and it is the device in charge of the conversion between 6LoWPAN and the standard IPv6 address. No more complex application layers are required to add Internet features in low-power devices. Figure 2.6 shows the 6LoWPAN protocols stack in comparison with the classic IP stack.

Since the edge router works at the network level, it is independent from any application protocols. Therefore, this create an end-to-end connection between nodes

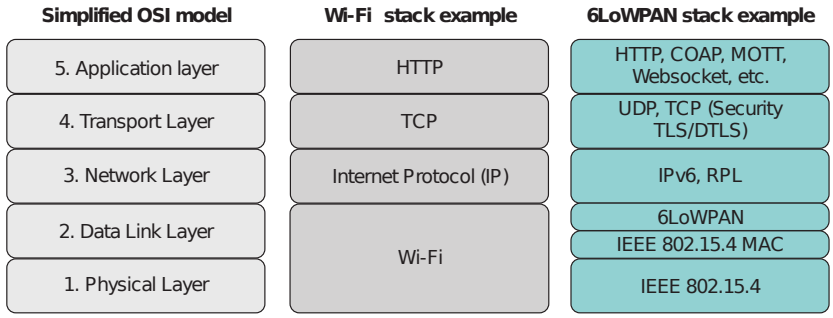

Figure 2.6: 6LoWPAN stack - taken from [Ols14]

that can take full advantage of the IP technologies, favoring open standards and interoperability [Ols14]. These benefits make the protocol ideal for new IoT systems.

While it was originally designed to support IEEE 802.15.4 wireless network only, 6LoWPAN protocol has been the adapted to several other low-power communication technologies like BLE, Sub-1 GHz and low-power WiFi [Ols14].

Chapter 3
Background Security

This chapter presents an introduction to the cryptographic algorithms and protocols used in classical computing but still relevant for constrained devices. In addition Diffie-Hellman keys exchange which incorporates some of the previous concepts is described. It is adopted by several wireless technologies as the main protocol to establish a connection.

3.1 Cryptographic Algorithms

This section presents the differences between symmetric and asymmetric cryptography, the basic ideas behind them, as well as some real implementations along with benefits and drawbacks.

3.1.1 Symmetric Key Cryptography

Symmetric cryptography (or Secret-Key cryptography) is a technique that use a single secret key to encrypt and decrypt a message. The message could be any kind of data like plain text, binary files, audio and video. The key which can be a number or a word is applied to the massage in order to hide the content. Only by applying the same algorithm and password it is possible to obtain the original message. In other terms, the key is a shared secret between two or more parties.

This technique is generally fast to compute and feasible even for devices with limited computational power. However, a single key is required for each couple of users that want to communicate which increase the number of keys to remember. In addition, the shared key has to be exchanged somehow in a secure way before the start of the communication. These are the main drawbacks of the symmetric cryptography. Anyway it is very used, in particular when it is combined with asymmetric cryptography [Bea].

During the years researches developed several implementations of this idea, each one with its own features and security benefits. The most known are DES, 3DES, CAST5 and AES. Currently, The last one is the most used and secure.

AES

Advanced Encryption Standard (AES) is the specification for a symmetric algorithm, established by NIST in 2001. In the last years, it has replaced DES and it has been adopted by several institutions like USA government due to its higher level of security.

AES operates with data block of 128 bits and variable key size of 128, 192 or 256 bits. Obviously, the longer keys are more secure, however it is harder to compute. The length of the key determines the number of cycle repetition that transform and hide the original content. For instance, 128 bits key provides 10 cycles of repetition. Each repetition is the combination of two mathematical operations, substitution and permutation.

AES reaches very high level of security, like reported by [Sec03]: *"The design and strength of all key lengths of the AES algorithm (i.e., 128, 192 and 256) are sufficient to protect classified information up to the SECRET level. TOP SECRET information will require use of either the 192 or 256 key lengths".*

3.1.2 Asymmetric Key Cryptography

Asymmetric cryptography (or Public-Key cryptography) is a technique that use a pair of keys mathematically linked to encrypt and decrypt a message. The idea behind differs from the symmetric cryptography. Any massage encrypted with a public key can only be decrypted by the relative private key and vice versa, applying always the same algorithm.

Any user needs a setup procedure where he gets the private and public keys from a Certification Authority (CA), considered as a Trusted Third Party (TTP). CA is in charge of generation and confirmation of user's keys.

The common uses of public-key cryptography are:

- Confidentiality: an user who wants to send an encrypted message can get the recipient's public key directly from him or from a public directory. The sender uses this key to encrypt the message and he sends it to the recipient. The recipient gets the message and he can decrypt it with his private key which no one else should know.

– Digital signature: an user encrypts the message with his private key and he sends the message. The recipient can decrypt it using the public key of the sender in order to prove his identity.

The great benefit of this cryptography if compared with the symmetric technique is that just the public/private keys pair is needed to communicate with any other user. It limits the number of keys to remember for the users. Moreover, the public key can be passed over the Internet which avoid a previous exchange of a shared secret over a secure channel. The private key is the only one that must be private. However, a problem is that this cryptography is slower than symmetric encryption. It requires far more processing power to both encrypt and decrypt the message's content. Because of the computational complexity, asymmetric encryption is typically used only to securely transfer a symmetric key.

In details, public-key concept relies on mathematical problems without an efficient solution. Problems like integer factorization, discrete logarithm and elliptic curve discrete logarithm are currently hard to compute. These problems form the base for public/private keys generation in several implementations like RSA and ECC. Therefore, it is easy for a user to generate the keys, but much harder to determine the private key from the corresponding public key.

RSA

RSA (named from the initial letters of the authors R. Rivers, A. Shamir and L. Adleman) is an asymmetric algorithm first described in 1977. RSA relies on the factoring problem where the operation of factoring the product of two large prime numbers is hard to compute. The difficulty of the problem is related to the current computational power available.

In details, two prime numbers are the starting point for the keys generation and they must be kept secret to ensure the reliability of the algorithm. After an elaboration of these numbers a pair of keys is produced which can be used to encrypt and decrypt the messages. The longer keys provided, the more secure is the encryption but it also increase the computational time.

ECC

ECC is an asymmetric algorithm proposed at the middle of 80s but recently started to be used. It relies on the elliptic curve discrete logarithm problem to produce a linked pair of private/public keys.

Practically, given an elliptic curve and a generator point **g** on the curve, the method bases its strength on a mathematical function that produces another point

on the same curve. We can repeat the function **n** times which produce **ng** as final point. It is easy to generate this final value but hard to obtain the discrete value **n** (private key), starting from **ng**, **g** and the elliptic curve (public key). This is the discrete logarithm problem on elliptic curve.

ECC is used in specific protocols like ECDH for key exchange or Elliptic Curve Digital Signature Algorithm (ECDSA) for digital signature. The USA National Security Agency (NSA) even suggests to employ it for classified information, using at least 384 bits key.

3.1.3 Summary

Compared to RSA, ECC provides the same security for smaller keys size and consequently less computational time. In addition, smaller keys reduce the storage and transmission requirements. For these reasons ECC is replacing RSA.

However, symmetric cryptography remains the best solution in term of computational time. As we can see on Table 3.1, at the same security level, AES requires smaller keys size. Therefore, symmetric cryptography is still large used in several communication protocols for its efficiency after a key exchange achieved by an asymmetric algorithm. The most relevant example of that is Diffie-Hellman protocol.

Table 3.1: Key length (bits) comparison at same security level - adapted from [Gro]

AES Key Length	RSA Key Length	ECC Key Length
128	3072	256
192	7680	384
256	15360	512

3.2 Diffie-Hellman Key Exchange

Diffie-Hellman (DH) key exchange, defined by Whitfield Diffie and Martin Hellman in 1976 [DH76], is a protocol to securely exchange a cryptographic key over an insecure channel. An insecure channel, as the Internet, is subjected to passive eavesdropping by malicious users who could read the massages exchanged in the channel. Cryptography protects the massages against passive eavesdropping but an insecure channel can compromise it since it cannot avoid the steal of the cryptographic keys during the initial setup of a communication. Therefore, DH algorithm protect the users from this kind of threat. The concept idea is to use asymmetric cryptography between the parties to establish a shared secret between them with very low risk. This shared secret can be used as a symmetric key to encrypt future messages or to derive other more secure keys.

DH relies on discrete logarithm problem. Introducing the original version of the algorithm, given **p** prime number and **g** a primitive root modulo **p** as public initial information, and given **a** an integer, it is easy to compute **A**

$$A = g^a \mod p$$

But it is hard to obtain **a** (discrete logarithm) starting from all the public information (**g**, **p**, **A**). In this case **a** represents the private key.

More in details, during an interaction User A and User B calculate his own public key **A** and **B**, after a previous exchange of the common base **g** and **p**. User A sends the key **A** to User B and vice versa. Both users can now calculate a new value **S**, using their relative private key. **S** represents the shared secret and only User A and User B can know it. A summary algorithm is presented in Fig 3.1.

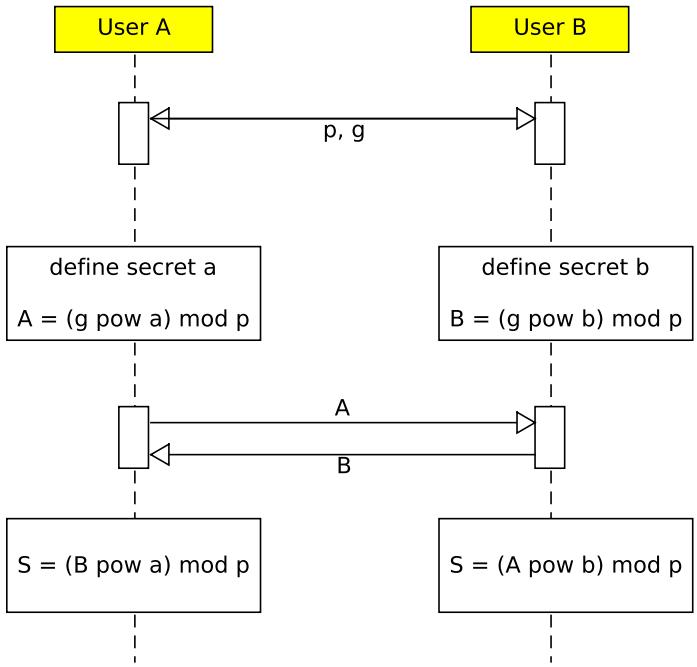

Figure 3.1: Diffie-Hellman key exchange diagram

However, a strong limit of the algorithm is the anonymity (or not-authentication) of the process. Two parties can communicate securely but no one can prove the identity of the other. For this reason, the protocol is generally extended by methods

to provide authentication. For instance, TLS communication protocol incorporates an extended version of DH algorithm.

3.2.1 ECDH

ECDH is a variation of Diffie-Hellman algorithm which changes the mathematical problem, including the benefits of ECC. In this case, given an elliptic curve **c** and a generator point in the curve **g** as public initial information, and taken an integer **a**, it is easy to compute **A**

$$A = n * fun(c, g)$$

But it is hard to obtain **a** (elliptic curve discrete logarithm) starting from all the public information (**c**, **p**, **A**). **a** represents the private key for an user.

Described the mathematical problem, the algorithm proceeds in the same way of DH. A summary algorithm is presented in Fig 3.2.

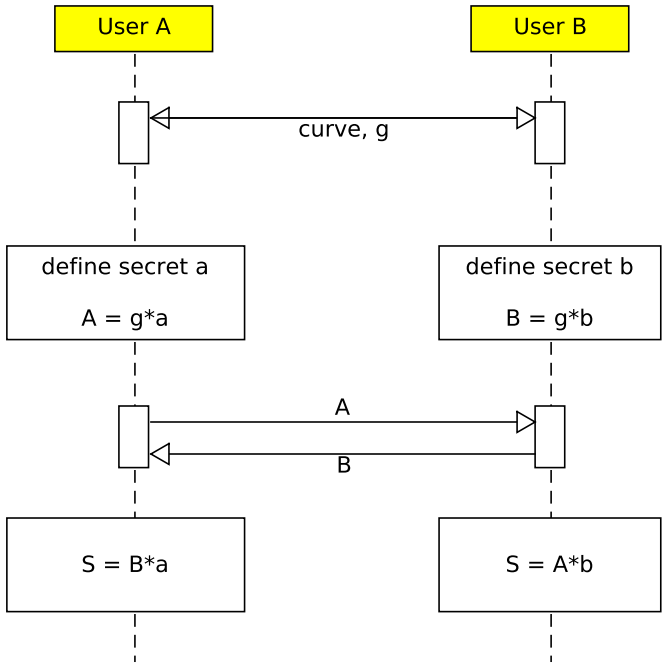

Figure 3.2: ECDH key exchange diagram

Chapter 4
Use cases

This chapter will present several IoT implementations for smart home category products. The purpose is to analyze the architecture and technologies used by each products and their setup mechanism, focusing on the fundamental key exchange method. For any solution we will also discuss the user-friendly level obtained.

This study will introduce some interesting issues for the actual IoT scenario which will be discussed in more details in the next chapter.

4.1 Electric Imp

Electric Imp, founded in 2011, provides a service platform that makes it simple to connect home devices to the Internet. It offers WiFi integrated hardware, software, OS, APIs, cloud services and security in order to decrease cost and time to build a new specific IoT solution. Several independent products already use it. For instance, Lockitron uses Electric imp's technologies for its smart door locker which is remotely controlled by smartphone without a physical key.

We had the possibility to test the setup procedure with the developer kit suite. All the results are presented in the next subsections.

4.1.1 Setup and Keys Exchange

The setup procedure begins with the creation of a new personal account on the official website, followed by the installation of the smartphone app. The app requires to insert the account credentials and the details of the chosen WiFi network. It only accepts WiFi network with a single authentication key (WPA and WPA2 Personal), avoiding credentials-based access. At this point we just need to send all the information stored in the smartphone to the imp WiFi module. In order to do that, Electric Imp provides BlinkUp. Once the imp WiFi has obtained the information, it can automatically contact the router to establish the connection.

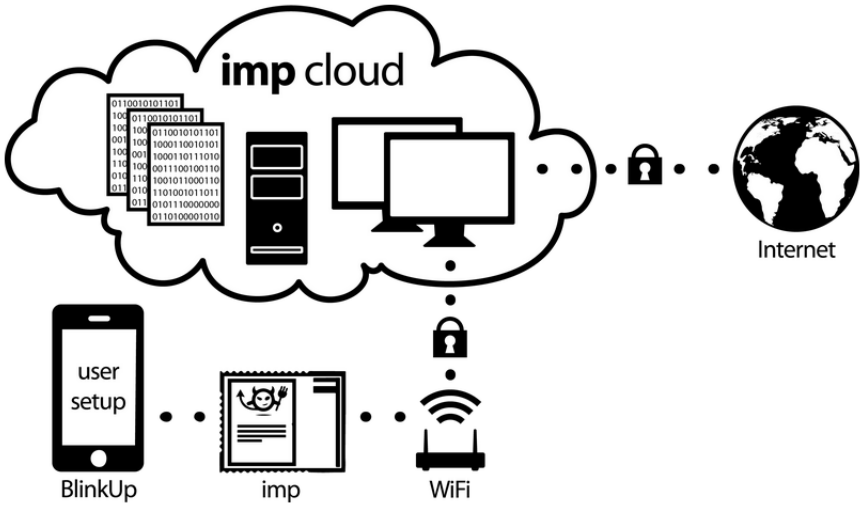

Figure 4.1: Electric imp - taken from [Imp]

BlinkUp is an innovative way that allows devices like smartphones or tablets to send information using the display light as an OOB channel. In other words, we just need to hold the sender device in front of a light sensor and in a few seconds the light blinking of the display can be transmitted and decode by the receiver.

The critical point of the entire procedure is the BlinkUp phase where WiFi key and credentials are exchanged between the parties. We can consider it enough secure since the difficulty to eavesdrop the light. It works in a very short working range which also ensure authentication. Only the users with physical access to the target device can configure it, excluding MITM risks.

4.1.2 User-Friendliness

During the test conducted the account creation, app installation and first configuration take around 10 minutes and they are simple to complete for a normal user. However, the BlinkUp step required four attempts to achieve the goal. In particular, it was necessary to switch off the room's light in order to make the imp sensor able to work.

In the end, although it is secure, the setup process using the light is not completely user-friendly in particular for people non familiar with electronic devices. It is also not recommended for scenario where an large amount of devices need to be installed.

4.2 Amazon Dash Button

Dash Button is the name of a new device and service from Amazon. It consists on a smart solution that provide an easy way for the user to reorder some commercial products available on Amazon online store. When the supplies of a product are going out, the user just presses the button of the specific device to buy and receive it at home in a few days.

Amazon Dash Button is a tiny device that uses the WiFi technology to take part of the Internet. It uses the interior sound sensor for the bootstrapping and commissioning phases which allow it to contact the home WiFi router. It also presents a sticky tape useful to place it wherever the user wants.

Figure 4.2: Amazon Dash Button - taken from [But]

Unfortunately we could not test the devices since no developer suite were available. All the information reported in the following are taken from the official website [But].

4.2.1 Setup and Key Exchange

The bootstrapping phase is similar to Electric Imp, it starts with an Amazon account creation and the download of a specific mobile app. They difference on the OOB method to send the required information. Dash Button uses ultrasonic tones as the OOB source produced by the smartphone speakers. Firstly, the user has to select the home WiFi network in the smartphone app and insert the linked passkey. Then the smartphone, placed close enough to the Dash Button, sends all the information using the ultrasonic tones. The Dash Button can now contact the WiFi router and create the connection. As final step the user can select the preferred product.

The procedure described is only valid for Apple phones. Amazon only provides a WiFi link between the smartphone and the Dash Button for Android phones since no OOB audio channel is supported.

Compare to the light used as OOB channel for Electric Imp even the sound does not presents risks of passive eavesdropping and MITM attacks. The working range is too short to allow malicious users to act on the devices. Therefore, we can consider the solution to be secure.

4.2.2 User-Friendliness

Same consideration can be exposed for the user-friendly level. The usage of the sound seems easy to use but it strongly depends on the environment conditions. Noisily places can disturb the ultrasonic tones and make the solution unfeasible. It is also not recommended for scenario where a large number of devices need to be installed.

4.3 Mother

"Mother" is a home device developed by Sen.se. It enriches everyday objects with additional functionalities. With accelerometers in tiny devices (called "cookies") it can detect, for instance, if a tooth brush has been used. The system can then give credits for good behavior, like regularly brushing your teeth.

The working kit consists of two parts.

- **Mother Base** is the main indoor gateway in charge of routing the messages. It is wired connected to the home router in order to take part of the Internet.
- **Motion Cookies** are small devices which can be affixed to almost anything. They include 3D accelerometer and thermometer sensors which allow smart features like analyze the movements of objects and people, control their presence in a chosen area or measure the surrounding temperature.

This implementation, different to the previous, does not use WiFi communication technology or other standard protocols like Bluetooth. The Cookies communicate with the Mother Base through a private radio signal at 868 Mhz.

Even for this system, we did not have the possibility to test it and all the information reported in the following are taken from the official website [Mot].

4.3.1 Setup and Key Exchange

Ses.se provides for its products a web-based initialization by using a personal cloud service. The first step is to create an account and add the Mother Base to it. After that it is possible to add as many Cookies as wished, making sure to place the device close enough to the Mother Base station. It is important to notice that, unlike other

Figure 4.3: Mother Base and Motion Cookies - taken from [Mot]

IoT products, the gateway is the initiator of the bootstrapping phase. Controlled by the web service, Mother Base contacts the closer Cookie recognized by the wireless signal strength and it starts the pairing process.

We can consider the security level inadequate during the bootstrapping phase. Since the website does not report details about the keys exchange protocols used, we are not in a position to know if the system is subjected to passive eavesdropping. But we can assume the risk of MITM attacks due to absence of an authentication phase. The Cookie's authentication using the only signal strength is not sufficient. A malicious user may use an alternative device working at the same wireless frequency but at an higher power level. Therefore, the gateway could identify the malicious device as the closest one.

4.3.2 User-Friendliness

As we can read on the official website, the web cloud and mobile application offer guided steps to complete the entire bootstrapping and commissioning. The user does not need to do particular actions directly to the devices. Therefore, the user-friendly level is based on the quality of the graphical user interface offered. Overall, the system presents a good level of user-friendliness.

4.4 Summary

After the analysis of three IoT systems for smart home, we can expose some conclusions. First, it is good to notice that all the devices developed to enrich the

physical objects with smart functionalities do not provide IO interfaces. Keyboards and displays would increase the cost and the dimension of the things.

Solutions that use an OOB channel for the bootstrapping phase are generally more secure. Electric Imp with the light and Amazon Dash Button with the sound exchange the authentication and cryptographic keys in a low range distance. It gets hard to eavesdrop the content. The low range also offers a secure authentication method which avoid the risk of MITM attacks. However, the main drawback of OOB channels is the more complex work for the final user who need to face the specific OOB implementation. It also depends on the environment conditions. For instance, a very bright or noisy room can make the systems unfeasible or really hard to setup.

OOB is also the only way to turn the target device as the initiator of the communication. Due to the absence of IO interfaces for the target, the only gateway can start the bootstrapping phase. In this term, we can imagine the OOB channel as the IO extension for the target. In fact, Electric Imp and Amazon Dash Button make the target devices able to contact the gateway and build the network. Mother, instead, does not offer OOB features and only the gateway (Mother Base) can be the initiator of the setup procedure.

The lack of security in Mother, due to the absence of OOB channel during the pairing process, is balanced by an high level of user-friendliness. An initiator gateway, programmed with easy to use graphical interfaces, is an effective way to keep the system usable by several types of user.

Table 4.1: Summary

	OOB	Initiator	Security	User-Friendliness
Electric Imp	Yes	Target	+	-
Dash Button	Yes	Target	+	-
Mother	No	Gateway	-	+

Table 4.1 exposes a summary and can be used as a good starting point for future analysis. In fact, the challenge is to find a good solution that mixes the security benefits of OOB channels with the easy to use steps offered by an initiator gateway.

Chapter 5
Proposed Scenario

After the discussion about the IoT systems for smart home (presented in Chapter 4) the first step for an improvement analysis was to define a relevant scenario for the next generation of products. Due to its large impact on everyday life, the idea was to keep the focus on the smart home category. It requires low/medium range of wireless connections which is enough to cover one or few more floors.

We chose BLE as primary communication technology since it is ideal for WPAN. It offers a quite good throughput combined with a very low power consumption, extending the battery life of the connected devices. Another feature supporting this choice is its large spread over the common devices like smartphones, tablets and laptops. This spread extends the spectrum of the possibilities. For instance, a common smartphone with BLE chip could be used by the user during the setup procedure or to send some private information.

Since we are focusing on IoT platforms, each device needs an Internet connection. This is easy to obtain with classic WiFi modules, as we saw for Electric Imp and Amazon Dash Button. However, this is not directly achievable on other technologies like BLE or ZigBee. Historically, the first solution to connect a low-power device to the Internet was to build a properly gateway in charge of handle and route the information. This solution works at application level. It functions well with small systems but it needs time and efforts in order to develop it. That is why researchers defined 6LoWPAN adaption protocol 2.3. It allows some low-power devices which not use WiFi to obtain an IPv6 address and take part of the Internet. In this way, the low-power device can use all the benefits of the TCP/IP protocols stack. It creates an end-to-end connection between nodes without the needed of a complex application gateway, favoring open standards and increasing the interoperability. 6LoWPAN is still fresh but ready to be used. Our scenario relies on the usage of this novel technology.

Obviously, the usage of BLE combined with 6LoWPAN requires a review of the

router. We cannot use the common WiFi router but we need a new platform enabled with the technologies required. Figure 5.1 presents a summary scheme of our working scenario. A *router/gateway* is placed in a fixed position and wired connected to the Internet. It includes BLE hardware as well as software support for 6LoWPAN. On the other side, several *target devices* (the specific IoT products) are paired to the gateway and able to take part of the Internet.

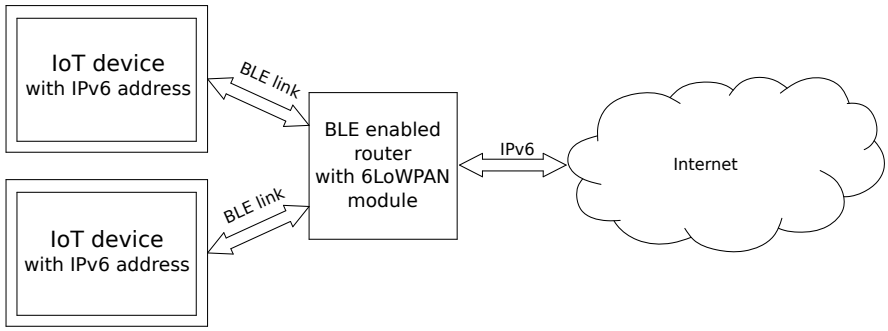

Figure 5.1: IoT - adapted from [Sema]

According to the idea to introduce smart devices in the everyday objects, they have to consider the constrains like low cost, low power and tiny dimension. Low-cost in order to not increase the price of the products, low-power not to increase the energy consumption and tiny dimension not to increase the volume. At the current state, these constraints converge in computational limited devices without IO interfaces. As we can expect, CPU and memory limits introduce doubts on their usage in combination with protocols like TLS or heavy asymmetric cryptography algorithms. Moreover, the absence of input/output interfaces introduces new challenges in term of security. For instance, classical authentication methods with passkey insertion cannot be used anymore in solutions with no physical interfaces like keyboards and displays.

Security The working scenario proposed refers to a IoT system. In such context, it is important to provide an high level of security and privacy. Typical implementations manage confidential information that the user wants to keep secret. Mother Sen.se, presented in Chapter 4, is an example of it. That is why security is a requirement in this work. At first we aim to ensure *confidentiality* which avoid passive eavesdropping of the messages. Then an adequate *authentication* step is required to avoid MITM attacks from malicious users.

User-Friendliness Beside to security, another requirement is the *user-friendliness*. Since the physical objects are used by every every kind of users, the introduction of

smart features should not exclude people not friendly with electronic devices and new technologies. The aim is to keep them easy to setup and use.

Goals According to the scenario proposed and the requirements to achieve, the focus of the next analysis is to study how to build the secure network of BLE constrained devices, operating on the Internet through 6LoWPAN. The critical aspect of the analysis is the method to exchange the cryptographic keys used to encrypt the channel and how to provide authentication for the user.

Chapter 6
Secure and User-Friendly Solutions

This chapter starts from the proposed scenario described in Chapter 5 to present some solutions for the commissioning and bootstrapping of constrained devices. As already said, the aims of the work are security and user-friendliness. With the former, we want to achieve a pairing without the risks of passive eavesdropping of the messages and MITM attacks. In other terms, we expect to securely exchange the cryptographic keys, used to maintain confidentiality and integrity in the channel, and offer authentication between the parties. With the latter, we want to keep the process as simple as possible, allowing a large amount of users to use such devices.

A lack in the initial keys exchange could compromise the entire communication, allowing malicious users to authenticate on behalf of victim's identity or to read/modify the messages content. This is considered a critical point of the current generation of IoT systems which justify the particular focus on the keys exchange.

We decided to split the analysis in two blocks. The first refers the case of movable target devices which the user can place without efforts. The second refers to the case of fixed target devices which are difficult or impossible to move. This requirement increases the security risks and the steps to complete the bootstrap.

In both scenarios, we first described the goal to reach, defining the steps and the user-friendly level. Then we discovered some practical solutions to implement them with particular regard for security. In other words, we fixed the user-friendly requirement and we tried to obtain some related solutions that also fit the confidentiality and authentication requirements.

In order to do that, we combined the results collected in the use cases analysis (presented in Chapter 4) with our working scenarios. Every use case spends particular effort in the bootstrap of the devices, proposing several methods to exchange the required information and keys. For instance, Electric Imp and Amazon Dash use an OOB channel to exchange information like the WiFi keys that the devices required

in the setup phase.

At first, in the following we present a review of the pairing methods offered by BLE, explaining the choice to use an OOB channel as unique authentication method. Then we list several OOB channels for BLE, focusing on NFC. In the end, we describe the solutions proposed. Section 6.3 studies the scenarios with movable target. Section 6.4, instead, analyze the case of fixed target devices.

6.1 BLE Pairing Overview

BLE Core Specification 4.2 Secure Connection offers several pairing methods. For a complete introduction of them and the BLE pairing algorithms please refer to Section 2.1. For our scenario, we can use only the *OOB* method. The other methods do not fit the security or user-friendliness requirement we fixed.

Just Works method does not offer authentication between the parties but only a simple way to make them on work. It is valid for products without security concerns but not for future IoT systems which could handle confidential informations about the users. *Passkey* method, instead, respects the security requirements and it is large used in Bluetooth applications. However, we consider it not enough user-friendly. Considering the current technologies available, a passkey manual insertion looks old and stressful for the users. We should avoid it in the next generation of products. Moreover, it is difficult to implement for constrained devices without IO interfaces, required in order to show or insert the passkey. *Numeric Comparison* method presents the same drawbacks of *Passkey*. It requires IO interfaces to compare and accept a common shared number. Although it sounds easy, the action of checking two displays is often unfeasible or uncomfortable for the users, therefore we discard also this solution.

As already said, BLE *OOB* pairing method is the only attractive way for our scenario. Similar to the others, it ensures protection versus passive eavesdropping by using the ECDH key agreement. An external user cannot derive the secret shared between two parties, maintaining confidentiality and integrity in the channel. Therefore one of the security requirement is considered to be met.

However, BLE delegates the authentication phase to the specific OOB channel chosen. We can consider the BLE interaction protected from MITM attack only if the specific OOB channel employed for the authentication is free of it. The challenge for the next part of the thesis is to find a secure OOB technology without the risk of MITM.

It is important to remember that the specific OOB channel can start the BLE pairing process. It is a great advantage for our scenario since we avoided IO interfaces

and typical bootstrap steps are not possible. We can see the OOB channel as an interface extension useful to start the communication. For a complete overview please read Chapter 2.

6.2 Out-Of-Band Channels

OOB refers to a communication technology out of the main channel that can be used to exchange confidential or specific information. The first advantage of OOB is in term of security since it hides private messages from the main transport link which increase the protection versus eavesdroppers. A potential attacker should know the transport mechanism used and get the equipment needed. It also offers an alternative way to send information, increasing the flexibility of the systems in case of problems or failures.

Assuming BLE as the primary channel for our scenario, we need a secondary link for the authentication phase. Several kinds of OOB technologies can be presented (the list is not exhaustive):

- Wired connections
- Radio frequency technologies (different from BLE)
- Visible light
- Sound waves
- Movements and vibrations

In Chapter 4 we discussed some of them, used in commercial IoT products. Electric Imp provides light sensors to transmit authentication keys and other information through the blinking light of a display. Amazon Dash Button, instead, does the same with sound waves produced by the speakers. In both solutions the main channel is a WiFi connection. As already said in the end of the cited chapter, these OOB channels cannot be considered user-friendly since they depend on the particular working environment. Noisily or bright rooms could affect the normal use, making them unusable. Same arguments for movements and vibrations, strongly related to the ambient conditions and the user's capabilities. Although wired connections are secure and stable, they are an old solution which force the user to use cables no more needed. Therefore, it does not represent an elegant solution at the current technologies level.

A different radio frequency channel seems the best choice for our purpose. Without noise, it is independent from the working environment and it does not require

particular action by the users. The main issues for wireless technologies are in terms of security since the information travel in the air. Therefore, our challenge is to find a wireless link free of risks that can be combined with BLE.

6.2.1 NFC

The choice pointed to NFC. A background on NFC is presented on Section 2.2. It is within the wireless technologies category one of the easier to use solution. The only requirement to make these devices work is to put them close enough to each other at the distance of typically 4 cm. It is also low-cost and low-power, ideal for the IoT systems we want to obtain.

Another significant advantage is that it fits well with the security requirements. According to [HB06], MITM attack is infeasible in NFC channel and issues like data corruption, data modification and data insertion can be easily avoided by several techniques. Passive eavesdropping is the only risk for the data exchanged in NFC. It is important to say that a NFC eavesdrop is difficult to obtain and they depend on several factors, like the equipment available by the attacker. However, to avoid any risks, it is possible to prevent it by the establishment of a NFC secure connection using key agreement protocol like Diffie-Hellman or ECDH.

In the following we assume NFC as the OOB channel for our BLE scenario. Its simplicity and the protection against MITM (the only security requirement remained open by BLE OOB pairing method) make it a good partner for Bluetooth devices for future implementations.

6.3 Movable Target Devices Scenario

The current scenario is relevant for target devices with limited dimensions and easy to move that the users can place without particular efforts. Light bulbs, alarm clocks and pills boxes are all examples of movable objects. We want to connect them to the Internet using BLE network and NFC as a support channel.

We assume the devices (target and gateway) are working in *listening mode*, ready to start the pairing process. This can be reached by a previous activation phase on both devices, like pressing a button or using a graphical user interface. We suggest an activation in order to limit the pairing time and increase the control of the system. However, this part is out of the scope of this work.

First we described how we would reach the commissioning and bootstrapping of a movable device. In this way, we set the user-friendliness level for future generation of products. We started considering a new product for the user who has no previous knowledge of it. The challenge is to put it on work easily, with few steps and

avoiding manual insertions of keys. Figure 6.1 describes the process we want to obtain, considering the case of a smart bulb with BLE and NFC chips.

Figure 6.1: Movable Target Device: [A] The user buys a new IoT product. [B] The user places the object close to the gateway for a few seconds to set up and authenticate the system. [C] The object is connected to the Internet and ready to work.

We consider this approach the best standard for future systems with movable targets. It lowers the stress and the knowledge required for the final users. Starting with this description, in the following we present a way to reach the goal.

6.3.1 Solution: BLE Out-of-Band Pairing Method

The proposed solution for the movable target scenario consists of a simple BLE OOB pairing method, where NFC is the specific OOB channel. As reported in BLE 4.2 Core Specification, we can use the specific OOB channel to start the BLE pairing process [PMoBS14]. This is ideal for our working scenario since we do not consider IO interfaces. The information exchanged through NFC are essential to the subsequent BLE connection. They are exchanged in the secondary channel due to their confidentiality. An eavesdrop of them could compromise the entire communication. In addition, the unique NFC interaction even includes the authentication phase.

The idea is to hold the target device close to the gateway in order to make the NFC chips (working at *listening mode*) able to communicate. A detailed description of the messages exchanged by the OOB channel is presented in Subsection 2.1.2. Figure 6.2 shows the communication diagram of the solution.

The solution does not require any support hardware. The target device, enabled with BLE 4.2 and NFC chips, includes all the features to start a communication with

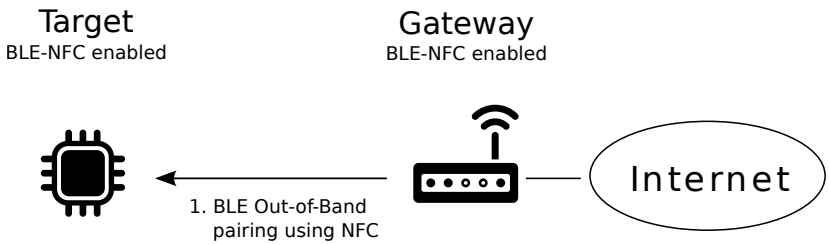

Figure 6.2: Communication diagram for movable target devices: [1] The gateway is the initiator of the communication. It uses NFC to exchanged the confidentially information needed for the subsequent BLE Out-of-Band pairing

the gateway.

BLE Roles Following the direction of the arrow, the gateway is the initiator of the connection and it represents the *Master* role in the BLE star network. The target, instead, responds to the gateway and it represents the BLE *Slave*.

Security Analysis BLE 4.2 Secure Connection offers a good security level. It provides ECDH key agreement in order to establish a shared secret between the parties. In this way, it prevents passive eavesdropping, ensuring the confidentiality/integrity requirement of our work. Active eavesdropping (MITM) is the only risk to consider in the BLE pairing process, depending on the authentication method chosen. Since we decided to use the OOB pairing method, MITM protection is delegated to the specific OOB channel.

NFC is out OOB channel. It is used to start the pairing and it is in charge to exchange the confidentially information. As presented in Subsection 6.2, NFC fits the security requirements since it is free of MITM attack [HB06] and offers an authentication step based on low-range contact.

Considerations We can consider the presented solution secure, offering an adequate level of authentication for the users and confidentiality/integrity for the messages exchanged in the link.

6.4 Fixed Target Devices Scenario

Even though the solution for movable devices above is secure and easy to implement, it represents a limited scenario. Considering the smart home environment, it is clear that in most of the case the target devices present large dimensions or they require a fixed location. Washing machines, fridges or doors are all examples. This make the

previous solution unfeasible. In this case we need a *support device* like smartphone or tablet to reach the goal.

Even in this case we assume the devices are working in *listening mode* and they are ready for the pairing process. We do not cover the activation phase.

Again, we described the user-friendliness level to obtain for a new target device. As primary step, we would connect the support device to the gateway without any manual insertion of credentials. In this way, we can figure the smartphone as a gateway's range extension. This step should be required once, independently from the number of target devices to bootstrap. After that, we would use the support device in the same way to allow the fixed target to contact the gateway. Figure 6.3 describes the process we want to obtain, considering the case of a smart washing machine with BLE and NFC chips.

We consider this approach a good standard, in terms of user-friendliness level, for future products with fixed target devices. Even though the distance between the gateway and the target increases the difficulty, the easy steps described above should not limit the number of users able to make the system on work. In the following we present two practical solutions to reach the goal. As support device we consider a smartphone with BLE and NFC chips.

6.4.1 First Solution: BLE Tunnel

A simple way to connect the target to the gateway is to establish a secure wireless tunnel between them, using the smartphone as an intermediary. The smartphone is a non-limited device able to provide routing features. The idea is to create the tunnel through a double BLE connection, a single *gateway/smartphone* and multiple *smartphone/target* connections. Using this BLE tunnel as an OOB channel, gateway and target can start a new secure connection between them. Figure 6.4 shows the BLE tunnel.

In details, at first we can use the BLE OOB pairing method between smartphone and gateway, using NFC as the secondary link. Once connected, the smartphone can act like the gateway's range extension without any risks of eavesdropping. Their link is protected by Bluetooth symmetric cryptography (AES-CCM). This step is required once. From now on, we can pair the smartphone to any target devices using the BLE OOB pairing method in the same way. Gateway and target are now linked with a complete BLE tunnel. This tunnel can act like a new OOB channel. We can use it securely exchange all the required information to start a direct link between target and gateway. Figure 6.5 shows the communication diagram for the solution.

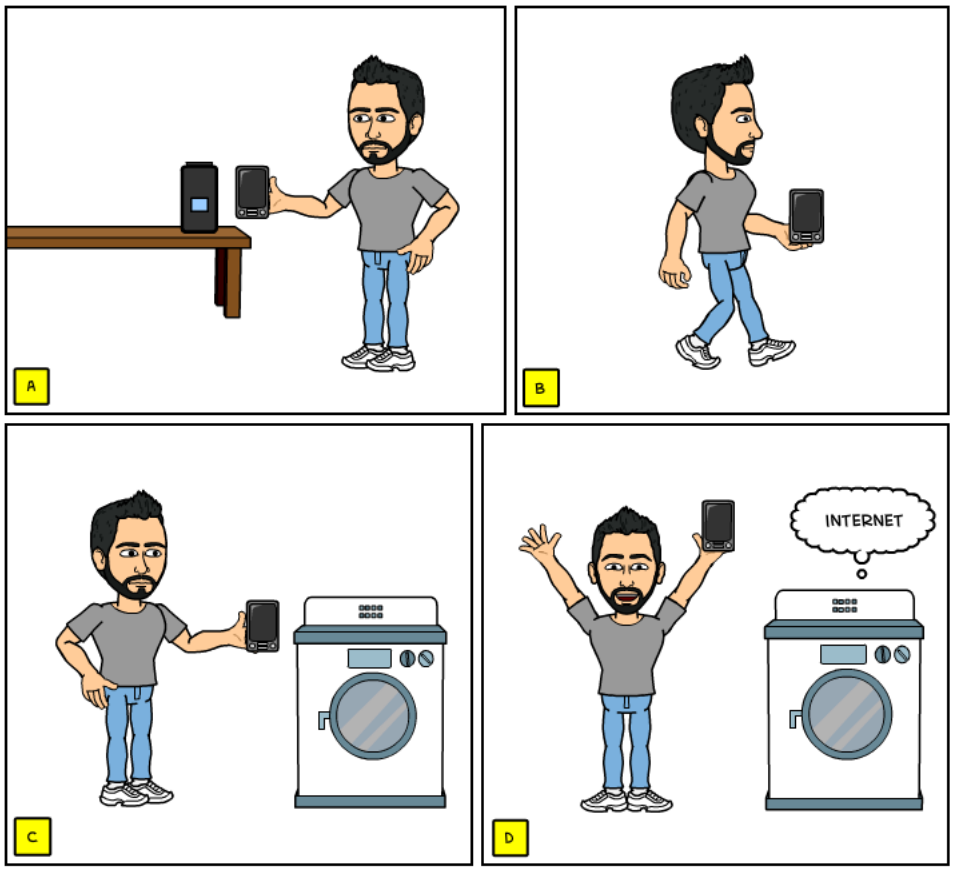

Figure 6.3: Fixed Target Device: [A] The user places the support device close to the gateway in order to securely pair them. The smartphone can be considered as a gateway's range extension. [B] The user can now reach the location of the fixed target device. [C] The user holds the smartphone close to the fixed target in order to exchange the needed information for a direct contact between target device and gateway. [D] The object is connected to the Internet and ready to work.

BLE Roles Considering Figure 6.5, in connection [1] the smartphone is the initiator and it represents the *Master* of the BLE connection. The gateway is the *Slave*. It is the same in connection [2], where the support device is still the *Master*. In the final connection [3], the gateway has to be the *Master*, in order to connect more *Slaves*. Figure 6.6 shows the BLE roles.

Even if the presented solution seems easy to obtain, we can affirm that it is unfeasible at the current BLE Core Specification 4.2. BLE supports only star network topology, where a single *Master* device can be connected to several *Slaves*.

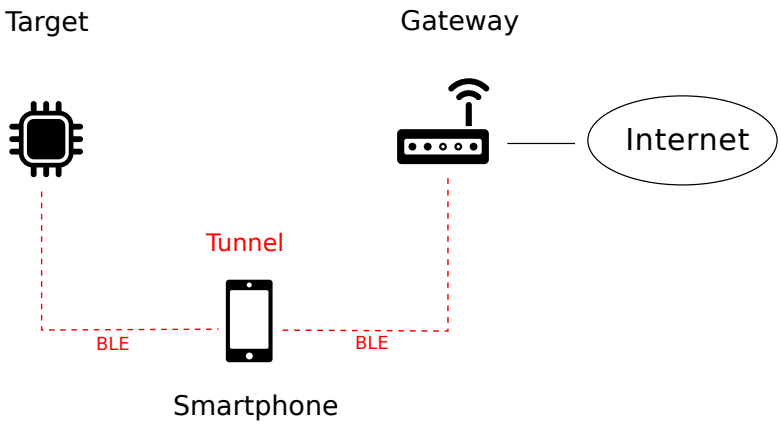

Figure 6.4: BLE tunnel

This creates two technical problems in our scheme. At first, a device cannot be Master and Slave at the same time. It happens for the gateway connected to the smartphone as *Slave* and to the target as *Master*. Secondarily, a device cannot be a Slave of more then one Master. This happens for the target device connected to the gateway and the smartphone as a *Slave*.

In the end, the solution proposed cannot be used at the moment. It could become relevant in the near future since Bluetooth SIG is working to add mesh network features into BLE [EET].

Security Analysis Also in this solution we want to present a security analysis. The entire process relies on the BLE OOB pairing method three times. The first two using NFC as OOB link. We already described, in movable target solution, the absence of risks on it. User authentication is provided by NFC physical contact between the devices, avoiding MITM attack from malicious users. The last connection, instead, uses a complete BLE tunnel, where all the information exchanged are encrypted by AES. The smartphone, working as the intermediary of the link, can ensure the authentication phase. It is physically controlled by the user.

However, this solution add a critical point in the connection chain. The smartphone, in fact, represents the central node in charge of routing the information and authenticate the parties. A malicious user could hack the smartphone or steal it to have physical access. In this case, confidentiality and integrity of the messages is not guarantee. We do not cover the security features of the smartphone in the following.

Considerations As already said, the solution is practically unfeasible due to the lack of support for star network topology by BLE. It represents a limit for Bluetooth

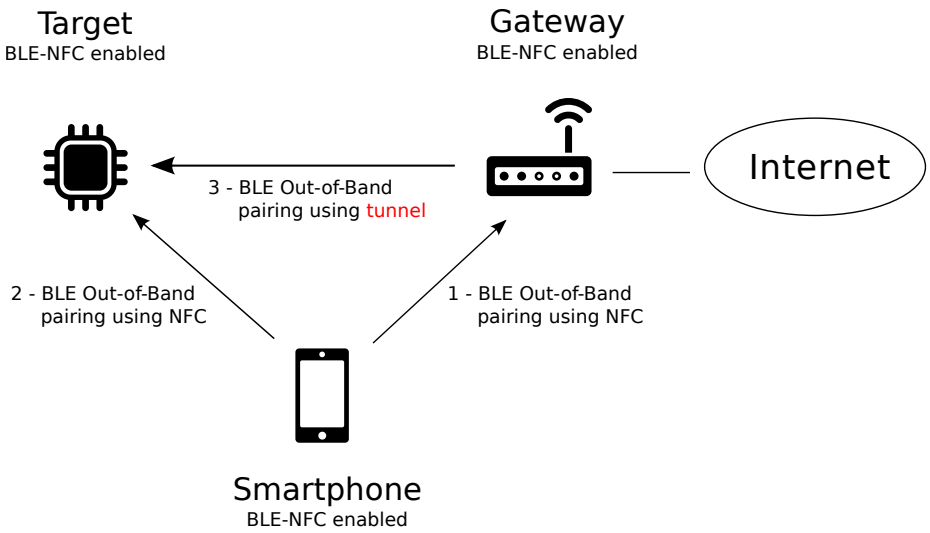

Figure 6.5: Communication diagram for fixed target device - BLE Tunnel: [1] The user can pair the support device with the gateway, using NFC as the starting point for the BLE OOB pairing method. It is required once. [2] The user can pair the support device to any target device available in the same way. [3] Once the tunnel is built, the user can start a new BLE OOB pairing method between target and gateway. The tunnel acts like the OOB channel and the smartphone is the controller.

technology. It may be overcome in next Core Specification by the introduction of mesh network support. It will make the presented solution achievable.

Such solution also presents another issue. Each target device has to perform the heavy ECDH key exchange protocol twice, one during the pairing with the smartphone and again during the subsequent connection with the gateway. Current generation of constrained devices could not be ready for such computational requirement, doubling the pairing time.

For these reasons we looked for another solution in order to fix the previous problems. We present it in the following.

6.4.2 Second Solution: BLE and NFC Tunnel

To solve the problems in the previous solution, we need to change one BLE connection with another wireless link. We chose a simple NFC communication as the best solution. It does not require additional hardware and it offers an easy to use channel. In this case, the tunnel between gateway and target device is formed by the combination of BLE and NFC. The support device is still the intermediary in charge to route the

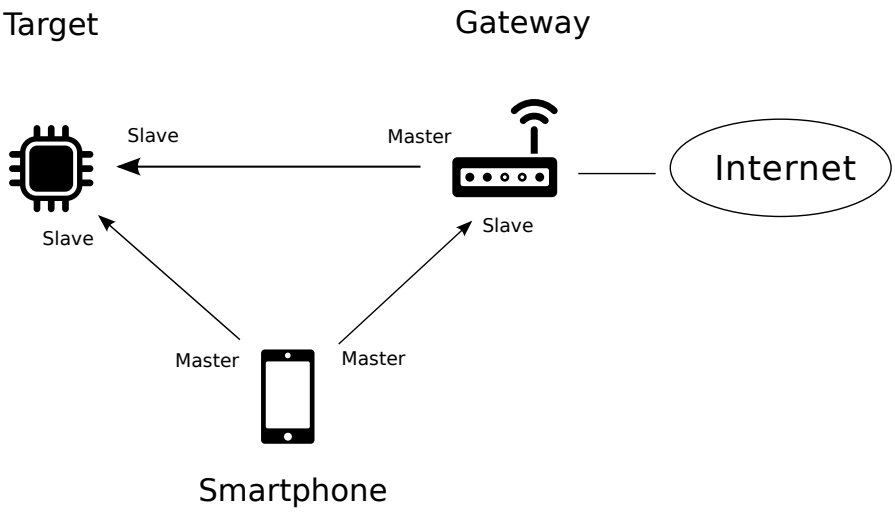

Figure 6.6: Fixed solution with BLE tunnel - BLE Roles

information. Figure 6.7 shows the tunnel.

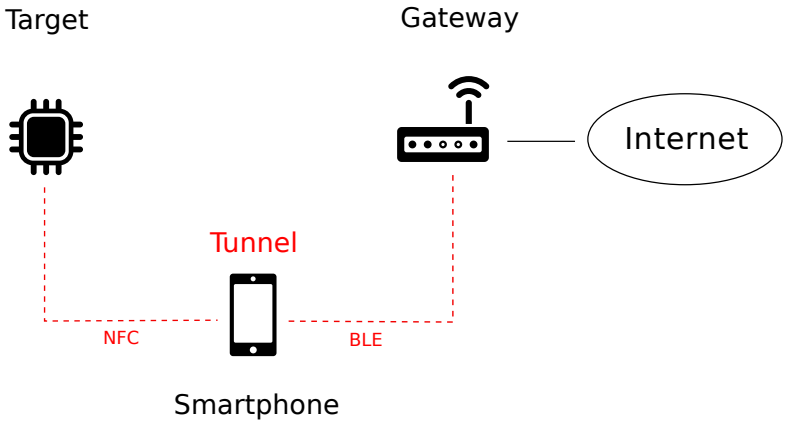

Figure 6.7: BLE-NFC tunnel

In details, Figure 6.8 describes the communication diagram for this solution.

BLE Roles In this solution, there are only two kind of BLE pairings, one between *gateway/smartphone* and the other one between *gateway/target*. In both interactions the gateway is can act like the initiator of the connection and it represents the BLE *Master*. Figure 6.9 show the BLE roles.

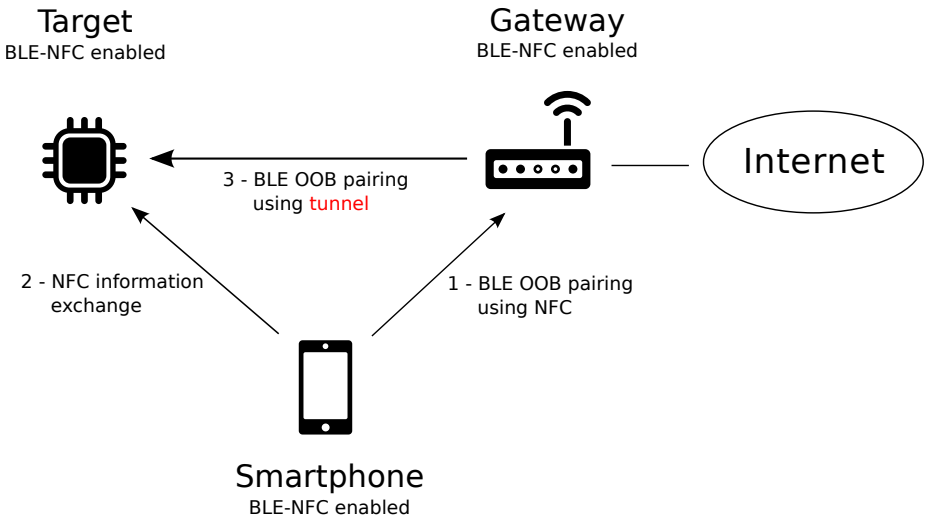

Figure 6.8: Communication diagram for fixed target device - BLE Tunnel: [1] The user can pair the support device with the gateway, using NFC as the starting point for the BLE OOB pairing method. It is required once. [2] The user can place the support device close to the target device to start a NFC interaction. In this communication the smartphone gets all the information to establish a BLE OOB connection with the target. [3] Once the tunnel is built, the user can start a new BLE OOB pairing method between target and gateway. The tunnel acts like the OOB channel and the smartphone is the controller.

The solution presents a star topology, where the gateway (*Master*) is linked with a smartphone and several target devices (*Slaves*). Therefore, the proposed scheme is practically feasible.

Security Analysis The security analysis follows the same arguments exposed in the previous solution. The substitution of the BLE connection with an NFC communication between smartphone and target device is the only difference. It does not add significant issues since we ensured the security level of the NFC link.

Considerations This solution represents out best option for future implementations with fixed devices. It fits all the security requirements of this work and reach an adequate user-friendly level.

Compared to the previous solution, it fixes all the problems encountered. At first, it is implementable by the current communication technology since it does not need a mesh network. At the moment, BLE Core Standard 4.2 supports only star network topology. Then, it increases the efficiency. The target device is no more

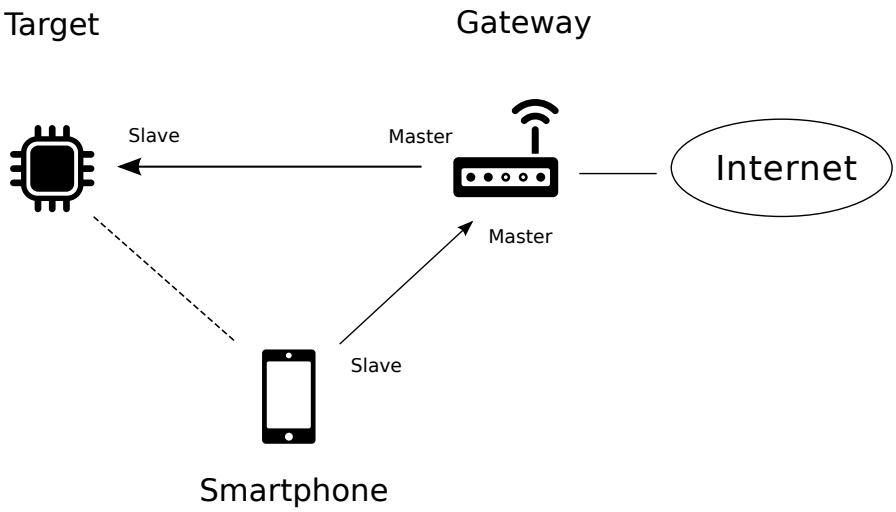

Figure 6.9: Fixed solution with BLE-NFC tunnel - BLE Roles

connected with the smartphone through a BLE secure channel. This releases the target device from an ECDH process, considered the heavier computational work in the chain. The target compute an ECDH process only during the pairing with the gateway, reducing the global time of bootstrapping.

Chapter 7
Discussion

In the previous chapter we presented some solutions for the commissioning and bootstrapping problem of constrained devices. In the solutions we presented several technologies but we have not discussed the actual support to them. In this chapter we want to complete the study analyzing the current products commercially available required to implement the solutions. In other terms, we want to show the feasibility of our conclusions.

As already exposed, in our scenarios we require three kinds of products: gateway, target devices and smartphone (as support device in fixed target device scenario). We present a brief introduction to all of them.

Target Devices The target device is any physical object expanded with smart features and connected to the Internet. It represents the specific IoT product. For our purpose, it has to include BLE and NFC hardware support.

We could combine two different chips to obtain the final device, one for each wireless technology. However, in order to lower the dimension and the price, we focused on System on a Chip (SoC) which include all the hardware needed. In particular we looked for BLE and NFC support but we considered also hardware accelerator to speed up the cryptography operations. After a research we selected two SoC for our scenario:

- **Nordic nRF52**: from Nordic Semiconductor, it was released in May 2015 in two version: the SoC and the Development Kit. It was specifically designed to support the spread of IoT systems. Due to recent release, only few information are available and the support for NFC is limited. Figure 7.1 shows the products.
- **Broadcom BCM20737**: it was released in March 2015. It presents similar features to Nordic nRF52 but it also introduce specific hardware support for security.

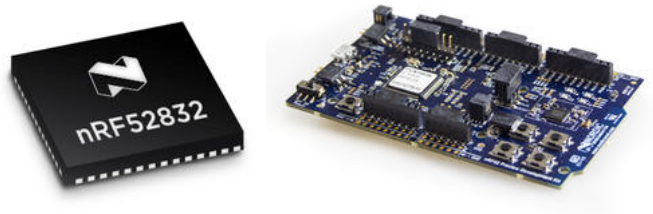

Figure 7.1: On the left the Nordic's SoC nRF52. On the right the Development Kit version - taken from [Semb]

Gateway The gateway does not require a specific product. It could be implemented by a simple router enabled with BLE and NFC chips. In addition it has to support 6LoWPAN protocol. Any router that fulfills these requirements can be used for our scenario. For example, Nordic Semiconductor suggests to use a Linux computer that includes bluetooth-6lowpan module in the Kernel [Sema]. bluetooth-6lowpan module was introduced in Linux Kernel v3.17.

Smartphone To complete the overview over the products required we present a list of the most used smartphone over the world. Even in this case, we want to show the wireless technologies support for our purpose.

Here a list of smartphones analyzed:

- Apple iPhone 6
- Nexus 6
- Samsung Galaxy S6

All of them provide BLE chip. The version supported differs within v4.0 and v4.1. In our solutions we considered the v4.2 since it introduces several improvements for security. It is too fresh for the actual generation of smartphone but probably it will be added in the next generation.

Also for NFC, all the smartphone listed support it. However, Apple iPhone 6 limits the usage to some applications but we suppose it will be more open in the near future.

Conclusion After this introduction of the devices available we can affirm that all the solutions presented in Chapter 6 are suitable. The gateway does not represent a technical issue. The most used smartphone are ready to act as a support device

required in fixed target devices scenario. The SoC for the target devices that combine BLE and NFC are available and low-cost.

As final consideration, the recent release of Nordic's and Broadcom's SoC indicate that we moved in the right direction. Our decision to combine BLE with NFC as secondary channel finds confirmation with the ideas of some of the most relevant companies producers of electronic devices.

Chapter 8
Conclusion

The starting point of the thesis was to analyze some IoT systems for smart home and discover drawbacks in term of security and usability. With the results and test obtained, the goal was to propose solutions for the commissioning and bootstrapping of constrained devices. Such solutions had to fit security requirements like authentication and confidentiality, as well as user-friendliness. The working scenario chosen includes target devices enabled with BLE 4.2 and able to communicate with a fixed router expanded with 6LoWPAN protocol.

We defined solutions for movable and fixed target devices. In both case, the idea was to combine BLE as the main communication technology with NFC as the OOB channel to provide authentication. BLE Core Specification 4.2 is the current standard and it offers a *Secure Connection* mode. It provides an high level of data security, ensuring confidentiality and integrity for the messages exchanged. It uses ECDH to exchanged the cryptographic keys and produce a shared secret in a insecure channel. The critical point of the procedure is the authentication phase. Here we introduced NFC. It offers an high level of security, avoiding risks of passive and active eavesdropping. In addition, it increases the user-friendliness level in the authentication phase for the final users, removing static passkey entry or other manual procedures.

For the movable target case, we propose a BLE 4.2 Secure Connection pairing started by an NFC interaction. From the user's point of view, it means placing the target device close enough to the gateway to start the Bluetooth pairing. Similar to the fixed target case, but it requires a support device like a smartphone. The smartphone, paired via BLE once with the gateway, is the intermediary device. It creates secure tunnels between gateway and target. The tunnel can be used to exchanged information for a subsequent connection of the parties without risks of eavesdrops and attacks.

Future IoT systems for smart home could implement the solutions proposed

in this work. For first, BLE with 6LoWPAN is an advance technology that allow tiny devices to take part of the Internet. Compare to WiFi products, it reduces cost and power consumption. It is a great advantage since these devices will be included in everyday life objects. In addition, the combination with NFC technology overcomes the traditional authentication methods, offering an easy to use way to securely bootstrap the network of things. The recent spread of such communication technologies in smartphone and tablet, as well as the market of new SoC combining BLE and NFC, confirm the validity of the results.

Future Work To expand the results obtained, I suggest some future works:

- Try to implement the solutions in order to discover possible technical issues and to analyze the performances of the current generation of devices.
- In the fixed target devices scenario we add the smartphone as an important requirement. This intermediary device introduces a critical point in the connection chain that we did not consider in this work. A study of the vulnerabilities and risks of the smartphone is necessary to have a better overview and avoid related risks.
- Move the focus to another low-power radio technology like ZigBee. The comparison between BLE and another low-power standard could be interesting to test and evaluate the real benefits of Bluetooth.

References

[Bea] Vangie Beal. Symmetric-key cryptography. URL http://www.webopedia.com/TERM/S/symmetric_key_cryptography.html [Accessed: 09/07/2015].

[But] Amazon Dash Button. Official website. URL http://www.amazon.com/b/?node=10667898011&lo=digital-text [Accessed: 15/06/2015].

[DH76] W. Diffie and M.E. Hellman. New Directions in Cryptography. *IEEE Transactions On Information Theory*, 1976.

[EET] EETimes. Mesh comes to bluetooth. URL http://www.eetimes.com/document.asp?doc_id=1325815 [Accessed: 02/10/2015].

[For] Near Field Communication Forum. Official website. URL http://nfc-forum.org/what-is-nfc/what-it-does/ [Accessed: 25/06/2015].

[Gro] OpenSSL Project Group. Elliptic curve cryptography. URL https://wiki.openssl.org/index.php/Elliptic_Curve_Cryptography [Accessed: 14/07/2015].

[HB06] Ernst Haselsteiner and Klemens Breitfuß. Security in Near Field Communication (NFC) Strengths and Weaknesses. *Semiconductors*, 11(71):71, 2006.

[Imp] Electric Imp. Official website. URL https://electricimp.com/ [Accessed: 15/06/2015].

[Ins] Texas Instruments. Overview for 6lowpan. URL http://www.ti.com/lsds/ti/wireless_connectivity/6lowpan/overview.page [Accessed: 25/06/2015].

[MC14] Adrian McEwen and Hakim Cassimally. *Designing the Internet of Things*. ISBN 978-1-118-43062-0. Wiley, March 2014.

[Mot] Mother. Official website. URL https://sen.se/store/mother/ [Accessed: 15/06/2015].

[Ols14] Jonas Olsson. 6lowpan demystified. *Texas Instruments*, October 2014.

[PMoBS14] Inc. Promoter Members of Bluetooth SIG. *Bluetooth Specification Version 4.2*, December 2014.

[SA00] Frank Stajano and Ross Anderson. The resurrecting duckling: Security issues for ad-hoc wireless networks. In Bruce Christianson, Bruno Crispo, JamesA. Malcolm, and Michael Roe, editors, *Security Protocols*, volume 1796 of *Lecture Notes in Computer Science*, pages 172–182. Springer Berlin Heidelberg, 2000.

[Sec03] CNSS Secretariat. National policy on the use of the advanced encryption standard (aes) to protect national security systems and national security information. June 2003.

[Sema] Nordic Semiconductor. nrf51 iot sdk. URL https://developer.nordicsemi.com/nRF51_IoT_SDK/doc/iot/html/index.html [Accessed: 15/07/2015].

[Semb] Nordic Semiconductor. nrf52 soc. URL https://www.nordicsemi.com/Products/nRF52-Series-SoC [Accessed: 02/10/2015].

[SOM+13] B. Sarikaya, Y. Ohba, R. Moskowitz, Z. Cao, and R. Cragie. Security Bootstrapping Solution for Resource-Constrained Devices. *Internet-Draft*, 2013.