SECURITY ANALYSIS AND ADVANCEMENT IN ZIGBEE COMMUNICATION

By

Ngonidzashe Gwata

Thesis submitted in partial fulfilment of the requirements for the degree

Master of Engineering: Electrical, Electronic and Computer Engineering

In the Faculty of Engineering

At the Cape Peninsula University of Technology

Supervisor: Prof Vipin Balyan

Bellville

February 2024

DECLARATION

I, Ngonidzashe Gwata, declare that the contents of this thesis represent my own unaided work and that the thesis has not previously been submitted for academic examination towards any qualification. Furthermore, it represents my own opinions and not necessarily those of the Cape Peninsula University of Technology.

Signed n.gwata   Date 29/02/2024

ABSTRACT

The rapid growth of the Internet of Things (IoT) has led to an increased demand for secure communication protocols. This has highlighted the importance of addressing vulnerabilities in protocols like Zigbee, one of the prominent communication protocols used in the IoT domain.
The purpose of this thesis is to analyse the security weaknesses of Zigbee and propose enhancements to mitigate these vulnerabilities.

This research was built upon many extensive studies conducted to examine the security of Zigbee, and it involved reviewing various literature sources that identified potential vulnerabilities in the Zigbee protocol. Additionally, different methods for exploiting these vulnerabilities were analysed to understand better how they can be addressed.

Zigbee is built on top of IEEE 804.1 and utilizes Advanced Encryption Standard (AES) encryption with a strong 128-bit key. This cipher has been thoroughly tested and proven secure. However, this research revealed that the weak point lies in key transportation and management within Zigbee. To address this issue, the proposed enhancements in this thesis focus on reinforcing security during key transportation by implementing Elliptic Curve Diffie-Hellman (ECDH) encryption during the network joining process. This approach aims to securely protect the network key against unauthorized access or manipulation.

The proposed solution is evaluated in terms of its computational, energy, and communication overhead. The results demonstrate that the suggested approach brings about minimal additional demand. This research contributes to IoT security by offering an enhanced approach to safeguarding Zigbee networks. This ultimately reinforces the safety measures for both IoT devices and secure data transmission.

ACKNOWLEDGEMENTS

I wish to thank:

- Doctor Balyan Vipin. I am deeply grateful to my supervisor, for his guidance and insightful input during the study.
- My family and friends. I've remained positive and motivated throughout this process because of their confidence in me.

DEDICATION

To Lemuel, Carmel, and Eden

Table of contents

LIST OF FIGURES

LIST OF TABLES

ABBREVIATIONS AND ACRONYMS

AES	Advanced Encryption Standard
APS	Application Support Sub-Layer
CCM	Cipher Block Chaining - Message Authentication Code
CSA	Connectivity Standards Alliance
CSMA-CA	Carrier Sense Multiple Access with Collision Avoidance
APL	Application Layer
DDoS	Distributed Denial of Service
DES	Data Encryption Standard
DoS	Denial of Service
ECC	Elliptic-Curve Cryptography
ECDH	Elliptic-curve Diffie-Hellman
FC	Frame Counter
ID	Identifier
IDE	Integrated Development Environment
IOT	Internet of Things
LoWPAN	Low-Power Personal Area Network
LPWAN	Low-Power Wide Area Network
M2M	Machine To Machine
MAC	Medium Access Control
MCU	Microcontroller Unit
MITM	Man-In-The-Middle
NWK	Network
PAN	Personal Area Network
PHY	Physical

RSA	Rivest–Shamir–Adleman
TC	Trust Center
TCLK	Trust Center Link Key
WPAN	Wireless Personal Area Network
WSN	Wireless Sensor Network
ZC	Zigbee Coordinator
ZED	Zigbee End Device
ZR	Zigbee Router

CHAPTER 1 INTRODUCTION

1.1. Introduction

Zigbee is an open standard for low-power Wireless Personal Area Network (WPAN) with a simple structure and a low cost. Since secure communication is essential, any communication protocol must ensure data confidentiality, integrity, and availability. As pointed out by (Fan et al., 2017: 1), attempting to keep devices inexpensive, low-power, and highly compatible resulted in compromised security in Zigbee, which gives rise to significant security risks. The goal of this research is to identify vulnerabilities within the Zigbee network's architecture affecting its level of security while also proposing improvement measures accordingly. This chapter will provide an overview of the research by first discussing the background, then the research problem, objectives, questions, significance, and finally the limitations.

1.2. Background

In light of a report by (Curryer, 2023) the Internet of Things (IoT) is undergoing a remarkable surge, boasting around 15.14 billion connected IoT devices in 2023. Forecasts anticipate a substantial escalation, surpassing 25 billion devices by the year 2030. The extensive expansion of IoT technologies and the growing desire for them initiated the creation of various IoT standards. Notably, Zigbee, as highlighted by (DFRobot, 2023), stands out as one of the prevailing wireless communication technologies in widespread use. Therefore, being one of the earliest and most widely adopted IoT standards, Zigbee will remain relevant. According to the Connectivity Standards Alliance (CSA), the Alliance will continue to develop Zigbee technology and will retain the Zigbee technology brand (Flaherty, 2021). An article published on its website, (Connectivity Standards Alliance, 2021) indicates that the number of Zigbee-certified products is growing, highlighting 2020 as the year of record-breaking certifications with an increase of 30% over the previous year. This reflects steady market demand, continuous deployment, and widespread industry adoption of the Zigbee technology. As a result of increasing earning popularity, the security of the Zigbee devices becomes critical, especially when devices have access to highly personalized and sensitive information.

Like any other wireless communication protocol, Zigbee technology is not immune to security threats. ZigBee wireless network devices, due to cost constraints, have limited computational power, small memory size, restrained energy consumption and communication capability. This makes it impractical to apply sophisticated security techniques like public key cryptography (Razouk et al., 2014: 376). Furthermore, because ZigBee is targeted for low-cost applications, the availability of tamper-resistant hardware is not warranted. If an intruder attains a device from an operating network that has no tamper resistance, they may acquire access to secret keying material and other confidential data, as well as access to the security software and hardware. For this reason, it is important to further study the concerns related to ZigBee security features and focus more research on this area.

The purpose of this research is to investigate the security of the Zigbee communication protocol, identify possible security loopholes, and find ways to enhance the security aspects of the protocol. The researcher argues that there is still a need to further analyse security in Zigbee protocol to identify methods of improving it.

1.3. Problem statement

According to researchers interested in securing IoT networks, many IoT-connected devices lack adequate security and are therefore vulnerable to cyber-attacks (Rana et al., 2018: 37). One of the communication technologies that is frequently utilised in Internet of Things platforms is ZigBee.

Despite extensive research and testing on the Zigbee standard, improvements have not been sufficient, with severe vulnerabilities in consumer-grade devices still present today (Van Leeuwen & Ayuk, 2019).

The existence of vulnerabilities within the Zigbee network poses a significant concern for individuals and organizations striving to guarantee the provision of end-user confidentiality and security in their network implementations.

1.4. Aim

This study aims to investigate security flaws in Zigbee communications and propose a solution to mitigate security vulnerabilities without increasing the system's cost or power consumption.

1.5. Objectives of the research

i. Identify security vulnerabilities in the Zigbee specification.
ii. Examine potential exploits for known Zigbee security flaws.
iii. Offer a solution to enhance network security for Zigbee-enabled devices.

1.6. Research questions

i. What are security vulnerabilities that can make Zigbee communication protocol unsafe for transmitting sensitive and personal data?
ii. How can attackers exploit Zigbee specification flaws to compromise the system?
iii. How can the protocol's security be improved?

1.7. Significance of the research

For wireless technology, secure communication is essential. IoT and wireless sensor network systems using the Zigbee protocol, particularly those where it is used to transmit private information, will benefit from this study.

1.8. Research design and methodology

The research will evaluate the Zigbee specification and its security mechanisms. An extensive study on wireless network security, in general, is carried out. A review of the literature on the security vulnerabilities in Zigbee and how they are exploited will be conducted. A method for enhancing Zigbee security is proposed. The research will finalize the solution with a prototype that uses the Zigbee protocol and applies the suggested approach.

1.9. Research outputs, results, and contributions

Lightweight Zigbee security solution, appropriate for slower and resource-constrained hardware, resulting in a more secure IoT and WSN.

One Journal paper

One Thesis

1.10. Delimitations

Only the information required to comprehend the Zigbee security architecture is provided. Any further details that are not important to this study may be omitted.
The prototype will be made up of the few available nodes and act as a foundational building component.

1.11. Thesis structure and outline

In **Chapter 1**, the context of the study has been introduced. The research objectives and questions have been identified, and the value of such research is argued. The delimitations of the study have also been discussed.

In **Chapter 2**, the Zigbee protocol is covered in detail, and the issues relating to the problem statement that inspired the research questions are identified.

Chapter 3 will cover the literature study on security in wireless communication. It will go through authentication, specifications for cryptography modules, and key management. The research concerns regarding creating a secure Zigbee network will be addressed in this chapter by comparing algorithms and solutions.

In **Chapter 4**, the existing literature will be reviewed to identify key flaws in Zigbee specification. How these weaknesses are exploited is also examined. The work previously conducted to address and improve the identified weaknesses is also reviewed.

Chapter 5 will explore the integration of the Elliptic Curve Diffie-Hellman (ECDH) cryptographic algorithm into a Zigbee protocol. The chapter will also discuss the implementation of ECDH in Zigbee and how it will be used to enhance the network's security by generating and exchanging a link key between the nodes using ECDH to encrypt the network key transmission.

Chapter 6 will provide the implementation of the suggested security solutions on the Zigbee network, as well as the development environment and methods.

Chapter 7 will discuss the results of the proposed security solution.

Chapter 8 will draw a conclusion to the research based on the discussions and the results obtained.

1.12. Summary

Whether in a business building, power grid, industrial plant, or home security system, the operation of crucial systems infrastructure must never be compromised. The fact that ZigBee is being utilised in more and more significant applications makes the additional security features built into Zigbee crucial. The gap between Zigbee's limited processing capabilities and the requirement to integrate additional effective security controls must be addressed swiftly. This study's findings will aid in resolving ZigBee security issues.

CHAPTER 2 ZIGBEE NETWORK

2.1. Introduction

The chapter presents the functions of devices in a Zigbee network, explains the structure of the Zigbee stack, and examines security measures like trust centers, network layer security, and application support layer security. Additionally, it covers the multiple keys utilized to safeguard data in distinct manners.

2.2. The Zigbee network

Zigbee is a wireless PAN (Personal Area Network) protocol that was intended to facilitate automation, machine-to-machine interaction, remote control, and monitoring of Internet of Things devices. It emerged from the IEEE 802.15.4 wireless standard and is maintained by the Connectivity Standards Alliance (formerly the Zigbee Alliance). The IEEE 802.15.4 standard specifies physical and data link layer requirements, whereas the Zigbee provides specifications from the network layer to the application layer. ZigBee devices are classified into three types:

i. **ZigBee Coordinator (ZC):** The Coordinator is the most competent device, forming the root of the network and potentially bridging additional networks. Each network has precisely one ZigBee Coordinator since it is the device that initiated the network. It maintains network information and serves as the Trust Center (TC) and repository for security keys.

ii. **ZigBee Router (ZR):** In addition to running an application, ZR may operate as an intermediary router, forwarding data from other devices. Since they are not intended to sleep, routers should typically stay operational as long as a network is active.

iii. **ZigBee End Device (ZED):** A device with only enough capabilities to communicate with the parent node, either the ZC or a ZR, it cannot relay data between devices. This connection enables the node to sleep for a considerable portion of the time, resulting in a long battery life.

2.3. Zigbee Stack Architecture

ZigBee is an architecture that was built on top of the IEEE 802.15.4 reference stack model and takes full advantage of its strong physical radio layer. IEEE 802.15.4 is a Low-Rate Wireless Personal Area Network (LR-WPAN) technology. The Zigbee architecture, illustrated in Figure 2.1, is divided into several layered components. Each layer provides a distinct set of services to the layer above it. The two lowest levels are defined by IEEE 802.15.4, the Physical (PHY) layer and the Medium Access Control (MAC) layer. The ZigBee Alliance expands on this basis by providing the network layer and the application layer framework. The Application Support sub-layer (APS) and the ZigBee Device Objects (ZDO) make up the application layer framework. The framework is used by application objects defined by the manufacturer, which share APS and security with the ZDO.

IEEE 802.15.4 contains two PHY layers which operate at different frequencies. The lower frequency PHY layer includes the 868 MHz band as well as the 915 MHz. The higher frequency PHY layer operates at 2.4 GHz. Using a CSMA-CA method, the IEEE 802.15.4 MAC regulates radio channel access. It may also be responsible for delivering beacon frames, synchronization, and delivering a dependable transmission method (Zigbee Alliance, 2017: 1–2).

The network layer is responsible for ensuring the correct operation of the IEEE 802.15.4 MAC as well as providing an acceptable service interface to the application layer. The network layer essentially contains two service entities that offer the necessary capability to interface with the application layer. The data service and the management service are the two service entities. A data entity is responsible for data transmission, whereas a management entity is responsible for all other services. Each service entity presents an interface to the higher layer via a Service Access Point (SAP), and each SAP supports a variety of service primitives to provide the necessary functionality.

The ZigBee Application (APL) layer is made up of the Application Support Sub-layer (APS) sublayer, the ZDO, and the application objects defined by the manufacturer (Wang et al., 2014).

The ZDO oversees the implementation of Zigbee End Devices, Zigbee Routers, and Zigbee Coordinators. ZDO serves as a bridge between application objects, device profiles, and applications. It compiles configuration data from the end APS to decide and implement device and service discovery, security management, network management, binding, node, and group management. A Zigbee Device Object maintains a device's security policies and security settings. It distinguishes three categories of logical devices in a network (coordinator, router, and end device), each with a distinct function.

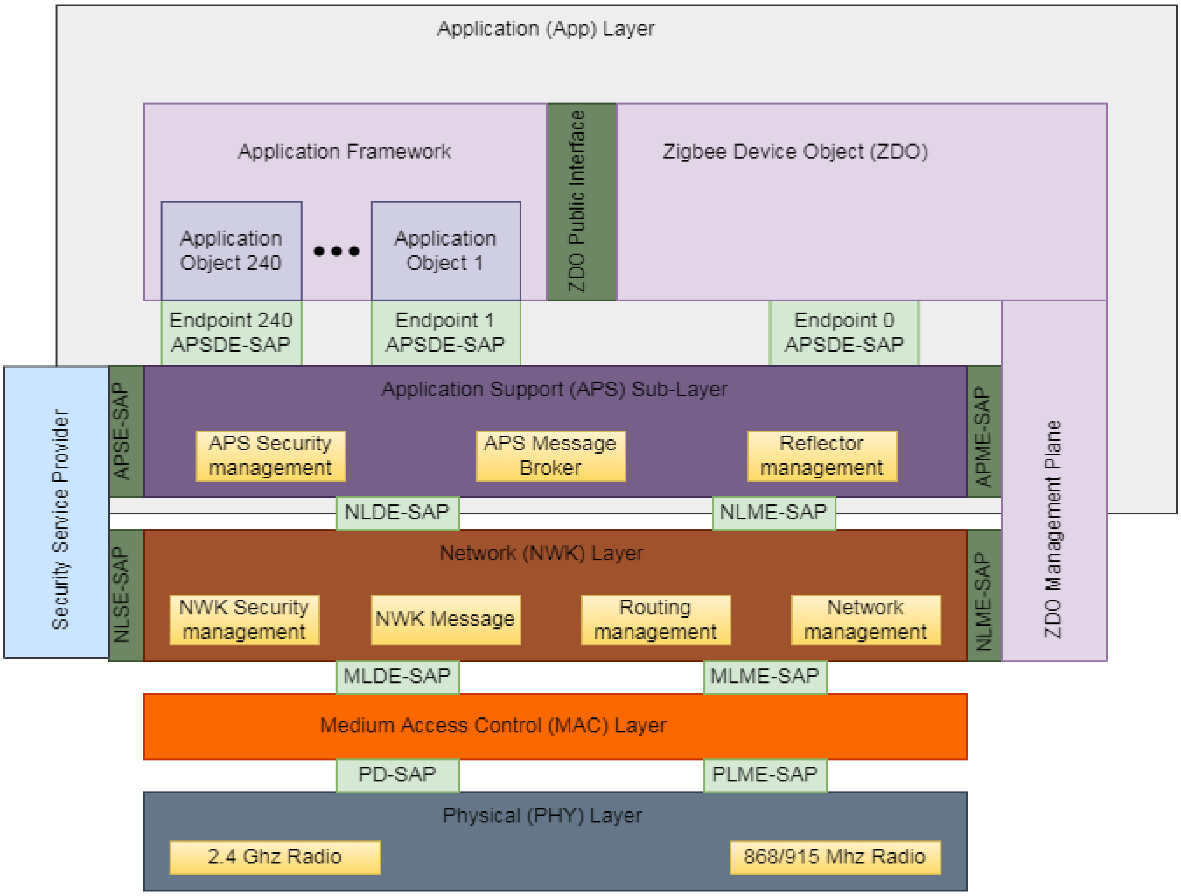


Figure 2.1: Zigbee Architecture

(Zigbee Alliance, 2017: 2)

The Application Support Sublayer (APS) connects the NWK and APL layers. It offers services for the formation and stabilization of security connections. Services are supplied via the APS Data Entity (APSDE) and the APS Management Entity (APSME) is in charge of data transmission services between application entities, whereas APSME is in charge of security, device binding, and group administration. The APS sublayer supports frame security based on link keys or the network key. It is also in charge of the processing procedures required to safely transmit and receive frames, and securely construct and preserve cryptographic keys (Ivezic, 2019).

2.4. Zigbee Network Topology

Personal Area Network (PAN) topologies supported by Zigbee networks include star, tree, and mesh. The topology option is considered at the network design stage and must be aligned with the network's objective. The topology chosen is further determined by device power supply options, estimated battery duration, network activity intensity, latency constraints, and network element (Ivezic, 2019).

The star topology, as shown in Figure 2.2, has no routers, and the coordinator is in charge of routing packets, starting and maintaining network devices. End devices can only communicate with one another through the coordinator.

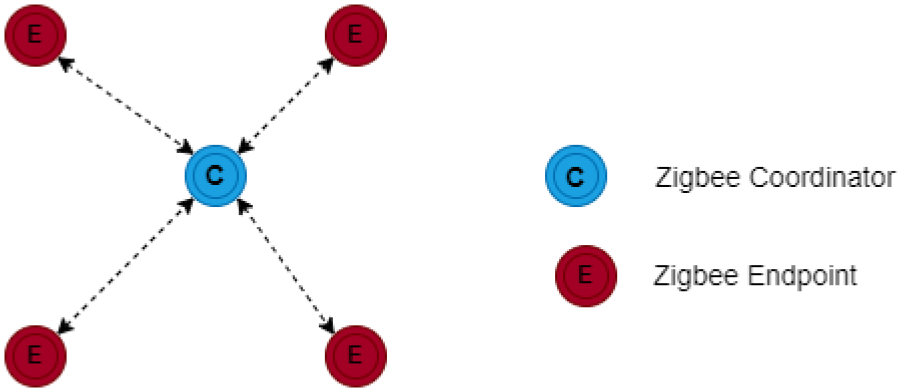


Figure 2.2: Star Topology

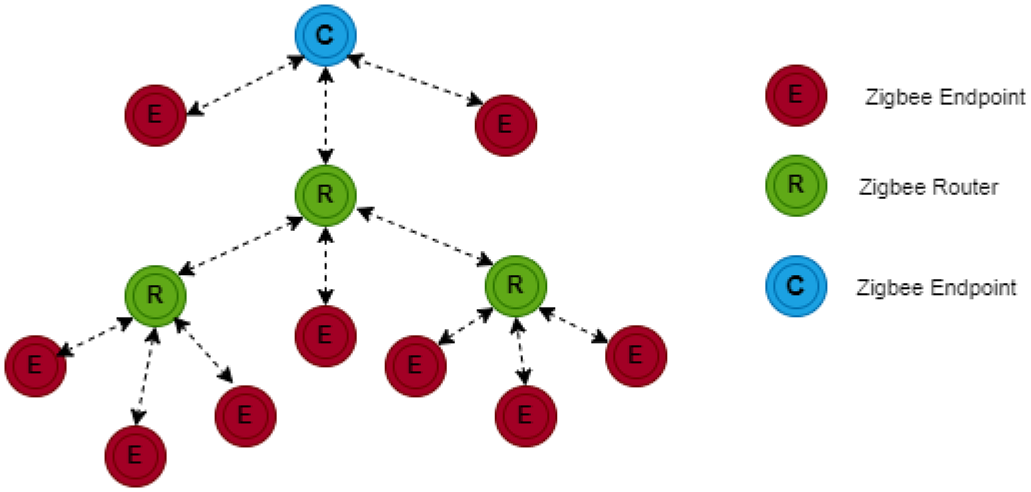


Figure 2.3: Tree Topology

Figure 2.3 depicts a tree topology. The coordinator is the master node in charge of constructing the network and selecting critical network settings. A router is responsible for relaying packets through the network using a hierarchical routing method and might be a child of the coordinator or another router. An end device can be a coordinator or a router's child, and it can only connect with another end device through a router or a coordinator. The IEEE 802.15.4 standard allows tree networks to use beacon-oriented communication.

Figure 2.4 illustrates the mesh topology. It enables complete peer-to-peer information exchange. It features a single coordinator, numerous routers for network extension, and optional end devices. The coordinator is in charge of building the network and selecting important network parameters. Routers cannot send beacons under this topology.

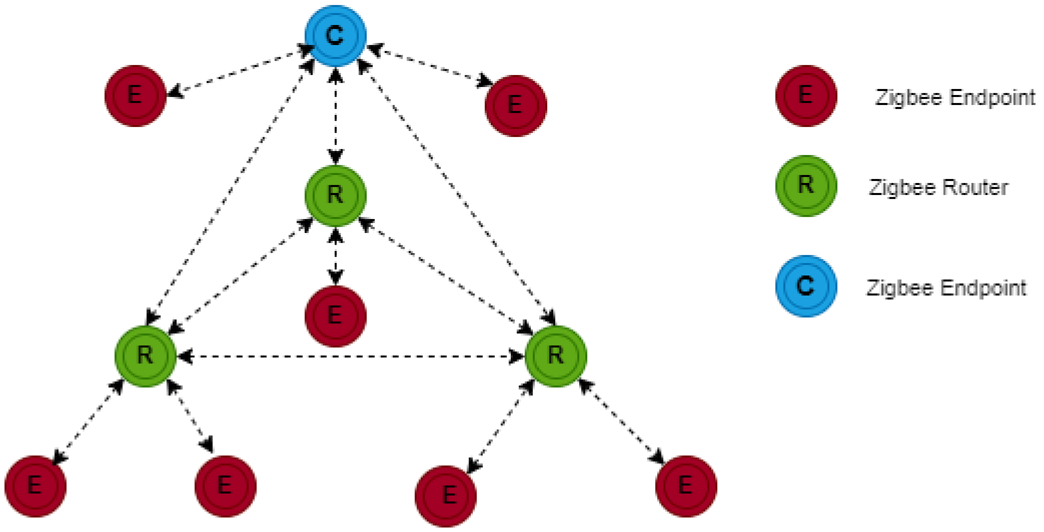


Figure 2.4: Mesh Topology

2.5. Zigbee Security

Methods for key establishment, key transport, frame protection, and device management are among the security services offered by ZigBee. The foundation for applying security policies inside a ZigBee device is formed by these services (Zigbee Alliance, 2017: 407).

There are two types of security networks: centralized security networks and distributed security networks. Centralized security utilizes a coordinator/trust center to establish the network and handle the distribution of network and link security keys to joining nodes. The trust center (TC) is the sole component of the centralized network that is permitted to distribute keys. There is no coordinator or trust center in a distributed security network, and the network is established by a router. Any Zigbee router node may then transfer keys to connecting devices using conventional transport key instructions or other out-of-band ways (Digi, n.d.).

The ZigBee security architecture's level of security is dependent on the symmetric keys being stored safely, the protection measures being used, and the correct application of the encryption techniques and related security policies.

Table 2.1: Security Keys

Key name	**Description**
Centralized security global trust center link key	Link key used for joining centralized security networks. It is used or supported by the device if no other link key is specified by the application at the time of joining
Distributed security global link key	Link key used for joining distributed security networks
Install code link key	Link key derived from install code from joining device to create unique trust center link key for joining
Application link key	Link key used between two devices for application layer encryption
Device Specific trust center link key	Link key used between the trust center and a device in the network. Used for trust center commands and application layer encryption.
Network Key	Network key is shared amongst every device in the network. It is used by a trust center to add end devices to the network

Link keys and a network key are used to secure a system of ZigBee devices. Single hop connection between two devices, none of which is the trust center, is protected by a 128-bit link key shared between the two devices, whereas broadcast and network layer communications are secured by a 128-bit network key shared by all devices in the network. The intended receiver understands whether a frame is encrypted with a link key or a network key. A device should obtain link keys through key transport or pre-installation (for example, during factory installation). A network key must be obtained by a device through key transfer. Some application profiles have also built out-of-band techniques or key negotiation protocols for producing link keys or network keys on devices. Finally, device security is dependent on secure key initialization and installation. Table 2.1 describes the various types of keys utilized.

2.6. Security Architecture

Security mechanisms are built into the ZigBee security architecture at two levels of the protocol stack. The NWK and APS layers are in charge of ensuring the safe transport of their corresponding frames. These layers oversee the processing steps required to securely transmit and receive frames. Upper layers manage security processing operations by configuring suitable keys and frame counters and determining which security level to employ. The APS sublayer offers services for establishing and maintaining security relationships. The ZDO manages a device's security policies and security configuration.

The Zigbee security relies upon keeping the secret key hidden from prying eyes. For a device, Zigbee adopts an open trust model; various levels of the communication stack and all applications operating on a single device trust one another. As a result, the security architecture relies on end-to-end encryption but no encryption between layers, and cryptographically securing just the interfaces between various devices. The open trust model assumes that the network key is kept hidden from unauthorized nodes.

Furthermore, each protocol layer (APS, NWK, and MAC) manages the security of the frames originated by that layer. Because of the open trust architecture, and because all protocol layers typically reside on the same node, security may be predicated on key reuse by each layer. On a single node, the APS, NWK, and MAC layers all utilize the same security key. This contributes to lower storage costs.

2.7. Network Layer Security

Network security ensures security regardless of the applications operating on a Zigbee node. It offers the fundamental access control for determining which nodes are permitted to engage in a certain Zigbee network. For encryption and decryption, network security makes use of a network-wide key. All network-enabled devices have a copy of the key that is used to encrypt and decrypt every network traffic. Every device in a secure Zigbee network has a record of the network key. It is critical that devices that possess a network key retain it securely. A leak of the network key may jeopardize the network's confidentiality, integrity, and availability (Silicon Labs, 2022b)

2.8. Application Layer Security

The purpose of APS security is to provide a method for securely sending messages inside a Zigbee network such that no other device can decode the data except the source and destination. This differs from network security, which simply provides hop-by-hop protection. In such an instance, every network device hears the packet being transmitted to its destination and decrypts it. APS security employs a shared key of which only the source and destination are aware of, ensuring end-to-end security. To encrypt the contents of a communication, both APS layer and network layer encryption can be employed concurrently. In this situation, APS layer security is used first, followed by network layer security.

2.9. Joining procedure of Node

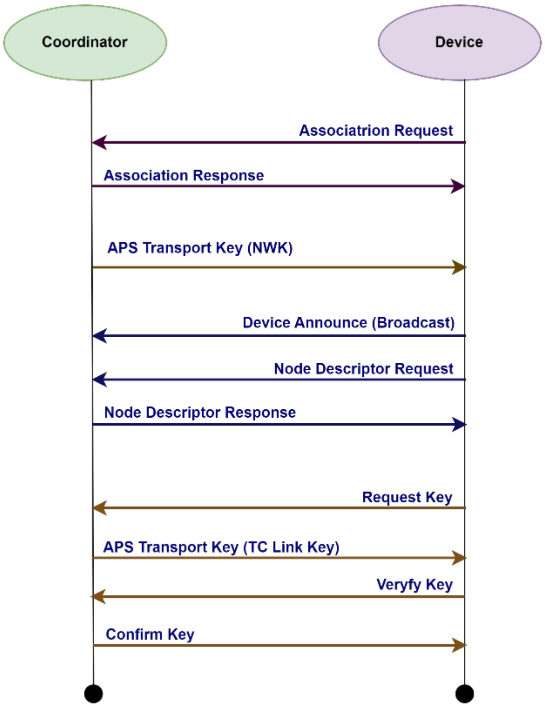


Figure 2.5: Joining procedure

A device joins a WPAN by associating with its coordinator. A device should initially scan channels for possible coordinators. After selecting a possible coordinator, the device sends an association request message to the coordinator. The coordinator will acknowledge receipt of the request. Then it will examine its resources to evaluate whether or not to accept this association request. To obtain the association result, the device will submit a data request to the coordinator. The coordinator can then send the outcome of the association to the device. Figure 2.5 summarizes the association method.

2.10. Summary

The APS layer may be decrypted by anyone using the Zigbee default TCLK. This becomes the most serious security flaw. When devices join, the network key is broadcast over the air encrypted using the default TCLK, enabling any attacker sniffing during that moment to access the network key. The network key can be employed to enable devices to connect to the network and decrypt all network packet transfers. Additionally, Zigbee security does not enable authentication: any device having access to the default TCLK can automatically join the network. To improve network security, the joining procedure must be adjusted to enable only authenticated and authorized devices to connect to the network. It is crucial to remember that the network key must be acquired before any communication between two devices may take place. The considerations raised above can be developed further to get increased security in answering the problem statement and achieving the objective of this study.

CHAPTER 3 NETWORK SECURITY

3.1. Introduction

With a better understanding of network security, we can build a secure network capable of identifying, responding to, and protecting against potential Zigbee vulnerabilities detailed in Chapter 4. This chapter will cover general network security, cryptography, cryptographic key management, and algorithms for encryption.

3.2. Network Security

In a wireless network, transmitted frames can be received by any listening device operating on the same frequency band. Several security issues with wireless networks include a range of potential threats that can compromise the confidentiality, integrity, and availability of data transmitted over these networks. The following are some of the issues:

a) Data confidentiality

The intruder device can access private or sensitive information by sniffing the transmitted packets. This problem can be solved by data encryption. An encryption algorithm, along with the encryption key, encodes the original information into an unreadable message. Only the intended recipient will be able to decrypt the message.

b) Data authenticity:

It is a way of confirming digital identification and keeping invalid users off networks, systems, and devices. Including a message integrity code (MIC) with each outgoing frame will allow the recipient to know whether the message has been changed in transit.

c) Data Integrity:

Protecting information within a network from being modified or deleted by unauthorized parties. The concern here is that the intruder may modify and resend one of the previously sent packets, even if the messages are encrypted.

d) Data Availability:

The assertion that information can be accessed by an authorized user whenever it is needed.

e) Data Accountability:

Data accountability is a crucial concept encompassing tracking of user actions entities in the network such that it is possible to trace back the actions to the entity that is responsible for them.

(Cisco, n.d.) defined network security as the protection of the underlying networking infrastructure from unauthorized access, misuse, or theft. It involves creating a secure infrastructure for devices, users, and applications to work in a secure manner. It is important to recognize possible security attacks that can occur on a network. This will help in implementing intense security policies and controls to block malicious actors from carrying out threats and exploits.

There are two forms of attacks that are related to security, namely passive and active attacks. Attackers may perform passive attacks to gather information and then use that information to perform active, vigorous attacks.

3.2.1. Passive Attack

The purpose of a passive attack is to gain information about the system being targeted; it does not involve any data alteration. Passive attacks are hard to detect because they do not involve any direct action on the target (TechTarget, 2021). Passive attacks come in various forms, including the following:

Traffic analysis: This involves analysing network traffic as it moves to and from the target systems. These types of attacks use statistical methods to analyse and interpret the patterns of communication exchanged over the network. Although these attacks can be performed on encrypted network traffic, they are more common on unencrypted traffic.

Eavesdropping occurs when an attacker captures, erases, or alters data that is being exchanged in a communication medium. Although eavesdropping is similar to snooping, snooping is limited to gaining access to data during transmission with the intention of discovering data that should not be visible or shared.

Footprinting: This is the process of gathering as much information as possible about the target's network and resources. It is usually the first step in gathering information for a penetration test.

Spying: An intruder might disguise as an authorized network user and spy without being noticed.

Dumpster diving. In this type of attack, intruders look for information stored on disposed devices and packaging. They can then use this information to secretly gain entry to a network.

3.2.2. Active Attack

An active attack is a network exploit during which the attackers modify the data stream or fabricate false messages and impact the system resource. The attackers attempt to disrupt and force the lock of the system. An active attack is easier to detect compared to a passive attack but is also difficult to perform compared to a passive attack.

Active attacks can be divided into the following four categories: masquerade, replay, modification of messages, and denial of service (Xiao et al., 2007: 104).

Masquerade: takes place when an intruder impersonates an authentic node.
It is usually combined with other active attacks by the adversary to deploy security attacks.

Replay: It is a passive capture of valid transmissions and retransmits to create an unwarranted effect.

Modification: is when a segment of a legitimate message is modified, or that message is delayed or reordered to produce an unwarranted effect.

Denial of service (DoS): is an attack that prevents the normal use of communication facilities. DoS attacks accomplish this by flooding the target with traffic to consume valuable network resources such as bandwidth or node resources such as memory or computation power.

Brute-force attack: attackers work through all possible combinations, hoping to guess encryption keys correctly.

3.3. Cryptography

Cryptography is an automated mathematical tool that secures communications from outside observers. It ensures data confidentiality and integrity, as well as user authentication and accountability. The cryptography technique consists of plaintext, ciphertext, an encryption key, and encryption and decryption algorithms. The encryption algorithms scramble plaintext and produce unreadable ciphertext. The key allows the intended receiver to restructure the original data using decryption algorithms (Sarkar et al., 2021: 18–29).

The three main categories of cryptographic methods are hash functions, symmetric-key cryptography, and asymmetric-key cryptography.

3.3.1. Symmetric-key cryptography

Symmetric cryptography, or Secret Key Cryptography, uses a single key to encrypt data. Both encryption and decryption in symmetric cryptography use the same key, making this the easiest form of cryptography. The cryptographic algorithm utilizes the key in a cipher to encrypt the data, and when the data must be accessed again, a person entrusted with the secret key can decrypt the data. Secret Key Cryptography can be used on both in-transit and at-rest data but is commonly only used on at-rest data, as sending the secret to the recipient of the message can lead to compromise.

Symmetric cryptography is split into block ciphers and stream ciphers. **Stream ciphers** encrypt bits individually. This is achieved by adding a bit from a key stream to a plaintext bit. There are synchronous stream ciphers where the key stream depends only on the key and asynchronous ones where the key stream also depends on the ciphertext (Paar & Pelzl, 2010: 29).

Block ciphers encrypt a complete block of plaintext bits at a time with the same key. The encryption of any plaintext bit in a given block depends on every other plaintext bit in the same block. Most block ciphers either have a block length of 128 bits (16 bytes), such as the advanced encryption standard (AES), or a block length of 64 bits (8 bytes), such as the data encryption standard (DES) or triple DES (3DES) algorithm (Paar & Pelzl, 2010: 30–31).

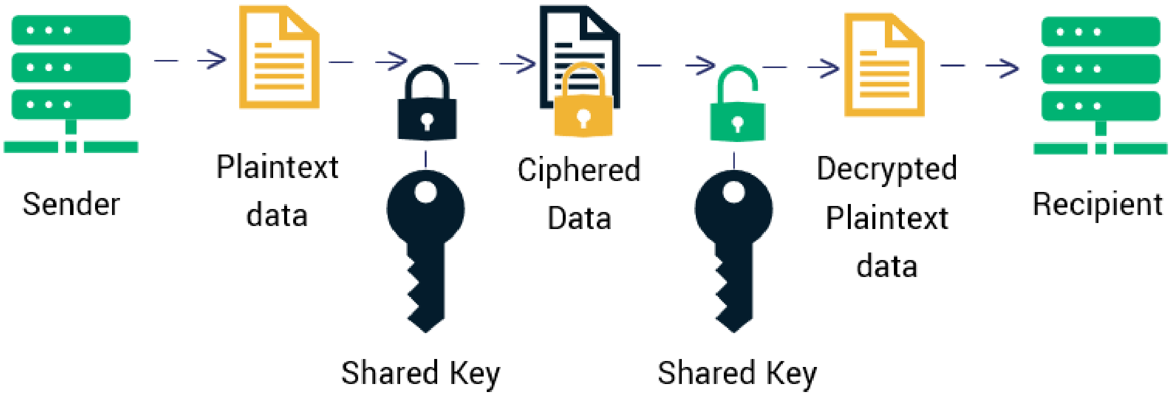


Figure 3.1: Symmetric Key Encryption

(Thakkar, 2020)

3.3.2. Asymmetric-key cryptography

One key is kept private and is called the "private key", while the other is shared publicly and can be used by anyone; hence, it is known as the "public key". The mathematical relation between the keys is such that the private key cannot be derived from the public key, but the public key can be derived from the private one. The private key should not be distributed and should remain with the owner only. The public key can be given to any other entity.

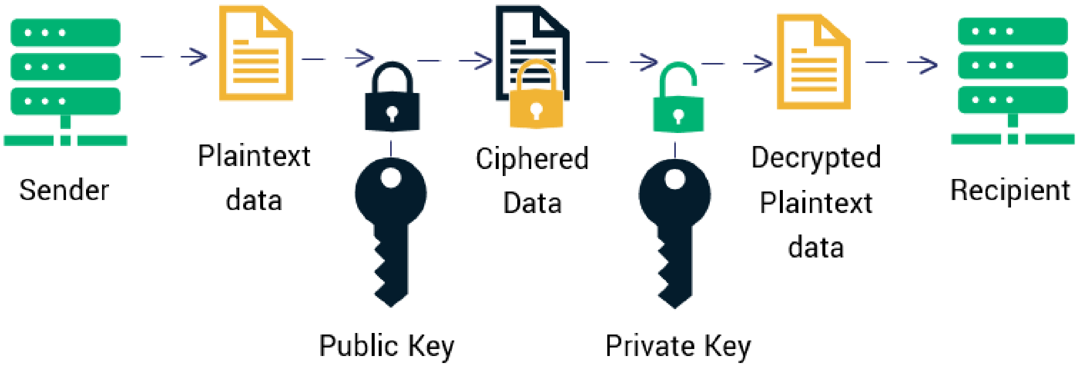


Figure 3.2: Asymmetric Key Encryption

(Thakkar, 2020)

3.3.3. Hash functions

Hash functions are irreversible, one-way functions that protect the data at the cost of not being able to recover the original message. Hashing is a way to transform a given string into a fixed-length string. A good hashing algorithm will produce unique outputs for each input given. The only way to crack a hash is by trying every input possible until you get the exact same hash. A hash can be used for hashing data, such as passwords, and in certificates.

3.4. Key management

Cryptographic keys are a crucial part of security systems and form the basis of all data security. Data is encrypted and decrypted via the use of encryption keys, which means the loss or compromise of any encryption key would invalidate the data security measures put in place. Proper management of keys and their related components can ensure the safety of confidential information. Key Management is the process of putting certain standards in place to ensure the security of cryptographic keys in an organization. It deals with the creation, exchange, storage, deletion, and refreshing of keys. Well-protected keys are only accessible to users who need them.

Key Management follows a lifecycle of operations that are needed to ensure the key is created, stored, used, and rotated securely. To safeguard secure key generation, storage, usage, and distribution, cryptographic keys follow a life cycle that involves:

Generation

The first step in ensuring that a key is secure is to generate it with a strong encryption algorithm and on a secure location. Key generators, AES encryption algorithms, or random number generators tend to be used for secure key generation.

Distribution

Ensures the security of the keys being distributed is maintained. Carries out a safe distribution of the keys to the required user via a secure connection.

Use

Once the key is distributed, it is used for cryptographic operations by authorized users.

Storage

Keys must be securely stored for encryption and decryption. One of the secure methods is through the Hardware Security Module (HSM).

Rotation

When the key expires, it is retired and replaced with a new key. It is necessary to rotate keys regularly, after a period known as cryptoperiod, to reduce the chances of it being stolen or compromised.

Revocation and Destruction

There are two ways to deal with a compromised key: revoking or destroying the key in question. Revocation of a key means the key can no longer be used to encrypt or decrypt data, even if its cryptoperiod is still valid. Destroying a key deletes it permanently from any key storage method. Deactivated keys may be kept in an archive to be used when old data encrypted in the past needs to be decrypted by that key or key pair.

3.5. Encryption Algorithms

Encryption algorithms are a mathematical formula which, with the help of a key, changes plaintext into ciphertext. They also make it possible to reverse the process and turn ciphertext back into plaintext.

The following table lists the common encryption algorithms: Add ECC and ECDH

3.5.1. AES

The Advanced Encryption Standard (AES) is an encryption algorithm published by the National Institute of Science and Technology (NIST) in 2001 (Stallings, 2006). AES is a symmetric block cipher with a very complex structure compared to public-key ciphers and other symmetric ciphers. It utilized a block cipher from the Rjindael cipher family and was designed to have the following characteristics: resistance against all known attacks, speed, and code compactness on a wide range of platforms, and design simplicity.

The key length in the AES specification allows 128, 192, or 256 bits, but the block length is limited to 128 bits. The bigger the key size, the more secure the encryption. Using a series of bitwise operations, blocks of data are encrypted using keys of a given length. If a 128-bit key is used, the encryption on the block is done 10 times. With 192, the encryption is done 12 times, and with 256, 14 times. Thus, 256-bit keys are the most secure, but for most encryption cases, 128-bit keys are sufficient. The higher the security level of the data, however, the higher the size of the key should be. AES is suitable for systems with limited resources because of its small footprint. It is the encryption method Zigbee uses for end-to-end communication.

Table 3.1: Encryption Algorithms

Algorithm	Type	Mode
AES	Symmetric	Block cipher
DES and Triple DES	Symmetric	Block cipher
RSA	Asymmetric	Block cipher
Diffie-Hellman	Asymmetric	Mathematical
ECC	Asymmetric	Mathematical
ECDH	Asymmetric	Mathematical

3.5.2. DES and 3DES

The Data Encryption Standard (DES) is a symmetric cipher that encrypts blocks of length of 64 bits with a key of size of 56 bits (Paar & Pelzl, 2010). The DES key space is too small, making it insecure against a determined attacker, but it is still used in legacy applications. 3DES, or triple DES, encrypts data three times in a row with DES and yields a very secure cipher, which is still implemented in some situations. Triple DES employs a variety of key selection approaches, including the following:

- In the first, all keys used are unique.
- Two keys are the same, and one is different in the second.
- All keys are the same in the third.

3.5.3. RSA

The RSA cryptographic scheme, sometimes referred to as the Rivest–Shamir–Adleman algorithm, is a widely used asymmetric cryptographic scheme. There are many applications for RSA, but in practice, it is most often used for the encryption of small pieces of data and for digital signatures (Stallings, 2006). However, RSA encryption is several times slower than symmetric ciphers such as AES. This is because of the many computations involved in performing RSA. Thus, the main use of the encryption feature is to securely exchange a key for a symmetric cipher. In practice, RSA is often used together with a symmetric cipher such as AES, where the symmetric cipher does the actual bulk data encryption.

The technical details of RSA work on the idea that it is easy to generate a number by multiplying two sufficiently large numbers together, but factorizing that number back into the original prime numbers is extremely difficult. The public and private keys are created with two numbers, one of which is a product of two large prime numbers. Both use the same two prime numbers to compute their value. RSA keys tend to be 1024 or 2048 bits in length, making them extremely difficult to factorize. RSA relies on the size of its key to be difficult to break. The longer an RSA key, the more secure it is (Puneet, 2020). As a result, it is unsuitable for the Zigbee protocol, which is designed for low-data transmission.

3.5.4. Diffie-Hellman

The Diffie–Hellman (DH) key exchange enables two parties to derive a common secret key by communicating over an insecure authenticated channel. DH securely generates a unique session key for encryption and decryption that has the additional property of forwarding secrecy. The trick is to use a mathematical function that's easy to calculate in one direction but very difficult to reverse, even when some of the aspects of the exchange are known (Saha, 2021). While using DH key exchange, the sender and receiver have no prior knowledge of each other, and communication can take place through an insecure channel. Since the Diffie–Hellman (DH) key exchange does not authenticate either party involved in the exchange, the channel needs to be authenticated.

3.5.5. ECC

Elliptic Curve Cryptography is a sophisticated type of public key cryptography that encrypts wireless data transmission using the elliptic curve's mathematical characteristics. Known for offering strong security while demanding smaller key lengths than more conventional encryption techniques like RSA or Diffie-Hellman, ECC is known for this property. Due to this benefit, ECC is a desirable option for environments with limited resources, such as embedded systems and Internet of Things applications. ECC's public key cryptography strategy is based on mathematical algorithms that control the algebraic structure of elliptic curves over finite fields, in contrast to RSA's use of prime factorization. Consequently, it is more challenging to numerically crack the keys produced by ECC (Filippone, 2023: 1).

3.5.6. ECDH

Elliptic Curve Diffie-Hellman Key Exchange is an anonymous key agreement scheme that enables two parties to create a shared secret across an insecure channel and each has an elliptic-curve public-private key pair. This shared secret can be used as a key on its own or to create more keys. Succeeding transmissions can then be encrypted with a symmetric-key cipher using the key, or the key that was derived from it. The secret shared key cannot be calculated by third parties using publicly available information without knowing the private information of both parties.

3.6. Key bits

A key length for symmetric crypto algorithms is only relevant if a brute-force attack is the best-known attack. If there is an analytical attack that works, a large key space does not help at all. Of course, if there is the possibility of social engineering or implementation attacks, a long key also does not help. The key lengths for symmetric and asymmetric algorithms are dramatically different. For instance, an 80-bit symmetric key provides roughly the same security as a 1024-bit RSA (RSA is a popular asymmetric algorithm) key. Table 3.2 gives a rough indication of the security of symmetric ciphers with respect to brute-force attacks, in accordance with (Paar & Pelzl, 2010). A large key space is a necessary but not sufficient condition for a secure symmetric cipher. The cipher must also be strong against analytical attacks.

Table 3.2: Symmetric Key Ciphers

Key Length	Security Estimation
56 – 64 bits	Short term: a few hours or a day
112 – 128 bits	Long term: several decades (assuming there are no quantum computers)
256 bits	Long term: several decades (even with quantum computers that run present known algorithms)

3.7. Cryptographic implications of computer advancements

It is difficult to know the kinds of computers that will be available to us in the next 10 years. We can apply Moore’s Law for medium-term predictions. Gordon Moore predicted that the computing power of our computers doubles every 18 months to 2 years while the costs stay constant. (Carla Tardi, 2022) states that even though Moore’s observation is neither a legal law nor a scientifically proven theory, it became known as Moore's Law.

This has an effect on cryptography. Given that computers increase in speed and performance over time, less effort, time, and money will be required to break a particular cipher in 10 years compared to today.

3.8. Summary

In conclusion, this chapter presented a comparison between symmetric and asymmetric encryption algorithms. The chapter explored the advantages and disadvantages of both types of encryption and highlighted the differences in their approach to security. After weighing the available options, the chapter concluded by recommending the use of the Elliptic Curve Diffie-Hellman (ECDH) algorithm in the Zigbee protocol. ECDH was chosen due to its lightweight nature and efficient key exchange capabilities. Going forward, Chapter 5 will focus on incorporating ECDH into the Zigbee protocol to enhance its security. Ultimately, this will lead to a more secure and robust Zigbee protocol that is better equipped to protect users' data and privacy.

CHAPTER 4 LITERATURE STUDY

4.1. Introduction

This chapter represents a comprehensive review of literature pertaining to the security considerations of Zigbee protocol since its inception. Several studies have been conducted that provide an overview of the security aspects regarding the Zigbee protocol and this literature review relies heavily on the extensive work by previous researchers. The emphasis will be on the technology's security shortcomings, how these security issues are exploited, and the work that has been done to address them. This enables us to comprehend how Zigbee security has evolved over the years since its introduction. The objective is to gather sufficient data to enable the researcher to come up with ways to improve Zigbee security. In order to add to the existing body of work, novel insights will be presented that can help provide a better understanding of current security issues in the Zigbee standard.

4.2. Evaluating the Zigbee Network's Attack Landscape

In accordance with (Khanji et al., 2019: 54–55), the analysis is performed based on factors such as the context of execution, the layer of the communication stack in which they are detected, the malicious node's affiliation with the network, and the specific area of the network being targeted.

It was mentioned in Chapter 3 that wireless network attacks can be passive or active in nature. Passive attackers can intercept and collect all data packets transmitted within the Zigbee network using a sniffing device. The attacker can learn about the network's devices by analyzing these packets. Active attackers, on the other hand, can introduce a malicious device that transmits Zigbee packets, acting as either a ZED to join the network or a ZC to create a new Zigbee network. Such attackers may also impersonate existing devices in order to trick other network components. This paper also looks at other Zigbee network vulnerabilities, such as exploiting device security features like pre-shared keys or installation codes via physical or remote access during or after manufacturing. The impacts of denial-of-service or resource exhaustion from intentionally disrupting network communication via interference with physical or network layer messages are also considered.

4.3. Investigated Attacks

The headers of the MAC and Network layers contain plaintext MAC and network addresses, PAN ID, and EPID. The attacker can retrieve this information by passively capturing ordinary Zigbee traffic. An attacker can then use this information to mimic one of the endpoints or possibly the coordinator and transmit fabricated messages with the intent to disturb the normal flow of communication. (Wang & Hao, 2022) detailed the steps that an attacker could take in the normal operation of Zigbee networks. They explained that after sniffing the network information, the attacker overwrites the physical address supplied by the attack device vendor and pretends to be a device in the Zigbee network. The attack device then impersonates the network coordinator to build packets and inject them into the Zigbee network, causing the target device to process the fabricated packets and become dysfunctional. The researchers discovered attacks that might cause Zigbee devices to disconnect from the network or expose encryption keys.

To maintain compatibility, some Zegbee devices transmit NWK in plain text or encrypted with GMK during the initial stages of communication. GMK is public knowledge. This has been identified as Zigbee's main vulnerability in earlier studies by (Fan et al., 2017) and (Khanji et al., 2019). In order to obtain the NWK key when a device joined the network, (Fan et al., 2017) conducted a packet sniffing experiment. The experiment worked and the NWK key was obtained. The NWK key allowed hackers to take control of the network and add more intelligent (and harmful) devices to it. The authors also managed to copy other devices' network parameters. They were able to duplicate the MAC address and add it to their malevolent device, which already had the network key, so that it could join the network because the TC mistook it for a simple reconnect due to the broken connection.

As reported by (Jamieson, 2016) this Zigbee network's security flaw is known to the Connectivity Standards Alliance, but they do not view it as a concern because it only affects the network briefly when the network key is being transmitted. They made the decision to increase the security options by configuring the install code for the 2017 Zigbee version. In an out-of-band setup, install code is used by the joining devices as a key to encrypt the transport command, allowing the network key to be transmitted over the air encrypted with it, more details in Section 4.4.1.

Another typical type of Zigbee network attack is the replay attack. Attackers can intercept network traffic and send it again as if they are the original sender, which is the basis of replay attacks (Olawumi et al., 2014). Replay attacks can be used against unencrypted Zigbee networks because an attacker could send the original frames more than once. (Farha & Ning, 2019) reported that (Zualkernan et al., 2009) had successfully run the replay attack on a standard Zigbee network. The attack was conducted with the aid of a program called killerBee. The authors were able to inject old captured frames into the standard Zigbee network because it had no encryption or message integrity checking.

Early researchers recommended deploying Frame Counters (FC) in secure Zigbee networks. (Mishra et al., 2012) propose that in order to avoid the replay attack, the Zigbee stack should be able to identify the frames by a sequence number and make sure that the received number is greater than the previously received frame. (Razouk et al., 2014: 378) expounded that in the Zigbee specification, The ZigBee specification provides a frame counter as a security measure, focusing on ensuring freshness. It uses an ordered sequence of inputs to discard previously sent frames. This indeed makes the network relatively more secure. The work of (Fan et al., 2017) proved that a Zigbee network with FC implemented is less susceptible to relay attacks. After they attempted a replay attack on the Zigbee network by injecting previously encrypted and transmitted messages on the network, they determined that the reason why a replay attack was ineffective was due to the implementation of a counter as part of the encryption and authentication system. However, this is true if the attacker lacks an encryption key. They can possibly alter the packets to modify the counter payload to a greater value than the present system counter if they have access to the encryption key. In this instance, an attacker can also fabricate a frame with a maximum value of 0xfffff, resulting in the rejection of subsequent frames and a denial of service.

When an FC is reset on a secure Zigbee network, a replay attack is achievable even without the encryption key. After resetting the FC, (Farha & Chen, 2018) were able to successfully launch a replay attack against a secure ZigBee network. Additionally, they advised changing the network key after the ZC reboots. It is a good solution, but there are some drawbacks to changing the network key when the ZC restarts. It will change the network key for a large number of devices, consuming power and causing significant network congestion.

Furthermore, this solution is incompatible with ZEDs that come pre-configured with fixed network keys.

The failure to properly address the forward security requirement is another flaw in the ZigBee security model. A node that has left the network can still access the connectivity because it still has the master and link keying material because a proper revocation has not been performed. The loss, abuse, or theft of one or more ZigBee devices is a very real possibility. (Goodspeed, 2009) demonstrates using real-world experiments that the extraction of security keys is feasible. Data extraction is therefore quite possible. Consequently, an adversary could take advantage of the circumstance and exploit this weakness if the keys stored on the devices are not properly revoked. Thereafter, it becomes very easily achievable to attack the network and application layer. Therefore, it is important to take this kind of attack seriously and not to undervalue it.

Another type of attack that the Zigbee network is vulnerable to is Denial of Service (DoS). After connecting to the network with a fake ZigBee device, (Rana et al., 2018) launched a DoS attack. They did not need to monitor the packages being sent for their experiment, but instead, they repeatedly introduced a malicious packet into the network. The network was bombarded with dummy messages during the DoS attack, which made the network freeze and then stop providing service. The dummy packet was injected into the network once every 100 milliseconds for two minutes, causing the network to freeze throughout the entirety of the attack. Nevertheless, the network returned to normal operation after they ceased injecting the packet. They conducted a second experiment to replicate the same attack for 10 minutes with the same packet injection interval. This time, the network crashed, and it was unable to recover. Both experiments are essential for use in the real world. Even though the network was able to recover quickly in the first experiment, the consequences of such attacks can be disastrous and pose a threat to human life if they are carried out on equipment used to monitor patients.

Check Point researchers have proven how to utilize a Zigbee-enabled Philips Hue smart lamp to distribute malware over a network, according to (Anderson, 2020). During the attack, the lamp is initially taken over, its firmware is patched with malicious code, and it is then forced to perform maliciously to trick a user into thinking the lightbulb is disconnected. Because the bulb appears unreachable in the control app, the user will attempt to reset it. After uninstalling the bulb from the app, the control bridge must be instructed to re-discover it. After the bridge discovers the compromised bulb, the user reconnects it to their network. The upgraded firmware on the hacker-controlled light bulb then begin to transmit a lot of messages resulting in a heap-based buffer overflow. This allows the hacker to install malware on the bridge. As the bridge is connected to the local TCP/IP network, the malware may now search for other devices such as computers on the network to infect.

Anderson, (2020) reported that in 2016, researchers regarded Philips Hue devices to be highly difficult targets to attack. Additionally, he emphasized the proliferation of low-cost IoT gadgets like cameras and household appliances. However, the fact that the Philips Hue devices were found to be susceptible while being labelled as extremely hard targets for detecting and exploiting software flaws, may cause one to stop for reflection. Given the growing number of IoT devices and household appliances that use the Zigbee protocol, there are bound to be plenty of simpler targets available.

Worm infection is another type of attack described by (Ronen et al., 2017). In their paper, they claim to have created a shred of solid evidence for a worm that can destroy Philips Hue lightbulbs throughout entire cities. The software uses symmetric encryption keys that are hardcoded to control devices in a Zigbee wireless networks. By utilizing only the built-in ZigBee wireless access and the geographical proximity of the lightbulbs, the worm spreads by hopping directly from one lightbulb to its neighbour nodes. The attacker can turn all the city lights on or off, completely brick them, or use them in a significant DDoS attack by simply plugging in one infected lightbulb anywhere in the city. Within minutes, the attack will have spread rapidly throughout the entire city. Since the attack creates the first native and autonomously self-spreading ZigBee worm, the researchers claim that their attack is much more powerful than all the earlier described attacks. It combines a takeover attack that enables remote full control of lamps without using specialized hardware with a Correlation Power Analysis (CPA) attack against the CCM mode used to encrypt and verify firmware updates. No prior knowledge of the targeted lamps is assumed, and even knowing the secret master key to the ZigBee Light Link (ZLL) is not necessary for the attack to succeed. Of course, it's important to keep in mind that the Philips Hue light bulbs were the only market leading IoT products that the researchers examined. It is very likely that there are several other Zigbee devices out there that haven't yet received attention but are probably just as vulnerable.

4.4. Related Work

The Zigbee Alliance incorporated further safety measures in their revisions in an effort to improve security in the Zigbee protocol. Furthermore, previous researchers have additionally suggested different cryptographic improvements to the system. Even so, these solutions are insufficient to fight against the vulnerabilities outlined in the previous section, or they require more resources than a standard Zigbee device. I'll go over some of the proposed solutions and their shortcomings.

4.4.1. Security features by Zigbee Alliance

First, let us look at the security improvements the Zigbee Alliance has proposed as well as their drawbacks.

Zigbee Link keys.

A link key can be generated between a joining device and the coordinator, as described in the Zigbee Standard (Zigbee Alliance, 2017: 333–348). In addition, an end device can request a link key from the coordinator to use while communicating with a neighboring end device. In response, the coordinator will generate and distribute to both devices a link key encrypted using the NWK. Because the link key is protected by the NWK, an attacker with the NWK can access the link key and undertake attacks. Although link keys enable unique encryption of messages between two devices, the mechanism by which these keys are exchanged does not prevent other devices, malevolent or not, from accessing them. Sending link keys encrypted with the NWK defeats the purpose.

Installation code.

An install code is provided to facilitate secure transmission of information from the centralized Trust Center device to the joining device when joining a Zigbee network. The code is a random value that is loaded on the device during production and encrypts the initial network key transport. It serves as an authorization token for the connecting device and is often printed on the packaging in either hexadecimal or encoded form, such as a barcode or QR code. The code, along with the device's IEEE MAC address, is transmitted to the Trust Center device or its online or cloud interface via an out-of-band mechanism. Device-specific data can be saved on a remote server or cloud-based system, which safely transmits it to the Trust Center, allowing the device to create security credentials. This occurs prior to the start of the in-band joining process (Silicon Labs, 2022a). Despite allowing the verification installation of a connecting device, the install code procedure remains vulnerable. The install code is written on the outside of a device, and anyone with access to the device can obtain it. After that, the attacker can decrypt the transport key message and retrieve the NWK. Moreover, even if the attacker is not present at the moment of joining, the technique does not offer forward secrecy, which implies that an attacker can record the encrypted transmission and later when they obtain the install code decrypt the communications. Moreover, even if the attacker is not present at the moment of joining, the technique does not offer forward secrecy, which implies that an attacker can record the encrypted transmission and later, when they obtain the installation code, decrypt the communications.

Touchlink.

Touchlink is a Zigbee feature that allows devices that are physically close to each other to communicate with each other without being on the same network. A connecting device is placed near the coordinator for verification. (Morgner et al., 2017) discovered this method to be capable of undermining the overall security of the Zigbee protocol. In their study, they showed the extraction of the network key from a distance of 130 meters by passively listening on a touchlink commissioning operation. They were then able to factory reset or permanently remove nodes from the authorized network. Their analysis demonstrates that the support for touchlink commissioning is adequate to undermine the security of the ZigBee applications.

The Zigbee protocol is continually evolving, In April 2023, the Connectivity Standards Alliance (CSA) unveiled the latest update - Zigbee PRO 2023. This new release includes advanced security measures that highlight device safety and compatibility within IoT development. In accordance with (PRNewswire, 2023), key features of this upgrade involve improved mechanisms for safeguarding networks during onboarding and operation stages to combat current security risks.

However, the challenges persist, as the upgraded protocol continues to suffer from a wide range of security weaknesses. As indicated by (Allakany et al., 2023), this is attributed to the incapacity of constrained wireless sensor network devices to employ conventional security protocols like asymmetric cryptographic mechanisms, which are resource-intensive and unsuitable for wireless sensor networks. Hence, there is a need to conduct further research in this area and increase research efforts to develop alternative security solutions.

4.4.2. Public Key Infrastructure (PKI)-based suggestions

Certificate-based model

The use of only symmetric cryptography in Zigbee is the primary cause of vulnerability, making it difficult to add trusted devices to the network while keeping untrusted devices out. (Misra et al., 2016) proposed implementing a public key infrastructure (PKI) as a solution, where each device would have a public and private key as well as a manufacturer-signed certificate. However, this method does not effectively distinguish between the desired device and the billions of other certified devices in an open ecosystem with multiple vendors, making it easy for attackers to include malicious devices. In such an ecosystem, a large number of devices can be legitimately certified. Which makes it easier for attackers to modify the software on a device or become a vendor in order to possess a certified but malicious device. Additionally, it can be difficult to install all of the vendor root certificates on resource-constrained Zigbee devices during production, which restricts the usability and capability of these devices when interacting with other devices when they do not have the vendor root certificates of those other devices.

Certificateless model

In order to generate and distribute the NWK, (Choi et al., 2012) used the elliptic curve Diffie-Hellman (ECDH) key distribution mechanism. They also suggest using subMAC for message authentication and to protect against man-in-the-middle attacks. They argue that their study indicates improved message authentication and a key mechanism that closes security holes in ZigBee Pro for wireless sensor networks. This method, however, is unable to spot a malicious device masquerading as a benign one. Furthermore, it is unclear how secret keys can function as NWK in their solutions because NWK must be distributed among all devices in order to decrypt broadcast messages. Additionally, (Hamza Kadhum, 2020: 67) found out that generating NWK with ECDH results in an unexpected outcome. The researcher experimented with using ECDH to authorize and acquire the NWK key for joining devices. Both the joining device and the TC have to exchange keys to compute the NWK key using the shared secret point G. However, the coordinator, which is also the TC, controls and distributes keys in the network, so it refused to collect the packet containing the public key of the joining node. The researchers discovered that the ECDH solution was not suitable for generating the NWK key since all devices in a Zigbee network must share the same NWKS. They attempted to use ECDH one-way authentication, with only the coordinator sending its public key, but this was dependent on each joining device having the same public and private keys to compute the same symmetric key. However, constant keys eliminate the benefits of ECDH for pairwise secure communication since each node should have a unique key from their private and public keys. Further to that, the extra power computation required to generate constant keys is unnecessary for NWK key generation, where the same key must be shared among all nodes for communication.

4.5. Summary

Many of the above-mentioned threats may be avoided with good key management and authentication. I suggest a new certificate-less Zigbee joining protocol that makes use of inexpensive public-key primitives in order to increase the security of Zigbee networks. This is introduced in the next chapter.

CHAPTER 5 PROPOSED SOLUTION

5.1. Introduction

Due to the flaws associated with NWK and TCLK, there is a need to re-evaluate how secret keys are shared and used among devices in the network. Also, the constraints imposed by the Zigbee environment, such as limited resources and efficiency requirements, need to be considered while coming up with solutions. The suggested fix aims to address the vulnerabilities that stem from using a single shared NWK to encrypt all network communication. To achieve this, link keys related to specific coordinator-to-device communication should be used instead of NWK. A secure arrangement on a shared link key, independent of the NWK's secrecy, should be established between each pair of communicating devices. This is done by employing a key exchange scheme based on the Elliptic Curve Diffie-Hellman (ECDH) during the join procedure. This would enhance the security of the network without compromising its efficiency requirements.

There are two major vulnerabilities in the Zigbee protocols that inspired the proposed solution:

i. Encrypting link keys with the network key, thereby negating the point of employing link keys.

ii. TCLK is used to encrypt the network key. The pre-installed TCLK can leak, putting the entire network at risk.

5.2. The fix

To address these vulnerabilities, a new key exchange procedure in the Zigbee protocol is proposed to prevent private information from being accessible by all network nodes. This is accomplished by switching the order of key exchange so that the link key goes before the NWK. The NWK in the transport key message is then encrypted using the link key as opposed to TCLK.

Link keys are used to protect end-to-end communication privacy. Encrypting broadcast messages with individual link keys for each device, on the other hand, would be inefficient in terms of both time and resources. NWK is retained in the proposed approach, however, it is now only used to encrypt and decrypt broadcast messages. The coordinator transmits NWK to new devices during the join procedure, however in this new proposal, the NWK is encrypted with the link key rather than the TCLK. Broadcast messages can only contain network-related commands, such as device announcements, and cannot contain commands particular to individual devices.

5.2.1. ECDH Key Exchange in Zigbee

ECDH Key Exchange is an anonymous key agreement mechanism that allows two parties, each with an elliptic-curve public-private key pair, to establish a shared secret across an unsecured channel. (Aikins-Bekoe & Ben, 2020: 2) stated that ECDH cryptography is considered the strongest method for key generation. This algorithm enables two parties to establish a shared secret key without ever transmitting the key itself, which makes it perfect for key generation and exchange.

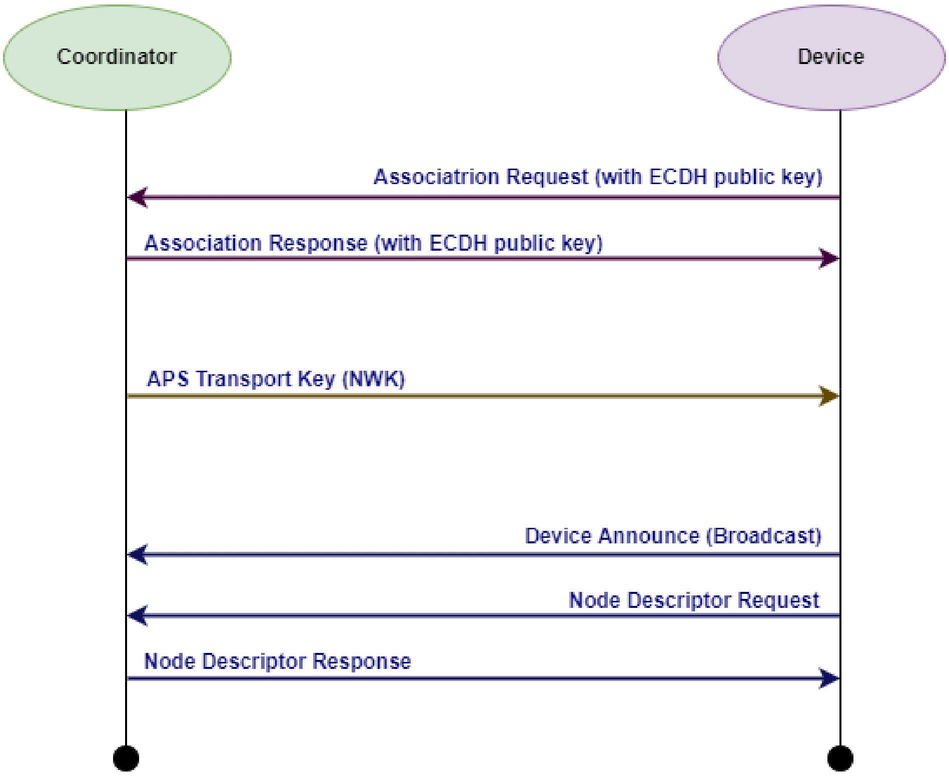
Incorporating it into Zigbee protocol, two Zigbee devices Dev1 and Dev2, can share a secret key as follows:

1. Dev1 and Dev2 have the parameters of the elliptic curve pre-installed, including the Generator Point (G), and an integer associated with the curve (n).
2. Dev1 will generate a random private key (d_1) in the range [1, n−1] by taking a point on the curve and computes a public key (Q_1); $Q_1=d_1\times G$.
3. Dev2 will do the same and generate its public key (Q_2) from its private key (d_2): $Q_2=d_2\times G$.
4. Dev1 and Dev2 exchange their public keys.
5. Dev1 will then use Dev2's public key and its private key to calculate the shared key (sharekey_dev1): sharekey_dev1 = $d_1\times Q_2 = d_1\times d_2\times G$.
6. Dev2 will then use Dev1’s public key and its private key to determine the shared key (sharekey_dev2): sharekey_dev2 = $d_2\times Q_1 = d_2\times d_1\times G$.

Since d_2 and d_1 are integers, $d_1 \times d_2 = d_2 \times d_1$, hence, for the properties of the elliptic curves, sharekey_dev1 = sharekey_dev2 = Shared Key (S).

As a result, dev1 and dev2 now share a secret key, S. The shared secret S is used as the link key to secure every end-to-end communication between two devices Dev1 and Dev2 in the network.

In Figure 5.1, we show the updated join procedure with ECDH. The coordinator and the joining device exchange values Q_1 and Q_2 during the joining process. These values can be included in the association request and response commands, eliminating the need for any additional messages in the joining operation. After acquiring each other's values, the shared secret key S is generated, allowing the coordinator to send the NWK symmetrically in the transport key message encrypted by S. Keeping the NWK inaccessible to eavesdroppers.



Coordinator
Device
Associatrion Request (with ECDH public key)
Association Response (with ECDH public key)
APS Transport Key (NWK)
Device Announce (Broadcast)
Node Descriptor Request
Node Descriptor Response

Figure 5.1: New Join Procedure with ECDH

An attacker who can listen in on the exchange cannot obtain the value of the key S and hence cannot extract the NWK. Since the key S is adopted as the link key in the proposed model, each ZC, ZR, or ZED has to generate a unique encryption key for each connection they establish. This technique has the following advantages:

- Protection from attackers with the TCLK.
- Inability of an opponent with the NWK to compromise devices due to messages being encrypted with link keys.
- Devices being unable to eavesdrop on conversations between other devices in the network.

5.3. Summary

The use of ECDH is proposed to be a valuable tool in enhancing the security of Zigbee networks. By modifying the join procedure to encrypt the network key with the security generated through ECDH, the network will be able to effectively protect itself from potential attackers. This method of key agreement offers secure, efficient, and easy real-time negotiation of a shared secret between two devices, allowing for a robust and secure connection. It can be concluded that the ECDH algorithm provides a reliable solution for improving the security of Zigbee networks, and its implementation is demonstrated in the next chapter.

CHAPTER 6 IMPLEMENTATION

6.1. Introduction

In this section, we go over how the suggested security enhancement was included in the Zigbee protocol. The objective is to analyze network connectivity with the security solution in place to confirm that devices will continue to function and communicate normally. Also, examining how the ZC handles communication across multiple nodes in the network can help us better understand how to make the network more effective.

6.2. Development Environment

The Zigbee protocol was developed by the Connectivity Standards Alliance (CSA). Several firms are involved in the organization, and the CSA certifies Zigbee-certified products, ensuring that each Zigbee stack supplied by an outside company complies with the standards and protocol. The alliance has certified a variety of products, ranging from commercial lights to development boards with microcontroller units that manage the Zigbee stack. The following are the hardware and software tools utilized in this experiment:

6.2.1. Hardware tools

This thesis uses the B85 Development Board and B85 Dongle, development kits based on Telink Semiconductor's TLSR8258 SoC, for the implementation of Zigbee and security analysis. See Figures 6.1 and 6.2. The TLSR8258 is an IEEE 802.15.4 multi-standard wireless SoC solution created by Telink that combines the features and capabilities required for all 2.4 GHz IoT standards into a single SoC. The chip supports the 2.4 GHz proprietary standard, Bluetooth Low Energy (up to Bluetooth 5), BLE Mesh, 6LoWPAN, Zigbee, RF4CE, HomeKit, and ANT (Telink Semiconductor, 2020). Appendix A contains a functional diagram of the TLSR8258 SoC.

The target board's firmware was downloaded using an adapter that was built around a Telink **TLSR8266-based burning EVK** shown in Figure 6.3. It uses a mini USB interface to connect to the PC and a serial wire debug interface to communicate with the target board's microcontroller.

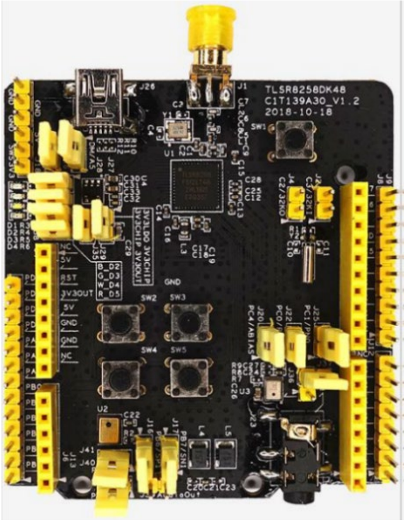


Figure 6.1: B85 Board

(Bin Yang, n.d.)

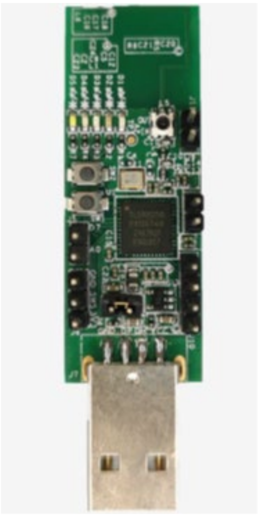


Figure 6.2: B85 Dongle

(Bin Yang, n.d.)

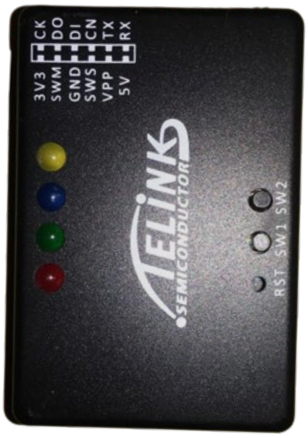
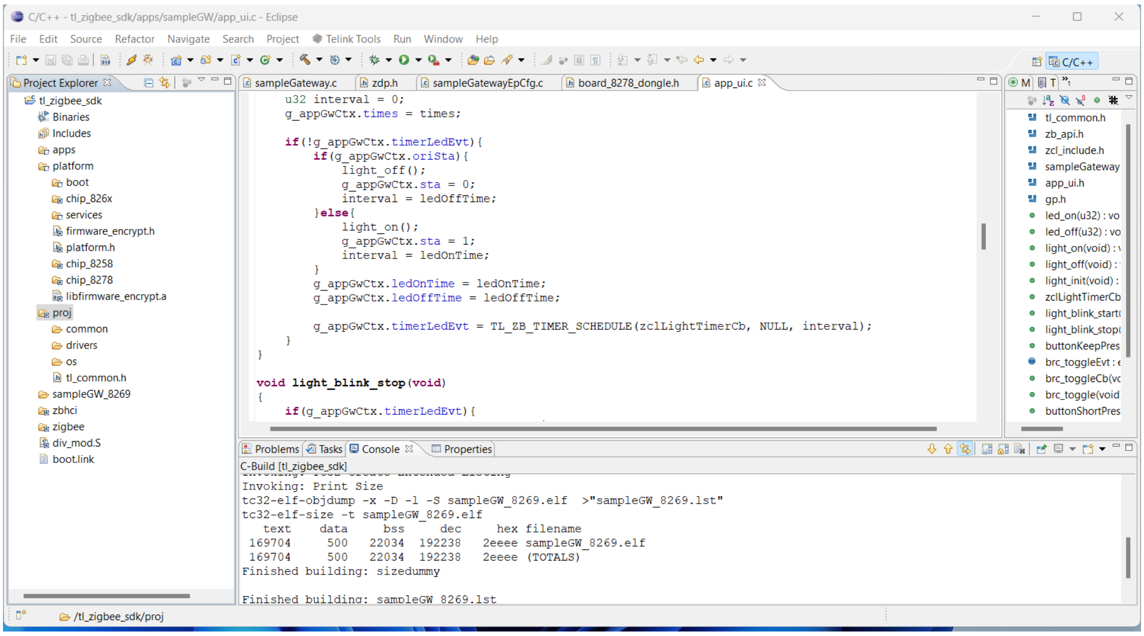


Figure 6.3: B85 Board

(Bin Yang, n.d.)

6.2.2. Software tools

The development boards are programmed and debugged using **Telink IDE for TC32**, an Eclipse-based Integrated Development Environment (IDE).

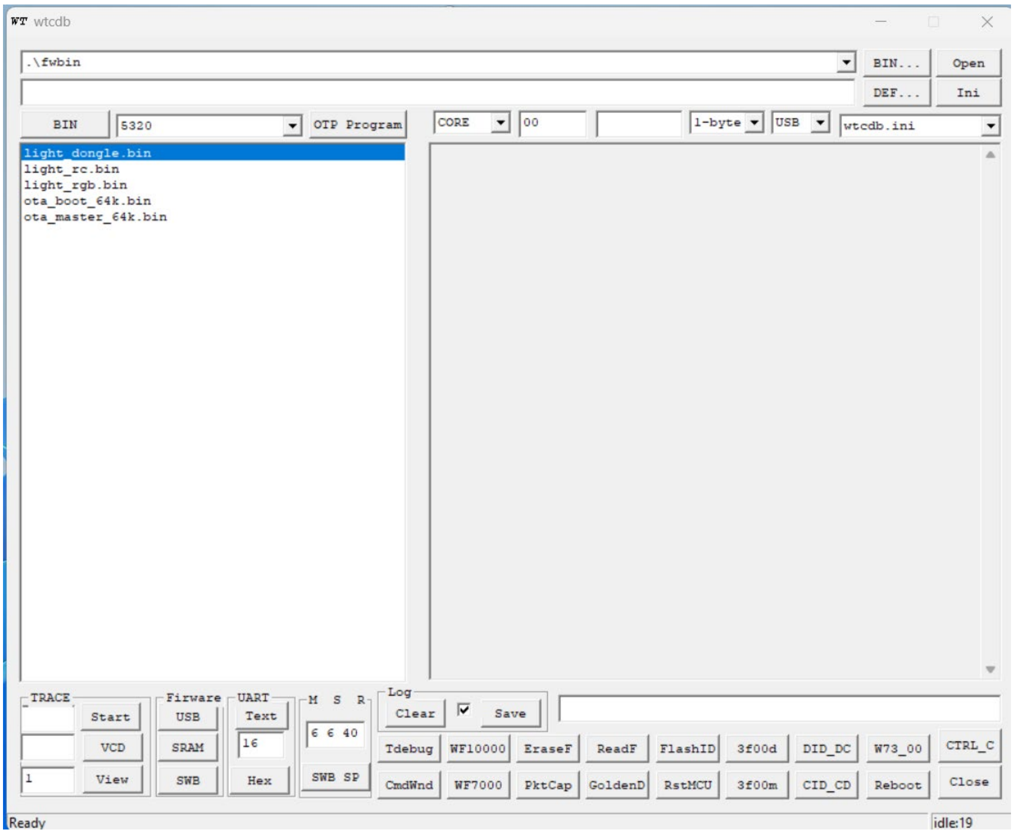


```
    u32 interval = 0;
    g_appGwCtx.times = times;

    if(!g_appGwCtx.timerLedEvt){
        if(g_appGwCtx.oriSta){
            light_off();
            g_appGwCtx.sta = 0;
            interval = ledOffTime;
        }else{
            light_on();
            g_appGwCtx.sta = 1;
            interval = ledOnTime;
        }
        g_appGwCtx.ledOnTime = ledOnTime;
        g_appGwCtx.ledOffTime = ledOffTime;

        g_appGwCtx.timerLedEvt = TL_ZB_TIMER_SCHEDULE(zclLightTimerCb, NULL, interval);
    }
}

void light_blink_stop(void)
{
    if(g_appGwCtx.timerLedEvt){
```

```
Invoking: Print Size
tc32-elf-objdump -x -D -l -S sampleGW_8269.elf  >"sampleGW_8269.lst"
tc32-elf-size -t sampleGW_8269.elf
   text    data     bss     dec     hex filename
 169704     500   22034  192238   2eeee sampleGW_8269.elf
 169704     500   22034  192238   2eeee (TOTALS)
Finished building: sizedummy

Finished building: sampleGW_8269.lst
```

Figure 6.4: Telink IDE

Programming of the boards was done using the **Telink Burning and Debugging Tool (BDT)**. With the BDT, Flash can be erased, firmware can be downloaded, the MCU can be activated when communication is lost, and other functions such as accessing memory areas like FLASH, CORE, ANALOG, and OTP are all possible with BDT. It can also be used to read/write global variables and view the USB log (Bin Yang, n.d.).



```
.\fwbin
5320
CORE  00  1-byte  USB  wtcdb.ini
light_dongle.bin
light_rc.bin
light_rgb.bin
ota_boot_64k.bin
ota_master_64k.bin
16
6 6 40
1
Ready                                              idle:19
```

Figure 6.5: Telink Burning and Debugging Tool

Zigbee Gateway Controller is a PC-based application that connects to the Zigbee Coordinator, used to issue commands to ZC.

6.3. Experiments

The experimental techniques used to implement the suggested solution are covered in this section. The experiment's and solutions' firmware was created using the Telink IDE and the Telink Zigbee SDK was installed. The Zigbee stack is a component of the Telink Zigbee SDK, a collection of Zigbee protocol stacks created in accordance with the Zigbee PRO specification. It complies with the Zigbee 3.0 application specification, has been certified by the Alliance platform Zigbee Pro R21, and supports Pro R22 (Telink Semiconductor, 2021). The Zigbee stack is modified to satisfy the requirements and reach the objective of network security. The code was written to work in accordance with the proposed solution. Flowcharts are used to illustrate the procedure for the firmware functionality. The network's respective APS layer of Zigbee includes all security and network-related solutions; for more information, see Figure 2.1 in Section 2.3. In Table 6.1, all the types of equipment required for the experiments are listed.

Table 6.1: Required Components

Component	Description
B58 Development Board/B58 USB Dongle	Development board configured as ZC, ZR or ZED
Telink Burning EVK	Telink Burning Board
Telink IDE for TC32	Integrated development environment used to write the firmware uploaded to
Telink Burning and Debugging Tools	Download debugging tool
TL OTA Tool	OTA code conversion tool
TI Packet Sniffer	Auxiliary tool to capture and analyze packet
tl_zigbee_sdk_v3	Software development kit for TLSR8258 SoCs. Consist of examples, drivers
micro-ecc	ECDH and ECDSA C library for 8-bit, 32-bit, and 64-bit processors developed by (site).
Zigbee_gateway_controller.exe	Auxiliary control software on PC (ZGC)

6.4. Application of ECDH Security

A B85 EVK board configured as ZC, one B85 USB dongle configured as ZR, and another dongle configured as ZED. Secp128r1, a 128-bit prime field Weierstrass curve, was used for the ECDH protocol's public-key algorithm solution. The choice was made because it is one of the curves supported by the ECDH library employed and can generate 128-bit key lengths that can fit in the Zigbee packet payload. The ECDH C library used is the micro-ecc, a small and fast ECDH and ECDSA implementation written in C and suitable for 8, 32, and 64-bit architectures. This library supports five standards; Secp128r1, secp192r1, secp224r1, secp256r1, and secp256k1 curves, and is resistant to known side-channel attacks (Ken MacKay, 2023).

In this proposed approach, I use the ECDH to generate a secret that is used as a link key. The elliptic curve parameters—the generator point G and n, an integer associated with the curve—are stored in the ZC and ZR. The ECDH drive provided by micro-ecc is used to generate public and private keys as well as compute the symmetric key from the generated public key and shared secret. The joining process flow for the ZC and ZR joining to form a network is depicted in Figures 6.7 and 6.8. The process begins with the ZC allowing devices to join its network. The ZR generates a private key d_r and computes a public key Q_r before sending the association request. The public key is then included in the association request payload. When the ZC receives an association request, it reads and stores Q_r, generates its private key d_c, and calculates the public key Q_c. The ZC then includes its public key as the payload when sending the association response. The ZR will read the coordinator's public key after receiving the association response. At this point, both devices compute the symmetric key S using the pre-defined secp128r1 curve with basepoint G, as well as the public and private keys. If the procedure is successful, they will have the same symmetric key. You can refer to Section 5.2.1 for more information on the ECDH key exchange.

The ZC coordinator then encrypts the NWK in the key transport command with the symmetric key S. Upon receiving the key transport command, because it has the same symmetric key as the coordinator, the router can decrypt the key transport payload and retrieve the NWK. The joining process continues, with the ZR broadcasting the device announce command and sending a node descriptor request. The ZC concludes the process by responding to the node descriptor request.

If the devices do not generate the same symmetric key, the router won't be able to decrypt the NWK, and other network devices are not going to process its device announce broadcast and node descriptor requests. As a result, the join is unsuccessful. The ZR will restart the process by looking for new networks to join. When a network is discovered, the router will send an association request and repeat the preceding process.

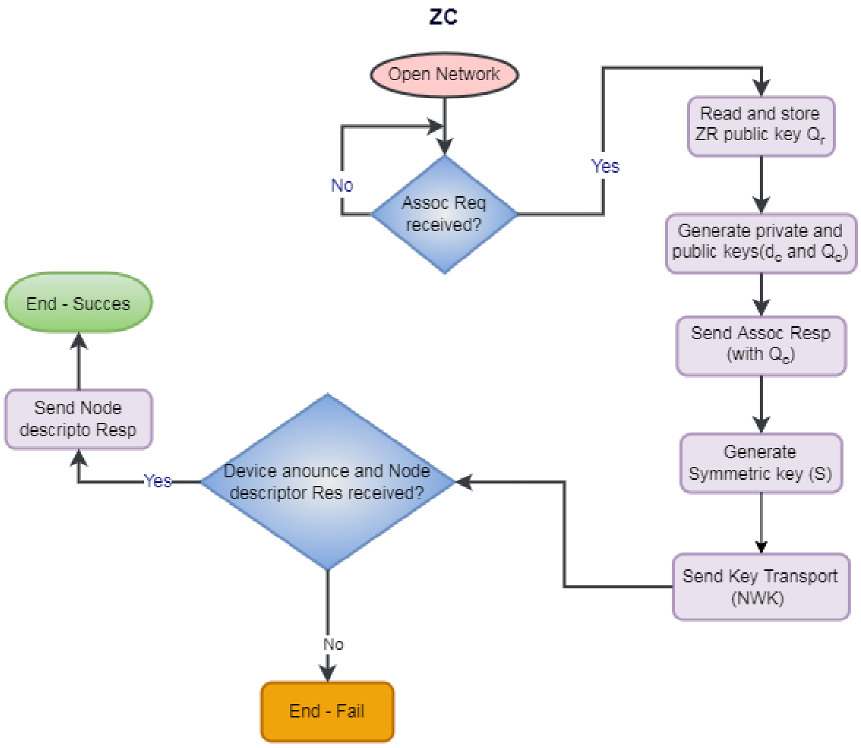

ZC
Open Network
No
Assoc Req received?
Yes
Read and store ZR public key Q_r
Generate private and public keys(d_c and Q_c)
Send Assoc Resp (with Q_c)
Generate Symmetric key (S)
Send Key Transport (NWK)
End - Succes
Send Node descripto Resp
Yes
Device anounce and Node descriptor Res received?
No
End - Fail

Figure 6.7: Zigbee Coordinator creating network

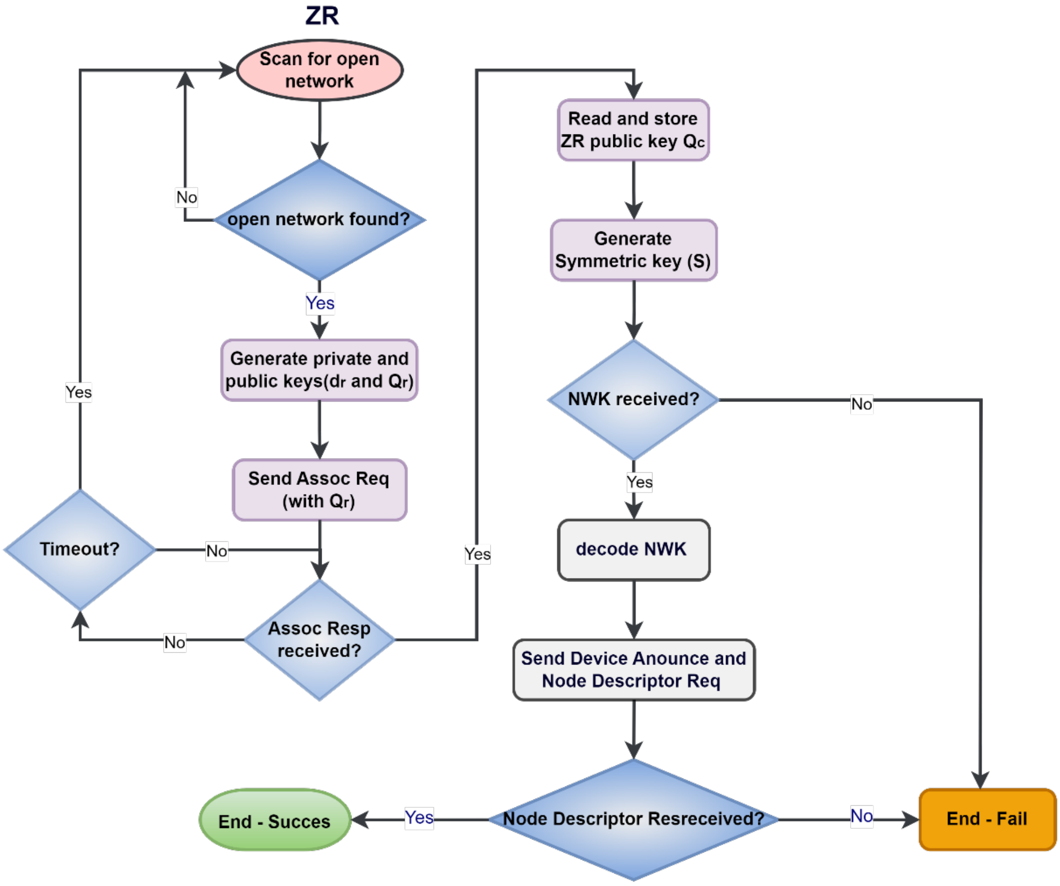

ZR
Scan for open network
No
open network found?
Yes
Generate private and public keys(dr and Qr)
Send Assoc Req (with Qr)
Yes
Timeout?
No
No
Assoc Resp received?
Yes
Read and store ZR public key Qc
Generate Symmetric key (S)
NWK received?
No
Yes
decode NWK
Send Device Anounce and Node Descriptor Req
End - Succes
Yes
Node Descriptor Resreceived?
No
End - Fail

Figure 6.8: Zigbee Router joining network

6.5. Method for Setting up a Zigbee Network

The Zigbee network was set up as a star network topology, which provides centralized security.

The boards were configured as follows to create nodes:

- B85 Dongle 1, as a Gateway (Coordinator)
- B85 Board, as a Light (Router)
- B85 Dongle 2, as a Switch (End Device)

6.5.1. Create network

When the SW1 button on the gateway board is pressed, it instructs the ZC to open or close the network. A green status LED will turn on to indicate that the network is open, at which point new devices can join the network. When it is turned off, the network is closed. When a gateway is powered on, if it is a new device, it will begin to build a network and allow devices to join. If it is a device that has previously established a network, the network will be restored, and the permit join state can be activated by pressing the button.

6.5.2. Join network

A red status LED is present on the B85 board, which is configured as a router. This red LED illuminates when the device is not connected to a network and turns off when it is. When the router is powered on as a new device with the gateway's network open, indicated by a green status LED, the network join will begin automatically. The red LED on the router will turn off after a successful join. If the device was previously connected to the network, the network will be restored. When the endpoint is turned on, a similar process occurs; if it is a new device, the network join is initiated automatically, and the green LED will blink after a successful join.

6.5.3. Functionality

When the end point's button (SW1) is pressed, a message is sent to the gateway. This endpoint serves as a switch. In turn, the gateway will send a toggle command to the router. In this arrangement, the router is a light node. When the router receives the toggle command, it will turn an LED on or off. This procedure is used to determine whether the proposed solution will allow Zigbee devices to function normally in the network.

6.6. Summary

This chapter has demonstrated the implementation of Elliptic Curve Diffie-Hellman (ECDH) in the Zigbee joining process. The tools required for the experiment were identified, and the setup was explained in detail, including the creation of a network comprising three nodes; a coordinator, a switch, and a light. The tools and setup used in this experiment can be replicated in future research to validate the effectiveness of ECDH in different IoT scenarios.

CHAPTER 7 RESULTS AND DISCUSSION

7.1. Introduction

This section covers the findings of the experiments for the suggested approach to improving Zigbee security in relation to the research objectives. The discussion in this section will lead to a conclusion, addressing the problem statement as well as the research questions to fulfill the goal of this thesis.

7.2. Computational Overhead

Only an ECDH secret key is included in both the end device's association request message and the coordinator's association response message in the proposed solution. The computation time for ECDH point multiplication has been determined to be approximately 275.8 milliseconds (refer to Appendix B for a detailed calculation). It should be noted that this time presupposes that both the coordinator and the end device can compute ECDH point multiplication at the same time, avoiding any overhead duplication associated with this operation. As a result, it takes an additional 275.8 milliseconds to establish a secure communication channel. It is essential to keep in mind, however, that these timings are dependent on specific hardware performance characteristics; for example, if a different system-on-a-chip (SoC) with either faster or slower performance is utilized, durations may deviate from these calculations accordingly.

7.3. Memory Overhead

In the new approach, extra memory space is needed for storing the constants and codes specifically related to ECDH. These constants take up 186 bytes while compiling the code adds another 5.6 KB (Table 5). The actual size of the Zigbee stack can vary depending on implementation details, resulting in varying memory usage. As an example, the total size of a light switch end device using TeLink's Zigbee stack exceeds 280 kB. Given this context, it is clear that the proposed method introduces very little memory overhead.

7.4. Communication Overhead

The proposed protocol sends ECDH public keys in the association request and response messages. This has a total communication overhead of 17 bytes, which falls within Zigbee's maximum payload of 127 bytes (Zigbee Alliance, 2017). This increased packet size provides a better solution compared to adding extra messages. Considering that Zigbee's standard data rate is 250 Kbit/sec and that more than 517 Bytes will be transmitted during the entire join process, the resulting communication overhead remains relatively minimal.

7.5. Energy Overhead

The typical procedure for joining Zigbee involves scanning for beacons and exchanging keys. The average energy consumption for a 5-volt battery-powered Zigbee device during this process is around 60 mW on Atmel mcu (Atmel, 2015). In the proposed solution, we have to perform two additional ECDH multiplications that consume around 0.57 mW, see Appendix C for how it was measured. This extra load accounts for less than 1% of the overall energy used during the join process and can be considered insignificant when compared to the long-lasting lithium CR2477 battery used in Zigbee devices. These batteries have a capacity of typically between 850 and 1000 mAh and usually operate continuously for three to seven years (batteryequivalents, n.d.). Again, The outcome relies on the network configuration as well as the frequency of radio transmission used by the node.

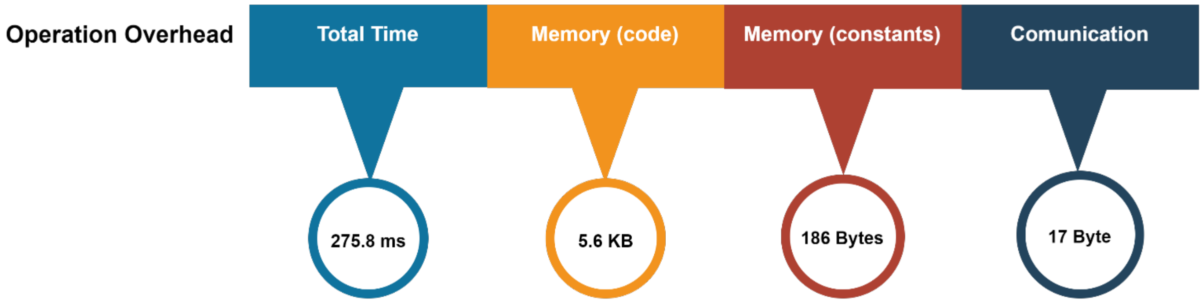


Figure 7.1 Total Operation Overhead

7.6. Comparison With Existing Proposals

In this section, I will assess the new protocol in contrast to the propositions explored in Section 4.4. I will evaluate various aspects, including:

- The additional messages needed for implementing the new protocol
- Whether a Certification Authority (CA) or Distributed Authority (DA) is necessary
- Shortcomings in preventing fake device injection
- Operation, memory, and communication overhead considerations of employing the proposed approach

7.6.1. Extra messages

In contrast to previous existing techniques, the solution presented in this work does not necessitate the use of additional messages. Consequently, compared to other methods, it has a lower communication overhead. Certain approaches, such as those described (Choi et al., 2012) and in (Tedeschi et al., 2020), add extra messages, which increases communication overhead and requires changes to the current protocol. For example, these references have 3 and 4 more messages than the current Zigbee protocol, respectively.

CA/DA requirement

The proposal presented in this paper offers a certificateless model, eliminating the need for a centralized CA or DA like (Misra et al., 2016) and (Tedeschi et al., 2020). Unlike protocols relying on CA/DA, using such models would significantly delay the joining process and fail to prevent unauthorized devices from entering the network. If an adversary succeeds in compromising either a device or the CA/DA system itself, they can easily become part of the network and acquire NWK access. Additionally, it can be difficult to set up a single authority for Zigbee networks with multiple vendors because of issues with mutual trust.

7.6.2. Operation, memory, and communication overhead

This solution offers improved efficiency in terms of operation, communication, code, and constant memory overheads. It outperforms the current solutions reviewed in this paper.

Additionally, its smaller memory footprint makes it a more cost-effective and practical choice for implementation.

7.7. Summary

The outcomes of the performance evaluation for securing the Zigbee join process were highlighted in this chapter. The findings demonstrated that, compared to existing proposals, the proposed solution exhibits reduced computational, memory, communication, and energy requirements. Moreover, a comparison was conducted between our proposal and other existing solutions based on factors such as message count, need for CA/DA, and operation, memory, and communication overheads. Contrary to previous suggestions, this suggested approach does not necessitate any supplementary messages. It also eliminates the requirement of a CA/DA. In addition to these advantages, our proposed solution further enhances operational efficiency while minimizing memory usage and communication demands when compared to other alternatives. All things considered, the findings of this performance evaluation show that the approach we suggest is effective and secure, making it a desirable choice for securing Zigbee networks by preventing unauthorized device access during network joining activities.

CHAPTER 8 CONCLUSION

Zigbee is a widely used wireless communication protocol in various fields, such as smart homes, industrial automation, and healthcare. However, previous studies have identified several security vulnerabilities in Zigbee networks. These include unauthorized access, message interception, and replay attacks. Such weaknesses pose significant risks to the confidentiality and integrity of data transmitted through Zigbee networks. This thesis aims to comprehensively examine these security issues while devising an efficient solution that effectively addresses them without imposing excessive costs or energy consumption on the system.

The implementation of the proposed solution was a success. It involved using ECDH to secure the Zigbee join process with the goal of creating a secure network that remained compatible with low-power, and low-cost devices. By modifying the join procedure to encrypt the network key through ECDH-generated security, an effective and efficient method for transporting keys was achieved. This showcased how utilizing public and symmetric key algorithms can greatly benefit Zigbee applications.

The findings of this thesis have implications for the security of Zigbee networks. ECDH can be used to secure the Zigbee join process, which is a critical step in ensuring overall network security. Implementing an ECDH solution to authenticate joining devices offers a reliable means of enhancing the security measures within Zigbee networks.

Future research could focus on how to detect malicious devices in the network and determine appropriate actions following a security breach.

Security is composed of various elements, among them the selection, management, and implementation of cryptographic algorithms and keys. Each element has its own advantages and disadvantages. For instance, a larger key increases security but also adds overhead and requires more resources on the device. The research questions were focused on assessing the strength of Zigbee security. This thesis revealed that the real weakness lies in key distribution and management. Both components—key distribution and management—are vital for establishing a secure network in any application seeking enhanced protection against threats. In this paper, it is concluded that the security of Zigbee can be improved. Additionally, a method to secure network key transmission has been proposed, which can be adopted on every device within the Zigbee WSN.

REFERENCES

Aikins-Bekoe, S. & Ben, J. 2020. Elliptic Curve Diffie-Hellman (ECDH) Analogy for Secured Wireless Sensor Networks. *International Journal of Computer Applications*, 176(10): 1–8.

Allakany, A., Saber, A., Mostafa, S.M., Alsabaan, M., Ibrahem, M.I. & Elwahsh, H. 2023. Enhancing Security in ZigBee Wireless Sensor Networks: A New Approach and Mutual Authentication Scheme for D2D Communication. *Sensors*, 23(12): 5703.

Anderson, T. 2020. Time to patch your lightbulb? Researchers demonstrate Philips Hue exploit. https://www.theregister.com/2020/02/05/time_to_patch_your_lightbulb_researchers_demonstrate_philips_hue_exploit/ 16 September 2022.

Atmel. 2015. AT03663: Power Consumption of ZigBee End Device.

batteryequivalents. Lithium CR2477 Battery - Equivalents and Replacements. https://www.batteryequivalents.com/lithium-cr2477-battery-equivalents-and-replacements.html.

Bin Yang. Telink Zigbee Overview. https://wiki.telink-semi.cn/tools_and_sdk/Demo/B91_Zigbee/Telink_Zigbee_Overview.pdf 5 November 2023.

Carla Tardi. 2022. What Is Moore's Law and Is It Still True? *Investopedia*. https://www.investopedia.com/terms/m/mooreslaw.asp 5 January 2023.

Choi, K., Yun, M., Chae, K. & Kim, M. 2012. An enhanced key management using ZigBee Pro for wireless sensor networks. In *The International Conference on Information Network 2012*. The International Conference on Information Network 2012. 399–403.

Connectivity Standards Alliance. 2021. Zigbee Certification Milestone. *CSA-IOT*. https://csa-iot.org/newsroom/zigbee-certification-milestone/ 18 September 2022.

Curryer, E. 2023. IoT in 2023 and beyond. *TechInformed*. https://techinformed.com/iot-in-2023-and-beyond/ 25 October 2023.

DFRobot. 2023. Comparison of Wireless Technologies: LoRaWAN and Zigbee , WiFi , NB-IoT - DFRobot. https://www.dfrobot.com/blog-1646.html 25 October 2023.

Digi. Zigbee Wireless Mesh Networking. https://www.digi.com/solutions/by-technology/zigbee-wireless-standard 22 September 2022.

Fan, X., Susan, F., Long, W. & Li, S. 2017. Security Analysis of Zigbee. https://courses.csail.mit.edu/6.857/2017/project/17.pdf.

Farha, F. & Chen, H. 2018. Mitigating replay attacks with ZigBee solutions. *Network Security*, 2018(1): 13–19.

Farha, F. & Ning, H. 2019. Enhanced Timestamp Scheme for Mitigating Replay Attacks in Secure ZigBee Networks. In *2019 IEEE International Conference on Smart Internet of Things (SmartIoT)*. 2019 IEEE International Conference on Smart Internet of Things (SmartIoT). 469–473.

Filippone, G. 2023. On the Discrete Logarithm Problem for elliptic curves over local fields. http://arxiv.org/abs/2304.14150 21 May 2023.

Flaherty, N. 2021. Zigbee Alliance changes name in Matter IoT launch eeNews Europe. *EENewsEurope*. https://www.eenewseurope.com/en/zigbee-alliance-changes-name-in-matter-iot-launch/ 18 September 2022.

Goodspeed, T. 2009. *Extracting Keys from Second Generation Zigbee Chips*. https://paper.seebug.org/papers/old_sebug_paper/Meeting-Documents/Blackhat-USA2009/BHUSA09-Goodspeed-ZigbeeChips-PAPER.pdf.

Hamza Kadhum. 2020. *Enhancing Zigbee Security for Industrial Implementation*. KTH ROYAL INSTITUTE OF TECHNOLOGY. http://kth.diva-portal.org/smash/get/diva2:1460818/FULLTEXT01.pdf.

Ivezic, M. 2019. Zigbee Security 101 (Non-5G IoT Connectivity Options). *5G Security by Marin Ivezic*. https://5g.security/5g-edge-miot-cybersecurity/zigbee-security-overview/ 27 September 2022.

Jamieson, P. 2016. Base Device Behavior Specification. https://zigbeealliance.org/wp-content/uploads/2019/12/docs-13-0402-13-00zi-Base-Device-Behavior-Specification-2-1.pdf.

Khanji, S., Iqbal, F. & Hung, P.C.K. 2019. ZigBee Security Vulnerabilities: Exploration and Evaluating. In *2019 10th International Conference on Information and Communication Systems (ICICS)*. 2019 10th International Conference on Information and Communication Systems (ICICS). Irbid, Jordan: IEEE: 52–57. https://ieeexplore.ieee.org/document/8809115/ 16 September 2022.

Misra, S., Goswami, S., Taneja, C. & Mukherjee, A. 2016. Design and implementation analysis of a public key infrastructure-enabled security framework for ZigBee sensor networks: PKI-ENABLED SECURITY FRAMEWORK FOR ZIGBEE SENSOR NETWORKS. *International Journal of Communication Systems*, 29(13): 1992–2014.

Morgner, P., Mattejat, S., Benenson, Z., Müller, C. & Armknecht, F. 2017. Insecure to the touch: attacking ZigBee 3.0 via touchlink commissioning. In *Proceedings of the 10th ACM Conference on Security and Privacy in Wireless and Mobile Networks*. WiSec '17: 10th ACM Conference on Security & Privacy in Wireless and Mobile Networks. Boston Massachusetts: ACM: 230–240. https://dl.acm.org/doi/10.1145/3098243.3098254 24 April 2023.

Olawumi, O., Haataja, K., Asikainen, M., Vidgren, N. & Toivanen, P. 2014. Three practical attacks against ZigBee security: Attack scenario definitions, practical experiments, countermeasures, and lessons learned. In *2014 14th International Conference on Hybrid Intelligent Systems*. 2014 14th International Conference on Hybrid Intelligent Systems (HIS). Kuwait, Kuwait: IEEE: 199–206. http://ieeexplore.ieee.org/document/7086198/ 16 September 2022.

Paar, C. & Pelzl, J. 2010. *Understanding cryptography: a textbook for students and practitioners*. Heidelberg ; New York: Springer.

PRNewswire. 2023. Zigbee PRO 2023 Improves Overall Security While Simplifying Experience. https://www.prnewswire.com/news-releases/zigbee-pro-2023-improves-overall-security-while-simplifying-experience-301795113.html 25 October 2023.

Puneet. 2020. What is RSA? How does an RSA work? *Encryption Consulting*. https://www.encryptionconsulting.com/education-center/what-is-rsa/ 5 January 2023.

Rana, S.M., Hoque, M.R. & Kabir, M.H. 2018. Evaluation of Security Threat of ZigBee Protocol to Enhance the Security of ZigBee based IoT Platform. , 11(1): 37–35.

Razouk, W., Crosby, G.V. & Sekkaki, A. 2014. New Security Approach for ZigBee Weaknesses. *Procedia Computer Science*, 37: 376–381.

Ronen, E., Shamir, A., Weingarten, A.-O. & O'Flynn, C. 2017. IoT Goes Nuclear: Creating a ZigBee Chain Reaction. In *2017 IEEE Symposium on Security and Privacy (SP)*. 2017 IEEE Symposium on Security and Privacy (SP). 195–212.

Saha, P. 2021. Everything You Need To Know About Diffie-Hellman Key Exchange Vs. RSA. *Encryption Consulting*. https://www.encryptionconsulting.com/diffie-hellman-key-exchange-vs-rsa/ 5 January 2023.

Sarkar, A., Chatterjee, S.R. & Chakraborty, M. 2021. Role of Cryptography in Network Security. In M. Chakraborty, M. Singh, V. E. Balas, & I. Mukhopadhyay, eds. *The 'Essence' of Network Security: An End-to-End Panorama*. Lecture Notes in Networks and Systems. Singapore: Springer Singapore: 103–143. http://link.springer.com/10.1007/978-981-15-9317-8_5 8 January 2023.

Silicon Labs. 2022a. AN1089: Using Installation Codes with Zigbee Devices. https://www.silabs.com/documents/public/application-notes/an1089-using-installation-codes-with-zigbee-devices.pdf.

Silicon Labs. 2022b. AN1233: Zigbee Security. https://www.silabs.com/documents/public/application-notes/an1233-zigbee-security.pdf.

Stallings, W. 2006. *Cryptography and network security: principles and practice*. 4th ed. Upper Saddle River, N.J: Pearson/Prentice Hall.

TechTarget. 2021. What is a passive attack? *WhatIs.com*. https://www.techtarget.com/whatis/definition/passive-attack 5 January 2023.

Tedeschi, P., Sciancalepore, S., Eliyan, A. & Di Pietro, R. 2020. LiKe: Lightweight Certificateless Key Agreement for Secure IoT Communications. *IEEE Internet of Things Journal*, 7(1): 621–638.

Telink Semiconductor. 2021. Application Note: Telink Zigbee SDK Development Manual.

Telink Semiconductor. 2020. Datasheet for Telink BLE + IEEE802.15.4 Multi-Standard Wireless SoC TLSR8258.

Thakkar, J. 2020. Types of Encryption: What to Know About Symmetric vs Asymmetric Encryption. *InfoSec Insights*. https://sectigostore.com/blog/types-of-encryption-what-to-know-about-symmetric-vs-asymmetric-encryption/ 8 October 2023.

Van Leeuwen, D. & Ayuk, L.T. 2019. *Security testing of the Zigbee communication protocol in consumer grade IoT devices*. http://urn.kb.se/resolve?urn=urn:nbn:se:hh:diva-40189 16 September 2022.

Wang, X. & Hao, S. 2022. Don't Kick Over the Beehive: Attacks and Security Analysis on Zigbee. In *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security*. CCS '22: 2022 ACM SIGSAC Conference on Computer and Communications Security. Los Angeles CA USA: ACM.

Xiao, Y., Shen, X. & Du, D. 2007. *Wireless Network Security*. Online-Ausg. Boston, MA: Springer Science+Business Media, LLC.

Zigbee Alliance. 2017. Zigbee Specification. https://csa-iot.org/wp-content/uploads/2022/01/docs-05-3474-22-0csg-zigbee-specification-1.pdf.

APPENDICES

Appendix A

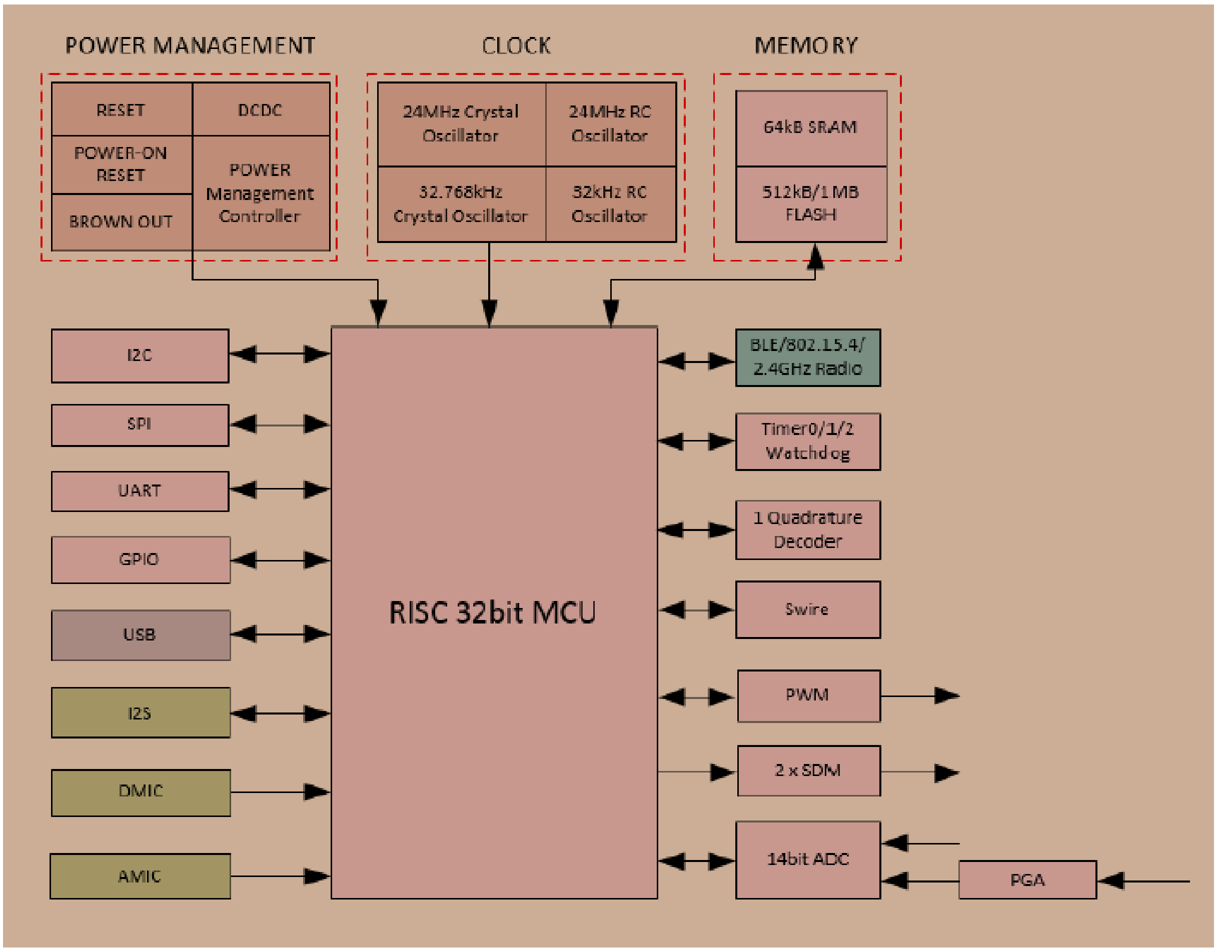


Figure A.1: Block diagram for TLSR8258 SoC

(Telink Semiconductor, 2020)

Appendix B

To measure the computation time for the ECDH (Elliptic Curve Diffie-Hellman) point multiplication, I employed a hardware timer, and here's a detailed explanation of the steps involved:

Step 1: Configuration and Initialization

Timer0 of the TLSR8261 MCU was configured into mode 3 (Tick mode) (Telink Semiconductor, 2020) to measure the time in high-resolution increments

The timer registers were initialized and set to appropriate prescaler values to achieve a resolution of 1 microsecond. The initial Tick value of Timer0 is zero, via address 0x630 ~ 0x633. Address 0x630 is the lowest byte, and address 0x633 is the highest byte.

Step 2: Start the timer.

Enable Timer0 immediately before the ECDH Point Multiplication begins. The timer tick (i.e., counting value) is increased by 1 on each positive edge of the system clock from the preset initial Tick value.

Step 3:

Measure the computation time by capturing the timer value at the end of the point multiplication. The value was measured to be 275 804. Since the time started counting from zero, this is the number of ticks it took the ECDH point multiplication to complete.

Step 4: Calculate the Computation time.

The computation time is obtained by multiplying the tick count with the timer resolution. That is, $\frac{1}{1000000} \times 275804 \approx 275{,}8$ms

Therefore, the computation time for the ECDH (Elliptic Curve Diffie-Hellman) point multiplication is 275.8 milliseconds.

Appendix C

In this experiment, the software is programmed to execute a loop of ECDH point calculations. The goal is to determine the amount of energy used during one calculation of an ECDH point. As shown in Figure A.2, we connect the oscilloscope probe to the shunt resistor which measures the potential difference generated by the current flowing through it.

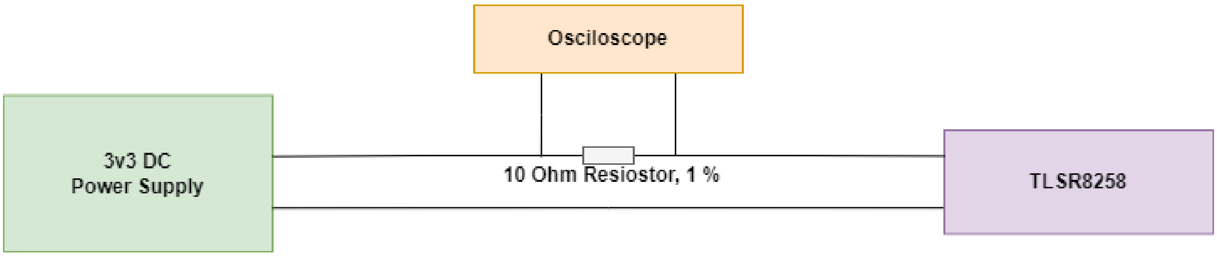
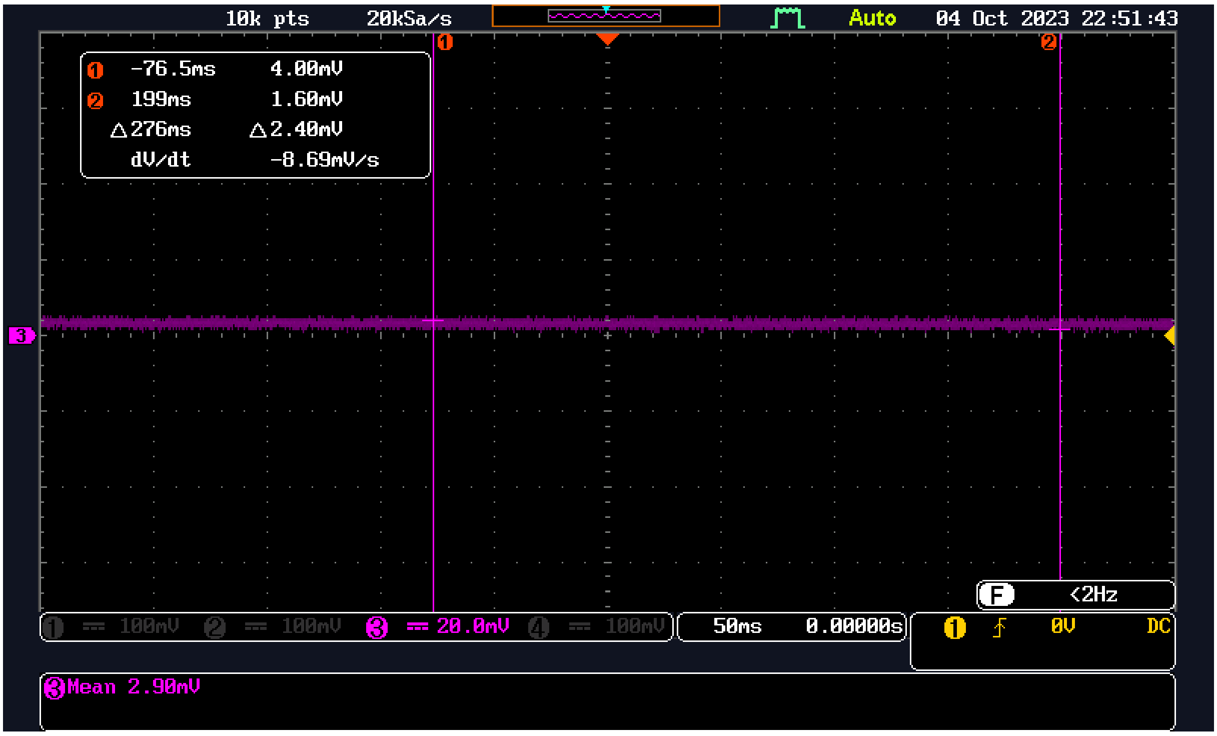


Figure A.2: Oscilloscope connection

The resistance of the shunt resistor used is known, and the potential difference across it is measured. By applying Ohm's law, Current equals voltage drop divided by the shunt resistor value.



```
10k pts   20kSa/s   Auto   04 Oct 2023 22:51:43

1   -76.5ms    4.00mV
2   199ms      1.60mV
    △276ms     △2.40mV
    dV/dt      -8.69mV/s

F  <2Hz
1 == 100mV  2 == 100mV  3 == 20.0mV  4 == 100mV   50ms  0.00000s   1  ⨍  0V  DC
3 Mean 2.90mV
```

Figure A.3: Oscilloscope voltage measurement

This is equivalent to the current flowing through the circuit. Based on these values, we can calculate the power and energy consumption.

The mean voltage refers to the average voltage level during the oscilloscope's acquisition period. It is determined by calculating the average value of all voltage samples between two cursor points (marked 1 and 2), as illustrated in Figure A.3. The measured mean voltage is 2.9 mV.

The total period is the time difference between two cursor points. It represents the interval from start to finish of the ECDH algorithm and was measured to be 275.8 milliseconds; additional information on how it was measured can be found in Appendix B.

The calculations are as below:

Average Current = Average Voltage / Resistance

= 0,0029/10

= 0,00029 Amps

= 0.29 mA.

Total Current Consumption [mA.ms] = Average Current x Total Period

= 0,29 * 275,8

= 79,982 (mA.ms)

Power = Voltage * Current

= 3 * 0,00029

= 0,00087 wats

= **0,87 mW**

Energy used = power * time *where time = 275,8 ms = 0,00007661 hours*

= 0,00087 * 0,00007661

= 0,0000000666507

= 66,6507 nWh

Thus, 66,6507nW is the energy consumed in a single ECDH point calculation.