An Assessment of Prevalent Security Issues on ZigBee 3.0 Networks

by

James W. Cato

A thesis submitted in partial fulfilment of the requirements for the degree of

Master of Information Security and Digital Forensics

Faculty of Design and Creative Technologies

School of Engineering, Computer and Mathematical Sciences

Auckland University of Technology

January 2022

Declaration

I hereby declare that this submission is my own work and that, to the best of my knowledge and belief, it contains no material previously published or written by another person (except where explicitly defined in the acknowledgements), nor material which to a substantial extent has been submitted for the award of any other degree or diploma of a university or other institution of higher learning.

…….

James W. Cato

31/01/2022

Abstract

ZigBee is a wireless technology standard for connecting Internet of Things (IoT) devices based on the IEEE 802.15.4 specification. Similarly to other IoT protocols, ZigBee faces numerous security issues that threaten the confidentiality, integrity and availability of its networks and services. ZigBee is implemented with a variant of the 128-bit Advanced Encryption Standard with symmetric keys for node authentication and data confidentiality. However, ZigBee's technology incorporates certain constraints, such as low cost and low power into its design, which has allowed certain security issues to persist across the protocol revisions over the years. These constraints raise concerns because ZigBee is often deployed in data-sensitive applications.

Although previous studies have addressed the main security issues found in the earlier protocol revisions, limited studies have been conducted on the latest 'ZigBee 3.0' standard. Therefore, this research contributes to addressing this research gap by investigating the impact of the identified and prevalent security issues against ZigBee 3.0 networks. Three core issues were investigated in this study based on the findings in the related literature: (a) 'Security of Symmetric Keys', which relates to how an attacker could obtain ZigBee's symmetric keys through exploiting known vulnerabilities and whether the implemented security mechanisms are sufficient to protect the keys; (b) 'Compromised Symmetric Keys', which concerns the breach against a network's confidentiality if one or more of its symmetric keys have been exposed by an attacker; and (c) 'Insufficient Denial of Service Protection Mechanisms', which enables the protocol to be susceptible to specific denial of service attacks.

The research was conducted as a practical undertaking against real ZigBee 3.0 networks comprising XBee 3 radio modules and ZigBee-compatible hardware. Attacks associated with each issue were performed to determine their impact, and where necessary, both security models provided by ZigBee 3.0 were evaluated separately. In addition, the study outlined the

security controls within the device's configuration, as well as best practices that can be applied to address or mitigate the attacks considered in this study and strengthen the network's security over symmetric keys. The compiled results revealed that certain attacks under each investigated security issue continue to affect the confidentiality or availability of ZigBee 3.0 networks. However, the enhancements made to the protocol's security controls combat the elements of each security issue, reducing their overall impact compared with its earlier revisions.

Acknowledgements

First and foremost, I would like to extend my sincere gratitude to my supervisor, Dr Akbar Ghobakhlou, for his support, guidance and insight throughout this project. His knowledge and expertise greatly assisted me in the preparation of this thesis. Acknowledgement must be made to my secondary supervisor, Dr Hakilo Sabit, for his feedback and assistance in sourcing equipment for this research. I would like to give thanks to my parents, Adam and Julie Cato, for the ongoing encouragement and support throughout my education. Finally, I am thankful to my loving wife, Jieun Yu, for all the patience and support while I completed my studies.

Table of Contents

List of Figures

List of Tables

List of Abbreviations

ACK	—	Acknowledgement
ACL	—	Access Control List
AES	—	Advanced Encryption Standard
API	—	Application Programming Interface
APL	—	Application
APS	—	Application Support Sublayer
ASCII	—	American Standard Code for Information Interchange
ASH	—	Auxiliary Security Header
AT	—	Attention
CBC-MAC	—	Cipher Block Chaining Message Authentication Code
CCM*	—	Counter with CBC-MAC (Extended)
CSM	—	Centralised Security Model
CSMA	—	Carrier Sense Multiple Access/Collison Avoidance
DDoS	—	Distributed Denial of Service
DoS	—	Denial of Service
DSM	—	Distributed Security Model
EO	—	Encryption Options
GTS	—	Guaranteed Timeslot
IEEE	—	Institute of Electrical and Electronics Engineers
IoT	—	Internet of Things
IV	—	Initial Vector
KY	—	Link Key
LR-WPAN	—	Low-Rate Wireless Personal Area Network
MAC	—	Media Access Control

MIC	—	Message Integrity Code
NIB	—	Network Layer Information Base
NIST	—	National Institute of Standards and Technology
NK	—	Network Key
NWK	—	Network
OTA	—	Over the Air
PAN	—	Personal Area Network
PAN-ID	—	Personal Area Network Identifier
PCAP	—	Packet Capture
PHY	—	Physical
PIB	—	PAN Information Base
SRP	—	Secure Remote Password
SSP	—	Security Service Provider
WPAN	—	Wireless Personal Area Network
WSN	—	Wireless Sensor Networks
ZDO	—	ZigBee Device Object

Chapter 1: Introduction

1.1 Background and Motivation

The Internet of Things (IoT) is a fast-growing, increasingly popular technological domain. ZigBee, part of this revolution, provides a standard for wireless personal area networks (WPANs) to enable connectivity between a wide range of IoT devices through its supported mesh, star and tree topologies. The standard is built upon the IEEE 802.15.4 specification. Its data rates, power consumption and cost are low in order to accommodate a full range of devices, including battery-operated wireless sensor nodes with potentially years of battery life (Dini & Tiloca, 2010). ZigBee currently holds a moderately competitive market share for its smart home, industrial and healthcare applications (Mordor Intelligence, 2020).

The demand for ZigBee's smart home applications has been steadily increasing, and its protocol has become a primary standard used in home automation. Major manufacturers, notably Amazon, Samsung and Phillips, have developed household appliances with the ZigBee protocol for lights, thermostats, door locks, motion sensors and alarms, which can be remotely monitored and controlled by devices on the internet (Carlsen, 2021). Within ZigBee's other domains, the protocol is used for wireless sensor nodes (WSNs) to monitor and collect data from their environmental surroundings (Matin & Islam, 2012).

ZigBee has various security controls to comply with the security requirements of its applications. For example, it monitors and collects patient data in healthcare and personal data in smart homes. Therefore, it often deals with sensitive and private data (Zillner & Strobl, 2015). While it is evident that data confidentiality is a concern, measures are also required to protect the integrity and authentication of its networks. ZigBee achieves these requirements through its security architecture that employees a simplified version of the 128-bit Advanced Encryption Standard (AES) for encryptions and the built-in elements of the IEEE 802.15.4 standard. These provide security services at each protocol stack layer via symmetric key

cryptography (X. Fan, Susan, Long, & Li, 2017). ZigBee's representative security services include data confidentiality, authentication and integrity; device authentication; and replay protection (Rudresh, 2017b).

Since ZigBee's first public release in 2005, it has gained unwanted attention around its security issues that are predominately enabled by its technology's low-cost, low-power design. As a result of its low-cost and low-power trade-offs, the ZigBee protocol is vulnerable to various network attacks that threaten its networks' and services' confidentiality, integrity and availability (Zillner & Strobl, 2015). Researchers over the years have expressed their concerns and have conducted practical experiments to understand these security issues further. Notably, in a Black Hat conference in 2015, Zillner and Strobl (2015) demonstrated the exploitation of several vulnerabilities and highlighted weaknesses they found in ZigBee systems. Additional research has been performed to create frameworks designed for exploiting ZigBee networks. In particular, Wright (2009) authored the KillerBee framework, which contains an arsenal of python-based attack scripts to exploit and sniff ZigBee networks.

Although efforts have been made to address security issues through ZigBee's protocol iterations over the years, specific issues are challenging to address owing to factors inherent in IoT devices and the compliance with ZigBee's low-cost, low-power design (Zillner & Strobl, 2015). This issue motivates the current study to determine ZigBee's current status in terms of its security issues and the likely impact of these issues on its networks.

The latest version of ZigBee is 'ZigBee 3.0', which was publicly released in 2016. ZigBee 3.0 improves on several aspects from the earlier 'ZigBee PRO' release and contains additional security services and reinforcements for its existing mechanisms (Texas Instruments, 2019). Therefore, this research is oriented towards the ZigBee 3.0 protocol and its stance against prevalent security issues.

1.2 Aims and Objectives

This thesis aims to conduct an up-to-date assessment of the ZigBee 3.0 protocol against security issues prevalent in the earlier protocol revisions. It aims to assess the impact of their associated attacks, and overall, ZigBee's current stance against these security issues. These are achieved through the following objectives:

- Survey prevalent security issues in ZigBee Systems.
- Construct and deploy appropriate testbed ZigBee 3.0 networks for evaluation.
- Perform practical attack experiments by exploiting weaknesses or concerns associated with each security issue and determine/measure their impact.
- Identify security measures to address or mitigate specific demonstrated attacks and strengthen the security of symmetric keys in ZigBee 3.0 networks.
- Combine results to determine the overall impact of each assessed security issue.

1.3 Thesis Structure

The thesis consists of six chapters. First, this chapter introduced and outlined the background and motivation behind the research topic. Second, it presented an overview of the aims and objectives that are the focus of this study.

Chapter 2 presents a literature review that builds a body of knowledge on the ZigBee protocol and its security concepts. It introduces WSNs and IEEE 802.15.4 technology concepts and provides an overview of ZigBee's technological workings. The main security components of ZigBee are thoroughly analysed, as well as how the protocol upholds security and its known security issues and weaknesses. The chapter identifies and discusses the security advancements made to the ZigBee 3.0 protocol. Last, it summarises five related studies in which researchers have analysed the ZigBee protocol against its main security issues through practical undertakings.

Chapter 3 outlines the research design and methodology. It identifies a research question and five supporting sub-questions that are formulated from the literature review. The research phases are outlined, describing the physical security testing approach for investigating ZigBee 3.0 and gathering the necessary data to answer the research questions.

Chapter 4 presents the findings and results gathered from the executed research approach. These relate to the impact inflicted against the testbed ZigBee 3.0 networks resulting from each practical attack associated with the security issues under analysis. Additional findings present security controls that can be applied to the device's security configuration to address or mitigate certain attacks and strengthen the security over symmetric keys.

Chapter 5 further discusses and analyses the findings and results gathered in Chapter 4. Each sub-question is answered based on the results. The answers for the sub-questions are ultimately combined to answer the primary research question.

Chapter 6 concludes the thesis. The research is summarised, and the adopted approach's potential limitations are identified and discussed. Last, it discusses future research that could be conducted to continue this research.

Chapter 2: Literature Review

2.1 Introduction

This chapter provides a literature review intended to build a body of knowledge on the ZigBee protocol and its security concepts. The literature review aims to survey the related literature to grasp the technical workings of the ZigBee protocol, its security features and the security issues that have been identified in the specification over the years.

The literature review is divided into six main sections. In the first two sections, 2.2 and 2.3, the concepts of WSNs and the IEEE 802.15.4 standard are introduced. These sections provide a brief overview of their technical workings and their associated security challenges. Then, Section 2.4 provides an overview of the ZigBee protocol, discussing its technology, ongoing advancements and real-life applications, and the different device types and their responsibilities in the various network topologies. Section 2.5 extensively analyses the security concepts of ZigBee, including how security is applied to each layer of ZigBee's protocol stack, the different security models and symmetric key types, the included security controls and countermeasures, and its known security issues and vulnerabilities. In Section 2.6, the security advancements made to the protocol's latest version 'ZigBee 3.0' are discussed. Section 2.7 summarises five related studies of practical undertakings against ZigBee's security issues. Last, Section 2.8 concludes this literature review.

2.2 Wireless Sensor Networks

WSNs are interconnected and infrastructure-less networks consisting of one or more sensor nodes that monitor the surrounding physical or environmental conditions for various applications (Matin & Islam, 2012). The development of WSNs is considered one of the most rapidly evolving technological domains, with its use expanding across a growing number of applications (Kandris, Nakas, Vomvas, & Koulouras, 2020). Sensor nodes are deployed to collect different types of data wirelessly, such as the temperature, vibration, sound, pressure

and motion. The collected data are then cooperatively routed to their primary destination for further observation and analysis. In a WSN topology, data are usually collected through a gateway. This information is then forwarded onto the parent/leader node or base station, termed a ‘sink’ (Carlos-Mancilla, López-Mellado, & Siller, 2016). WSNs are deployed across many industries through several application fields. Some of the most relevant and common application fields for WSNs include the environmental, industrial, health and military fields, each containing extensive subcategories (Kandris et al., 2020).

WSNs are vastly scalable networks that usually comprise thousands of connected sensor nodes across a single network. Once a node connects to the network, it becomes responsible for self-organising its network infrastructure to adapt to its surrounding network environment and routing data between nodes (Matin & Islam, 2012). As part of their defining characteristics, WSN nodes are designed to meet specific requirements that allow them to be mobile, conservative and scalable (Ahmed, Huang, Sharma, & Cui, 2012). Sensor nodes are typically low cost and designed with resource constraints, including limitations to processing speeds, storage capabilities and communication bandwidth. These limitations allow devices to conserve and maximise battery life (Ahmed et al., 2012). Furthermore, the low-power design of these sensor nodes can allow for years of operation on a single battery. In a WSN, sensor nodes can be programmed to sleep when idle for extended periods and power back on when their function is required. This feature allows devices to be deployed in outdoor environments for great lengths of time with little need for physical maintenance (Engmann, Katsriku, Abdulai, Adu-Manu, & Banaseka, 2018).

Inherent in all forms of WSN technologies are problems that cannot easily be addressed. Security is an issue prevalent across all WSN technologies. In some instances where data confidentiality is crucial, for example, in health care, the security of the WSN devices used must be considered carefully before deployment. Security challenges are relevant to WSNs,

both in logical and physical aspects. Logical security challenges primarily relate to ensuring secure data transmission between nodes. Cryptographic techniques, including symmetric and asymmetric keys, are implemented to establish confidential communications between devices. However, specific WSN devices can only use less secure cryptography implementations owing to resource constraints (Shanmugapriya, Kousalya, Rajeshkumar, & Nandhini, 2019). In addition to ensuring secure data transmission, measures should be established to prevent unauthorised or compromised devices from joining a network.

The physical security of WSN devices is another prevalent security challenge. Physical security relates to the physical protection of a node against unauthorised access to its software or hardware integrity. Sensor nodes are commonly deployed in hostile outdoor environments; therefore, measures are often required to prevent unauthorised individuals from physically accessing these devices. A physically compromised device could introduce numerous security issues across a WSN. For instance, if an attacker can extract the data from a compromised device, then the confidentiality of all data transmitted across the network is at risk (Barbareschi, Battista, Mazzeo, & Venkatesan, 2014).

2.3 IEEE 802.15.4 Standard and Networks

IEEE 802.15.4 is a standard defined in 2003 for the operation of low-rate wireless personal area networks (LR-WPAN). This standard defines the physical (PHY) and media access control (MAC) layers of LR-WPAN devices and is maintained by the IEEE 802.15.5 working group (Lu, Krishnamachari, & Raghavendra, 2004). The requirements defined by this standard include specifications for low-data rate wireless connectivity for fixed, portable and moving devices with no battery or very little battery consumption. Moreover, this standard defines the PHY layer for devices operating in various geographical locations (Lu et al., 2004).

IEEE 802.15.4 provides the fundamental network infrastructure and the lower layers for technologies that incorporate this standard, including ZigBee, ISA100.11a, WirelessHART,

MiWi, 6LoWPAN, Thread and SNAP. These technologies all extend the standard by developing their own upper layers and implementing security mechanisms in addition to those IEEE 802.15.4 provides (Adams, 2006).

Two main network topologies are supported by IEEE 802.15.4, which may be used for various applications. One is the star topology, which consists of one central node through which all other nodes communicate. The other is a peer-to-peer topology. This network still consists of a central node; however, the other nodes may communicate with each other directly rather than through the coordinator (Salman, Rasool, & Kemp, 2010).

2.4 ZigBee Technology Overview

First, this section discusses the technical workings of ZigBee and its developments over the years. ZigBee's real-life applications are identified and discussed. Next, it presents a discussion of each ZigBee device type, including the ZigBee coordinator, router and end device, which includes a description of their roles and responsibilities in a ZigBee network. Last, the three different possible ZigBee network topologies are analysed.

2.4.1 ZigBee Technology

ZigBee is an IoT technology designed to address low-cost, low-power industrial requirements. Its physical radio operates on the IEEE 802.15.4 specification with signal bands including 2.4 GHz, 900 MHz and 868 MHz. ZigBee's specification protocol suite allows its devices to communicate through various network topologies, and its battery life is optimised to last up to several years (Ramya, Shanmugaraj, & Prabakaran, 2011).

ZigBee technology is commonly used in applications where low bandwidth is adequate. ZigBee's low power consumption limits its physical range from 10 to 100 meters depending on various factors, including the power outlet and environmental conditions. However, ZigBee can transmit over long distances by routing data through intermediate devices over a mesh network. ZigBee can accommodate up to 65,000 nodes over a single network. Among ZigBee's

three licensed signal bands, its most common signal, 2.4 GHz, can transfer data at up to 250 kbps, whereas 915 MHz transfers at 40 kbps and 856 MHz supports up to 20 kbps (Ramya et al., 2011).

The ZigBee protocol was created after the ratification of IEEE 802.15.4 in 2004. It has since been developed and maintained by member companies of the ZigBee Alliance. The ZigBee Alliance membership consists of more than 300 semiconductor manufacturers, technology firms, original equipment manufacturers and service companies that have provided ongoing advancements to ZigBee's technology and has improved its capabilities (Digi International, n.d.-c). ZigBee's specification has had several releases since 2004 and can be profiled as follows (Lea, 2018, p. 156):

- 2005: ZigBee 2004 released.
- 2006: ZigBee 2006 released.
- 2007: ZigBee 2007 released. This is also known as ZigBee PRO, and it introduced cluster libraries and backward compatibility constraints with ZigBee 2004 and 2006.

ZigBee 3.0 is the current standard of the ZigBee protocol that is implemented into the ZigBee PRO 2015 (or newer) specification and was released to the public in December 2016 (Morgner, Mattejat, Benenson, Müller, & Armknecht, 2017). The major updates to ZigBee 3.0 include a child device management feature, improvements to existing and additional security features and support for the optional DSM (ZigBee Alliance, n.d.-b).

2.4.2 ZigBee Applications

ZigBee is a standard for personal area networks (PAN) used in a wide range of applications. Its design incorporates low-cost, low-power consumption, and reliability set to fulfil the requirements of many industrial standards (Digi International, n.d.-c). ZigBee is not used in situations that require high mobility among nodes. However, it is deployable in geographically challenging areas. ZigBee's primary applications include:

- Smart home: ZigBee is used in households to provide home automation. Its smart devices are controlled through a central hub over a network designed to improve comfort and convenience in households. Common ZigBee smart home devices include smart lightbulbs and security/motion cameras that can be remotely controlled through a smartphone or tablet (Wheeler, 2007).
- Commercial: ZigBee is used in commercial applications in a variety of industries, including medical, hospitality, education, retail and manufacturing industries. ZigBee's significant use is building automation to provide connected lighting, efficient energy control, climate and HVAC control, daylight and window blind systems, access control and safety (ZigBee Alliance, n.d.-b).
- Utility: ZigBee is used in utility applications to monitor, control, inform and automate the delivery and use of water, gas and energy for households and buildings (ZigBee Alliance, n.d.-a).

2.4.3 ZigBee Devices

The ZigBee protocol consists of three types of logical devices: the coordinator, the router and the end device. Each device has specific roles and responsibilities that differentiate it from its counterparts. Furthermore, the behaviour of each type of device towards routing data packets and communicating with other devices on the network differs (Ramya et al., 2011).

2.4.3.1 Coordinator Node

ZigBee coordinators are parent nodes responsible for establishing the network, setting network parameters and managing the overall network. A coordinator has all the routing capabilities of a ZigBee router; however, it is the only ZigBee device that can form a network (prior to ZigBee 3.0). Some of its key responsibilities include selecting the channel and the Personal Area Network Identifiers (PAN-IDs) and managing the network's security model. A

ZigBee network must always consist of one and only one coordinator for centralised networks (Aju, 2015). However, a distributed network model has no coordinator present.

Coordinators are required to operate consistently; therefore, they cannot be battery operated or enter a sleep mode. Once a coordinator creates a network, other correctly configured devices can join the network (Digi International, 2017). The coordinator keeps an up-to-date list of currently associated devices and facilities each device into its routing table as they join the network. Furthermore, it supports an orphan scan that enables previously associated devices to rejoin the network (Rudresh, 2017a). Coordinators route network traffic and can communicate with other devices on the network.

The coordinator begins by establishing a network. This process chooses an operating channel and generates two unique PAN-IDs, 16-bit and 64-bit, standard across all network devices. Depending on the security configuration, a device must either be preconfigured with PAN-IDs or discover nearby networks and select the PAN-ID to join the network (Digi International, 2017). When the coordinator chooses its operating channel and PAN-ID, it performs a series of scan functions to ensure that other nearby devices do not use these. Furthermore, it has a 'PAN-ID Conflict' mechanism, a frequency agility feature that migrates the network to a new PAN-ID when it detects the same PAN-ID in a broadcast request (Mukherji & Sadu, 2016). Part of the coordinator's functionality is to manage the security model of the network. This function includes setting the encryption options and updating/distributing the network key used for end-to-end encryptions across the network. By default, the coordinator is configured as the 'Trust Centre', an application on a ZigBee device that manages and distributes the network key. Depending on the chosen security model, the coordinator can elect another router node on the network to act as the trust centre (Rudresh, 2017a).

2.4.3.2 Router Node

A ZigBee router is an intermediate node responsible for routing traffic between the end devices and the coordinator on the network. Routers require permission to join the network from the Trust Centre or the coordinator, and in addition to their routing capabilities, they can also serve as end devices. There is no limitation to the number of routers that can operate on a ZigBee network. Similarly to coordinators, routers must be operating consistently; therefore, it is not suitable for them to be battery operated, and they cannot enter a sleep mode (Rudresh, 2017a).

A router's routing capabilities allow it to transmit/receive data packets and communicate with other devices on the network. In specific ZigBee network configurations, routers can allow other routers or end devices to join the network. They also maintain a list of all currently associated devices and support the orphan scan to allow previously associated devices to rejoin the network. Routers are also responsible for storing packets on behalf of their sleeping children (Tomar, 2011).

2.4.3.3 End Device Node

A ZigBee end device is a child node with limited networking capabilities and is most commonly a low-power, battery-operated device. End devices can join existing networks and can send, receive and route information. However, unlike coordinators or routers, end devices cannot act as intermediate nodes between devices or allow other devices to join the network. Moreover, given their inability to relay messages, end devices can only communicate within the network through their parent nodes (X. Fan et al., 2017).

Standard ZigBee end devices include sensor nodes used to collect and monitor environmental data and smart lightbulbs used in smart home applications. Because these devices are not required to relay information between nodes consistently, end devices can be configured to enter a non-responsive sleep mode to conserve battery temporarily. End devices

can receive packets sent from the parent device while they are asleep and are responsible for finding a new parent node if the link to their old parent is lost through a network rejoin (Gislason, 2008, pp. 234–235). ZigBee networks do not have a limitation regarding the number of end devices.

2.4.4 ZigBee Network Topologies

ZigBee supports three types of PAN topologies: star, tree and mesh. Each topology varies in complexity, and the topology choice may reflect on the network requirements. These could include the amount of network traffic required, the latency requirements and the solution cost (Rudresh, 2017a). Every ZigBee network topology must consist of only one coordinator.

2.4.4.1 Mesh Topology

A mesh topology (see Figure 2.1) consists of one coordinator and multiple routers and end devices and allows complete peer-to-peer communication. The nodes in a mesh network are interconnected and can therefore communicate through multiple pathways. The pathways between the nodes are dynamically updated and optimised through the built-in mesh routing table (Digi International, n.d.-c). In a mesh topology, the coordinator establishes the network and sets specific networking parameters. Routers extend the network coverage and route traffic between the source and destination within the network. Furthermore, routers can serve as end devices but cannot emit beacons (Rudresh, 2017a). Figure 2.1 shows the structure of a typical ZigBee mesh topology.

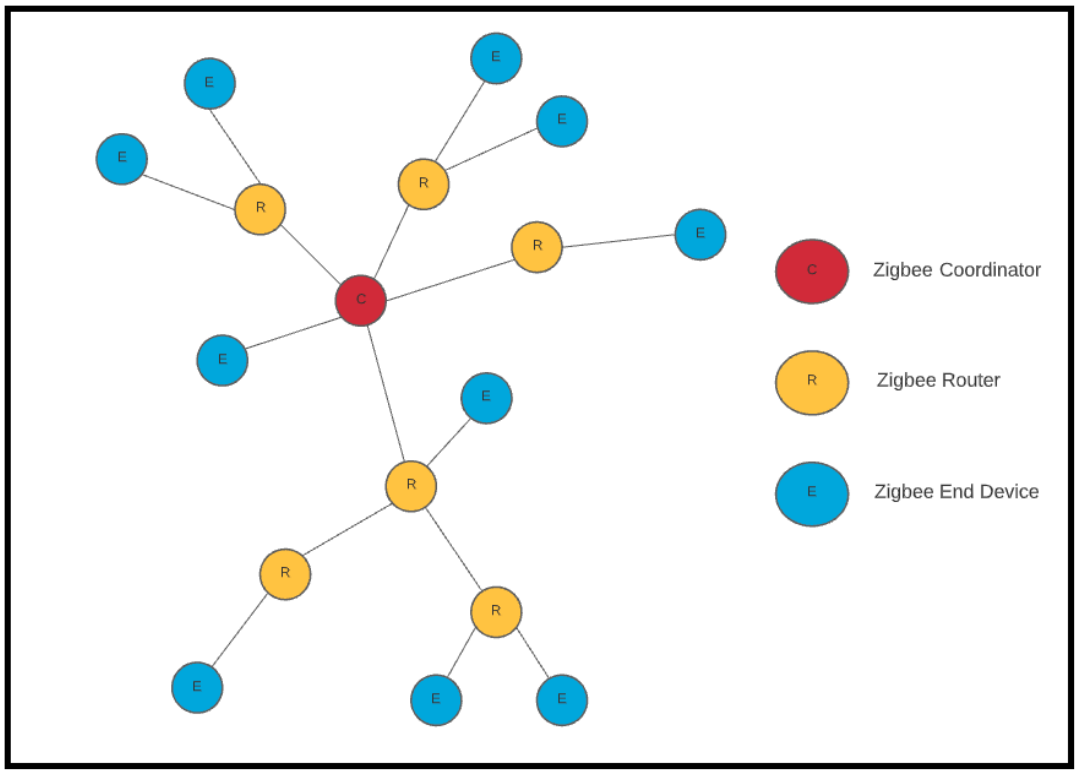

Figure 2.1. ZigBee mesh network. Adapted from 'ZigBee Topology: A Survey', by T. Kumar and P. Mane, 2016, *2016 International Conference on Control, instrumentation, communication and Computational Technologies (ICCICCT)*, p. 165. Copyright 2016 by IEEE.

The advantage of mesh topologies is that the pathways between the nodes are self-healing. If a link failure occurs, devices transmitting messages will find an alternative path to reach their destination, eliminating redundancy. ZigBee devices operating in a mesh network are equipped with a discovery feature that determines the best route for exchanging messages. Moreover, the failure of the coordinator does not result in a single point of failure (Khanji, Iqbal, & Hung, 2019). Mesh networks are ideal for medium to large-scale networks based on robust multi-hop communication, scalability and latency. However, its complexity is a drawback for it is more difficult to set up, and nodes have additional overheads (Rudresh, 2017a).

2.4.4.2 Star Topology

A star topology (see Figure 2.2) has one coordinator and several end devices. The end devices communicate directly with the coordinator, for there are no routers. In this topology, the coordinator is solely responsible for routing packets and establishing and managing network devices. End devices can only communicate through the coordinator (Kumar & Mane, 2016). Figure 2.2 shows a simple ZigBee star topology.

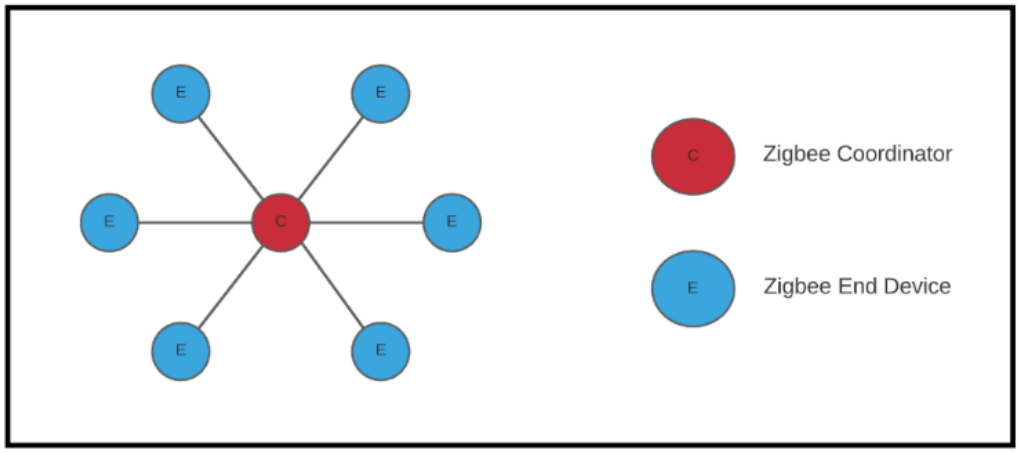

Figure 2.2. ZigBee star topology. Adapted from 'ZigBee Topology: A Survey', by T. Kumar and P. Mane, 2016, *2016 International Conference on Control, instrumentation, communication and Computational Technologies (ICCICCT)*, p. 165. Copyright 2016 by IEEE.

Star topologies are an effective solution for small-scale networks for they are simple to deploy and manage, and all packets must only go through a maximum of two hops to reach their destination. However, this topology is impractical for large-scale networks based on some of its distinct drawbacks. The main disadvantage is that the whole network can shut down if the coordinator fails and goes offline. Furthermore, the coordinator's bandwidth may become bottlenecked because of the lack of an alternative path between the network traffic source and destination (Kumar & Mane, 2016).

2.4.4.3 Tree Topology

A tree topology (see Figure 2.3) consists of one coordinator and multiple routers and end devices. In this topology, the coordinator is the central (root node) responsible for establishing the network and setting the network parameters (Rudresh, 2017a). Coordinators can be a parent node to routers as well as end devices. The primary function of routers in this topology is to extend the network coverage and to move data and control messages across the network using hierarchical routing strategies (Elahi & Gschwender, 2009). Figure 2.3 shows the structure of a basic ZigBee tree topology.

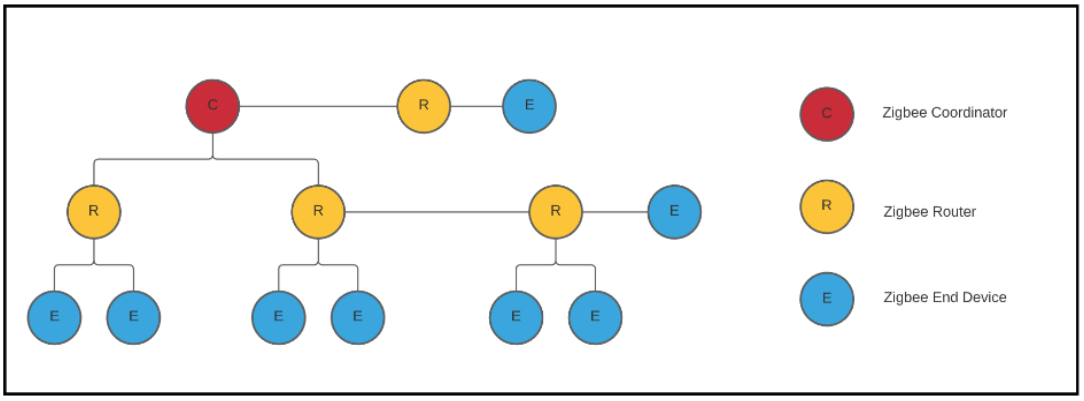

Figure 2.3. ZigBee tree topology. Adapted from 'ZigBee Topology: A Survey', by T. Kumar and P. Mane, 2016, *2016 International Conference on Control, instrumentation, communication and Computational Technologies (ICCICCT)*, p. 165. Copyright 2016 by IEEE.

Tree topologies are practical for larger-sized networks based on their high scalability and centralised monitoring. The disadvantage of this topology is that when a parent node becomes inactive, all the child nodes connected to that parent node become unreachable (Elahi & Gschwender, 2009).

2.5 ZigBee Security Overview

The ZigBee protocol defines several security services to maintain the confidentiality, authentication and integrity of its data between devices. ZigBee was built with security in mind,

by adopting the secure 128-bit AES-based encryption suite and implementing essential security services. However, its low-cost and low-power design comes as a trade-off to its overall security (Zillner, 2015).

This section discusses how security is applied to each layer of ZigBee's protocol stack. The traditional security model and ZigBee 3.0's alternative security model is then analysed. Next, the symmetric keys used by the ZigBee devices for encryption are discussed, followed by ZigBee's primary security controls and countermeasures. Last, this section analyses the known security issues and vulnerabilities of ZigBee that are prevalent in the protocol.

2.5.1 Security Architecture

In ZigBee's protocol stack, the Network (NWK) and the Application (APL) layers are built on top of the IEEE-defined PHY and MAC layers. ZigBee's APL layer consists of the Application Support Sublayer (APS), the ZigBee Device Object (ZDO) and the Application Framework, each having its own security services (Vasseur & Dunkels, 2010, p. 297). ZigBee uses an open trust model in which each layer shares trust; therefore, cryptographic protection only exists between devices and not the different layers of the protocol stack. This allows the same symmetric key to be used across all layers of a device. The IEEE 802.15.4 standard sets the encryption algorithm used by ZigBee, which is AES with 128-bit key lengths; however, ZigBee's upper layers define the ways in which the keys are managed or the authentication policies that are applied (Gascón, 2009). Figure 2.4 illustrates the layers of ZigBee's protocol stack:

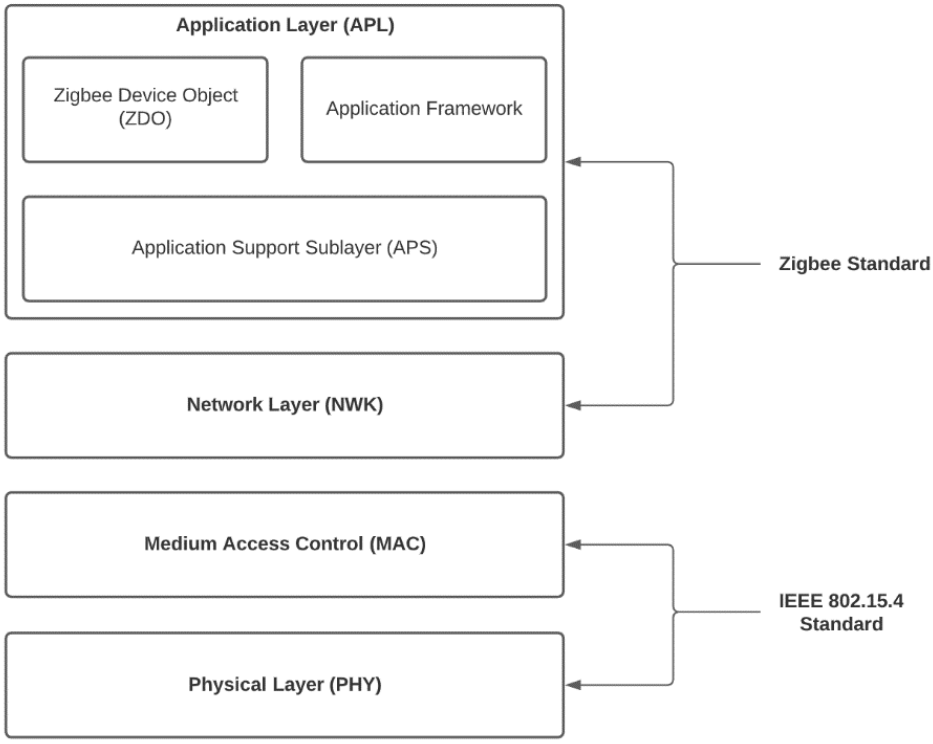

Figure 2.4. ZigBee's protocol stack. Adapted from 'Time sensitive IEEE 802.15.4 protocol', by A. Koubâa, M. Alves and E. Tovar, 2006, *Sensor Networks and Configurations: Fundamentals, standards, platforms, and applications*, p. 21. Copyright 2007 by Springer-Verlag Berlin Heidelberg. Adapted with permission.

ZigBee's security architecture includes security mechanisms incorporated into the MAC, NWK and APL layers (Fan et al., 2017). The next sections discuss the application of security to each layer of the stack.

2.5.1.1 MAC Layer Security

In ZigBee, the MAC layer's security is based on the IEEE 802.15.4 standard, which implements several features used by the upper layers in the ZigBee protocol. The MAC layer is augmented with a version of CCM (counter with cipher block chaining message authentication code) called CCM*, which offers encryption and integrity capabilities only (Rudresh, 2017a). ZigBee's MAC layer uses one key for all CCM* security levels (the MAC, NWK and APS layers). The upper layers of the protocol stack determine whether the MAC

layer should use security services, and they provide the keying information and information on the security level to use (X. Fan et al., 2017). Figure 2.5 outlines the security of an IEEE 802.15.4 MAC frame, which has specific fields related to its security controls.

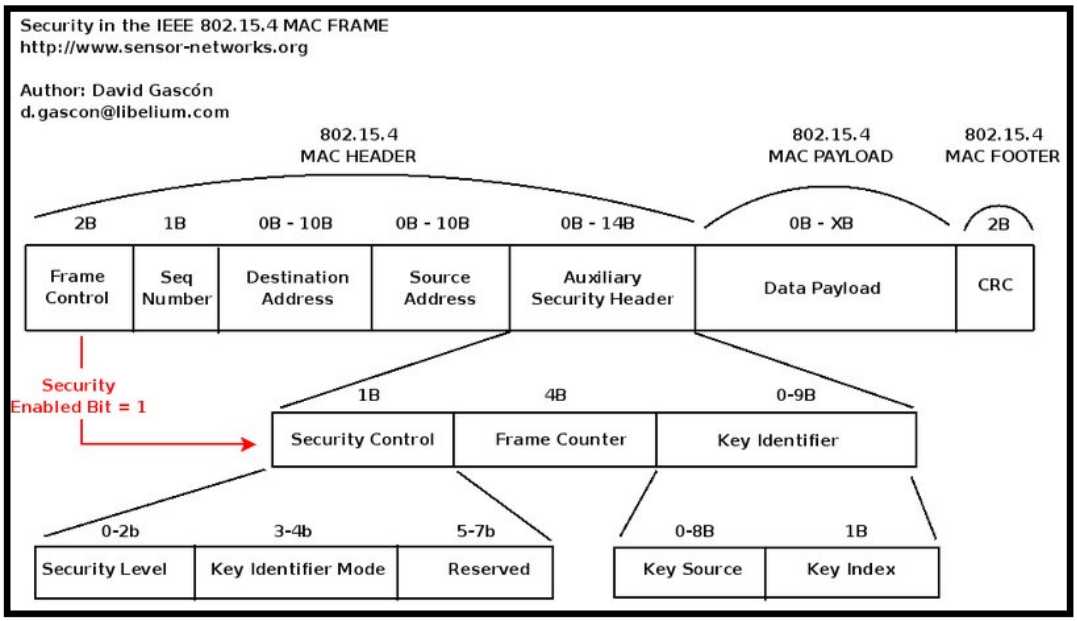

Figure 2.5. Security in the IEEE 802.15.4 MAC Frame. From Gascón, 2009. Security in 802.15.4 and ZigBee networks. Retrieved from https://www.libelium.com/libeliumworld/security-802-15-4-zigbee/

A ZigBee MAC layer frame is composed of the MAC Header, the MAC Payload and the MAC Footer. An IEEE 802.15.4 MAC frame has three fields that control how security is processed in ZigBee. These are the Frame Control, the Auxiliary Security Header (ASH) and the Data Payload. The Frame Control field in the MAC header has a 'Security Enabled' subfield, which determines whether the outgoing frame has security controls enabled (Gascón, 2009).

2.5.1.1.1 Auxiliary Security Header

The ASH is a field located in the MAC header, which is only enabled if the Security Enabled subfield of the Frame Control is set to 1. This field has three subfields related to security. Its Security Control subfield determines the type of protection that is used, including

whether encryption is enabled, and the integrity level used to protect a frame. This is controlled through its Security Level subfield with eight different values that determine the frame's security (Gascón, 2009). Moreover, the 'Key Identifier Mode' is a subfield of Security Control that sets the type of key (implicit or explicit) used by the sender or receiver. The Frame Counter subfield of ASH is a counter value set by the current frame source to prevent the frame from being replayed on the network. Last, the Key Identifier is another subfield of ASH that specifies the keying information needed to communicate with other nodes on the network (Gascón, 2009).

2.5.1.1.2 Data Payload

The IEEE 802.15.4 Data Payload field is located in the MAC payload and can have three different configurations that determine how AES is applied to the frame to protect data. This is dependent on the previously defined security fields, and the configurations include AES-CTR, AES-CBC-MAC and AES-CCM (Gascón, 2009). However, as previously discussed, ZigBee applies AES-CCM*, a slightly modified version of AES-CCM.

2.5.1.1.3 Access Control List

The MAC layer maintains an Access Control List (ACL) to prevent unauthorised devices from participating in the network. An ACL list is stored in the MAC PAN Information Base (PIB) and contains specific fields that allow devices to verify whether a packet's source or destination is trusted or not (Rudresh, 2017b). When a device is sending or receiving a packet, it looks up its ACL. Appropriate security measures will be applied if the associated node is trusted, and the packet will be sent/received. Otherwise, the packet will be dropped (Gascón, 2009). Each IEEE 802.15.4 device is responsible for storing the following fields into its ACL:

- Address: the MAC addresses of network nodes;

- Security Suite: the security suite in use; for example, AES-CCM* for ZigBee and AES-CTR, AES-CCM, AES-CBC-MAC for other IEEE 802.15.4 technologies;
- Key: the 128-bit key used in the AES algorithm; and
- Last Initial Vector (IV) and Replay Counter: The last IV is used by the source address and the Replay Counter with the destination address to prevent replay attacks (Gascón, 2009).

2.5.1.2 Network Layer Security

The ZigBee standard defines the NWK layer responsible for the processing steps required to transmit and receive frames securely. ZigBee uses a frame protection mechanism to secure frames originating from the NWK layer, such as broadcast frames intended to be received and processed by every node on the network. Moreover, AES-CCM* with 128-bit key lengths are applied to the NWK layers' frame protection mechanism to provide security to its frames. ZigBee's upper layers manage the NWK layer security by setting up the active and alternative network keys and the frame counter and by establishing the security level to use (ZigBee Alliance, 2017, p. 412). Figure 2.6 shows the security fields that are included in an NWK frame:

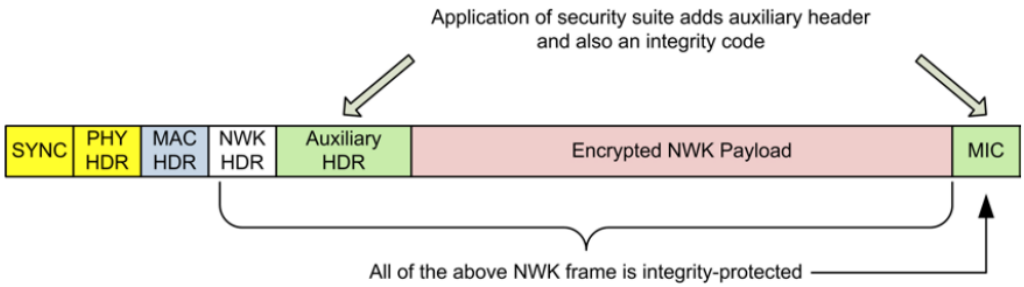

Figure 2.6. ZigBee NWK layer security frame. Reprinted from 'ZigBee Specification (Document No. 05-3474-22)' (p. 410), by ZigBee Alliance, 2017. Copyright 2017 by ZigBee Alliance.

2.5.1.2.1 Network Layer Outgoing Frame Security

For outgoing frames, security is processed when its corresponding security headers indicate that the frame requires protection. The NWK layer's critical security processing steps for outgoing frames are as follows (ZigBee Alliance, 2017, p. 412):

1. The NWK layer obtains the active network key, the outgoing frame counter and the key sequence number from the Network Layer Information Base (NIB). If the outgoing frame counter is equal to 2^32-1 (Max Value), or the key cannot be obtained, the security processing will fail and no further security processing on the frame will be performed.
2. After these values are obtained, the auxiliary header is constructed. CCM* is processed on the frame for encryption and authenticity based on the security level set.
3. Last, the outgoing frame counter is incremented by one. At any time, if a security processing step fails, then all security processing will be stopped for that frame.

2.5.1.2.2 Network Layer Incoming Frame Security

When the NWK layer receives a secured frame as indicated by the security subfield of the NWK header Frame Control field, it will perform the following critical security processing steps to ensure that the frame is securely received (ZigBee Alliance, 2017, p. 413):

1. The NWK layer will determine several attributes for the incoming frame, including the security level, the sequence number, the sender address and the received frame count. If the received frame count is equal to 2^32-1, then the frame will be flagged with a 'bad frame counter' to the upper layers, and no further security processing will be performed for that frame.
2. The NWK layer will then obtain the appropriate security material, including the key information and other attributes, by matching the frame's sequence number to any key in the nwkSecurityMaterialSet attribute in the NIB.

3. CCM* is then processed on the frame for the decryption and authentication checking operation.
4. Last, the frame counter is set to the received frame counter +1 and is stored in the NIB along with the sender address.
5. If a security process fails at any time for these steps, then no further security processing will be conducted for that frame.

2.5.1.3 Application Layer Security

The APL layer defined by the ZigBee standard contains the ZDO, the Application Framework and the APS. The ZDO is responsible for initialising the APS layer, the NWK layer and the Security Service Provider (SSP). However, the security mechanisms of the APL are handled by the APS sublayer.

2.5.1.3.1 ZigBee Device Object

The ZDO comprises applications that employ the NWK and APS layer primitives to implement ZigBee coordinators, routers and end devices. The ZDO assembles the configuration information from end applications to determine and implement functions to the device. These include service discovery, security manager (transport key, request key, update the device, remove a device and switch key), network manager, blinding manager, node manager and group manager (ZigBee Alliance, 2017, p. 201). Furthermore, the ZDO manages the security policies and security configurations of a device (B. Fan, 2017).

2.5.1.3.2 Application Support Sublayer

The APS layer security protects frames originating from the APL layer using frame security based on the link keys or the network key. Unlike the NWK layer security, the APS layer security is optional and provides end-to-end security between devices using an APS link key known by only the source and the destination devices (Silicon Labs, n.d.). Furthermore, the APS layer provides an interface between the NWK and APL layers and provides services

for establishing and maintaining security relationships (Rudresh, 2017a). The APS layer is responsible for the processing steps required to securely transmit and receive outgoing and incoming frames and to establish and manage the symmetric keys. Figure 2.7 shows the security fields included in an APS frame:

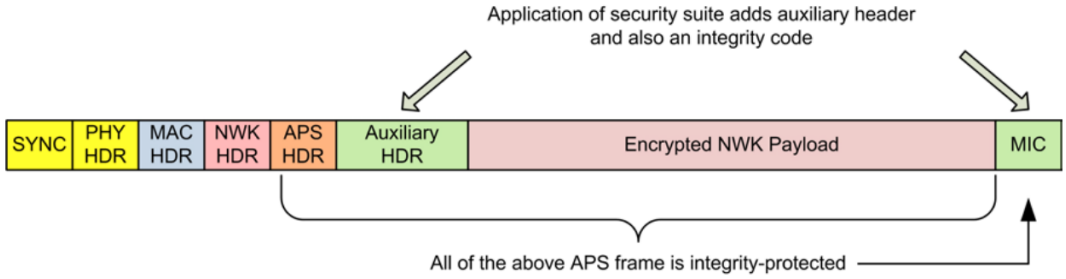

Figure 2.7. ZigBee APL layer security frame. Reprinted from 'ZigBee Specification (Document No. 05-3474-22)' (p. 410), by ZigBee Alliance, 2017. Copyright 2017 by ZigBee Alliance.

The APS layer's frame security processing steps work similarly to the NWK layer's frame protection. CCM* is applied to an outgoing frame for the encryption and authentication operation and to an incoming frame for decryption and authentication checking. If any security processing step fails, no further security processing will be performed for that frame (ZigBee Alliance, 2017, p. 418). Among its other security services, the APS layer is responsible for providing the ZDO and applications with keying services and device management services. Its keying services offer a secure solution to establish, transport, request, switch, verify and confirm the symmetric keys over the network. Furthermore, its device management services provide a secure means to update and remove devices from the network (ZigBee Alliance, 2017, pp. 410-411).

2.5.2 Security Models

With the introduction of ZigBee 3.0, the ZigBee protocol supports two models for key management: the Centralised Security Model (CSM) and the Distributed Security Model

(DSM). These security models differ in device authentication and message protection mechanisms (X. Fan et al., 2017).

2.5.2.1 Centralised Security Model

A CSM is the traditional security model used in ZigBee and has been improved in ZigBee 3.0, in which security is organised through a single node known as the trust centre. The trust centre is responsible for maintaining the network's overall security, and its primary role is to authenticate devices joining the network and distribute the symmetric keys. Furthermore, it has a view of every authenticated device on the network (NXP Semiconductors, 2017). Some of its other core security roles include establishing and managing the symmetric keys used for encryption on the network. The trust centre can be configured to revoke and generate a new network key when required or at set intervals, reducing the impact of a compromised network key. Another responsibility is to set a link key to authenticate devices and securely exchange the network key (NXP Semiconductors, 2017).

When creating a ZigBee network and choosing to implement a CSM, electing the trust centre and its security policy is crucial, depending on the network's security requirements. By default, the coordinator node is selected as the trust centre since it is already a centralised node responsible for forming the network and setting symmetric keys for encryption. A CSM is complex; however, it is the most implemented security model for ZigBee networks. It is also considered the most secure model because all security preferences and symmetric keys are managed and distributed by a single node (Rudresh, 2017b).

2.5.2.2 Distributed Security Model (ZigBee 3.0)

Introduced into the ZigBee 3.0 protocol is support for DSM networks. A DSM is considered a less secure security model and has no centralised coordinator or single trust centre to maintain security. Instead, all ZigBee routers in the network act as trust centres and are responsible for authenticating and joining other router nodes or end devices into the network

as their child nodes. Router nodes also distribute the network key to the newly joining devices, given that they are correctly configured with the link key (NXP Semiconductors, 2017). DSM networks have a lifetime of approximately 4.3 billion packets for the network key is fixed and cannot be updated (Texas Instruments, 2019). Moreover, a DSM incorporates a simplified security structure, which makes it non-ideal for networks containing highly sensitive information.

2.5.3 Symmetric Keys

The ZigBee standard supports the two primary 128-bit symmetric keys, including the 'Link Key' and the 'Network Key' along with a 'Master Key' to establish the link key. These keys are used in a ZigBee network to ensure that the devices can securely communicate using AES 128-bit encryption to maintain the confidentiality of exchanged messages. Next, ZigBee's symmetric keys are discussed.

2.5.3.1 Link Key

The link key is a secret session key used to secure unicast communications and is applied by the APS. This key is only shared between two devices, the trust centre/coordinator and the router/end device and is acquired through key transportation, key establishment or pre-installation (Zillner, 2015). ZigBee has two types of link keys, global and unique. A global link key is a known key used by all nodes on the network and is created by the ZigBee Alliance or otherwise defined by the manufacturer of specific ZigBee devices. The link key defined by the ZigBee Alliance is applied when no other link keys are specified by the APS when a device joins the network. This key is known as the 'default global trust centre link key' (Zillner, 2015), and its default value in hexadecimal equals '5A6967426565416C69616E63653039', which has a corresponding char value in ASCII (American Standard Code for Information Interchange) as follows:

ZigBeeAlliance09

Other specific ZigBee devices are factory preconfigured with a global link key defined by their manufacturer, allowing interaction between devices of the same manufacturer (Rudresh, 2017b). The trust centre is usually configured to encrypt the 128-bit network key using the link key in a CSM network before transmitting it to pairing devices. On DSM networks, nodes are typically factory-programmed with a manufacturer global link key or preconfigured with a link key (NXP Semiconductors, 2017).

2.5.3.2 Network Key

The network key is used in a ZigBee network to secure broadcast communications and is applied by the NWK and APL layers of the protocol stack. This key is shared between all devices on the network to secure transmitted frames between devices. The network key is the minimal requirement for establishing security on a ZigBee network. It protects transmitted frames and prevents both unauthorised joining and illegitimate ZigBee devices from using the network (Masica, 2007). A ZigBee network can have one of two types of networks keys: standard and high security. The key type generally defines how the network key is distributed over the network—that is, whether it is encrypted or not. A standard network key is distributed to pairing devices unencrypted, whereas the trust centre encrypts a high-security network key with the link key before key transport (X. Fan et al., 2017).

In a CSM, the trust centre is responsible for generating the network key and distributing it to all devices on the network. Moreover, devices can acquire the network key through key transport or pre-installation (Rudresh, 2017b). The ZigBee standard supports the trust centre storing multiple network keys for key rotation and key update purposes. However, only one network key can be active at one time, which is identified by a sequence number (Zillner, 2015). On a DSM, the routers distribute the network key to newly joining routers and end devices.

2.5.3.3 Master Key

The APS uses the master key to establish long-term security between two devices. Devices use the master key to generate the link key and to ensure that the link key exchange between two nodes is confidential. ZigBee's Symmetric-Key Key Establishment Protocol (SKKE) uses the master key to create a secure key exchange, which increases ZigBee's overall security (Radmand et al., 2010).

2.5.4 Security Controls and Countermeasures

The ZigBee standard and its security specification have built-in security controls to ensure the confidentiality, authentication and integrity of its data and devices. In this section, ZigBee's primary security controls and countermeasures are discussed.

2.5.4.1 Data Confidentiality, Authentication and Integrity

ZigBee achieves data confidentiality, authentication and integrity through the AES-CCM* security suite, which uses 128-bit key lengths. In a ZigBee network, frames are optionally encrypted on the NWK and APL layers. ZigBee devices use the symmetric keys to encrypt frames through security steps using AES-CCM* on the NWK and APS layers. The device on the receiving end applies AES-CCM* and the shared 128-bit symmetric key for decrypting and authenticating the frames (Yang, 2009). Figure 2.8 shows the operation of AES-CCM* to provide data confidentiality and authentication:

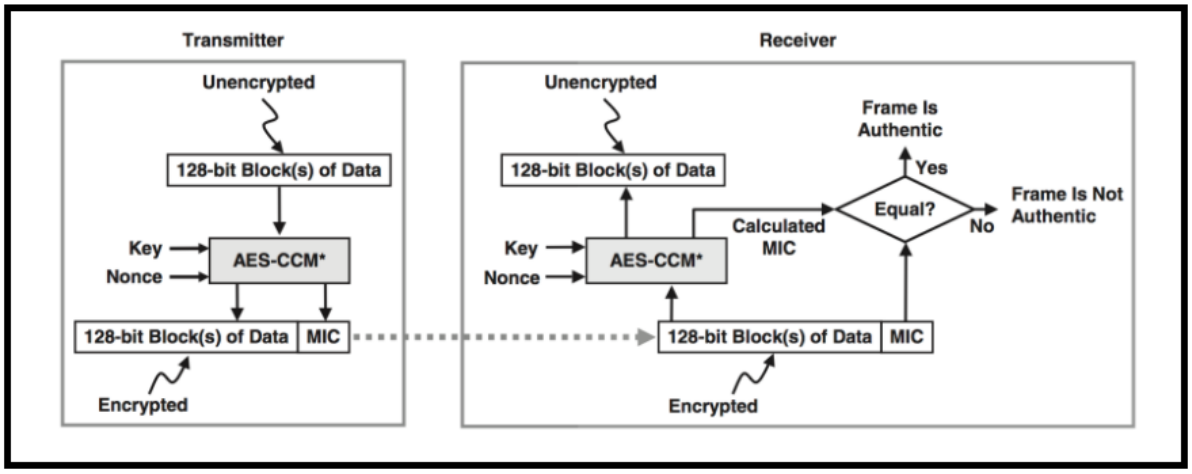

Figure 2.8. AES-CCM* operation in ZigBee. Reprinted from 'ZigBee and IEEE 802.15.4 Protocol Layers' (p. 126), by S. Farahani, 2008b, in S. Farahani (Ed.), *ZigBee Wireless Networks and Transceivers*, Burlington, MA, Elsevier. Copyright 2008 by Elsevier. Reprinted with permission.

As shown in Figure 2.8, the transmitter encrypts the data through the AES-CCM* operation. ASM-CCM* uses three inputs: the data to be encrypted, the security key and the nonce. The nonce is a 13-octet string consisting of the auxiliary headers' security control, frame counter and source address fields. AES-CCM* encrypts the data using the security key and generates an associated Message Integrity Code (MIC) sent to the receiver along with the frame (Farahani, 2008b). The receiver then decrypts the data using the AES-CCM* operation and the security key. A MIC is generated from the received frame and is compared with the received MIC. If the generated MIC is equal to the received MIC, then the frame is authentic. Otherwise, if the values are not equal, the frame is unauthentic and dropped (Farahani, 2008b).

2.5.4.2 Authentication

One of ZigBee's primary features is device and data authentication. This process involves ensuring that new devices joining the network are confirmed to be authentic. As discussed in section 2.5.4.1, authentication is processed through the CCM* operation; however, the trust centre/coordinator manages the process. Devices are authenticated by the trust centre

and receive the network key and other specific networking parameters before being admitted into the network (Rudresh, 2017b). Authentication is essential for it prevents unauthorised devices from joining the network and hacked devices from impersonating legitimate devices. The first level of authentication uses the standard 128-bit network key to provide authentication on the network level. This process prevents outside attacks with lower memory costs. Authentication can also be achieved on the APS layer between two devices by using unique 128-bit link keys. Establishing authentication between two paired devices prevents both inside and outside attacks but results in higher memory costs (Reddy, 2005).

2.5.4.3 Message Integrity

ZigBee provides message integrity to prevent an attacker from modifying a packet in transit. This step is achieved through the MIC or Message Authentication Code and is processed through the CCM* operation. The MIC or Message Authentication Code is embedded into a frame before it is sent to ensure the integrity of the MAC header and payload data (Farahani, 2008b). Integrity can be applied through the 0, 32, 64 or 128-bit integrity code (defaulting at 64 bits). The integrity option can be set while specifying the ZigBee network's security policy; however, higher integrity options offer a trade-off between message protection and message overhead (Reddy, 2005).

2.5.4.4 Security Levels

The ZigBee standard has eight defined security levels that are applied to the NWK and APS layers to indicate how an outgoing or incoming frame is secured. An identifier represents the security level and determines whether the payload of the transmitted frames is encrypted. Furthermore, it determines the amount of data authenticity provided over a frame, which is reflected by the MIC length (ZigBee Alliance, 2017, p. 425). Table 2.1, from ZigBee Alliance (2015), links each of the security levels applied to the NWK and APS layers:

Table 2.1

ZigBee's Security Levels

Security Level Identifier	Security Level Sub-Field	Security Attributes	Data Encryption	Frame Integrity (length M of MIC, in Number of Octets)
0x00	'000'	None	OFF	NO (M=0)
0x01	'001'	MIC-32	OFF	YES (M=4)
0x02	'010'	MIC-64	OFF	YES (M=8)
0x03	'011'	MIC-128	OFF	YES (M=16)
0x04	'100'	ENC	ON	NO (M=0)
0x05	'101'	ENC-MIC-32	ON	YES (M=4)
0x06	'110'	ENC-MIC-64	ON	YES (M=8)
0x07	'111'	ENC-MIC-128	ON	YES (M=16)

Note. Reprinted from 'ZigBee Specification (Document No. 05-3474-22)' (p. 456), by ZigBee Alliance, 2017. Copyright 2017 by ZigBee Alliance.

2.5.4.5 Replay Protection

ZigBee has a replay protection mechanism that prevents replay attacks, in which an attacker retransmits previously captured frames across a ZigBee network. ZigBee frames are embedded with a 32-bit (default) incoming and outgoing counter that determines the packet freshness. When a legitimate device receives a packet with an unsynchronised counter (equal or less than the previously received frame), the packet will be dropped (Ocenasek, 2009). The maximum value of a frame counter is 0XFFFFFFFF, and when the counter reaches this value, then no frame transmission is possible. However, it is unlikely that devices will reach this value in their lifetime. In a ZigBee network, the frame counter is only reset to 0 when the network key is updated, or when the network is reinitialised (Rudresh, 2017b) and has since been reinforced in ZigBee 3.0 (see Section 2.6).

2.5.4.6 Frequency Agility

ZigBee is implemented with a frequency agility security mechanism that is designed to protect the network availability in case of an interference problem, a jamming attack or specific denial of service (DoS) attacks. This process enables the ZigBee network to migrate to a new frequency channel to address these problems (Sarijari, Abdullah, Lo, & Rashid, 2014). As part of the coordinator's core functionality in establishing a network, it conducts a proximity scan to detect existing ZigBee networks and devices within its vicinity and to determine its frequency channel. The coordinator dynamically continues this process after the network has been established by consistently monitoring for signals that could indicate a threat to network availability (Wagh, More, & Kharote, 2015).

Frequency agility is very relevant to ZigBee applications that could potentially endure much harmful interference. For example, ZigBee smart home applications generally operate near various wireless technologies that can operate on the same 2.4Ghz frequency as ZigBee devices, particularly Wi-Fi and certain home appliances (Sarijari et al., 2014). ZigBee uses the frequency agility mechanism as an interference mitigation technique to improve network performance and address potential jamming attacks. This mechanism actively monitors the network for interference on its current operating frequency channel and other channels within its vicinity. If needed, ZigBee will migrate to a new frequency channel with the least level of interference (Sarijari et al., 2014).

Frequency agility is also used in preventing PAN-ID conflicts in ZigBee networks. PAN-ID conflicts can occur naturally when two existing separate ZigBee networks are within each other's range or can result from a DoS attack (Sajjad & Yousaf, 2014). The network manager/coordinator responds to a PAN-ID conflict through a process that migrates the network to a new 16-bit PAN-ID when it detects the same PAN-ID within its vicinity (Farahani, 2008a).

2.5.5 Security Issues and Vulnerabilities

ZigBee's security features have been enhanced since its introduction in 2005. However, its low-cost, low-power design comes at a trade-off that makes it more susceptible to various network attacks (Zillner, 2015). This section discusses concerning security issues and vulnerabilities in the ZigBee protocol. These can be identified as prevalent security issues across the different ZigBee releases.

2.5.5.1 Symmetric Key Issues

Various weaknesses regarding the security of ZigBee's symmetric keys have been identified. ZigBee deploys the AES algorithm with CCM* encryption mode, which is considered adequate. However, because of its memory and processing speed constraints resulting from its low-power design, ZigBee simplifies the encryption process by reusing the same key at each level of the ZigBee protocol stack. Therefore, the exposure of a single symmetric key could compromise the entire network's security (Khanji et al., 2019).

Each network node uses the same active network key. Although secured ZigBee networks are configured to encrypt the network key by using a preconfigured link key before transmitting it, the link key is likely of the 'global' type and is susceptible to exposure (NXP Semiconductors, 2017). The ZigBee standard uses a default value for the link key, which ensures interoperability between ZigBee devices from different manufacturers. However, this aspect introduces the vulnerability of an attack authenticating an unauthorised device onto the network using a default global link key (Zillner, 2015).

The security of ZigBee's symmetric keys is based on the assumption that the keys are securely stored and that devices are preconfigured with the keys so that they are not sent over the network unencrypted (ZigBee Alliance, 2017, pp. 407–408). However, there are exemptions to this assumption. Depending on the security policy, specific ZigBee devices may not be configured with a preconfigured link key, which means a single network key may be

transmitted over the network without protection. This causes a brief moment of vulnerability because an attacker could intercept the network key as it is transported to a pairing device. Although the timing of the pairing phase is narrow, techniques can be applied to trick a user into resetting a device to cause the key to be resent over the network unprotected (Zillner, 2015). Another exemption reflects on the low-cost, low-power design of the nodes, which makes them susceptible to physical attacks for it cannot always be assumed that the hardware is built to be tamper-resistant. Therefore, an attacker with physical access to a ZigBee device may be able to extract its keying material or other privileged information (Zillner, 2015).

2.5.5.2 Denial of Service Issues

ZigBee is known to be susceptible to a range of DoS and distributed denial of service (DDoS) attacks because of its lack of protection mechanisms to mitigate these attacks. These attacks can be targeted towards the different layers of ZigBee's protocol stack and depend on whether the attacker is part of the network (internal) or outside the network (external) (Radmand et al., 2010).

DoS attacks inside the network can be targeted towards the APS/NWK/MAC/PHY layers. For example, an insider attacker could target the APS layer by flooding the network with legitimate messages to interrupt message processing (Chaitanya & Arindam, 2011). Against the NWK layer, DoS can be achieved by eliminating the routing path between nodes, causing alterations to the routing protocol or data loss. Insider attacks are difficult to prevent for the protocol provides limited internal security features once a device joins the network (Radmand et al., 2010). External DoS attacks target the MAC and PHY layers of ZigBee's protocol stack. ZigBee's MAC layer, defined by the IEEE 802.15.4 standard, contains inherent properties that are susceptible to a range of attacks. For example, the IEEE 802.15.4's MAC layer mechanism's Carrier Sense Multiple Access/Collison Avoidance (CSMA), guaranteed timeslot (GTS), and PAN-ID conflict can be exploited to achieve DoS (Stelte & Rodosek,

2013). DoS can be performed on the PHY layer through selective jamming techniques and the physical destruction of nodes (Radmand et al., 2010).

2.6 ZigBee 3.0 Security Advancements

ZigBee 3.0 introduces additional security features and improvements over those included in the previous ZigBee specifications. The ZigBee 3.0 specification is implemented into ZigBee PRO 2015 (or newer) models and provides child device management, a DSM option (discussed in Section 2.5.2) and improved and added security features (Texas Instruments, 2019). ZigBee 3.0's noticeable security updates relevant to this study are discussed next.

2.6.1 Trust Centre Link Key Updates

Introduced into ZigBee 3.0 is an overhaul to the security procedures that request and change keys when a device joins the network. As discussed in Section 2.5.1, encryption is applied on the NWK layer and the APS sublayer. For previous ZigBee standards, devices were not required to update their APS layer encryption key (Link Key) after joining the network. However, ZigBee 3.0 mandates that all devices be updated with a trust centre link key on CSM networks. Upon joining, devices are required to request a randomly generated trust centre link key for encryption of all ongoing APS layer encrypted communications (Silicon Labs, n.d.). This improved feature adds an additional layer of security to the network because a device will not compromise the network key when it leaves the network and rejoins. Moreover, ZigBee 3.0 coordinators have a configurable feature that enables them to accept or reject legacy ZigBee devices that do not initiate the trust centre link key update (Texas Instruments, 2019).

2.6.2 Link Keys Derived from Install Code

ZigBee 3.0 has an additional security feature that provides the option to authenticate with a link key derived from the joining device's install code. In this configuration, the install codes, consisting of 16 bytes of random data + 2-byte cyclic redundancy check, are passed

through a hash function to generate a random trust centre link key (NXP Semiconductors, 2017). Every device supporting ZigBee 3.0 is required to have an install code. Furthermore, it currently provides the most secure method for generating link keys. It allows the user to individually identify joining nodes to the trust centre and guarantees that each joining device has a random link key (Digi International, 2018). Link keys derived from the install code also increase the security of the overall network by eliminating the use of global link keys, which are well known to be vulnerable to exposure (NXP Semiconductors, 2017).

2.6.3 Additional Relay Protection

ZigBee 3.0 has undergone improvements to its NWK frame counter to prevent replay attacks. The protocol now has a persistent NWK frame counter that does not reset its value during a standard or factory reset over the air (OTA). This additional layer of security to the frame counter mechanism prevents an attacker from initiating a network reset to perform a replay attack (Texas Instruments, 2019).

2.7 Related Studies

This section summarises related studies that have been conducted on the security of ZigBee and its revisions over the years. In these studies, researchers have investigated the ZigBee protocol by performing a range of symmetric key- and DoS-related attacks to assess their impact.

A total of five studies were identified that are relevant to this thesis. In Sections 2.7.1–2.7.4, practical studies on the main security components of ZigBee are discussed. Section 2.7.5 analyses a study conducted on the exploitation of ZigBee's remote AT commands to achieve DoS.

2.7.1 X. Fan, Susan, Long and Li (2017)

X. Fan et al. (2017) demonstrated known symmetric key and DoS attacks against the ZigBee protocol. For their experiments, they formed a simple network consisting of three

ZigBee devices, a Samsung SmartThings Hub v2 (coordinator), a Centralite Smart Outlet (router) and an Iris Contact Sensor (end device), each configured with a standard security configuration and default symmetric key values. The researchers used the ZigBee exploitation framework 'KillerBee' and an Atmel Razen RZUSB stick to conduct practical security experiments against their constructed ZigBee network.

In their first experiment, X. Fan et al. (2017) attempted to capture the network key through a man-in-the-middle attack over Wireshark as it was sent to the Centralite Smart Outlet and Iris Contact Sensor OTA. In Wireshark, they captured the encrypted APS command frame sent from the coordinator node containing the network key. Knowing that the network authenticates with a default global link key, they decrypted the encrypted payload of the APS frame using an AES decryption tool to obtain the plain-text network key. The compromised network key could then be applied to decrypt NWK layer communications and interact with the legitimate devices on the network.

In their second experiment, X. Fan et al. (2017) attempted to cause devices to crash through an association flooding attack using the KillerBee framework. As the attack was in motion, they used the SmartThings iOS application to assess how the attack would affect the network's functionality by accessing data from the Iris Contact Sensor powering the Centralite Outlet on and off. Although numerous association request packets were transmitted to each device, the network's functionality did not suffer and it operated as expected.

In their last experiment, X. Fan et al. (2017) attempted to induce a network key transport. They created a spoofed data packet for this attack based on an associated request packet captured from a previous experiment. They inserted the MAC address of their RZUSB device into the extended source field and the corresponding hex values of their malicious packet and sent it over the network. However, because of a hardware limitation they could not verify that the attack worked. The instant response to the spoofed packet did not leave enough

time to transition the single RZUSB stick from a transmitting state to a listening state to capture the induced network key.

2.7.2 Vidgren, Haataja, Patino-Andres, Ramirez-Sanchis and Toivanen (2013)

Vidgren et al. (2013) evaluated several vulnerabilities found in the main security components of ZigBee technology. They proposed two practical attacks that can be performed against ZigBee based on these vulnerabilities, the latter of which they undertook in this study. That is, their practical undertaking was to demonstrate their second proposed attack, the 'ZigBee network sniffing attack', that exploits the unencrypted network key transport vulnerability found in ZigBee networks configured with a standard security level. The researchers aimed to raise discussion on this vulnerability through their attack scenario in the hope of removing the standard security level from the ZigBee specification altogether.

Vidgren et al. (2013) devised an experiment using freely available software tools and cheap ZigBee hardware to keep their experiment practical and straightforward. While a ZigBee coordinator was deployed, they captured the network traffic as an end device joined to the network using a Texas Instruments CC2531 USB dongle. After capturing the joining session, the data were converted from PSD-format (Packet Sniffer Data) to PCAP (Packet Capture) so that the KillerBee tool 'zbdsniff' could interpret the data. The researchers then passed the PCAP file into the zbdsniff tool to search and extract the APS key transport command frame containing the network key. They successfully extracted the unencrypted network key, concluding that the standard security level provides insufficient security and should be avoided in security-critical ZigBee-enabled systems.

2.7.3 Olawumi, Haataja, Asikainen, Vidgren and Toivanen (2014)

Olawumi et al. (2014) conducted a practical study on the security of ZigBee by undertaking three attack experiments using the KillerBee framework. Their experiments focused on exploiting several vulnerabilities found in ZigBee technology and on networks that

do not employ encryption mechanisms to protect their data. They used an Atmel RZ Raven USB stick and KillerBee firmware for sniffing and injecting on ZigBee networks.

In their first experiment, they conducted an information gathering attack to discover ZigBee networks within range and the configuration details of the corresponding ZigBee devices. They used KillerBee's 'zbstumbler' tool with the RZ USB to scan ZigBee channels and discover their ZigBee network constructed in the laboratory. The victim ZigBee networks operating channel, PAN-IDs and stack version were discovered through this attack, providing a foundation for future exploitation.

Then, in their second experiment, Olawumi et al. (2014) moved on to actively capture network data across the discovered operating channel. In this attack, they used KillerBee's 'zbdump' tool to capture packets over a period and save the output capture file for further analysis in Wireshark. Their observation determined that an attacker could decode sensitive captured data over an unencrypted network.

In their final experiment, they used the previously captured data to conduct a simple replay attack. They used the KillerBee tool 'zbreplay' to perform the replay attack by reading the capture file and retransmitting the frames at a pre-specified delay. As their network was unencrypted, the frames did not undergo integrity checking and were therefore acknowledged by the devices on the victim network. The researchers concluded that replay attacks are a straightforward process because an attacker could easily manipulate capture files to retransmit only the necessary frames.

2.7.4 Azzi (2016)

Part of Azzi's (2016) research investigated known vulnerabilities related to the ZigBee symmetric key and DoS through practical experimentation against a constructed XBee/ZigBee network. The attacks performed by this researcher were based on a combination of attacks, which led to various other attacks from the acquired information. The experiments performed

were against a testbed ZigBee network consisting of three nodes, a coordinator, a router and a 'compromised' router node. The ZigBee hardware chosen included three XBee S2 modules that are integrated with the ZigBee PRO version of the protocol.

In the first attack experiment, Azzi (2016) connected the XBee module of the 'compromised' router node to XCTU software and read the AT parameters stored in memory. Azzi performed this attack as a physical information gathering technique to acquire the device and network configurations and potentially keying material. After accessing the device's configuration through XCTU, Azzi found that all configuration parameters, including the security configuration, could be read apart from the symmetric key values, which were 'write-only'. The information gathered in this attack was applied to the subsequent experiments.

Then, Azzi (2016) conducted an attack experiment that exploits the unencrypted network key vulnerability in unsecured ZigBee networks. For this experiment, the researcher deliberately left the link key unconfigured and set the encryptions option as 0 on an XBee module to force the coordinator to send the network key unencrypted OTA to joining nodes. Furthermore, while security is enabled in this configuration, the network is open, allowing specific devices to join without requiring the link key for authentication. Azzi configured a dummy node with a similar security configuration and the matching PAN-ID of the coordinator and placed it within proximity of the ZigBee network. As expected, the attacker's node successfully detected and received the unencrypted network key from the coordinator and joined the network. The attacker's node could then be positioned to perform internal attacks against the network.

In the final attack experiment, Azzi (2016) conducted an internal flooding DoS attack against a victim node. The researcher created a packet within the XCTU console with random data as its payload for this attack. The packet was then injected into the network every 100 milliseconds over 2 minutes, causing the entire network to freeze over the duration of the

attack. Although the nodes came to a standstill, they could operate normally immediately after ending the attack. Azzi concluded that these attacks could be applied to real-world applications, which could have unforgivable consequences.

2.7.5 Vaccari, Cambiaso and Aiello (2017)

Vaccari et al. (2017) conducted a study on the exploitation of ZigBee's remote AT commands. The focus of their research was to cause DoS against a 'target' sensor node using remote AT commands sent from an internal 'attacker' node. They formed a testbed ZigBee network for their experiment, which contained one coordinator node, two end device nodes and one malicious 'attacker' node. Each node functioned on an XBee S2 module integrated with the ZigBee PRO version of the protocol. The coordinator and attacker nodes were connected to a PC through a Raspberry Pi 3 and an XBee USB board, while the sensor nodes were remotely operating on an Arduino UNO R3 and XBee shield. This hardware implementation allowed the researchers to perform the attack experiment and monitor the impact against the targeted sensor nodes.

In their experiment, Vaccari et al. (2017) sent remote AT command packets to a single sensor node to disrupt its ability to transmit readings every 35 seconds to the coordinator without affecting the other nodes. An external ZigBee device captured data over the same channel to measure the attack's impact for a total of 120 seconds over two phases, passive and active. The passive phase monitored the network traffic flow over 50 seconds before initiating the attack against the targeted sensor node. In the active phase, the traffic flow was monitored for 70 seconds while the attack was in motion. The researchers found that shortly after transmitting the malicious remote AT commands, the targeted sensor node became disconnected from the network, and its communication with the coordinator came to a complete halt. However, the other devices on the network could continue to operate normally without any detected interference. Vaccari et al. concluded that the number of packets sent from the

attacker was minimal; therefore, it is not easy to detect a remote AT command attack without undertaking a deep packet inspection.

2.8 Conclusion

This chapter has built a body of knowledge on the ZigBee protocol and its security concepts. It identified how the protocol applies security and cryptographic mechanisms to protect its data, and the prevalent security issues that threaten the confidentiality, integrity and availability of its network and services. The literature review revealed limitations in the research conducted on the latest revision of the protocol, ZigBee 3.0. Although the literature has widely documented prevalent security issues found in the earlier revisions of the protocol, limited studies have reviewed these security issues against ZigBee 3.0, especially as a practical undertaking. Security issues related to the symmetric key and DoS were identified among the prevalent issues found in the main security components of ZigBee.

Therefore, the direction of this study is to investigate the ZigBee 3.0 protocol against these security issues found in the earlier revisions of ZigBee. The research methodology adopted for this study is outlined and presented in Chapter 3. Chapter 3 defines the research questions and approach for investigating the ZigBee 3.0 protocol against the identified prevalent security issues.

Chapter 3: Research Design and Methodology

3.1 Introduction

Chapter 2 surveyed a range of relevant literature on the security concepts of ZigBee and its security issues. In addition, five similar studies through which researchers have undertaken practical approaches to investigate the security of ZigBee against security issues related to the symmetric key and DoS were summarised. The aim of Chapter 3 is to establish an effective research design and methodology to investigate the security of ZigBee 3.0 against the protocol's prevalent security issues.

In Section 3.2, a research question and five supporting sub-questions are outlined that are formulated from existing literature and related studies. Section 3.3 describes the research design that consists of four main phases to answer the proposed research questions. Section 3.4 outlines the security testing design components that apply to the research design to investigate the ZigBee 3.0 protocol as a practical undertaking. This section outlines each security issue and the associated attacks to be tested against ZigBee 3.0, the processes of the utilised security testing framework, the testing environments, the data collection and procedures and the data analysis activities. Section 3.5 concludes the chapter.

3.2 Research Questions

The focus of this study is based on a primary research question formulated through the existing literature and related studies discussed in Chapter 2. It was identified that the ZigBee protocol has security issues that threaten the confidentiality, integrity and availability of its network and services. The most significant concerns prevalent against earlier revisions of ZigBee were the lack of, and limitation to, security services resulting from the low-cost, low-power design of its technology, which make ZigBee-enabled systems susceptible to several symmetric key and DoS attacks. While existing literature has widely covered security issues on ZigBee's earlier revisions, it was identified that limited research has been conducted on the

protocol's latest revision, ZigBee 3.0, particularly in a practical undertaking. Furthermore, the ZigBee 3.0 protocol has improvements to its security services and includes additional security features (NXP Semiconductors, 2017). This research contributes to addressing this limitation in the literature by investigating the impact of symmetric key and DoS security issues on ZigBee 3.0 networks. Thus, the primary research question is as follows:

Research Question 1 (RQ1): ***What impact do symmetric key and denial of service security issues that are prevalent against earlier revisions of ZigBee pose against ZigBee 3.0 networks?***

- **Research Sub-Questions**

To extensively analyse the impact of security issues related to the symmetric key and DoS against ZigBee 3.0 networks, five sub-questions have been derived from the primary research question. These sub-questions, designed to evaluate and explore the security services and features included in the ZigBee 3.0 protocol, are as follows:

Sub-Question 1 (SQ1): *What impact do the exploitation of ZigBee and IEEE 802.15.4's known symmetric key vulnerabilities pose against the security of symmetric keys in ZigBee 3.0 networks?*

Sub-Question 2 (SQ2): *What impact do the exploitation of ZigBee and IEEE 802.15.4's known denial of service vulnerabilities pose against the availability of ZigBee 3.0 networks?*

Sub-Question 3 (SQ3): *What impact do compromised ZigBee and IEEE 802.15.4 symmetric keys pose against the confidentiality of ZigBee 3.0 networks?*

Sub-Question 4 (SQ4): *What methods can be applied to strengthen the security of symmetric keys on ZigBee 3.0 networks?*

Sub-Question 5 (SQ5): *What are the security limitations regarding the security of symmetric keys for 'Distributed Security Model' networks compared with 'Centralised Security Model' networks in ZigBee 3.0?*

3.3 Research Design

The ZigBee 3.0 protocol is investigated against the proposed sub-questions to answer RQ1. Four research phases have been established to achieve this aim, as shown in Figure 3.1. In addition, this research uses a combination of qualitative and quantitative analysis techniques. Qualitative analysis techniques are applied as a primary method to identify, interpret and explain a phenomenon that occurs by exploring the issues that each sub-question attempts to investigate (Creswell & Creswell, 2018). Quantitative analysis is utilised as a secondary technique to select and analyse data sources, predominantly in experiments that attempt to measure an effect numerically on a dependent variable (Morgan, 2014). Primary data are collected through experimentation in this study and are analysed through these research methods to interpret the underlying impact of the identified symmetric key and DoS security issues on ZigBee 3.0 networks. Figure 3.1 represents the research phases applied to this research:

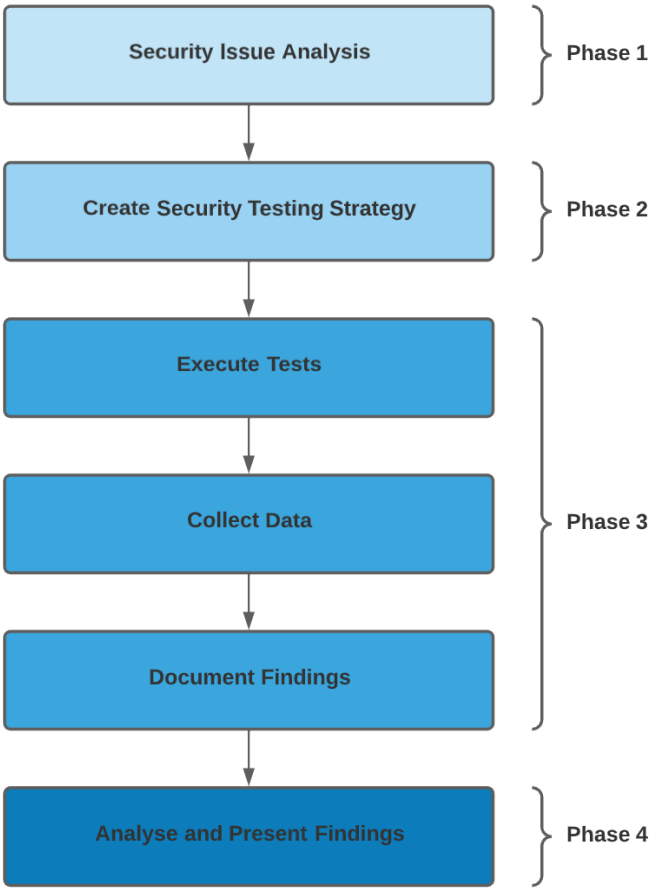

Figure 3.1. Research phases.

In the first research phase, a preliminary investigation is conducted based on the findings of other researchers to identify and analyse the protocol's prevalent symmetric key and DoS security issues. In this phase, the security issues are expanded and categorised with their associated attacks, attack surface and potential impact based on a defined scope. This phase aims to establish an attack model of these security issues that can be implemented into the design and execution of the security testing strategy.

A security testing strategy is created in phase 2. This phase outlines how the security issues and associated attacks will be tested against ZigBee 3.0 as practical experiments. The ZigBee 3.0 hardware and supported software that will be used to undergo analysis for the experiments are identified. In this phase, ZigBee 3.0 networks are designed and constructed as a testbed, and their base configuration using manufacturer-provided documentation is created.

Moreover, the hardware and software tools utilised to perform the security testing experiments are identified.

In phase 3, security testing experiments are performed against ZigBee 3.0 networks following the security testing strategy established in phase 2. Qualitative and quantitative data are collected through conducting these experiments, and findings are generated and documented to interpret the data and prepare for analysis in the next phase.

The acquired data and documented findings are analysed and presented in phase 4. Qualitative and quantitative analysis methods are applied to the gathered data and findings to produce written reports or graphical displays of data that describe the impact inflicted by the security testing experiments against ZigBee 3.0. Last, the findings related to each sub-question are combined to support the main research question and determine the overall impact that prevalent symmetric key and DoS security issues pose against ZigBee 3.0 networks.

3.4 ZigBee 3.0 Security Testing Design

A series of security tests are performed against ZigBee 3.0 networks to gather the necessary data to answer RQ1 and each sub-question. These tests are designed to evaluate the impact of symmetric key- and DoS-related attacks on ZigBee 3.0 that were prevalent against the earlier revisions of the ZigBee protocol. Practical attacks are performed against testbed ZigBee 3.0 networks, and in necessary symmetric key experiments, both security models (CSM and DSM) are evaluated. The data and findings gathered through these tests will be used to investigate how the security configuration of each security model type can mitigate or address the symmetric key security issues and increase the overall security of symmetric keys across ZigBee 3.0 networks. Furthermore, the security limitations of the symmetric keys can be compared across the two ZigBee 3.0 security models.

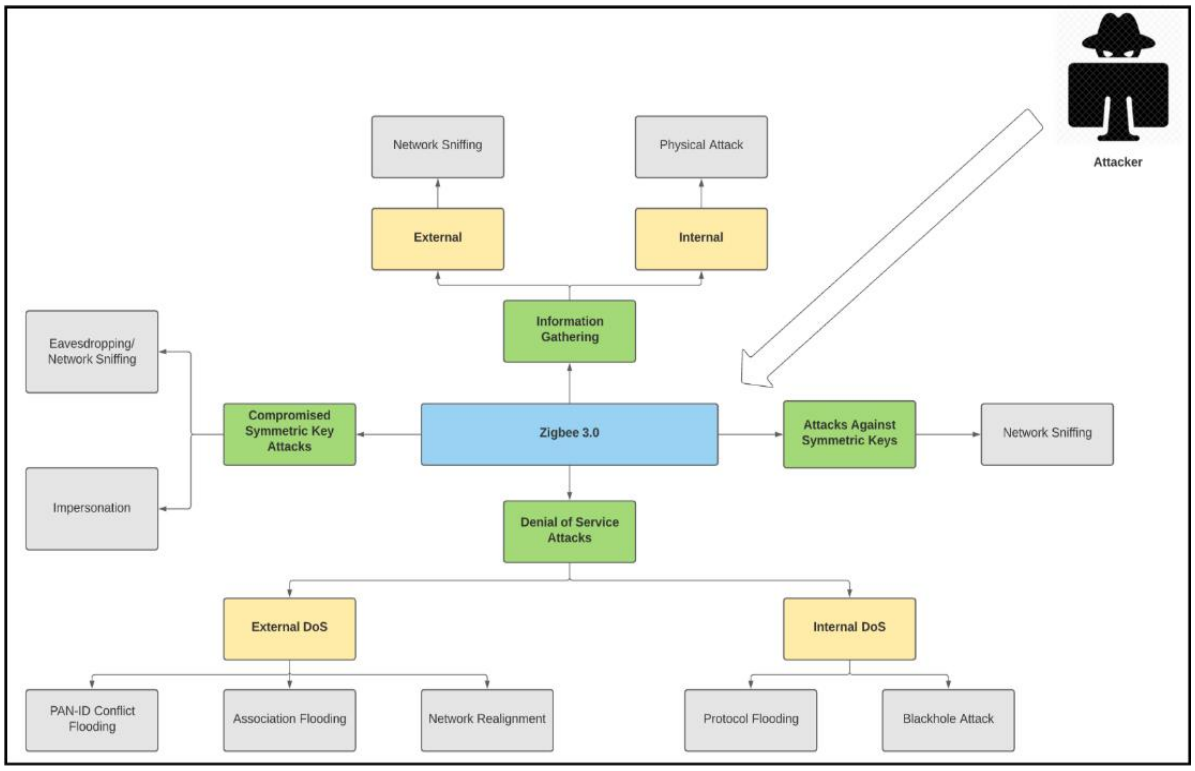

Figure 3.2. ZigBee symmetric key and DoS attack model.

Figure 3.2 portrays an attack model derived from the analysis of security issues in Section 3.4.1. The model shows the different types of attacks and their surface resulting from the symmetric key and DoS security issues in the ZigBee protocol.

3.4.1 Scope of Security Testing Experiments

The following scope has been established to define the extent of attacks to be tested against ZigBee 3.0 in the security testing experiments. This scope applies to the analysis and categorisation of associated attacks listed under each security issue in Section 3.4.2:

- **In-scope attacks are**
  - attacks that target known symmetric key and DoS vulnerabilities,
  - internal and external attacks that are within the capabilities of the exploitation hardware and software outlined in Section 3.4.4.3, and
  - security testing experiments, which are to be limited against the ZigBee 3.0 hardware outlined in Section 3.4.4.1.

- **Out-of-scope attacks are**
    - physical attacks that risk damaging the equipment (node destruction, firmware extraction),
    - replay and similar attacks that have knowingly been addressed/reinforced in the ZigBee 3.0 protocol, and
    - attacks beyond the capabilities of the utilised hardware and software (radio jamming/interference attacks).

3.4.2 Security Issue Analysis

By examining the existing literature and related studies in Chapter 2, the symmetric key and DoS security issues prevalent in the ZigBee protocol could be defined and categorised. The security issues are separated into three categories, 'Security of Symmetric Keys', 'Compromised Symmetric Keys' and 'Insufficient Denial of Service Protection Mechanisms'. Each security issue categorises the possible attacks within the defined scope that can be performed against ZigBee networks.

3.4.2.1 Security Issue 1: Security of Symmetric Keys

It was identified that an attacker could maliciously obtain the symmetric keys in several ways by exploiting known vulnerabilities in unsecured networks. Table 3.1 outlines each of the known attacks and vulnerabilities that pose a threat to the security of ZigBee's symmetric keys.

Table 3.1

Attacks Against Symmetric Keys

Attack	Vulnerability	Description	Possible Impact
Unauthorised network access: Authenticating an unauthorised device	Default link key values	Networks that authenticate using default link-key values are vulnerable to unauthorised network joining. An attacker could	Compromised network data

Attack	Vulnerability	Description	Possible Impact
		authenticate a rogue device using a default link key (Zillner, 2015)	
Network sniffing: Intercepting and decrypting the network key	Default link key values	The network key is generally encrypted with the link key during the authentication process before transmitting OTA to joining devices. An attacker could capture and decrypt the network key as it is sent to a joining device using a default global link key (NXP Semiconductors, 2017)	Compromised network key
Network sniffing: Intercepting the unencrypted network key	Unencrypted network key transport (OTA)	ZigBee networks that do not have a link key configured by the trust centre pose a threat to the security of the network key. The network key will be sent unencrypted to devices joining the network with a matching PAN-ID and the channel on which the network is operating (Vidgren, Haataja, Patino-Andres, Ramirez-Sanchis & Toivanen, 2013)	Compromised network key

Note. OTA = Over the Air, PAN-ID = Personal Area Network Identifier.

3.4.2.2 Security Issue 2: Compromised Symmetric Keys

If an attacker were to obtain ZigBee's symmetric keys through one or more techniques, the confidentiality of the network could be breached (Zillner, 2015). Table 3.2 outlines each of the identified attacks that could be inflicted against a ZigBee network using compromised symmetric keys.

Table 3.2

Compromised Symmetric Key Attacks

Attack	Compromised Key	Description	Possible Impact
Eavesdropping/Network sniffing	Network key	ZigBee encrypts broadcast messages with the network key shared between all devices on the network.	Compromised broadcast/NWK layer communications

Attack	Compromised Key	Description	Possible Impact
		An attacker with this key could capture and decrypt all communications broadcast on the network (Radmand et al., 2010)	
Eavesdropping/Network sniffing	Link key	ZigBee encrypts unicast data with the link key that is shared between two devices. An attacker with the link key could capture and decrypt unicast communications between devices (Radmand et al., 2010)	Compromised unicast/APS layer communications
Impersonation	Network key	An attacker could impersonate the identity of a legitimate node by spoofing broadcast messages with the network key (Radmand et al., 2010)	Compromised network data

Note. APS = Application Support Sublayer, NWK = Network.

3.4.2.3 Security Issue 3: Insufficient Denial of Service Protection Mechanisms

Researchers have identified that ZigBee is susceptible to several DoS attacks owing to its lack of protection mechanisms. DoS attacks can be performed against the different layers of ZigBee's protocol stack and depend on whether the attacker is part of the network (internal) or outside the network (external; Radmand et al., 2010). Table 3.3 outlines various DoS attacks that are applicable to ZigBee networks and their possible impact against network availability.

Table 3.3

Denial of Service Attacks

Attack	Internal/External	Description	Possible Impact
PAN-ID conflict flooding (KillerBee attack)	External	Exploiting ZigBee's frequency agility mechanism by manipulating PAN-ID changes. In this attack, the network is flooded with found PAN-IDs to trigger PAN-ID changes on	• Affects coordinator and router node capabilities • Crashes nodes

Attack	Internal/External	Description	Possible Impact
		the network manager (River Loop Security, n.d.-b).	
Association flooding (KillerBee attack)	External	Repeatedly transmitting spoofed association request packets to the target PAN-ID (River Loop Security, n.d.-b).	• Affects coordinator and router node capabilities • Crashes nodes
Network realignment (KillerBee attack)	External	Spoofing an IEEE 802.15.4 realignment frame from the coordinator to a target device (River Loop Security, n.d.-b).	• Disconnects victim node (resets PAN-ID/Channel) • Causes data loss
Protocol flooding	Internal	Flooding a victim node with legitimate messages from inside the network (Chaitanya & Arindam, 2011).	• Leads to unfair network resource consumption • Causes data loss • Crashes victim node
Blackhole Attack (Exploiting Remote AT Commands)	Internal	Paralysing a victim node's ability to relay or receive packets from its neighbouring nodes. This attack can be performed by exploiting remote AT commands from inside the network (Vidgren, Haataja, Patino-Andres, Ramirez-Sanchis & Toivanen, 2013).	• Causes data loss • Disconnects victim node from the network • Change the routing structure

Note. AT = Attention, NWK = Network, PAN-ID = Personal Area Network Identifier.

3.4.3 Security Testing Framework

The processes of the security testing framework of the National Institute of Standards and Technology (NIST) have been adopted in this research as a guideline to ensure validity in testing and reporting. Among the different security testing frameworks, that of NIST was chosen because it provides a simple foundation for security testing that can be applied to ZigBee through its four stages (see Figure 3.3).

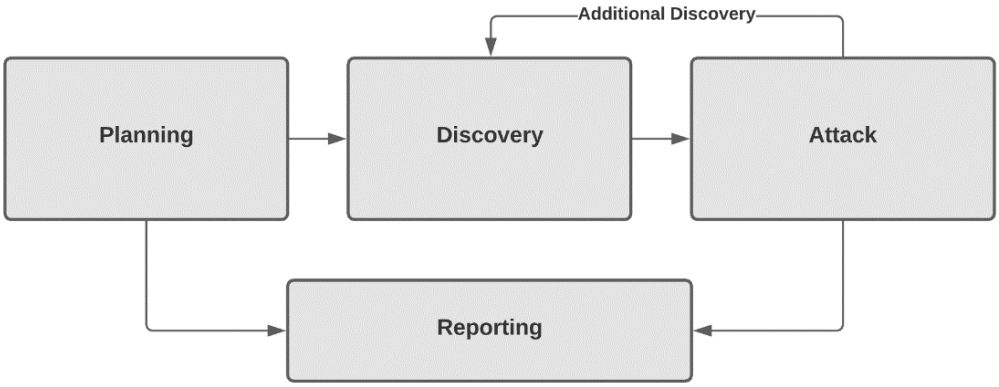

Figure 3.3. Four-stage penetration testing methodology. Adapted from 'Technical Guide to Information Security Testing and Assessment', by K. Scarfone, M. Souppaya, A. Cody, and A. Orebaugh, 2008, *National Institute of Standards & Technology Special Publication, 800*(115), p. 37.

The details of the four phases are as follows:

- **Planning:** In this phase, the groundwork for the security test is established. The test to be performed on ZigBee is planned and outlined, and the steps of engagement are defined. The ZigBee network is deployed and configured in a manner that is suitable for the test. No actual testing is performed in this phase.
- **Discovery:** In this phase, information is collected from an attacker's perspective that is required to perform the attacks. Information gathering techniques are applied in this phase against the victim network to gather:
    - the network's operating channel,
    - PAN-IDs,
    - device MAC addresses, and
    - keying material (where applicable).
- **Attack:** In this phase, the attack is executed using the information collected in the discovery phase. Network analysis tools will assist in verifying that the attack was

successful. Where necessary, solutions can be identified to mitigate or address the attack.

- **Reporting:** Reporting is an ongoing process that is performed simultaneously with the three other phases. The report will outline each step of the test and the impact caused by the attack.

3.4.4 Testing Environments

The security testing experiments are performed against a ZigBee 3.0 network, and where necessary, against the two security models: CSM and DSM (see Figure 3.4), which differ in device authentication and message protection mechanisms (X. Fan et al., 2017). For this research, it is appropriate to perform symmetric key-related experiments against both security models. Moreover, the number of nodes for each experiment type will vary to control the amount of generated traffic and network data.

In this section, the ZigBee hardware and software used to construct the testbed ZigBee 3.0 networks are identified, and their base configuration that is applied to each experiment is outlined. The exploitation hardware and software tools to perform the security testing experiments are then outlined and discussed.

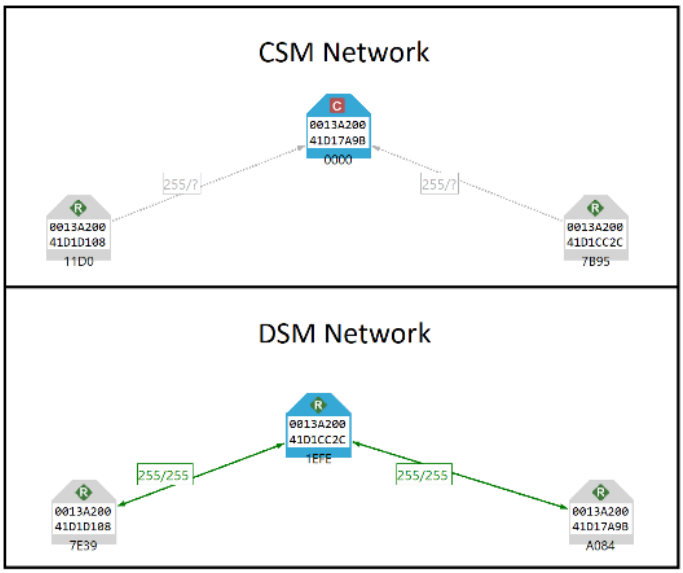

Figure 3.4. ZigBee 3.0 security models shown in XCTU network scan.

3.4.4.1 ZigBee Hardware and Software Setup

This section outlines the hardware and software used to construct the testbed ZigBee 3.0 networks supporting the CSM and DSM security models. A general overview of their workings is provided in the next sections.

3.4.4.1.1 ZigBee Hardware Setup

The following hardware selected for this research is appropriate for it allows devices to be configured and programmed to suit each experiment. Table 3.4 provides an overview of the hardware used to build each node.

Table 3.4

ZigBee Node Hardware

Coordinator Node	Router Nodes	End Device Nodes
• XBee 3 Pro Module • XBee Development Board • Antenna	• XBee 3 Pro Module • XBee Development Board • Antenna	• XBee 3/XBee 3 Pro Modules • Waspmote v1.5 Development Board • Antenna

The details of the hardware used are as follows:

- **Computer (Windows 10):**

A desktop computer hosting Windows 10 is used to configure, monitor and maintain the ZigBee 3.0 testbed networks. The coordinator and router nodes connect to the computer's USB ports to establish a gateway between the ZigBee network and the PC.

- **XBee 3:**

A total of five XBee 3 Pro (XB3-24Z8ST) and three XBee 3 (XB3-24Z8PT-J) modules are used to construct the testbed ZigBee 3.0 networks. These radio modules allow the supported hardware to operate as a coordinator, a router or an end device node on the network through the configured Application Programming Interface (API) mode. XBee 3 modules are integrated with the ZigBee 3.0 protocol and operate on the ISM 2.4 GHz frequency (Digi International,

n.d.-a). Furthermore, they are compatible with several third-party devices that support the ZigBee protocol, including the end device's Waspmote v1.5 development boards used in this research.

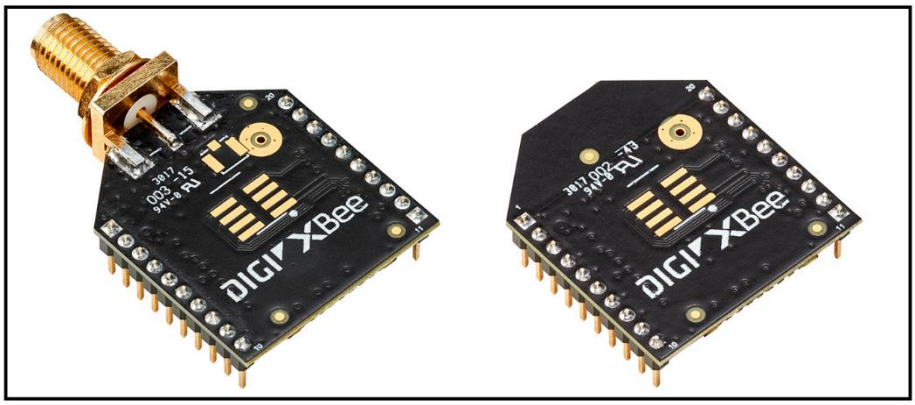

Figure 3.5. XBee 3 modules.

The gateway (coordinator and router) nodes utilise the pro series of XBee 3 for optimal range and data throughput. However, the security specification of ZigBee 3.0 between the pro and standard models are the same and include 128-bit AES encryption over 16 channels (Digi International, n.d.-a).

- **XBee Development Board:**

Three XBee Grove Development Boards are used to construct the gateway nodes for this research. These boards allow the data flowing between the ZigBee network and PC to be collected through a standard USB port. Furthermore, they act as a data bridge or access point between the ZigBee network and receiving equipment (Libelium, n.d.-a). The XBee 3 modules connect to the boards through the grove connectors, which can then be evaluated through the XCTU software on a PC. Each node is initially configured with the XBee development boards. During the experiments, the gateway nodes consistently operate on the XBee development boards connected to the computer's USB port, allowing the network data to be collected through the XCTU software.

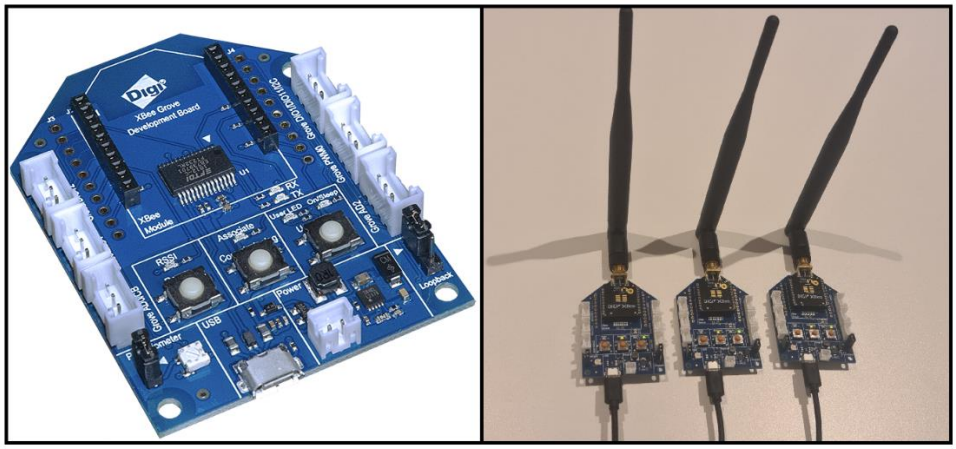

Figure 3.6. XBee development board and gateway nodes.

- **Waspmote v1.5 Development Board:**

The Waspmote v1.5 is hardware designed to integrate with several IoT technologies, including the ZigBee protocol (Libelium, n.d.-b). These boards operate on a battery pack and contain a program uploaded through the Waspmote IDE software, which allows them to run completely autonomously. The ZigBee 3.0 testbed networks of this research contain up to five end device nodes that are created with a Waspmote v1.5 board and an XBee 3 module, as shown in Figure 3.7.

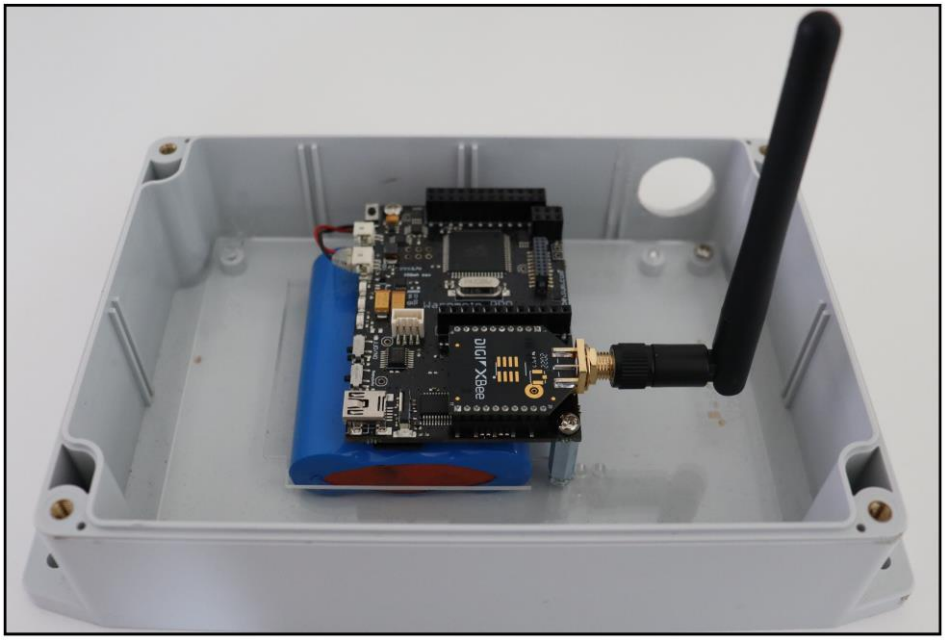

Figure 3.7. End device (Waspmote) node.

3.4.4.1.2 ZigBee Software Setup

The software implementation of this research is used to work alongside the ZigBee hardware to construct, maintain and monitor the ZigBee 3.0 networks. The utilised software is as follows:

- **XCTU:**

Digi XCTU is a configuration and test utility software used extensively in this research to update, configure and manage the XBee 3 radio modules. XCTU is used to set the node's parameters, including the network's security configuration and policy, along with other essential settings that determine how nodes communicate. XCTU locally connects the XBee 3 modules via the USB interface of the XBee development boards.

The XCTU software has a network scan function that is used to monitor the network internally with its graphical display of nodes, pathways and their respective signal strengths (see Figure 3.8; Digi International, n.d.-b). In each experiment, the XCTU mapping function is initially used to verify that the network is correctly formed. The mapping function will then be used in specific experiments from gateway nodes to assess changes to the network over time. Another function of XCTU used in this research is the frames generator tool, which creates and sends custom API frames from locally connected XBee 3 modules. This tool can be used to create any frame supported by the ZigBee 3.0 protocol and is used in specific experiments to verify the security of a sent frame or initiate an internal attack.

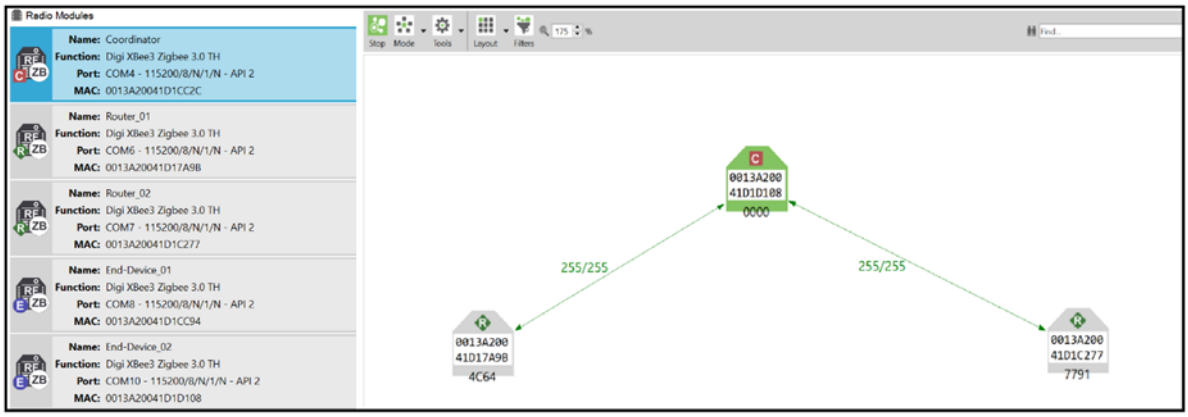

Figure 3.8. XCTU locally connected modules and network scan.

- **Waspmote IDE:**

The Waspmote IDE software is used to programme functionality onto the Waspmote v1.5 development boards allowing the end device nodes to be autonomous. The IDE contains a library of example codes that can be easily modified and uploaded to the Waspmote boards.

3.4.4.2 ZigBee Base Configuration

The testbed ZigBee 3.0 networks are configured with an initial base configuration that does not change throughout the experiments (see Appendix A for the complete device configuration). The security configuration and security model type (CSM and DSM) are not included in the base configuration since these will be configured to suit each experiment. Furthermore, features that are exclusive to XBee 3 or are not part of the ZigBee 3.0 protocol are excluded from the base configuration. In detail:

- **Firmware:**

  The XBee 3 modules in this research are programmed with the following firmware:

  - product family: XB3-24,
  - function set: Digi XBee3 ZigBee 3.0 TH, and
  - firmware version: 100D.

- **XBee 3 Base Configuration:**

The base configuration enables nodes to operate in API Mode With Escapes (AP = [2]), allowing every node to send and receive data. The end device nodes can enter a cyclic sleep (SM = [4]) when their operation is not needed, and sleep modes are disabled on gateway nodes (SM = [0]). The baud rate for the end device nodes is adjusted to 115200 (BD = [7]) for compatibility between the Waspmote v1.5 development board and the XBee 3 module. Older generation devices are unable to join the network (C8 = 0).

Table 3.5

XBee 3 Base Configuration

	Forming Node	Joining Node (Gateway)	Joining Node (End device)
Networking:	CE = Form Network [1] ID = 0 CR = 3 JV = Disabled [0] DC = 0 C8 = 0	CE = Join Network [0] ID = (ID from Forming Node) CR = 3 JV = Disabled [0] DC = 0 C8 = 0	CE = Join Network [0] ID = (ID from Forming Node) CR = 3 JV = Disabled [0] DC = 0 C8 = 0
Discovery options:	NI = (Node Name)	NI = (Node Name)	NI = (Node Name)
Security:	-	-	-
Sleep settings:	SM = No Sleep (Router) [0]	SM = No Sleep (Router) [0]	SM = Cyclic Sleep [4]
API configuration:	AP = API Mode With Escapes [2]	AP = API Mode With Escapes [2]	AP = API Mode With Escapes [2]
UART interface:	BD = 9600 [3]	BD = 9600 [3]	BD = 115200 [7]

Note. Adapted from XBee3 802.15.4 RF Module User Guide, by Digi International, 2020.

The forming node creates the network (CE = [1]) and generates a random 64-bit extended PAN-ID (ID = 0) that other radio modules will join. The joining nodes join (CE = [0]) the network of the defined PAN-ID (ID) set by the forming node.

3.4.4.3 Security Testing Setup

This section identifies the hardware and software used to perform the security testing experiments against the testbed ZigBee 3.0 networks (see Table 3.6). A general overview of their workings is given.

Table 3.6

Security Testing Hardware and Software

Component	External Attacks/Network Analysis	Internal Attacks (Compromised Node)
Hardware	• Laptop (Kali Linux) • 2x ApiMote (Flashed with KillerBee) with Antenna • 1x CC2531 USB Dongle (Flashed with ZBOSS)	• XBee 3 Pro • XBee Development Board • Antenna
Software	• Kali Linux 2018-3 • KillerBee • Wireshark (ZBOSS)	• XCTU

3.4.4.3.1 Security Testing Hardware

The following hardware is used alongside the software to perform the security testing experiments against the testbed ZigBee 3.0 networks:

- **Research Laptop:**

A generic research laptop powered by Kali Linux OS (see Figure 3.9) is used to conduct external attacks. The laptop hosts the required software to perform the security tests and monitor/analyse the ZigBee networks.

- **ApiMote:**

ApiMote is a ZigBee security research hardware and exploitation tool designed to evaluate the security of ZigBee/IEEE 802.15.4 networks. ApiMote is developed and designed by River Loop Security and is pre-flashed with the python-based KillerBee framework. This hardware can sniff and inject on ZigBee/IEEE 802.15.4 networks (River Loop Security, n.d.-

a). This research utilises two ApiMote v.4 BETAs to simultaneously capture and inject packets against the ZigBee 3.0 networks. Each ApiMote has a screw-on antenna for optimal range and connects to a computer powered by Kali Linux via a mini-USB (see Figure 3.9).

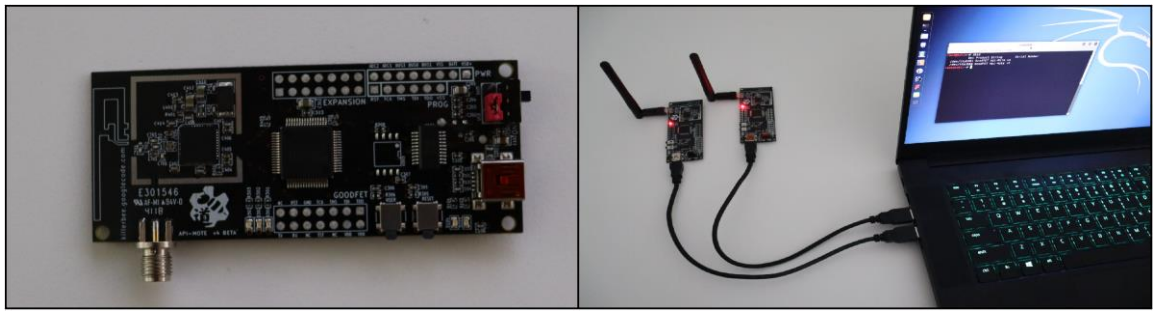

Figure 3.9. ApiMote v4 and research laptop.

- **CC2531 USB Dongle:**

The ZigBee network traffic is externally captured over Wireshark using a CC2531 USB dongle. With a Texas Instruments CC Debugger, the CC2541 USB dongle was initially flashed with ZBOSS sniffer firmware to support packet capturing over ZigBee channels through Wireshark.

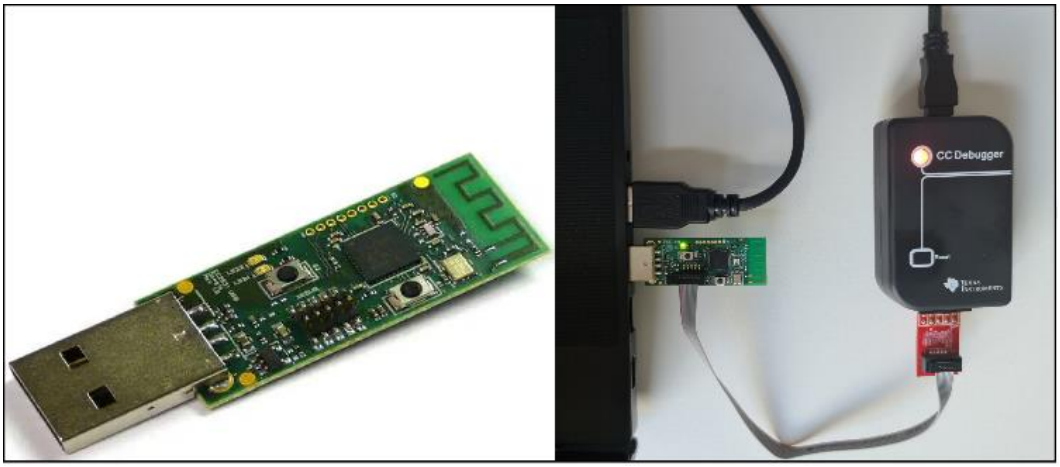

Figure 3.10. CC2531 USB dongle and Flashing ZBOSS Firmware with CC Debugger.

- **Compromised Nodes:**

In specific experiments, attacks are initiated inside the network (internal) from a compromised node. The nodes consist of an XBee 3 module and an XBee development board

(see Section 3.4.4.1) and already contain the network and security configuration of the victim network, allowing it to be easily tampered with to perform internal attacks.

3.4.4.3.2 Security Testing Software

The following software is used alongside the hardware to perform the security testing experiments against the testbed ZigBee 3.0 networks:

- **Kali Linux 2018-3:**

The external attack experiments are initiated from a single Kali Linux virtual machine. Kali Linux was chosen because it is preconfigured with the software dependencies required to run the KillerBee framework. Moreover, it includes the packet analysing software Wireshark that assists in evaluating ZigBee 3.0 networks. This research uses Kali Linux 2018-3 release for its stability with the KillerBee framework.

- **KillerBee Framework:**

KillerBee is a python-based, open-source framework and software tool designed for exploring and evaluating the security of ZigBee and IEEE 802.15.4 networks (River Loop Security, n.d. -b). The framework, first authored by Wright (2009), has since had contributors over the years, who have improved the framework and added further capabilities to the KillerBee arsenal (River Loop Security, n.d.-b). The requirements to use this framework include a Linux system and a transceiver compatible with KillerBee, including ApiMote.

- **Wireshark:**

Wireshark is software that is extensively used in this research for network and packet analysis. This software is an open-source packet analysing tool capable of actively capturing packets across ZigBee/IEEE 802.15.4 specified channels through a ZBOSS flashed CC2541 USB interface. The capture sessions can be saved as a '.pcap' file for later analysis.

3.4.5 Data Collection

The primary data collected through experimentation consist of:

External Network Data: Wireshark Capture Sessions

Internal Network Data: XCTU Network Scans and Gateway Console Logs

The external network data consist of the network traffic captured from outside the network using the hardware and software described in Section 3.4.4.3. The internal network data consist of observational findings shown in XCTU network scans and console logs retrieved from XCTU on the gateway nodes. The data collection procedures conducted in each experiment to ensure valid data collection within their respective NIST framework phase are outlined in Table 3.7.

Table 3.7

Data Collection Procedures

Phase	Procedure
Planning	• Reset XBee modules: Ensure hardware does not contain pre-existing data. • Configure XBee modules: Apply base configuration and appropriate security configuration to suit the experiment. • Deploy and monitor nodes: Ensure the network is operating correctly through XCTU network scanning.
Discovery	• Gather information: Acquire necessary network information to execute the attack. • Start data collection: Externally/Internally capture network data.
Attack	• Execute attack: Start the attack with the acquired data from the discovery phase. • Monitor network: When necessary, perform an XCTU network scan to monitor changes to the network and nodes. • End data collection: Save the captured data for later analysis.
Reporting	• Report findings: Evaluate the acquired data and ongoing processes of each phase. • Check validity: If errors occur at any stage, the experiment will be reset to the planning phase to ensure valid data collection.

3.4.6 Data Analysis

The data collected through the security testing experiments are analysed through a combination of qualitative and quantitative methods. Qualitative methods are applied to produce written findings, primarily through the direct observation of the experiments, and are

applied to each security issue under assessment. In addition, quantitative analysis methods are used to create graphical data displays of experiments that attempt to measure the impact of a DoS attack numerically. Table 3.8 summarises the connection between the applied research method and the security issue, and it shows the activities and outputs during data analysis.

Table 3.8

Data Analysis Methods, Activities and Outputs

Security Issue	Qualitative Analysis:		Quantitative Analysis	
	Activity	*Output*	*Activity*	*Output*
Security of symmetric keys	• Observing the processes that lead to symmetric key compromise through externally captured data	Written findings	N/A	N/A
Compromised symmetric keys	• Observing the external network data that becomes exposed resulting from a compromised symmetric key • Observing the impact of an attack through internal network data	Written findings	N/A	N/A
Insufficient denial of service protection mechanisms	• Observing the processes of an attack through external network data • Observing the impact of an attack through internal network data	Written findings	Measuring the impact of an attack through internal network data	Graphical findings

3.5 Conclusion

Chapter 3 presented the research methodology and design that applies to the research components of this study. A primary research question and five supporting sub-questions were established to undergo investigation using the proposed methods. The chapter discussed the security testing design components that outlined how ZigBee 3.0 will be tested against the

prevalent security issues through practical experimentation. Next, Chapter 4 presents the research findings of the security testing experiments against the ZigBee 3.0 testbed networks.

Chapter 4: Research Findings

4.1 Introduction

Chapter 3 outlined the research methodology and security testing strategy for evaluating ZigBee 3.0 against the identified and prevalent security issues affecting the earlier revisions of ZigBee. A primary research question was established, along with five supporting sub-questions that are based on the review of relevant literature and similar studies conducted in Chapter 2.

In Chapter 4, the findings gathered from the security testing experiments that analyse the impact of prevalent security issues on ZigBee 3.0 are presented in a graphical and narrative format. Section 4.2 provides an overview of each experiment, showing the relevant research sub-questions that each experiment investigates. The first set of preliminary experiments is presented in Section 4.3, which demonstrates the information gathering techniques performed in the discovery phase to acquire the necessary information for the future exploitation of ZigBee 3.0. In Section 4.4, the findings regarding the security of symmetric keys are presented, after investigating known symmetric key vulnerabilities, and a method to secure the keys in ZigBee 3.0 is proposed. Section 4.5 presents findings on the impact of compromised symmetric keys on ZigBee 3.0 networks. Last, Section 4.6 presents the results of several DoS attacks performed to investigate ZigBee's lack of DoS protection mechanisms.

4.2 Overview of Experiments

A series of security testing experiments are conducted against ZigBee 3.0 networks in accordance with the research methodology and security testing strategy outlined in Chapter 3. The experiments of this study analyse each of the identified security issues by performing attacks against ZigBee 3.0 through various tests (see Appendix B for individual security test processes). Furthermore, the number of nodes and the security model vary between experiments and the security issue under assessment.

The tables in this section provide an overview of each experiment presented in this chapter to analyse the identified security issues. Moreover, the relevant sub-question (SQ) that each experiment attempts to investigate is linked to the experiments.

4.2.1 Discovery Phase: Information Gathering

The discovery phase consists of preliminary experiments to set the foundation of this study (see Table 4.1). These demonstrate information gathering techniques that can be applied to all post experiments to gather vital network information required to exploit ZigBee 3.0 networks:

Table 4.1

Discovery Phase Experiments

Experiment	Section	Relevant Research Question(s)
1. External Information Gathering	4.3.1	N/A
2. Internal/Physical Information Gathering	4.3.2	N/A

4.2.2 Security Issue 1: Security of Symmetric Keys

Table 4.2 outlines the experiments to analyse the security of symmetric keys in ZigBee 3.0, which are described in detail in Section 4.4. Experiments 3 and 4 are attack experiments that exploit known vulnerabilities on unsecured ZigBee 3.0 networks to compromise their symmetric keys. Experiment 5 demonstrates a method that can significantly improve the security of symmetric keys in ZigBee 3.0.

Table 4.2

Security Issue 1 Experiments

Experiment	Section	Relevant Research Question(s)
3. Unencrypted Network Key Attacks	4.4.1.1	SQ1 SQ5
4. Default Link Key Attacks	4.4.1.2	SQ1
5. Securing Symmetric Keys with Install Codes	4.4.2	SQ4

Experiment	Section	Relevant Research Question(s)
		SQ5

4.2.3 Security Issue 2: Compromised Symmetric Keys

As outlined in Table 4.3, a series of experiments are performed against ZigBee 3.0 networks in Section 4.5 based on the assumption that an attacker has already obtained one or more of the victim network's symmetric keys. These experiments analyse the likely impact of a compromised symmetric key on the confidentiality of ZigBee 3.0 networks.

Table 4.3

Security Issue 2 Experiments

Experiment	Section	Relevant Research Question(s)
6. Key Sniffing (Eavesdropping)	4.5.1.1	SQ3 SQ4 SQ5
7. Packet Decryption (Eavesdropping)	4.5.1.2	SQ3 SQ4 SQ5
8. Node Impersonation Attack	4.5.2	SQ3 SQ4 SQ5

4.2.4 Security Issue 3: Insufficient Denial of Service Protection Mechanisms

Table 4.4 outlines the DoS attack experiments in Section 4.6 against a ZigBee 3.0 CSM network. These experiments attempt to exploit ZigBee's lack of DoS protection mechanisms from outside (external) and inside (internal) the network to evaluate their impact on ZigBee 3.0.

Table 4.4

Security Issue 3 Experiments

Experiment	Section	Relevant Research Question(s)
9. PAN-ID Conflict Flooding (External DoS)	4.6.1.1	SQ2
10. Association Flooding (External DoS)	4.6.1.2	SQ2
11. Network Realignment Attack (External DoS)	4.6.1.3	SQ2
12. Protocol Flooding (Internal DoS)	4.6.2.1	SQ2
13. Blackhole Attack Using Remote AT Commands (Internal DoS)	4.6.2.2	SQ2

4.3 Information Gathering on ZigBee 3.0

Among the five phases of the NIST framework, information gathering is conducted in the discovery phase. Information gathering involves applying techniques to actively interact with the victim ZigBee network to gather as much information as possible, including its operating channel, PAN-IDS, MAC addresses and network configuration. The network information collected in this phase can be used for future exploitation of the network. Furthermore, the information gathering techniques demonstrated in this section are applied to each experiment performed in this research. The experiments of this section are preliminary and were performed against a ZigBee 3.0 CSM with encryption enabled:

Table 4.5

Information-Gathering Experiment Descriptions

Test ID	Network	Test Description
Experiment 1: External Information Gathering		
T01	CSM	Externally interacting with the victim network to gather information for future exploitation
Experiment 2: Internal/Physical Information Gathering		
T02	CSM	Reading AT parameters stored in the memory of compromised ZigBee devices

4.3.1 External Information Gathering

External information gathering attacks are performed outside of the network against the victim ZigBee 3.0 network. The techniques displayed in this section are initiated from a Kali Linux powered laptop, two ApiMotes (flashed with KillerBee firmware) and a CC2531 dongle (flashed with ZBOSS firmware).

4.3.1.1 Experiment 1: External Information Gathering

4.3.1.1.1 Network Discovery and Operating Channel

The first stage in the active information gathering phase is to discover the victim network and its operating channel. In ZigBee, channels can range from 11 to 26. The KillerBee tool ‘zbstumbler’ is used with the ApiMotes to discover the operating channel of the network set up in the lab. The tool ‘zbstumbler’ works by actively sending out beacon requests across a channel, and then waiting momentarily for a response. After a defined interval, the tool hops to the next channel and repeats this process (River Loop Security, n.d.-b). The attacker will know the operating channel when a ZigBee device responds to a beacon.

Figure 4.1 show the execution of the tools ‘zbid’ and ‘zbstumbler’ in the Kali Linux terminal. The tool ‘zbid’ is executed first to identify the interfaces of locally connected ApiMotes. A response on an ApiMote is then received through ‘zbstumbler’ on channel 13, indicating that a ZigBee network is active on that channel:

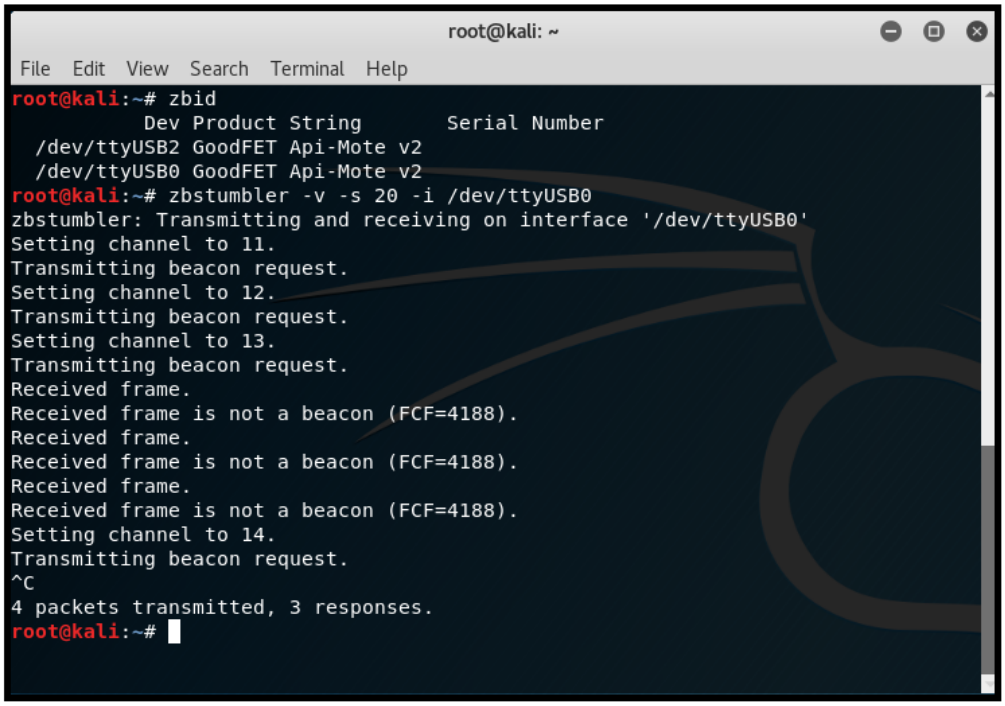

Figure 4.1. Discovering ZigBee network's operating channel with KillerBee.

4.3.1.1.2 Network Sniffing

Wireshark can actively capture packets over the discovered ZigBee network operating channel using the CC2541 dongle flashed with ZBOSS firmware. Figure 4.2 shows the ZigBee broadcast communications captured originating from the coordinator and router (gateway) nodes. When inspecting the contents of a broadcast packet, the source and extended MAC addresses and 16-bit PAN-ID can be extracted. Moreover, it is shown within the security header that the payload is encrypted with a network key:

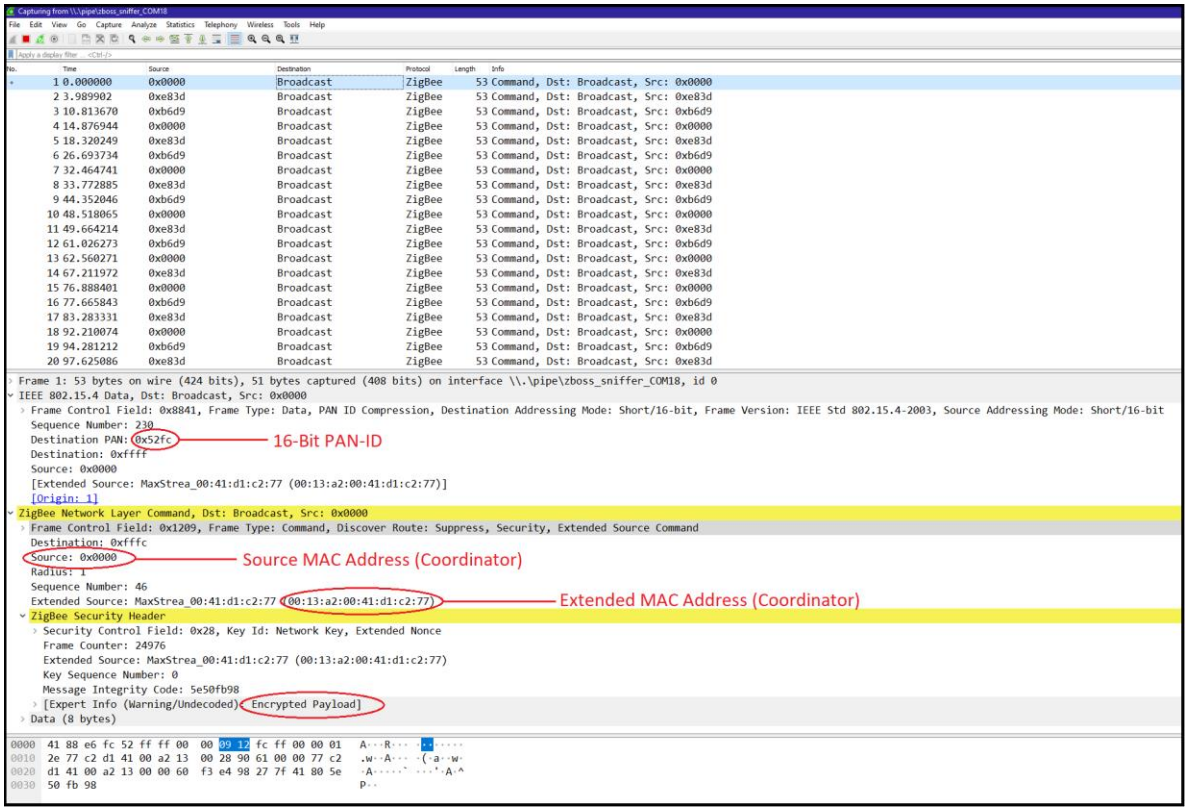

Figure 4.2. Obtaining network information over Wireshark.

The MAC addresses of end devices can be discovered in various data packets sent from the device, as shown in Figure 4.3.

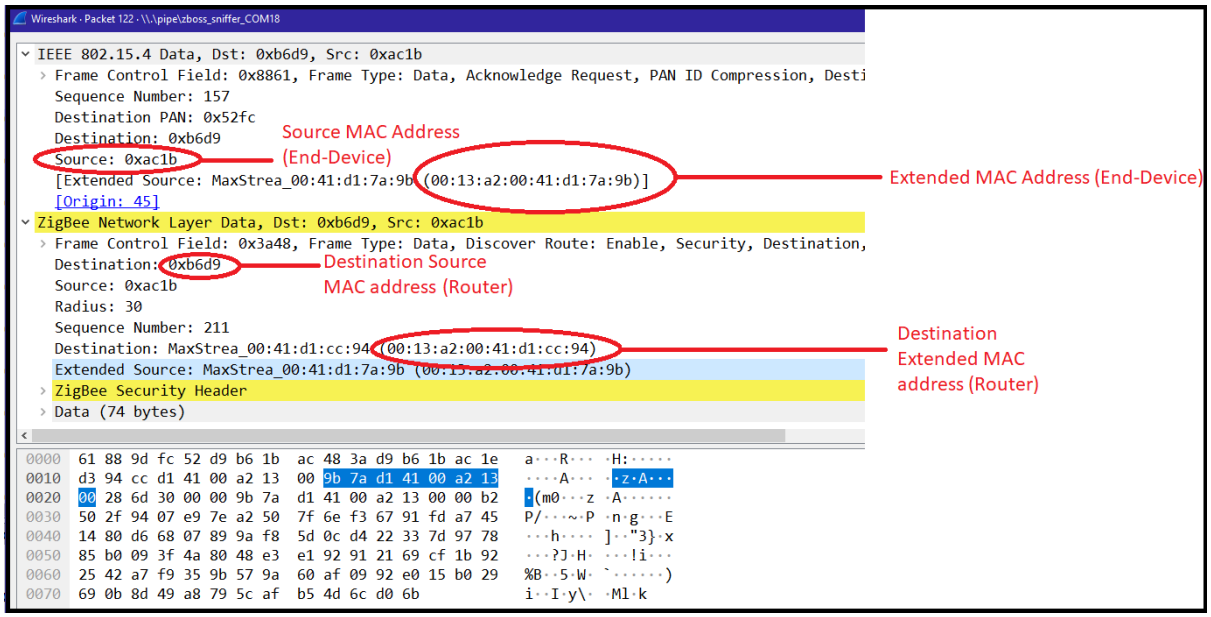

Figure 4.3. Capturing end device MAC addresses.

In ZigBee networks, the extended PAN-ID (64-bit PAN-ID) is transmitted over the network when a beacon request is sent from a device attempting to join or rejoin the network. This can be manipulated with the ‘zbstumbler’ tool by transmitting beacon request packets over the channel from an ApiMote and capturing the beacon response packets through Wireshark (see Figure 4.4).

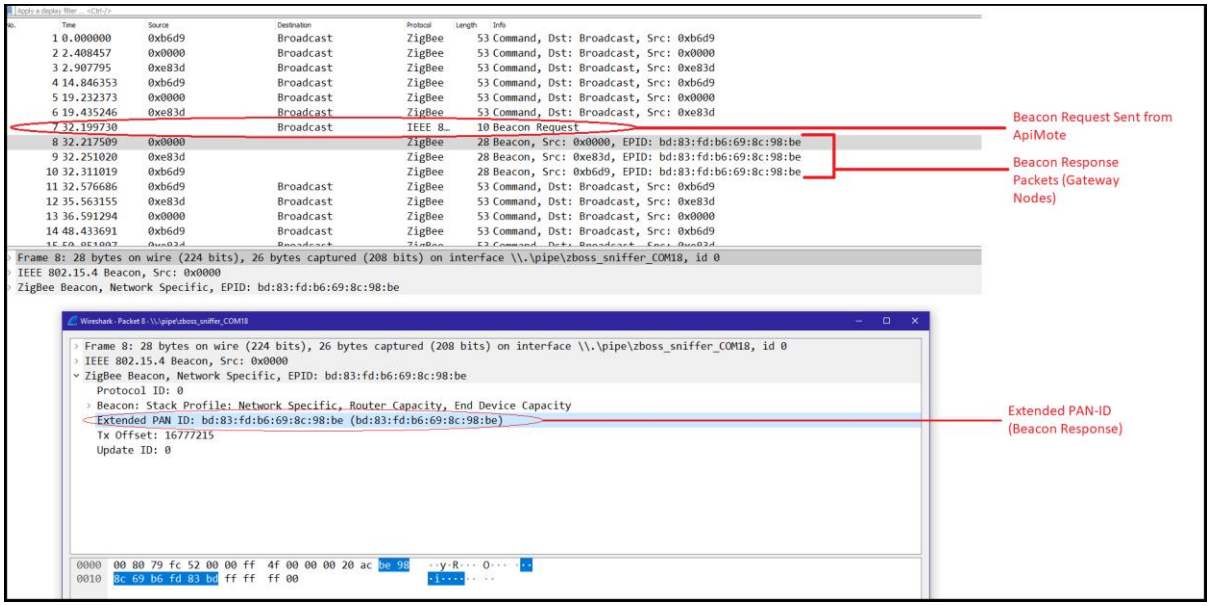

Figure 4.4. Capturing extended PAN-ID.

4.3.1.1.3 Monitoring Join Window

By default, XBee 3 and the ZigBee 3.0 protocol do not support an open joining model where devices can join the network at any given time. Instead, XBee 3 specifies a default join window (NJ) of 254 seconds, which allows devices to join the network within this timeframe. The join window can be opened on XBee 3 only when the commissioning button is pressed twice on a gateway node or by issuing a CB2 AT command (Digi International, 2018). The join window can be monitored externally using the ‘zbstumbler’ tool with two ApiMote devices transmitting beacon requests and listening for a response (see Figure 4.5).

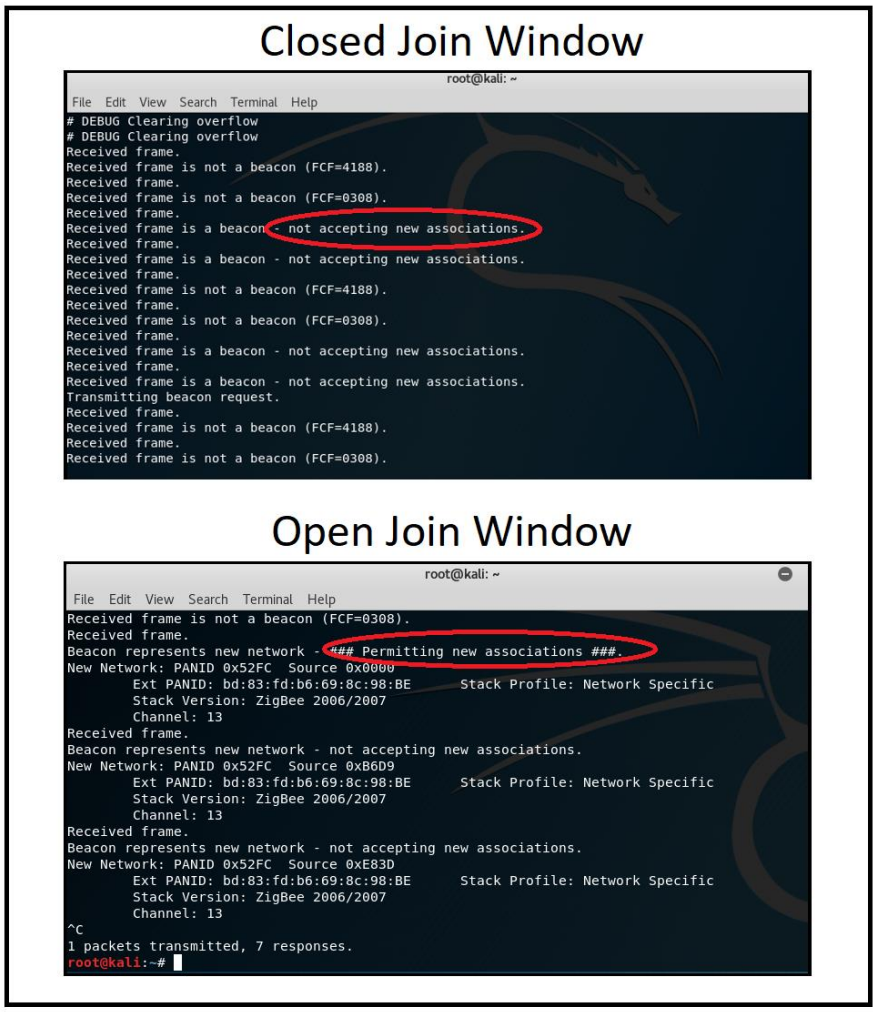

Figure 4.5. Monitoring ZigBee join window.

Figure 4.5 indicates that the network's join window is closed on the gateway nodes. After pressing the commissioning button twice on the coordinator node to open the join window, 'zbstumbler' captures the beacon response, which indicates that the join window is open on the coordinator node (source: 0x000) and closed on the router nodes.

4.3.2 Internal/Physical Information Gathering

The following experiment assumes that an attacker can physically compromise an XBee 3 device and connect it to the XCTU software. When XCTU detects the device, the XBee's AT parameters stored in memory are read and displayed. This experiment demonstrates the network information that an attacker would obtain through a compromised end device or router

node and by remotely connecting to a coordinator. The main points of interest that would be useful to an attacker are presented next.

4.3.2.1 Experiment 2: Internal/Physical Information Gathering

4.3.2.1.1 Compromised End Device Node

When connecting the compromised end device XBee 3 to XCTU software, it was found that an attacker has access to every configuration parameter apart from the keying material and trust centre configurations. The symmetric keys ‘KY’ and ‘NK’ cannot be read in the security configuration because their parameters are ‘write-only’ values to protect the keys. The encryption options (EO) is set to 2, indicating that the network uses a centralised trust centre and does not permit default link keys. Since this device is not a trust centre, the other security configuration parameters are of no relevance.

When initiating an XCTU network scan from the end device node, no network nodes could be discovered for end devices are child nodes and cannot relay messages between devices.

4.3.2.1.2 Compromised Router Node and Remote Coordinator

Similarly, an attacker can access every configuration parameter apart from the keying material and trust centre configurations on a compromised router node. However, because a router is a parent node, network nodes will appear when initiating an XCTU network scan. Furthermore, the default configuration of XBee 3 allows devices to be remotely configurated with AT commands. The compromised router node can remotely read and configure each node in the network, including the coordinator node (see Figure 4.6).

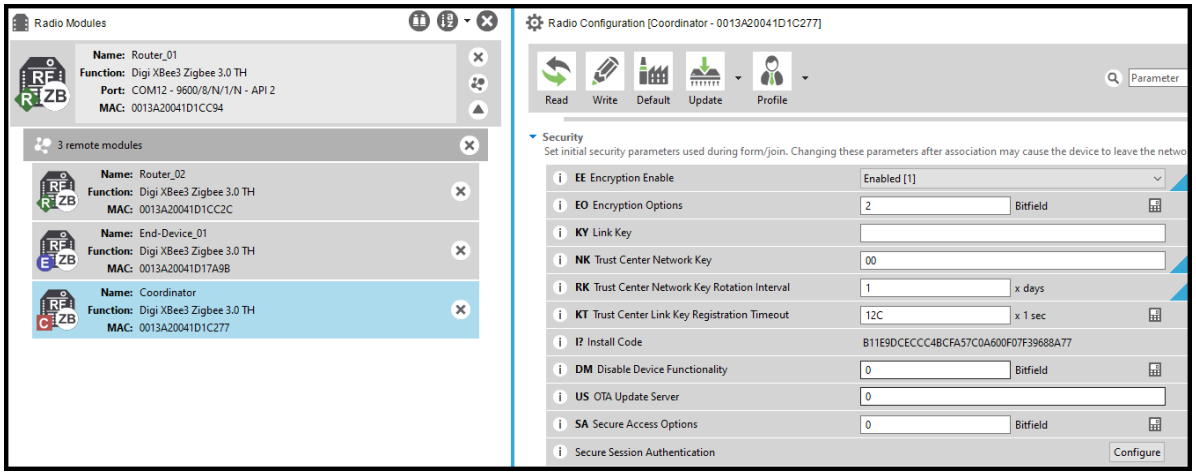

Figure 4.6. Remotely accessing coordinators' configuration.

In Figure 4.6, the compromised router remotely connects to the coordinator node and accesses its security configuration. The parameters only configured on the trust centre are displayed, including the network key rotation (RK) that specifies that the network key is rotated once per day. All other parameters in the trust centre's security configuration remain at their default values.

4.3.3 Summary of Information Gathering Findings

4.3.3.1.1 External Information Gathering

Section 4.3.1 demonstrated the techniques that an attacker can apply to gather vital network information for future exploitation of ZigBee 3.0 networks. The information obtained in this phase could be discovered without prior knowledge of the network and configuration and did not require key compromise. The information gathered externally through KillerBee tools and network sniffing against encrypted ZigBee 3.0 networks are as follows:

- operating channel,
- PAN-IDs,
- MAC addresses, and
- Network Join Window Open/Close State.

4.3.3.1.2 Internal Information Gathering

Section 4.3.2 demonstrated the information an attacker would obtain from a physically compromised device residing inside the network by connecting the device to XCTU and reading the AT parameters stored in memory.

In addition to information that could be externally captured, an attacker can read the device's configuration parameters, including its security configuration. However, the keying material is write-only and cannot be displayed through reading the module's AT parameters. The information that was extracted internally is as follows:

- operating channel;
- PAN-IDs;
- MAC addresses;
- security configuration (excluding keying material); the parameters read off a trust centre differ those from an end device or router node;
- routing structure (through initiating an XCTU network scan from router and coordinator nodes); and
- device configuration.

4.4 Security Issue 1: Security of Symmetric Keys

First, the security of symmetric keys on ZigBee 3.0 is analysed by exploiting known vulnerabilities on a deliberately unsecured network. Second, a method to secure symmetric keys in the ZigBee 3.0 protocol is identified and demonstrated.

Table 4.6

Descriptions of Security of Symmetric Keys Experiments

Test ID	Network	Test Description
Experiment 3: Unencrypted Network Key Attacks		
T03	CSM	Capturing the unencrypted network key through Wireshark on a CSM network

T04	DSM	Capturing the unencrypted network key through Wireshark on a DSM network
Experiment 4: Default Link Key Attacks		
T05	CSM	Intercepting and decrypting the network key through Wireshark using a default link key
T06	CSM	Authenticating an unauthorised device onto a CSM network with the well-known default link key
Experiment 5: Securing Symmetric Keys with Install Codes		
T07	CSM	Securely registering a joining node to the trust centre using an install code

4.4.1 Attacks Against Symmetric Keys

In this section, attacks are performed against the symmetric keys of ZigBee 3.0 networks to compromise the keys. For the following experiments, the network is configured to contain symmetric key vulnerabilities existing in earlier revisions of ZigBee. These vulnerabilities are possible in ZigBee 3.0 networks; however, in XBee 3, the default encryption options (EO=2) prevent these vulnerabilities. ZigBee vendors highly discourage enabling these encryption options shown in this section, because these significantly affect the security of the symmetric keys and network (Digi International, 2018).

4.4.1.1 Experiment 3: Unencrypted Network Key Attacks

In ZigBee networks, the network key can be transported from a trust centre to a joining device OTA in plain text. This vulnerability can be enabled in the security configuration of XBee 3 through the encryption options and is tested on both security models:

4.4.1.1.1 Security Configurations

The following security configurations shown in Figure 4.7 are applied to the CSM and DSM networks to enable the unencrypted network key transport vulnerability.

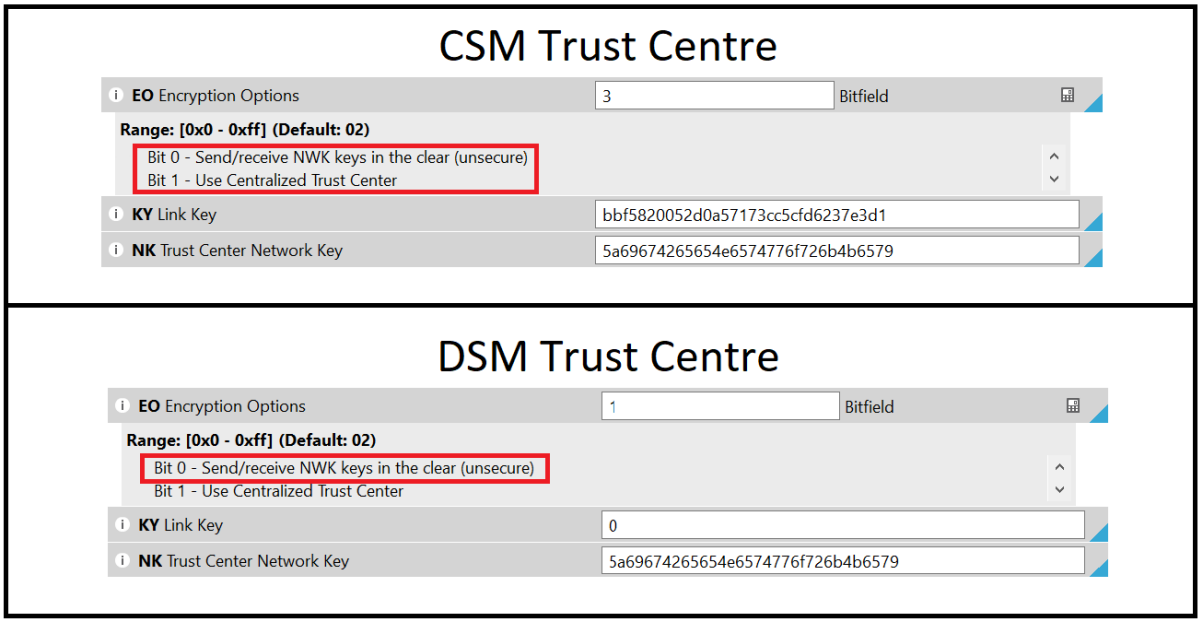

Figure 4.7. Unencrypted network key configuration on XBee 3.

On the CSM network, Encryption Options (EO) is set to 3 to enable bit 0, allowing the network key to be sent/received unencrypted, and to bit 1 to use a centralised trust centre. The link key (KY) is preconfigured on each node (a requirement for device authentication in CSM networks), and the network key (NK) is preconfigured on the coordinator node with the ASCII char value 'ZigBeeNetworkKey!' for demonstration purposes.

The DSM network sets EO=1 to allow the network key to be sent/received unencrypted. No preconfigured link key (KY) is set.

4.4.1.1.2 CSM Network Findings

The router node successfully joins the network when the join window opens and receives the network key. However, it is found that the network key was encrypted with the preconfigured link key, despite EO bit 0 being enabled (see Figure 4.8).

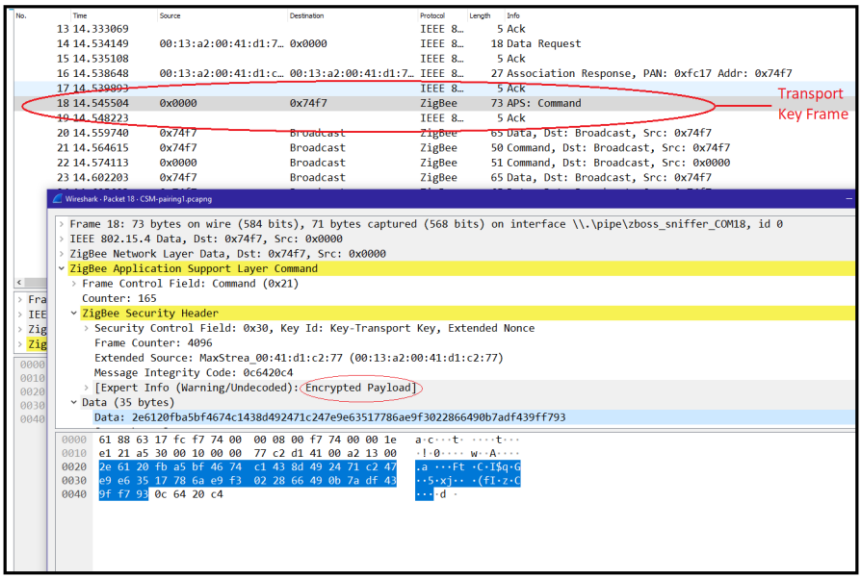

Figure 4.8. Encrypted transport key packet on CSM network.

4.4.1.1.3 DSM Network Findings

When the join window is opened on a gateway node, the router joins the DSM network and receives the unencrypted network key from a trust centre. An unencrypted 'Transport Key' packet is captured, and the plain-text network key can be extracted from its unencrypted payload (see Figure 4.9):

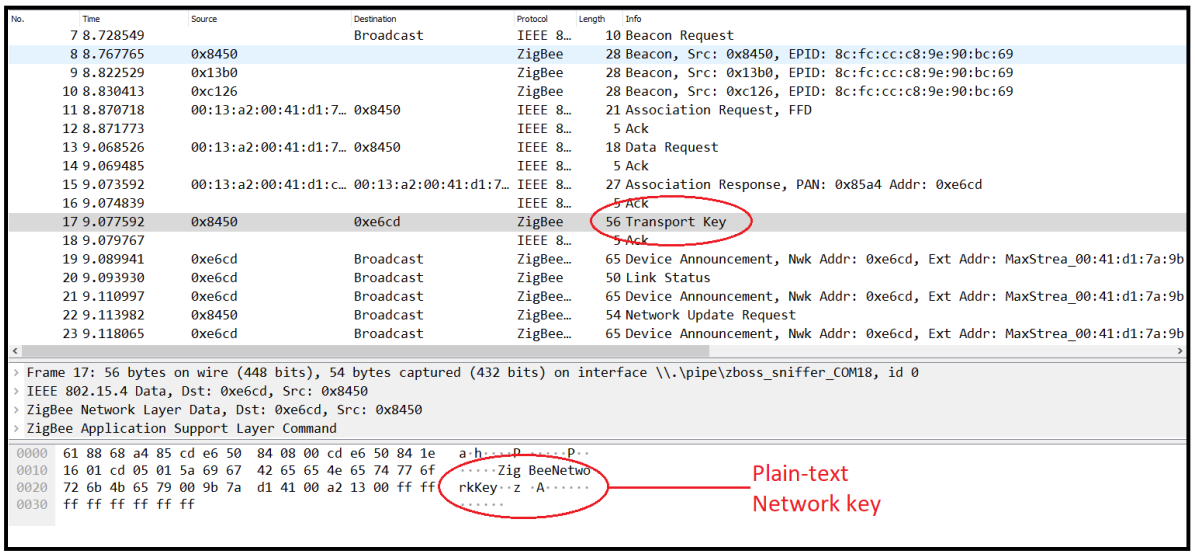

Figure 4.9. Unencrypted transport key packet on DSM network.

4.4.1.2 Experiment 4: Default Link Key Attacks

The following experiments evaluate how default link key values pose a threat to the network's security. These experiments are performed against a CSM network, and it is assumed that the attacker has knowledge of the trust centre authenticating with the well-known default link key.

4.4.1.2.1 Security Configuration

The security configuration shown in Figure 4.10 is applied to the CSM network to allow joining via default link keys.

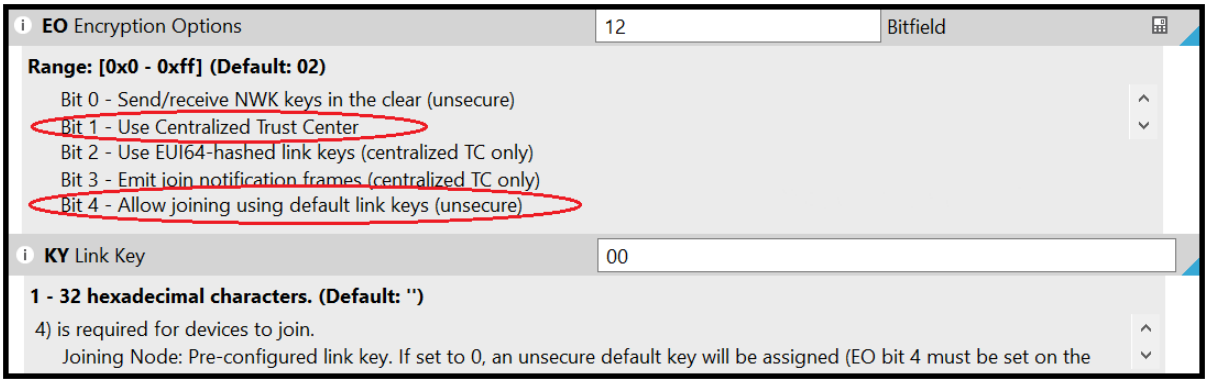

Figure 4.10. Default link key configuration.

The EO is set to 12 to enable bitfield options 1 and 4. The network utilises a centralised trust centre and permits the use of default link keys. The KY parameter is set to 0, enabling the trust centre to fall back to the default link key for authentication. The trust centre will generate a random network key (NK = 0) and encrypt it with the well-known default link key '5A 69 67 42 65 65 41 6C 69 61 6E 63 65 30 39'.

4.4.1.2.2 Network Sniffing

With the well-known default link key added to ZigBee's protocol preferences in Wireshark, the frames as the router join the network are captured. Among the captured frames is the 'Transport Key' packet encrypted with the default link key (see Figure 4.11).

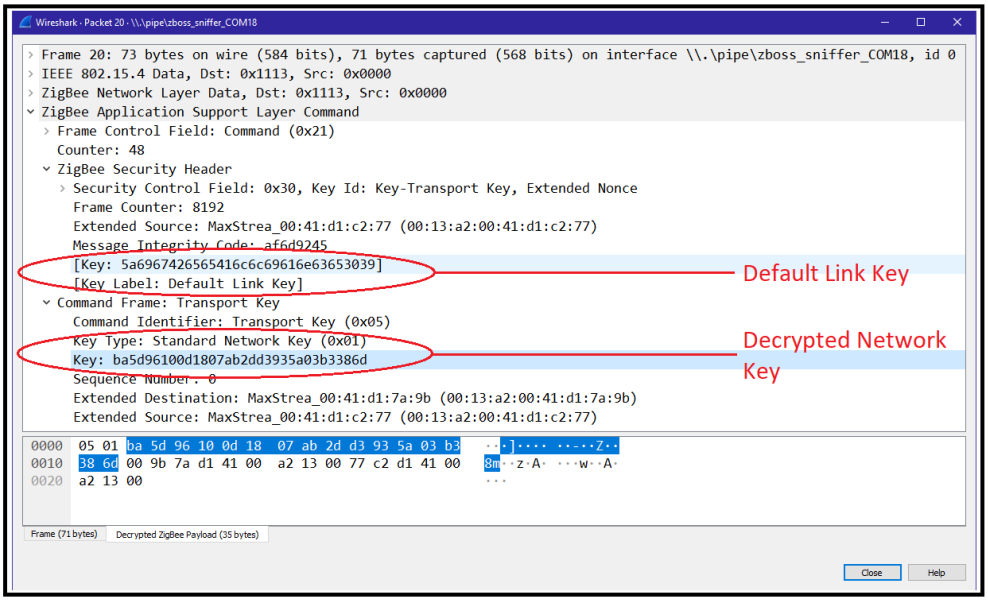

Figure 4.11. Decrypting network key with well-known default link key.

Figure 4.11 shows the contents of the ‘Transport Key’ packet. The default link key used to encrypt the payload is within the APS layer’s security header, and the network key can be extracted from the decrypted APS payload.

4.4.1.2.3 Unauthorised Network Joining

An unauthorised device is configured with the victim network’s extended PAN-ID and security configuration. In the device’s security configuration, the KY parameter is set to the value of the well-known default link key ‘5A 69 67 42 65 65 41 6C 69 61 6E 63 65 30 39’. Optionally, the KY can be set to 0 to assign a default key (see Figure 4.12).

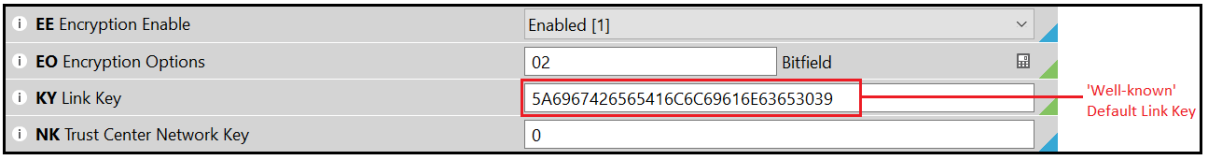

Figure 4.12. Unauthorised XBee 3 device with default link key.

The configured unauthorised device can join the network once the join window opens on a gateway node opens. The ‘zbstumbler’ tool can be utilised to monitor the network until a gateway node permits new associations. Once the join window opens, the unauthorised device

is successfully authenticated into the network and can remotely access network nodes by default (see Figure 4.13).

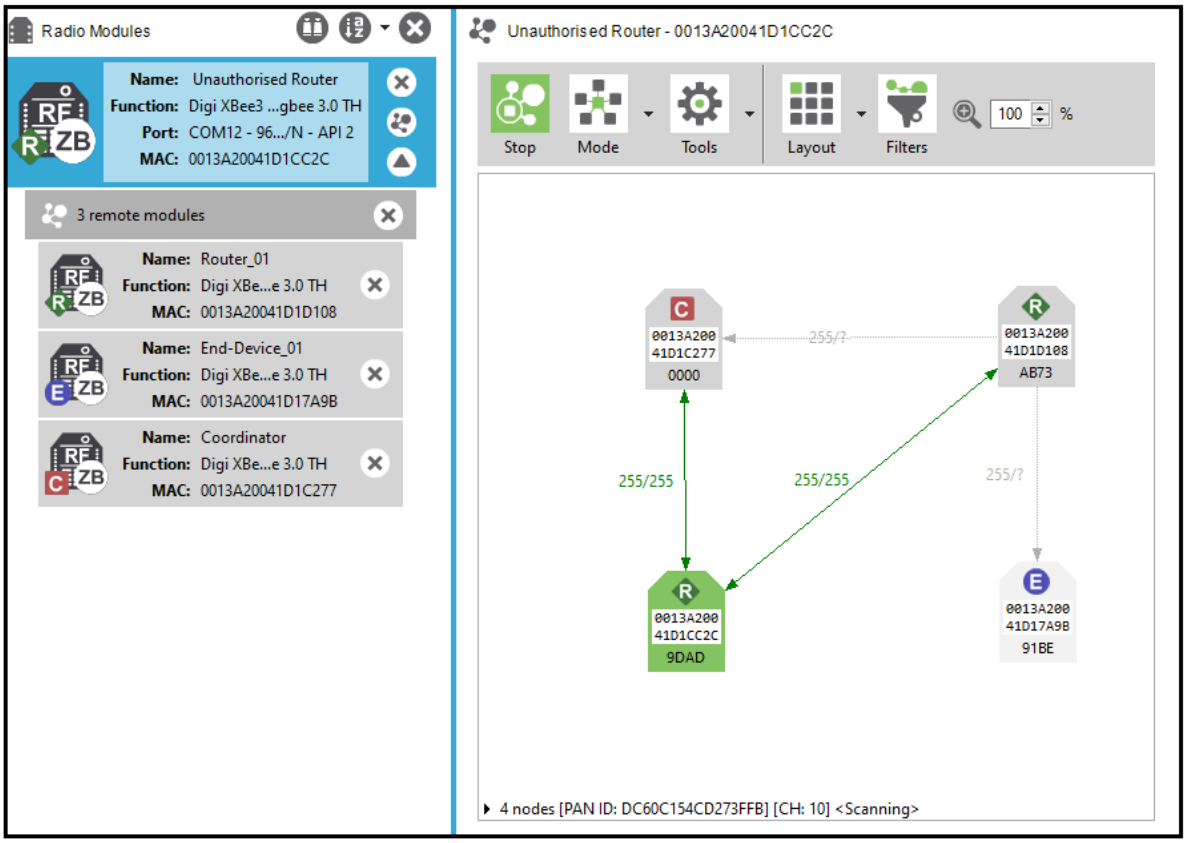

Figure 4.13. Remotely connecting to network nodes from unauthorised device.

4.4.2 Securing Symmetric Keys in ZigBee 3.0

ZigBee 3.0 devices have the option of joining the network securely with an install code. Device registration via install codes provides a high level of security level to symmetric keys. It guarantees that each joining device has a random link key (Digi International, 2018) and eliminates authenticating with a global link key.

4.4.2.1 Experiment 5: Securing Symmetric Keys with Install Codes

4.4.2.1.1 Install Code Configuration

The configurations shown in Figure 4.14 are applied to the joining node and trust centre to enable joining via install codes.

Figure 4.14. ZigBee 3.0 install code joining configuration on XBee 3.

In the security configuration, the encryption options (EO) is set to 2, enabling the network to use a centralised trust centre. The link key (KY) on the trust centre is set to 0 to increase the security further, for this generates a random link key that cannot be read and requires every node to be individually registered to the trust centre. In the networking configuration on the joining node, 'Device Controls' (DC) is set to 1 (bit 0) to enable authentication with a link key derived from its install code.

4.4.2.1.2 Registering Joining Device with Install Code

On the trust centre, a 0x24 (Register Joining Device) frame is created that contains the joining node's install code and MAC address. When transmitted from the trust centre, the joining node receives the 0x24 frame and successfully joins the network. The node securely receives the network key from the trust centre (see Figure 4.15).

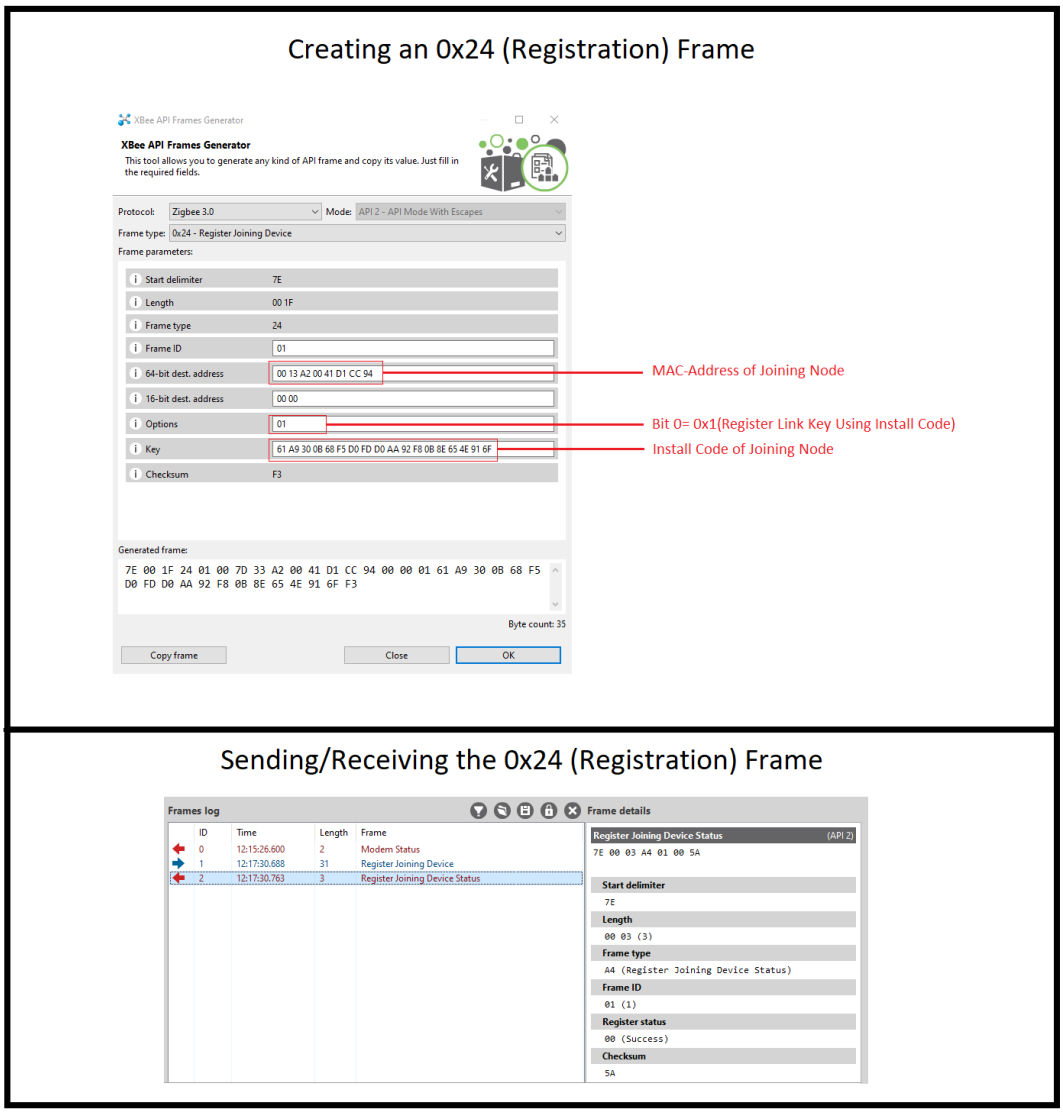

Figure 4.15. Creating and sending 0x24 frame with install code.

4.4.3 Summary of Security Issue 1 Findings

4.4.3.1 Attacks Against Symmetric Keys

The findings of the experiments that exploited symmetric key vulnerabilities on ZigBee 3.0 networks are summarised as follows:

- **Unencrypted Network Key:**

The unencrypted network key vulnerability can only exist in DSM ZigBee 3.0 networks when EO=1 (bit 0) is set. With the setting enabled, the network key is sent to the joining device in plain text and can be externally captured with a packet sniffer. On CSM networks, the network key is encrypted with the preconfigured link key despite having EO bit 0 enabled.

- **Default Link Key:**

The vulnerability of a trust centre authenticating devices with a default link key can exist on both security models when EO bit 4 is enabled. A default link key could allow an attacker to capture and decrypt the network key or authenticate an unauthorised device when the join window opens, as demonstrated in Section 4.4.1.2.

4.4.3.2 Securing Symmetric Keys in ZigBee 3.0

As demonstrated in Section 4.4.2, a joining node can be individually registered to the trust centre with a random link key derived from the device's install code. This method provides the highest level of security to symmetric keys for it ensures the link key is completely random on each device and protects the network key from being exposed from a compromised global or default link key.

4.5 Security Issue 2: Compromised Symmetric Keys

The experiments related to security issue 2 are a study based on the assumption that an attacker has compromised ZigBee's symmetric keys. Where necessary, these experiments are conducted against both security models to evaluate the impact of a compromised symmetric key on the confidentiality of ZigBee 3.0 networks.

Table 4.7

Compromised Symmetric Key Experiment Descriptions

Test ID	Network	Test Description
Experiment 6: Key Sniffing/Eavesdropping Attacks with Compromised Link Key		
T08	CSM	Capturing and decrypting symmetric keys transmitted to a joining router on a CSM network with a compromised link key
T09	CSM	Capturing the network key rotation on a CSM network
T10	DSM	Capturing and decrypting symmetric keys transmitted to a joining router on a DSM network with a compromised link key

Experiment 7: Packet Decryption/Eavesdropping Attacks		
T11	CSM	Capturing and decrypting network layer (broadcast) communications on a CSM network
T12	CSM	Capturing and decrypting APS layer (unicast) communications on a CSM network
Experiment 8: Node Impersonation Attacks		
T13	CSM	Impersonating a legitimate Coordinator: Attempting to realign the victim node to the attacker's network using compromised symmetric keys on a CSM network
T14	DSM	Impersonating a legitimate Coordinator: Attempting to realign the victim node to the attacker's network using compromised symmetric keys on a DSM network

- **Security Configuration and Setup:**

Two separate ZigBee 3.0 networks were constructed for the compromised symmetric key attack experiments with three gateway nodes. The networks are restricted to three nodes to reduce the overall amount of traffic and unwanted data generated.

  - ***CSM Network Configuration***:

The CSM network consists of one coordinator and two router nodes. The coordinator is configured as the centralised trust centre and is responsible for managing and setting the security policy on the network. The security configuration of the trust centre and router nodes are shown in Figure 4.16.

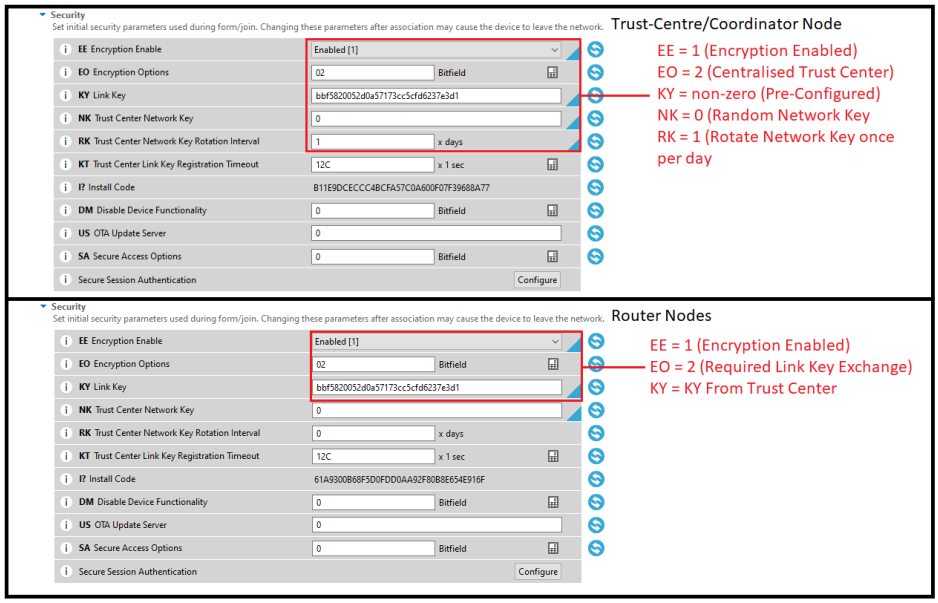

Figure 4.16. CSM security configuration for Security Issue 2 experiments.

- ***DSM Network Configuration:***

The DSM network consists of three router nodes. Each router node acts as a trust centre and contains a copy of the network key to authenticate joining devices. The security configuration of the DSM trust centre/router nodes is shown in Figure 4.17.

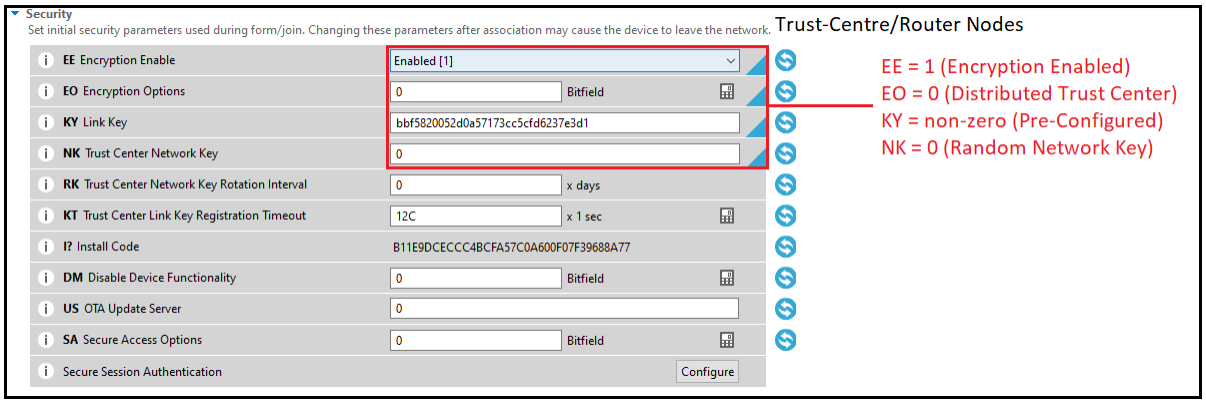

Figure 4.17. DSM security configuration for Security Issue 2 experiments.

In the DSM network, encryption is enabled (EE=1). To prevent using a centralised trust centre, the EO is set to 0 (no bitfield options). Devices are preconfigured with a unique link key for authentication. The NK is set to 0 to generate a random network key (key generated on Router_01), and the other nodes will obtain a copy of the key upon joining.

4.5.1 Eavesdropping Attacks

The following experiments are eavesdropping (network sniffing) attacks against the ZigBee 3.0 networks. These attacks involve actively capturing traffic on the victim network operating channel and using the compromised symmetric keys to decrypt sensitive network information.

4.5.1.1 Experiment 6: Key Sniffing/Eavesdropping Attacks with Compromised Link Key

In ZigBee networks, the network key is encrypted and transmitted to joining devices configured with the correct link key. Key sniffing is an attack against a network involving the interception of symmetric keys as they are shared to joining devices or otherwise when the keys are rotated. For the key sniffing experiments, it will be assumed that the attacker has already compromised the preconfigured link key through various techniques.

4.5.1.1.1 Wireshark Preparation for Key Sniffing

The compromised link key is initially added to ZigBee's protocol preferences in Wireshark. The security level for decryption is set to AES 128-bit encryption and 32-bit integrity protection. Once compromised, the other symmetric keys can be added to Wireshark for further network decryption (see Figure 4.18).

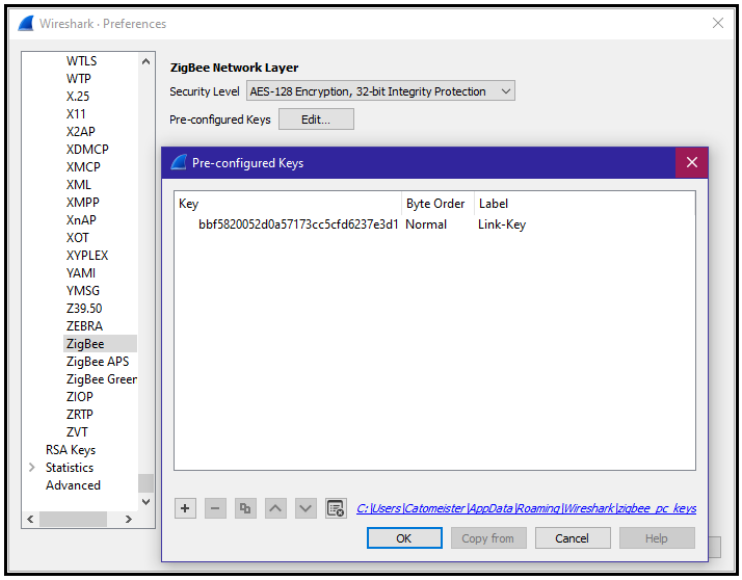

Figure 4.18. Adding a compromised symmetric key to Wireshark.

4.5.1.1.2 CSM Network Findings

The findings related to the CSM network are as follows:

- **Capturing Network Join:**

When the join window on the CSM network opens, the router joins the network, and two ‘Transport Key’ packets are sent to the device originating from the trust centre/coordinator node. Each ‘Transport Key’ frame contains a symmetric key (see Figure 4.19).

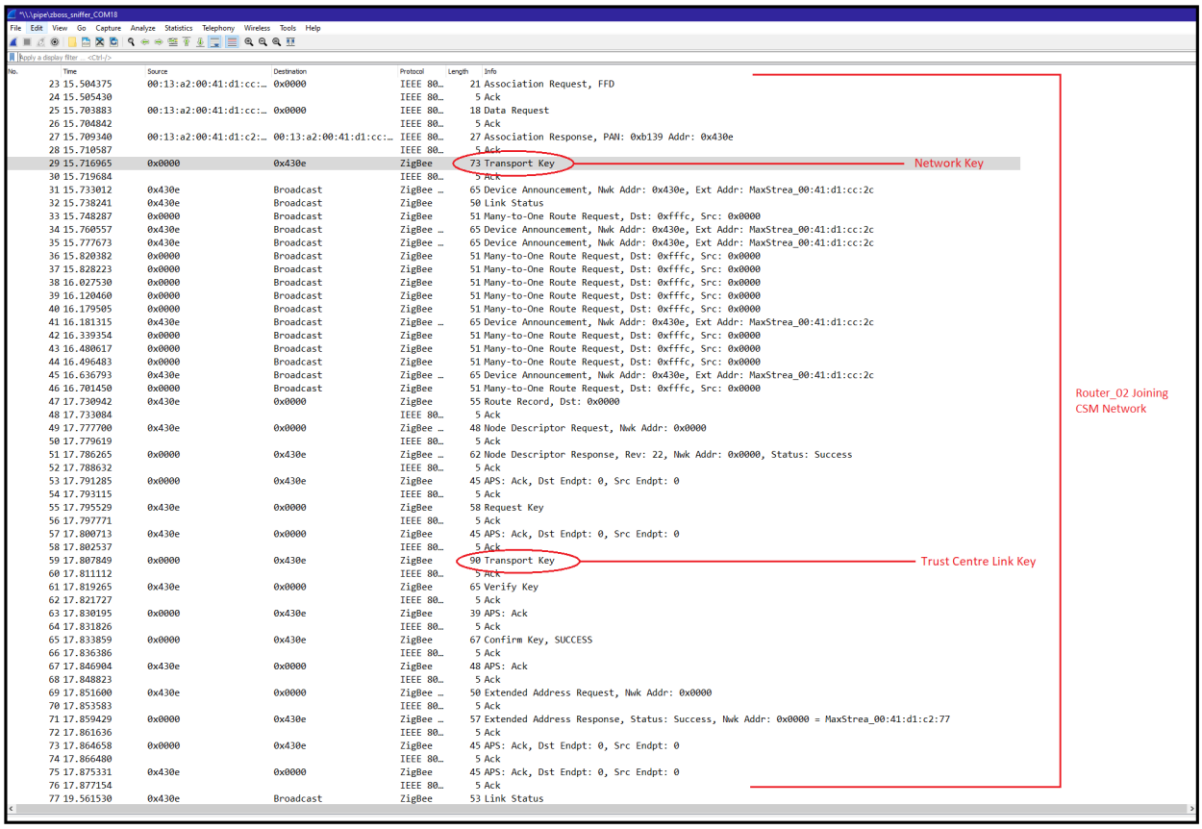

Figure 4.19. Capturing symmetric keys on a CSM network.

In the joining process on the CSM network, the joining router first received the encrypted network key. The router then requested and received an updated trust centre link key from the trust centre/coordinator for all future unicast communications. A key hash was generated and sent from the router as a ‘Verify Key’ frame to verify the integrity of the received trust centre link key. The trust centre/coordinator checked the key hash in the ‘Verify Key’ frame and confirmed with a ‘Confirm Key’ frame (see Figure 4.20). Both the captured

‘Transport Key’ frames contained the symmetric keys, which are encrypted with the initial preconfigured link key known to the attacker.

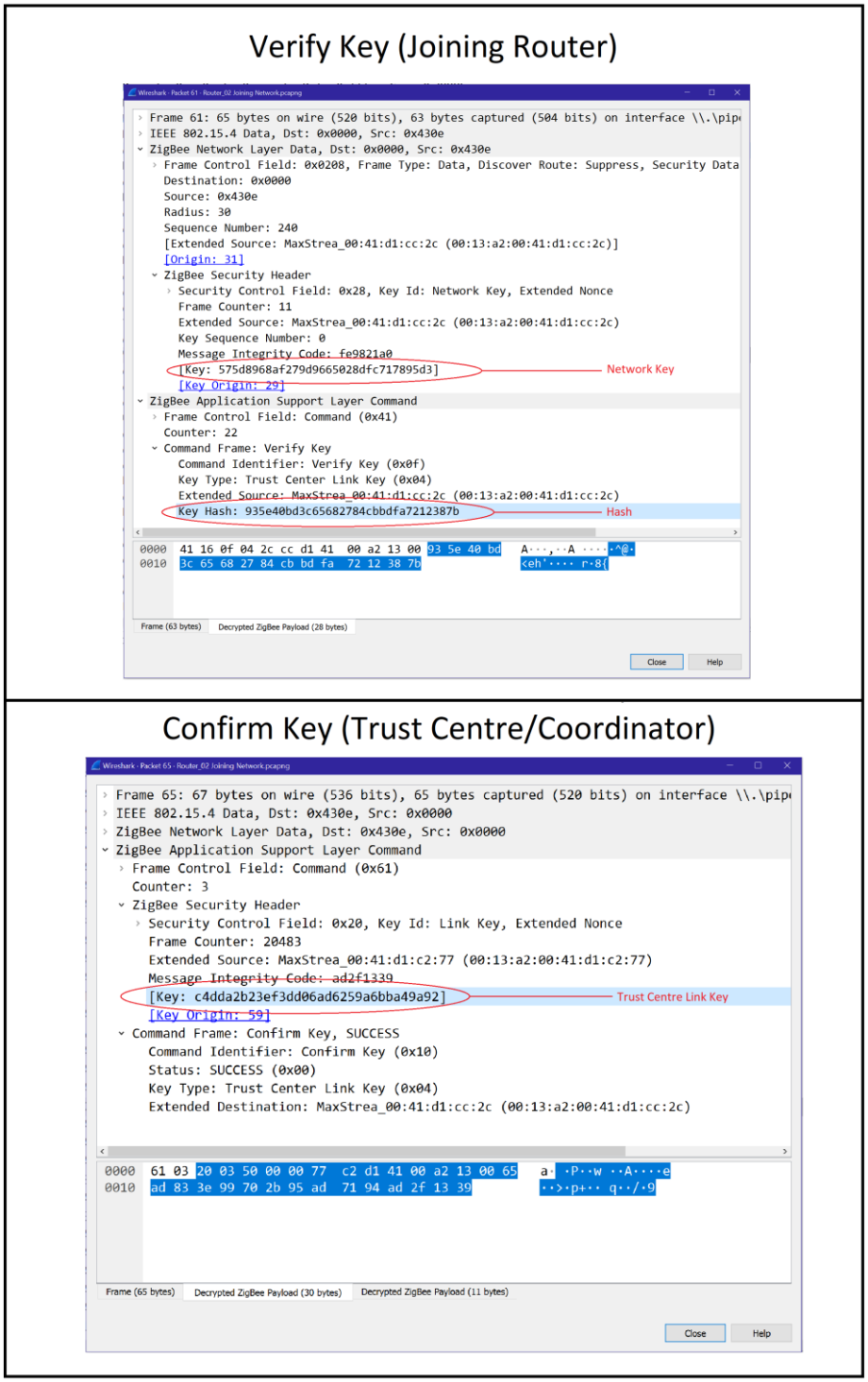

Figure 4.20. Symmetric key verification on CSM Network.

The captured symmetric keys can then be used for further decryption of network data on the CSM network (see Figure 4.21).

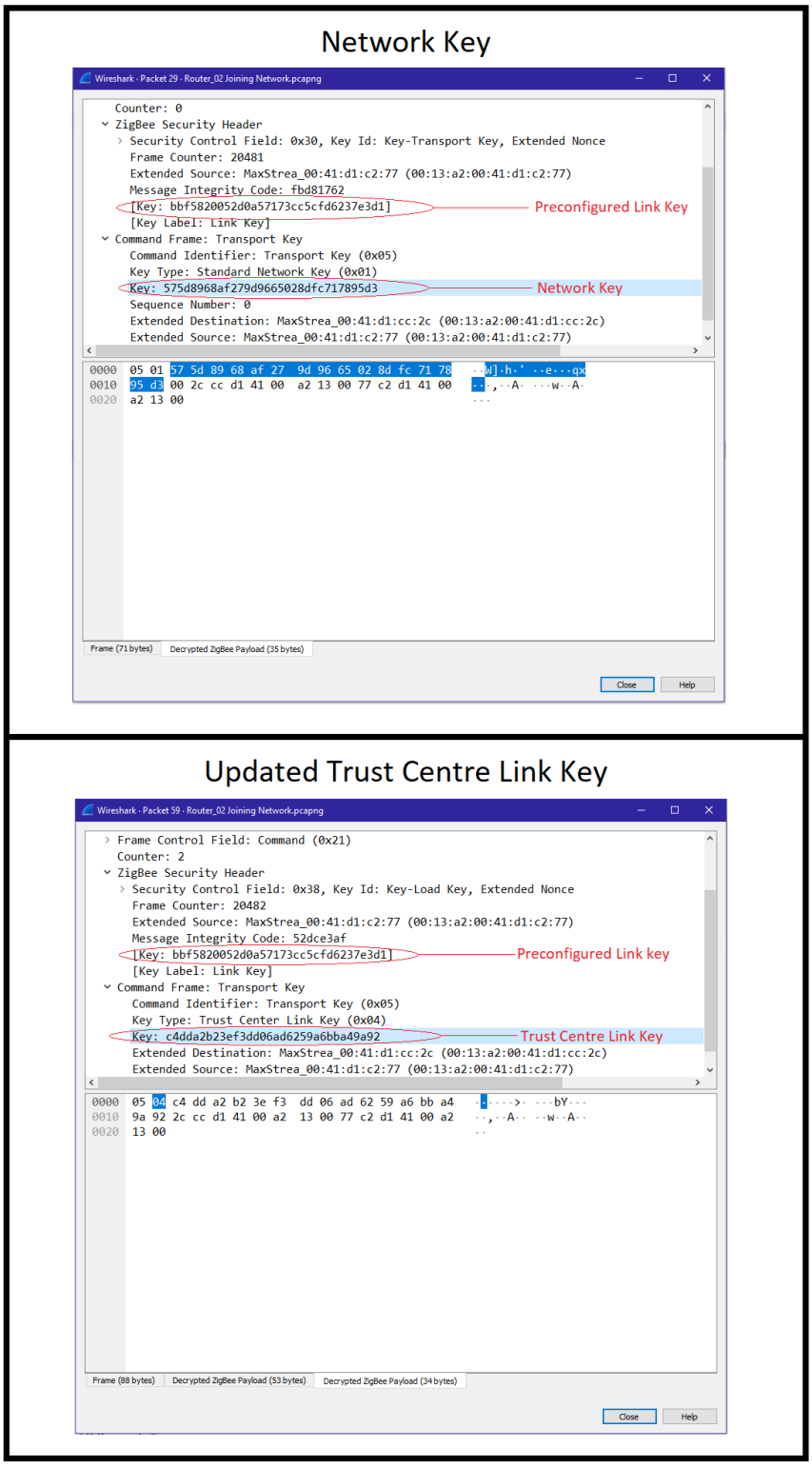

Figure 4.21. Decrypting symmetric keys on a CSM network.

- **Capturing Network Key Rotation:**

The network key rotation was captured roughly 26 hours after the network started. The trust centre/coordinator broadcasts ‘Transport Key’ packets containing the updated network key and notifies network nodes to ‘Switch Key’. However, the updated network key is encrypted with the old network (compromised) network key (see Figure 4.22).

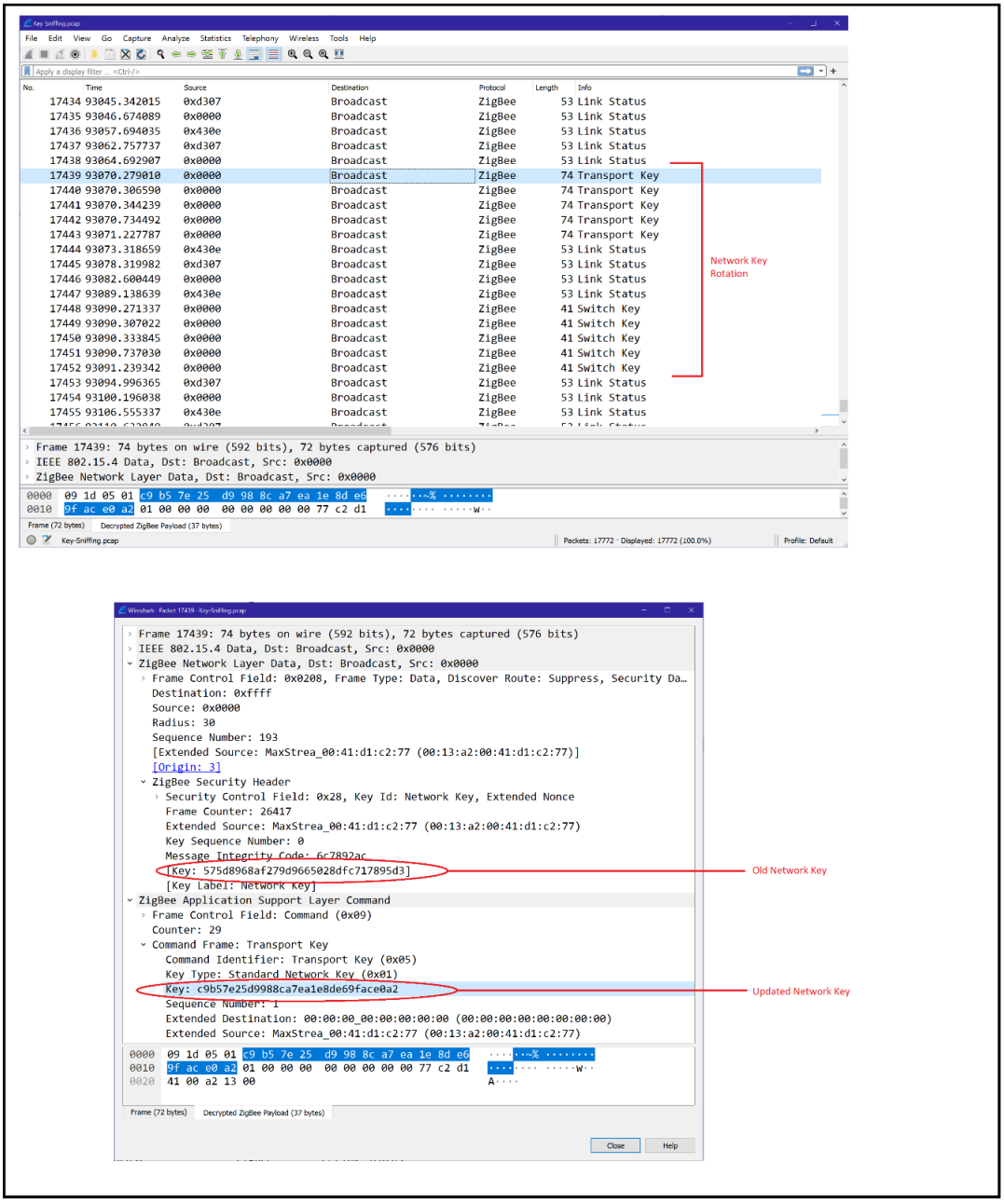

Figure 4.22. Capturing network key rotation on a CSM network.

4.5.1.1.3 DSM Network Findings

When the join window on the DSM network opens, the router joins the network, and a single symmetric key is sent to the device originating from a trust centre/router node. The ‘Transport Key’ frame containing the symmetric key is encrypted with the preconfigured and compromised link key. The captured symmetric keys can be used for further decryption of network data on the DSM network (see Figure 4.23).

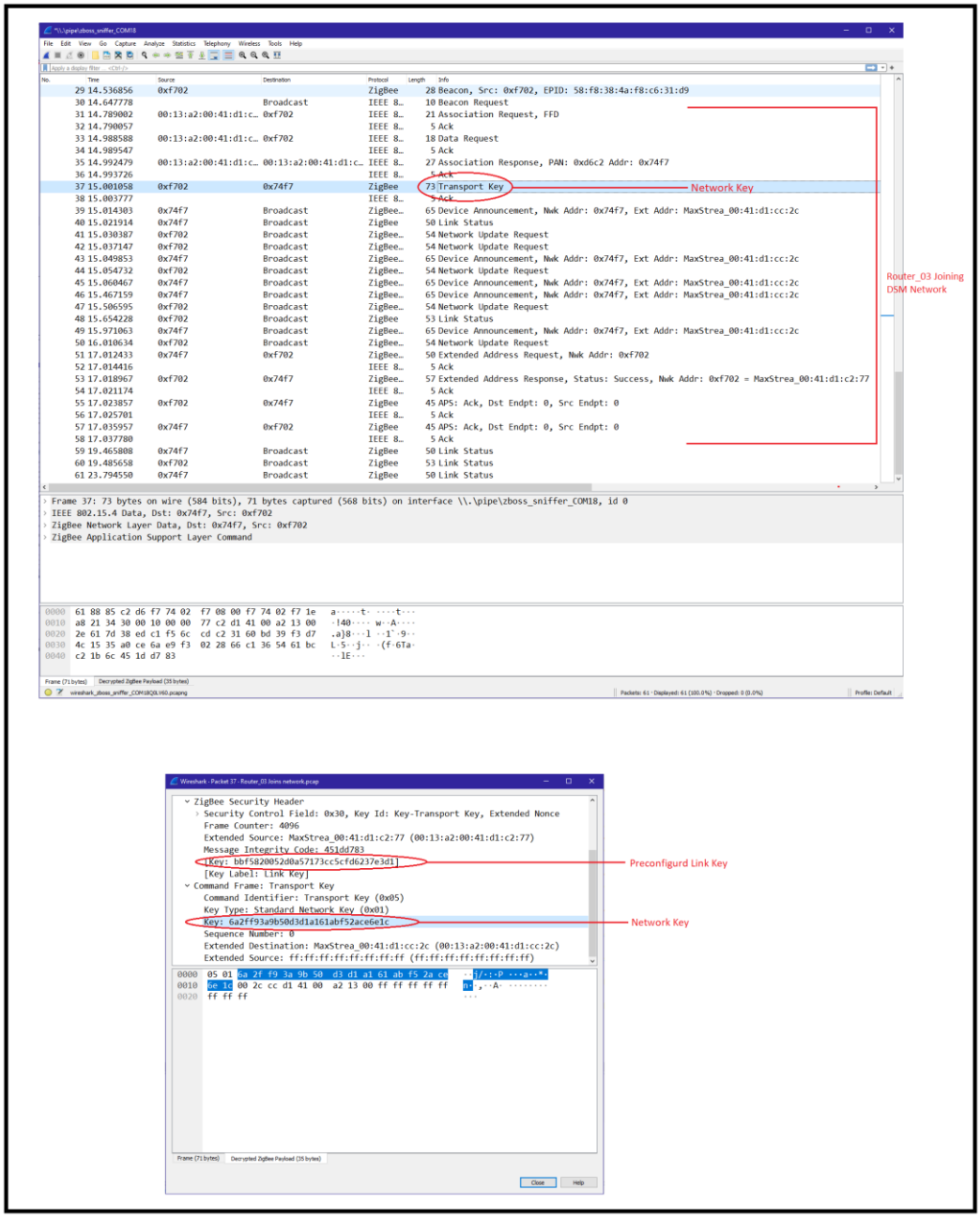

Figure 4.23. Capturing symmetric keys on a DSM network.

4.5.1.2 Experiment 7: Packet Decryption/Eavesdropping Attacks

After an attacker has compromised a ZigBee network's symmetric keys, the network data can be externally captured and decrypted. In ZigBee, the NWK layer (broadcast) encryptions are encrypted with the network key, and APS (unicast) communications are encrypted with the link key (Radmand et al., 2010). This experiment assumes that an attacker has obtained each of the victim network's symmetric keys through techniques shown in Section 4.5.1.1. This experiment is only performed on a single CSM network because encryption works similarly on a DSM network.

4.5.1.2.1 Decrypting NWK Layer/Broadcast Communications

Broadcast packets are transmitted on the network from gateway nodes. These packets can be decrypted with the network key. When sending a generic transmit request packet from the coordinator to a router node, encryption is applied on the NWK layer using the network key (see Figure 4.24).

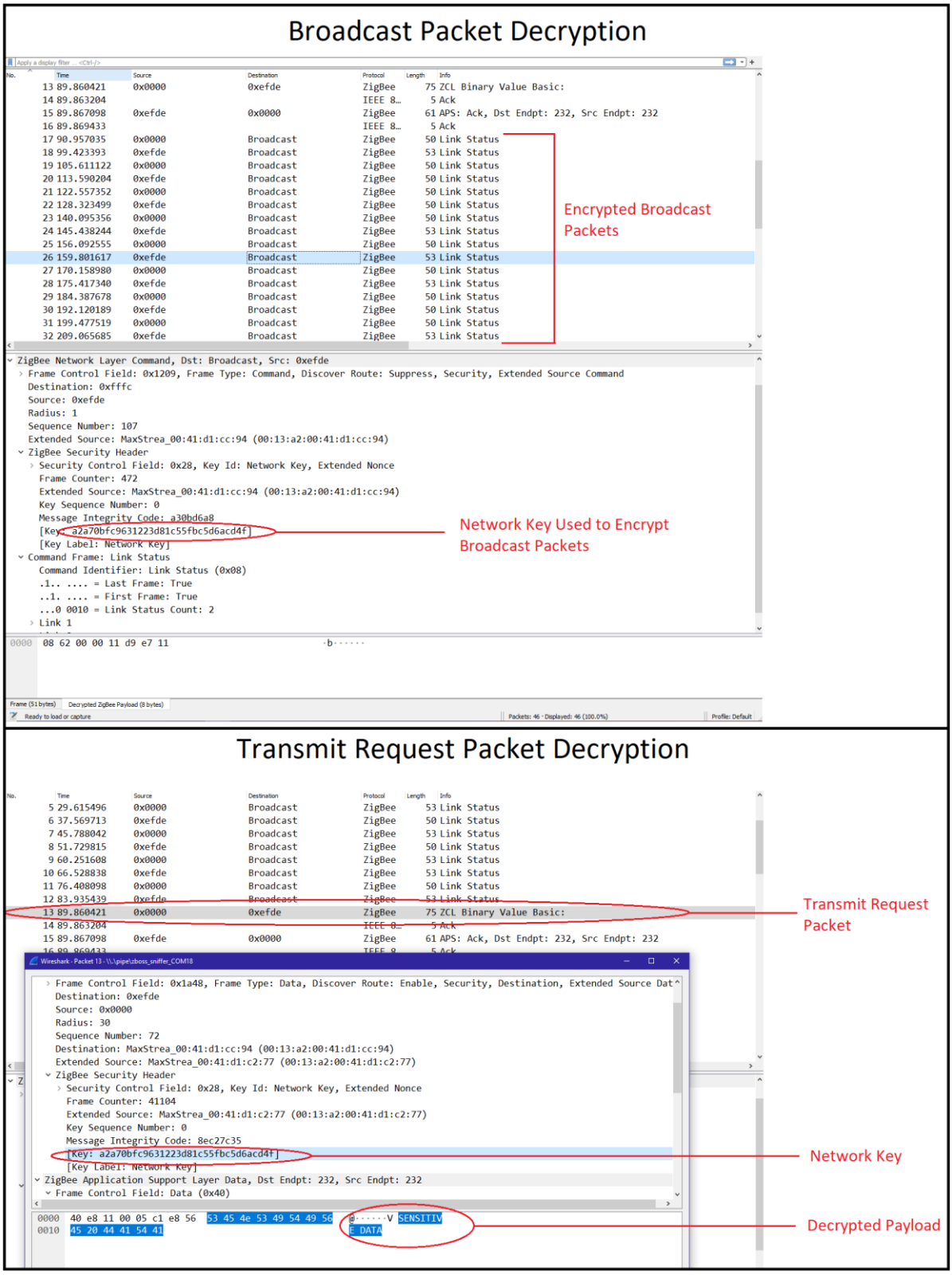

Figure 4.24. NWK layer decryptions on ZigBee 3.0.

4.5.1.2.2 Decrypting APS Layer (Unicast) Communications

A transmit request frame with APS encryption enabled is created in the XBee API Frames Generator on the coordinator node and transmitted to a router node. After the frame is sent, the packet is captured in Wireshark and decrypted with the updated trust centre link key (see Figure 4.25).

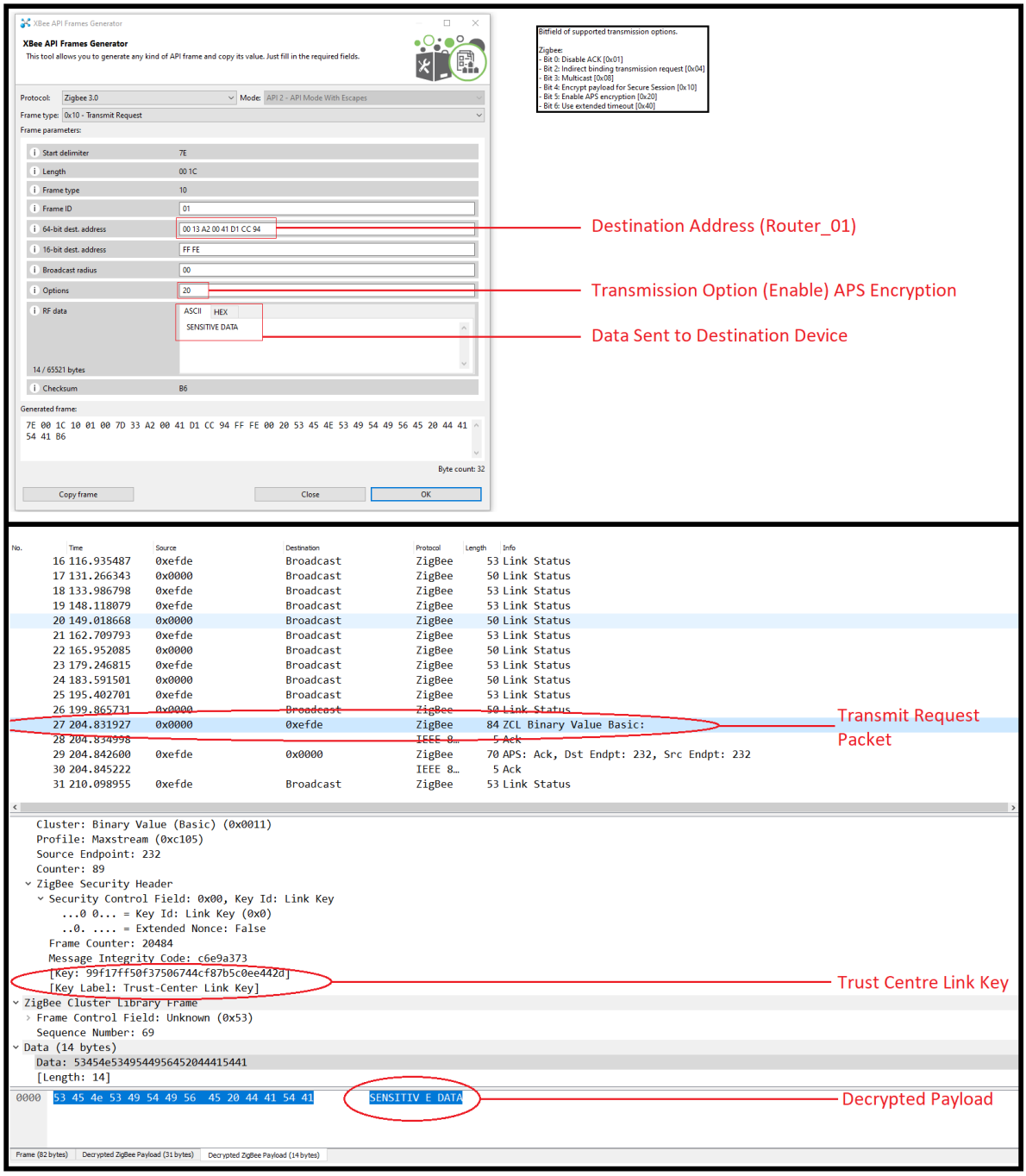

Figure 4.25. APS layer decryptions on ZigBee 3.0.

On a DSM Network, APS secured frames are encrypted with the preconfigured link key because devices do not receive an updated trust centre link key upon joining.

4.5.2 Node Impersonation Attack

A node impersonation attack can be performed against ZigBee 3.0 networks using the compromised symmetric keys. For this experiment, an attacker impersonates a legitimate coordinator node configured with the captured network information and the symmetric keys of the victim network. The attacker will attempt to realign a victim node to the attacker's network using a spoofed coordinator realignment frame. A successful attack would unwillingly cause the victim node to join the attacker's network, resulting in data compromise or DoS. For the following node impersonation attack experiment, it is assumed that the attacker has already obtained the network information and compromised the symmetric keys.

4.5.2.1 Experiment 8: Node Impersonation Attacks

4.5.2.1.1 Attack Setup and Execution

A coordinator (attacker) node is configured with the obtained victim network information and symmetric keys (see Figure 4.26).

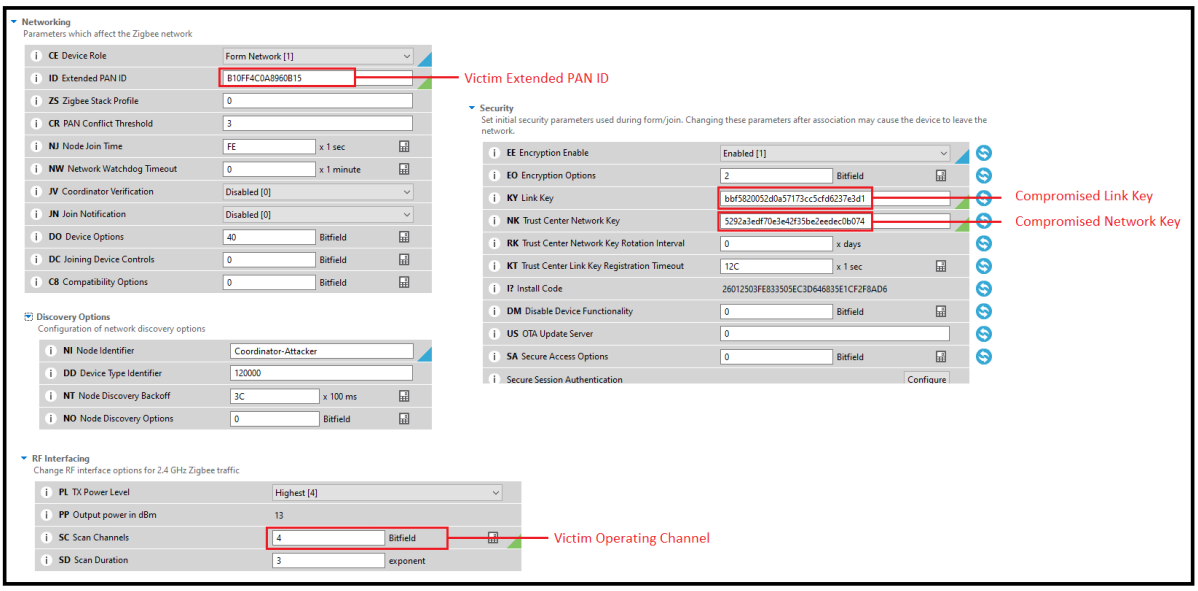

Figure 4.26. Configuration of attacker node for impersonation attack.

Figure 4.26 shows the configuration of the impersonated coordinator, for targeting a CSM network. The device is configured with the extended PAN-ID and compromised symmetric keys to match the victim network's configuration. Moreover, its channel scan is limited to match the operating channel of the victim network.

The attack is initiated from Kali Linux using the 'zbrealign' tool to create and send a spoofed coordinator realignment frame to the victim network (see Figure 4.27).

Figure 4.27. Executing 'zbrealign' script from Kali Linux.

4.5.2.1.2 CSM Network Findings

Figure 4.28 shows the victim router node (Router_02) targeted in this attack and the networking parameters established on the attacker node. A network is formed on the attacker node using the same channel and extended PAN-ID as the victim node:

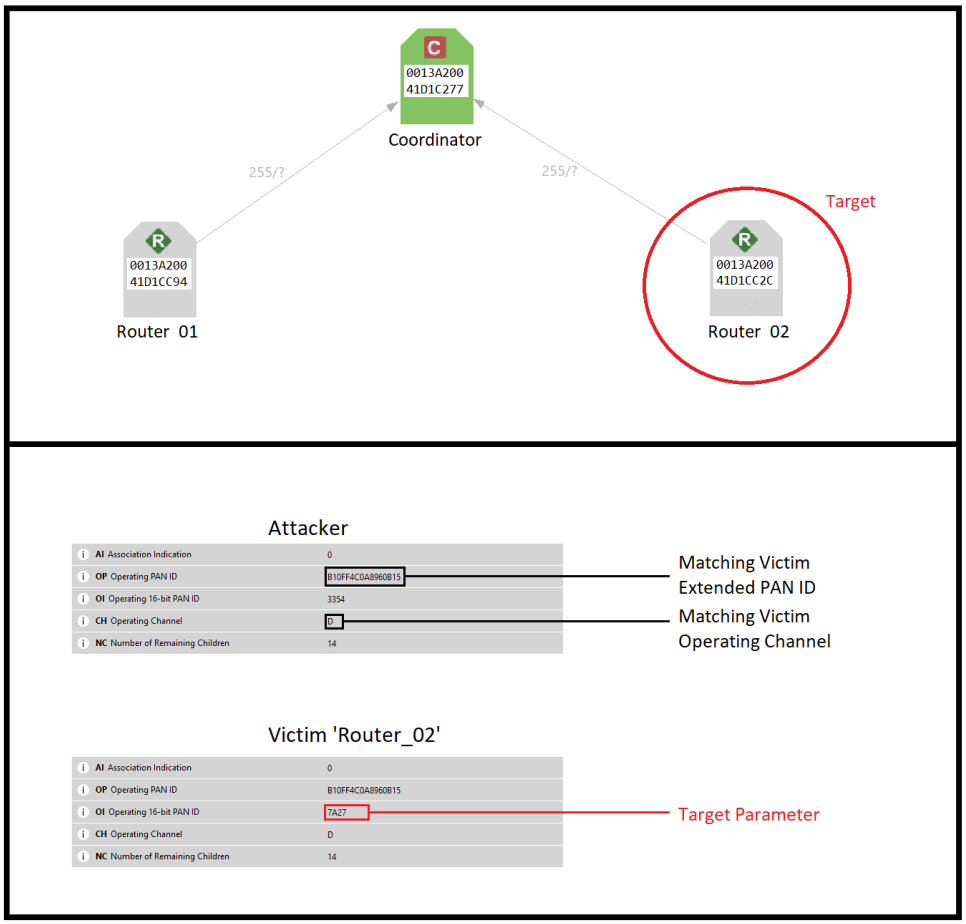

Figure 4.28. Pre-attack network scan and operating parameters for node impersonation attack (CSM network).

While the attacker node is idly waiting with its join window open, the Kali Linux machine transmits a spoofed coordinator realignment frame to the victim node. It is found that the victim node's operating parameters are realigned to match the attacker's parameters but it does not join the attacker's network (see Figure 4.29).

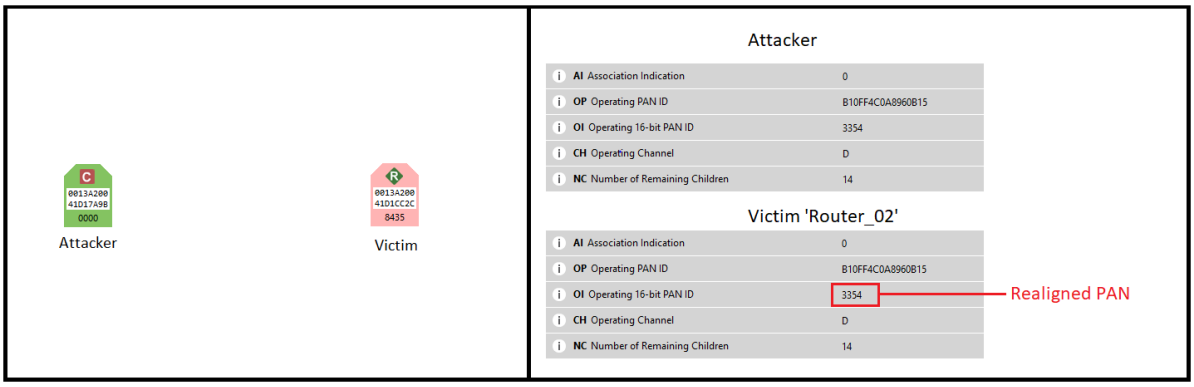

Figure 4.29. Post-attack network scan and operating parameters for node impersonation attack (CSM network).

4.5.2.1.3 DSM Network Findings

On the DSM network, Router_03 is the victim node for this attack (see Figure 4.30).

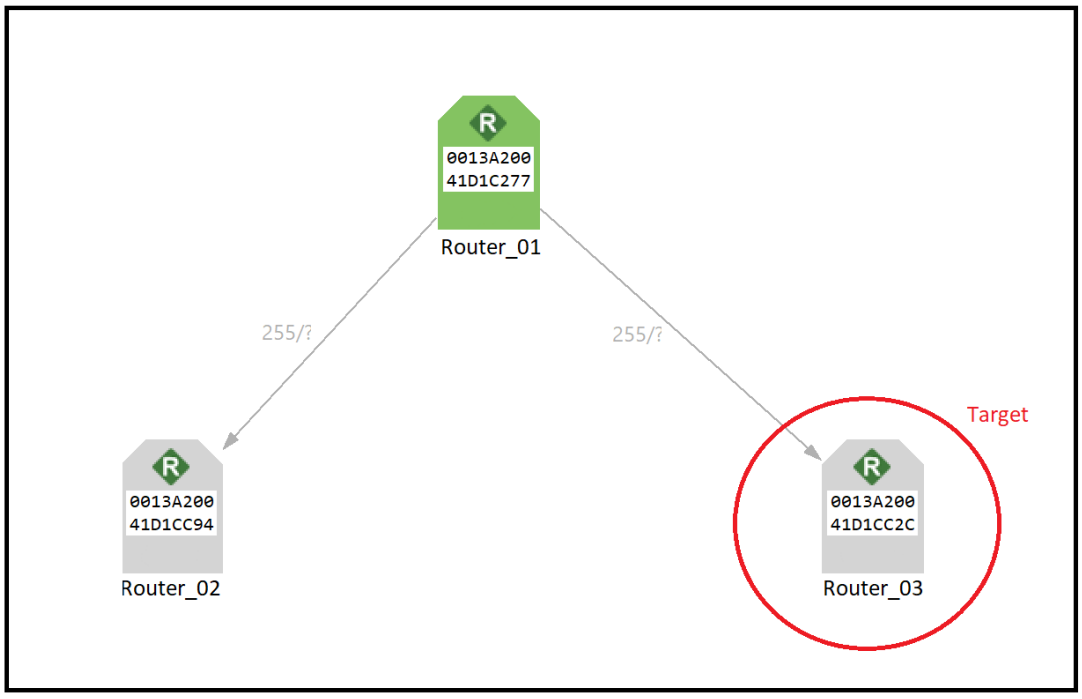

Figure 4.30. Pre-attack network scan for node impersonation attack (DSM network).

With the attacker node idly waiting with its join window open, the spoofed coordinator realignment frame is transmitted from the Kali machine to ‘Router_03’. It is found that immediately after launching the attack, ‘Router_03’ leaves its initial network and joins the

attacker's network. The other nodes of the victim network appear when initiating an XCTU network scan from the attacker node but do not join the network (see Figure 4.31).

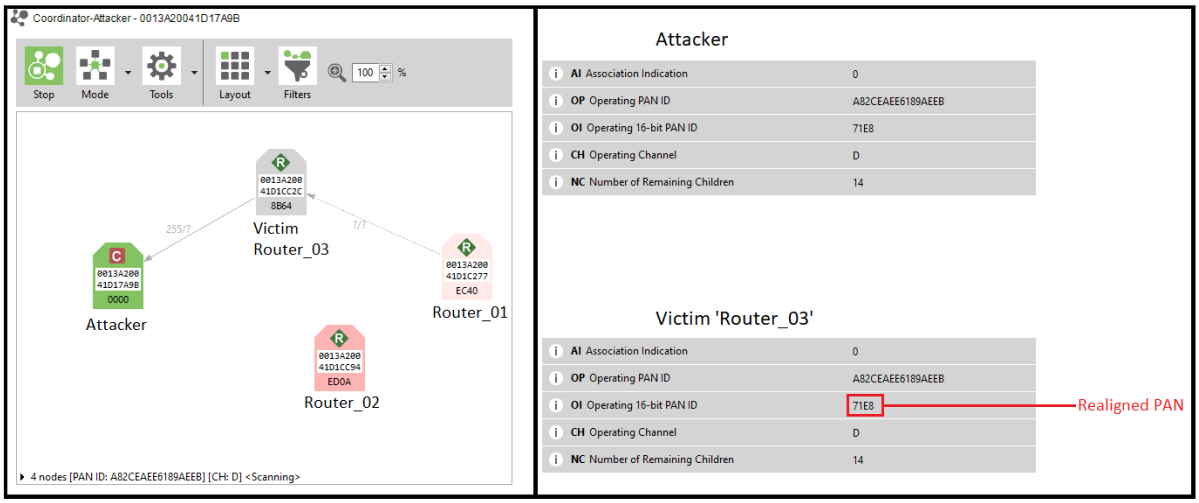

Figure 4.31. Post-attack network scan and operating parameters for node impersonation attack (CSM network).

4.5.3 Summary of Security Issue 2 Findings

4.5.3.1 Eavesdropping Attacks

The findings of the experiments that investigated eavesdropping attacks using compromised symmetric keys on ZigBee 3.0 networks are summarised as follows:

- **Key Sniffing:**

A compromised link key can be used to decrypt the symmetric keys sent to a joining device on secured ZigBee 3.0 networks, as shown in Section 4.5.1.1. The compromised link key was used on the CSM network to capture and decrypt the network key and the updated trust centre link key. Furthermore, when the trust centre initiated a network key rotation, the network key was encrypted with the previous (compromised) network key. In contrast, only a single symmetric key (network key) was sent to the joining device on the DSM network. The network key was decrypted with the compromised link key. Once the symmetric keys of the

DSM network are obtained, the network is indefinitely compromised because the keys are fixed and cannot be changed or rotated.

- **Packet Decryption**:

In the ZigBee protocol, broadcast/NWK layer communications are encrypted with the network key (Radmand et al., 2010). If an attacker obtains the network key, these communications can be decrypted, as demonstrated in Section 4.5.1.2. Furthermore, when a generic ZigBee packet, including a transmit request, is sent from one device without APS encryptions, encryption is applied to the NWK layer using the network key.

The ZigBee protocol encrypts unicast/APS layer communications with the link key. On the CSM network, it is found that the updated trust centre link key was used to encrypt ongoing APS encrypted packets after the device joins the network. However, the DSM network continues to encrypt with the preconfigured link key since there are no additional symmetric keys upon joining.

4.5.3.2 Node Impersonation Attack

The node impersonation attack utilised the compromised symmetric keys (excluding the updated trust centre link key) to configure an impersonated coordinator and hijack a victim node from a targeted network, as discussed in Section 4.5.2. A spoofed coordinator realignment frame was sent to a victim node while the join window of the impersonating coordinator was open. Against the CSM network, the victim node left its initial network but did not join the attacker node. On the DSM network, the victim node left and joined the impersonated coordinator network.

4.6 Security Issue 2: Insufficient Denial of Service Protection Mechanisms

The experiments of security issue 3 analyse the sufficiency of DoS protection mechanisms on ZigBee 3.0 through executing a series of attacks and evaluating their impact against the availability of the network and services. The DoS attack experiments are performed

against a single CSM network internally and externally, and the impact is evaluated through observational (qualitative) findings and numerical (quantitative) findings.

- **Security Configuration and Setup:**

The DoS attack experiments are performed against a single ZigBee 3.0 CSM network consisting of eight nodes, including three gateway nodes and five end device nodes (see Figure 4.32).

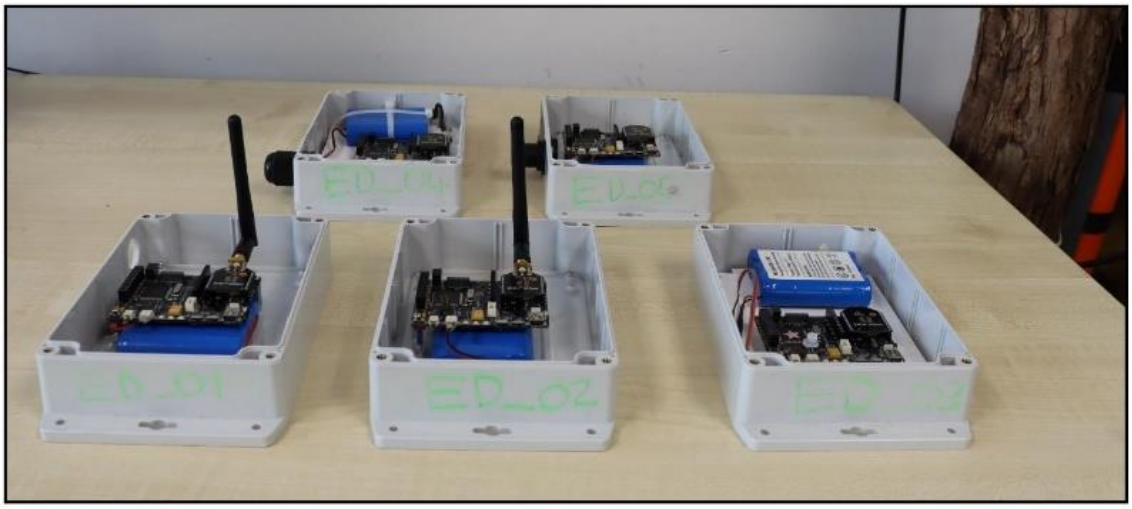

Figure 4.32. End device nodes for DoS experiments.

- ***Security Configuration*:**

The network uses a centralised trust centre (EO = 2), and each device is initially authenticated with a preconfigured link key (see Figure 4.33):

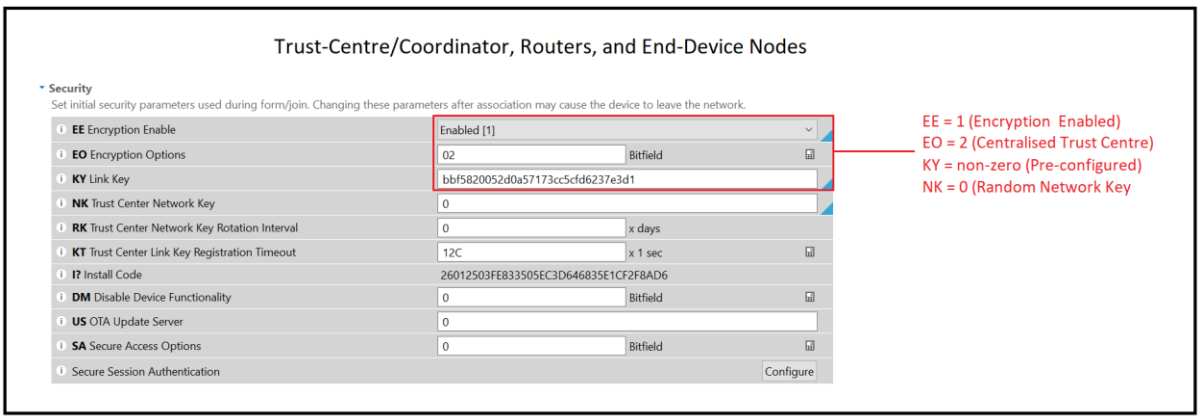

Figure 4.33. Security configuration for Security Issue 3 (DoS) experiments.

- ***End Device Functionality*:**

Each end device node is programmed to report its battery level by generating and transmitting a packet to a router (gateway) node every 3 seconds. The programme uploaded to each Waspmote v.1.5 Board is a modified 'XBee' communications example code found in the Waspmote IDE (see Appendix C for end device code). Each end device transmits a packet to the following router node:

1. End-Device_01: Sends a Packet to Router_01 every 3 seconds.
2. End-Device_02: Sends a Packet to Router_01 every 3 seconds.
3. End-Device_03: Sends a Packet to Router_01 every 3 seconds.
4. End-Device_04: Sends a Packet to Router_02 every 3 seconds.
5. End-Device_05: Sends a Packet to Router_02 every 3 seconds.

- ***Gateway Functionality:***

The gateway nodes idly relay the messages between nodes across the network. In specific experiments, a router node is configured to send packets to the other router node at defined intervals (see Appendix D for packet creation in XCTU).

4.6.1 External DoS Attacks

The external DoS attacks are externally performed against the network using the Kali Linux machine and ApiMote hardware to exploit the functionality of the ZigBee protocol and attempt to affect the availability of the testbed network and services. The first two experiments (Experiments 9 and 10) are flooding attacks that utilise KillerBee tools and ApiMote hardware to rapidly generate and transmit spoofed packets using the obtained network information across the victim network's operating channel. The third experiment (Experiment 11) exploits the protocol with a spoofed realignment packet in an attempt to isolate a victim node from its network.

Table 4.8

External DoS Experiment Descriptions

Test ID	Network	Test Description
Experiment 9: PAN-ID Conflict Flooding (External)		
T15	CSM	Measuring the number of successfully received packets on router nodes sent from the end device nodes
T16	CSM	Measuring the number of successfully received packets by Router_02 sent from Router_01
T17	CSM	Testing XCTU (software) functionality and the trust centre's ability to authenticate nodes
T18	CSM	Testing the impact of an extended PAN-ID Conflict Flooding attack against a ZigBee 3.0 network
T19	CSM	Testing whether adjusting the network's PAN Conflict Threshold will mitigate and reduce the number of PAN conflict reports
Experiment 10: Association Flooding (External)		
T20	CSM	Measuring the number of successfully received packets on router nodes sent from the end device nodes
T21	CSM	Testing XCTU (software) functionality and the trust centre's ability to authenticate nodes
Experiment 11: Network Realignment Attack (External)		
T22	CSM	Realigning the PAN-ID of a victim node to halt its ability to route/receive data and leave the network

4.6.1.1 Experiment 9: PAN-ID Conflict Flooding

In ZigBee networks, when a node receives a beacon request frame with the same 16-bit PAN-ID, it responds by reporting a PAN-ID conflict to the network manager (Coordinator). This function is exploited on the ZigBee 3.0 network using the KillerBee tool 'zbpanidconflictflood' along with two ApiMote interfaces.

4.6.1.1.1 Attack Execution

Figure 4.34 shows the syntax and execution of the KillerBee script 'zbpanidconflictflood' to trigger PAN-ID changes on the ZigBee 3.0 network. After launching the attack, the coordinator receives a PAN-ID conflict report and migrates the network to a new 16-bit PAN-ID (see Figure 4.34).

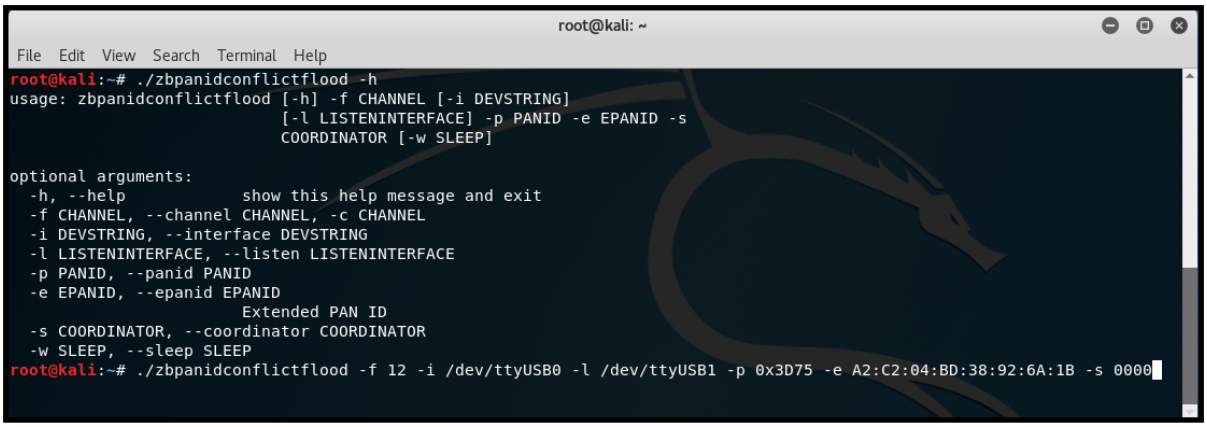

Figure 4.34. Executing ‘zbpanidconflictflood’ script in Kali Linux.

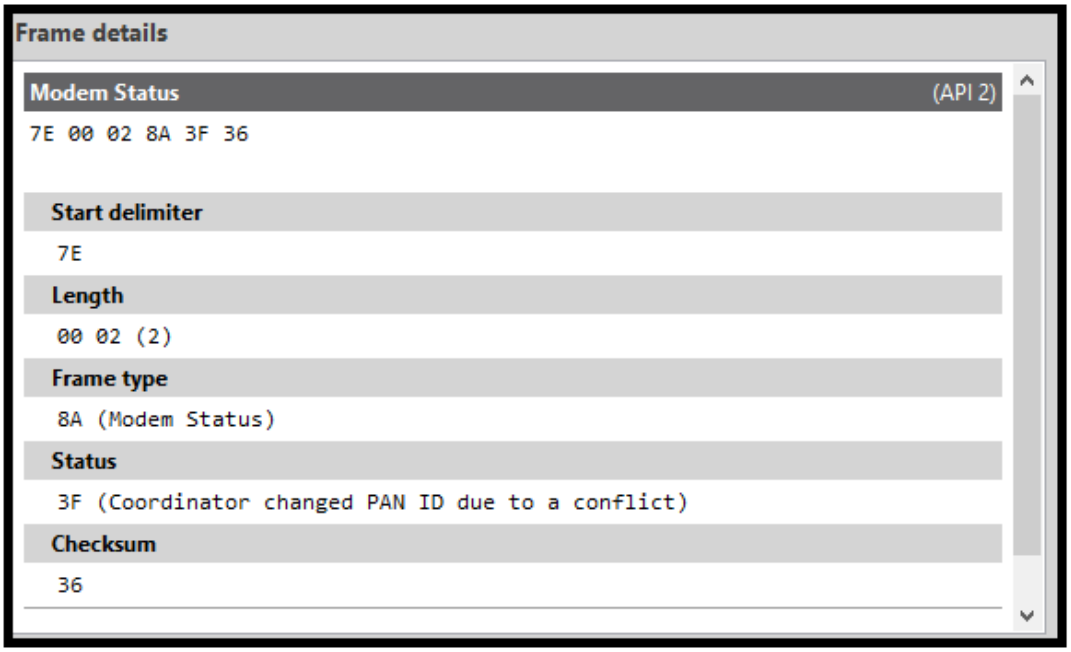

Figure 4.35. Coordinator PAN-ID change.

4.6.1.1.2 Impact on Received Packets (Router Nodes)

Figure 4.36 shows the number of successfully received packets on router nodes sent from the end device nodes over 15 minutes.

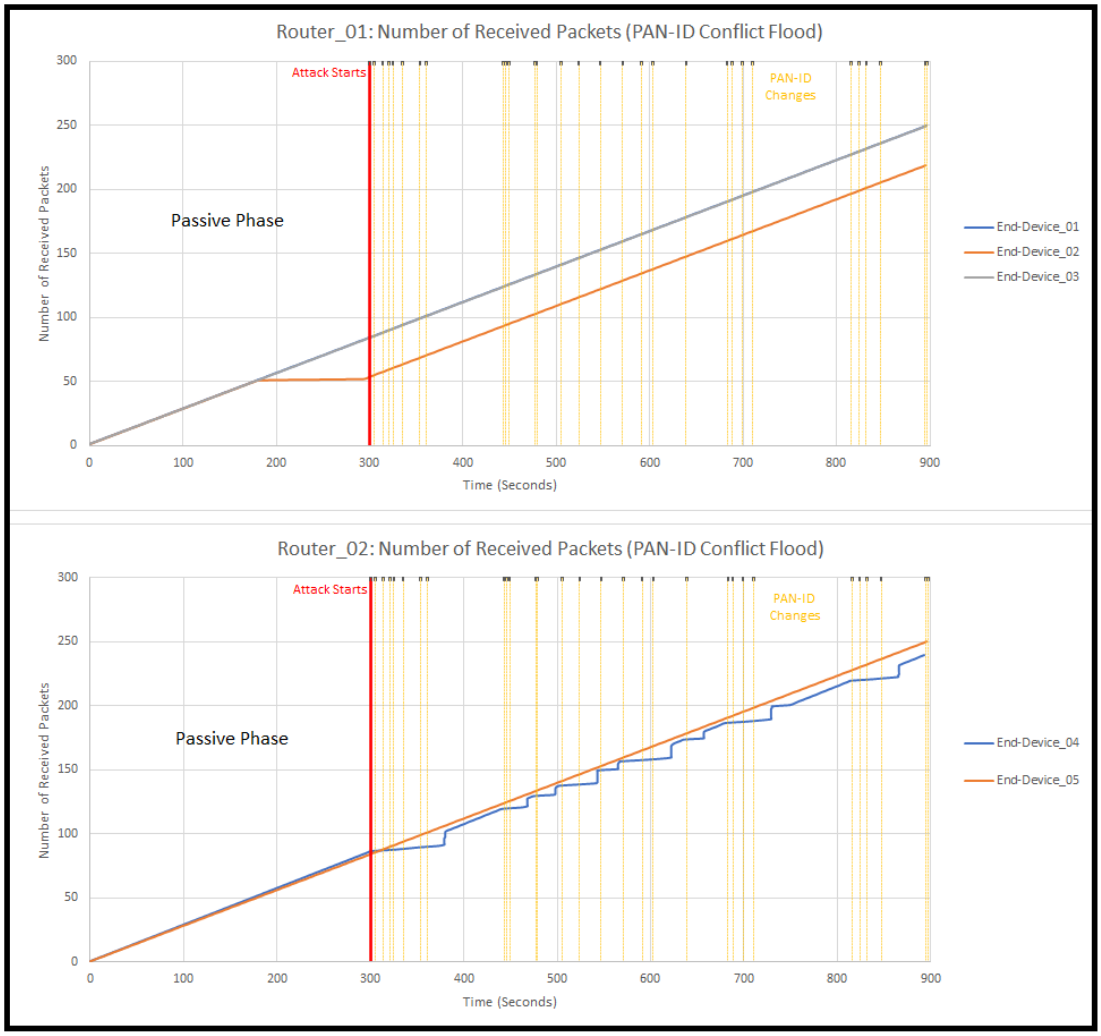

Figure 4.36. Number of received packets on router nodes (PAN-ID flooding).

After launching the attack, the router nodes continued to receive packets from the end devices without significant delays or dropped packets. On Router_02, packets sent from End-Device_04 experienced slight delays during the attack.

When sending packets (sent at 3-second internals) from one router to another over 15 minutes, noticeable effects were detected on the packet receive rate when the attack started (see Figure 4.37). The intervals where the router nodes undergo a PAN-ID change show the delays in receiving packets.

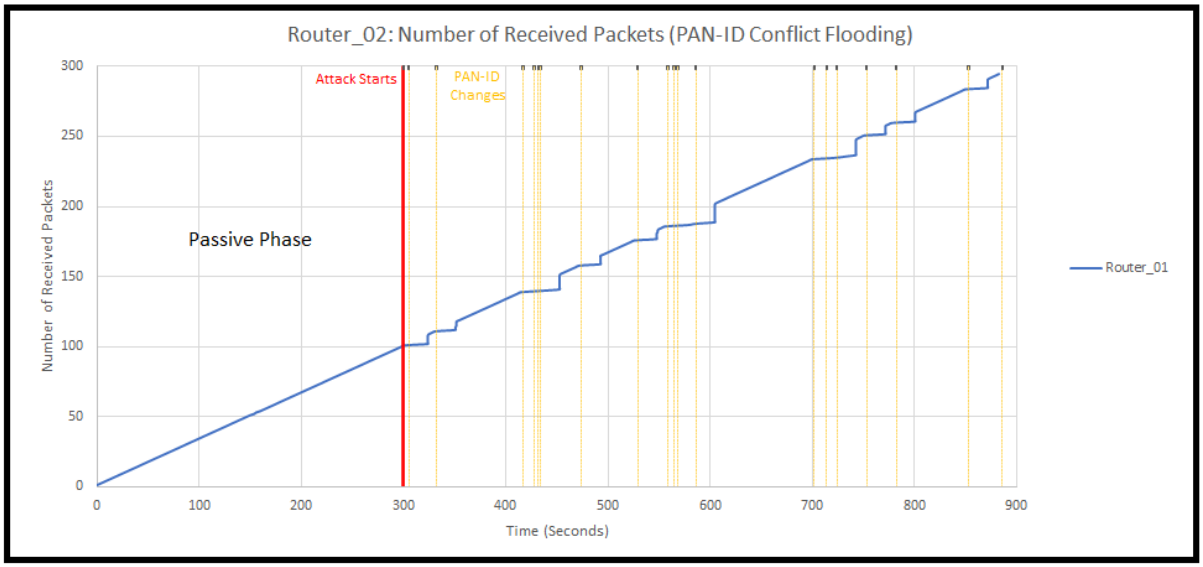

Figure 4.37. Number of received packets on Router_02 (PAN-ID conflict flooding).

4.6.1.1.3 Impact on XCTU Functionality and Device Authentication

When the join window opened with the attack in motion, End-Device_01 was able to rejoin the network, and End-Device_02 could successfully authenticate and join the network through the coordinator node. In addition, the coordinator can discover and map nodes without delays when initiating an XCTU network scan. However, the router nodes were able to discover nodes in the network only once the PAN-ID changes took effect (see Figure 4.38).

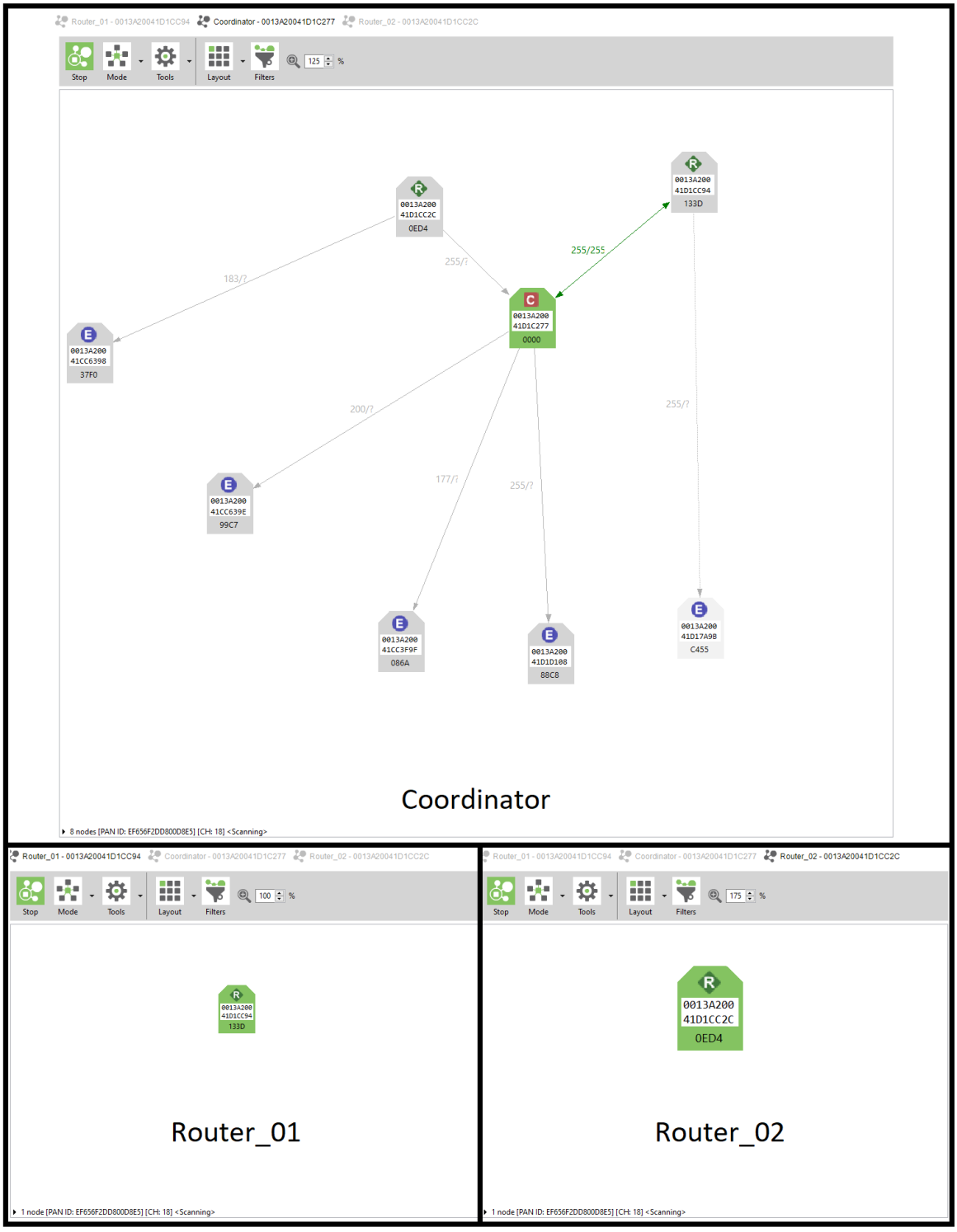

Figure 4.38. XCTU network scan from gateway nodes (PAN-ID conflict flooding).

4.6.1.1.4 Impact of an Extended (12 Hours) Attack

Figure 4.39 shows XCTU network scans performed on the coordinator node before and after the 12-hour PAN-ID conflict flooding attack.

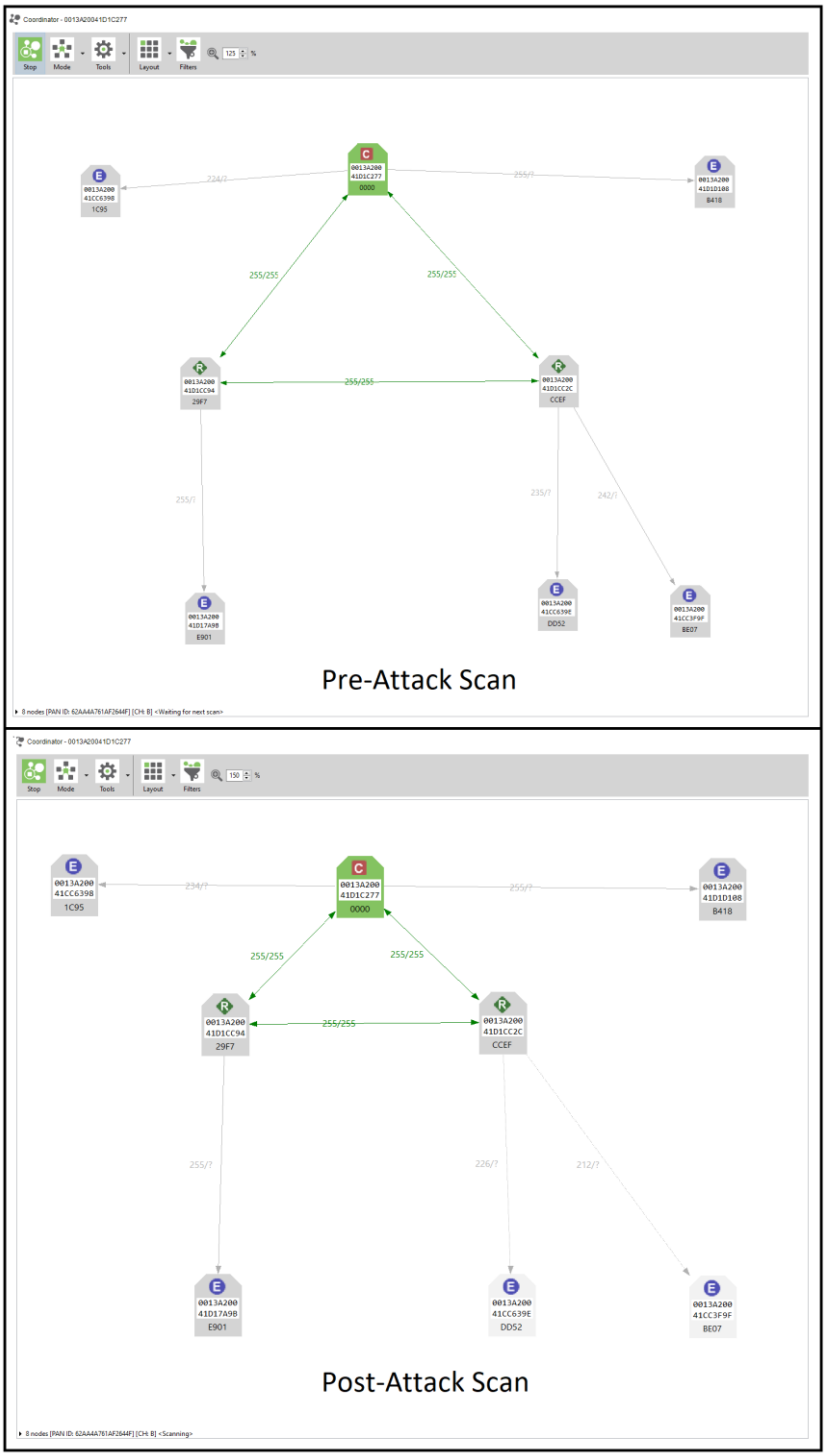

Figure 4.39. XCTU network scans before and after extended attack (PAN-ID conflict flooding).

The extended attack caused no change to the network's initial routing structure, and each node continued to operate without crashing. Throughout the attack (12 hours), the coordinator changed the PAN-ID a total of 1,517 times.

4.6.1.1.5 Mitigating PAN-ID Conflict Flooding Attacks

PAN-ID conflict flooding can be addressed or mitigated on ZigBee 3.0 networks by adjusting the ‘PAN Conflict Threshold’ (CR) on the Coordinator/Network Manager node (see Figure 4.40). The CR determines the number of PAN-ID conflict reports that must be received within one minute in order to trigger a PAN-ID change, and its default value is 3.

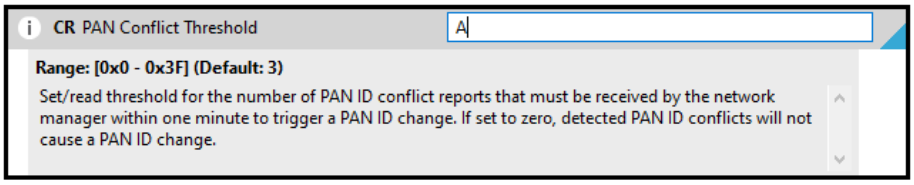

Figure 4.40. PAN conflict threshold on XBee 3.

Increasing the CR to A (10) significantly reduced the number of PAN-ID changes over a 10-minute interval. The coordinator changed the PAN-ID only once after 5 minutes into the attack and caused minimal disruption to the router nodes. Alternatively, the CR can be set to 0 to disable the PAN-ID change feature.

4.6.1.2 Experiment 10: Association Flooding

Association flooding is an attack that attempts to overwhelm a ZigBee device from too many connecting stations. This attack works by repeatedly sending association requests to the discovered PAN-ID of a victim network (River Loop Security, n.d.-b). In ZigBee 3.0, the join window must be open for the devices to respond to the attack, and the default join window of XBee 3 is 254 seconds (NJ=FE). The commissioning button is pressed on a gateway node to open the join window after the attack commences. Association requests are injected from both ApiMote tools to maximise the number of requests with the utilised hardware.

4.6.1.2.1 Attack Execution

Figure 4.41 shows the syntax and execution of the KillerBee script ‘zbassociationflood’ to inject association requests against the target 16-bit PAN-ID.

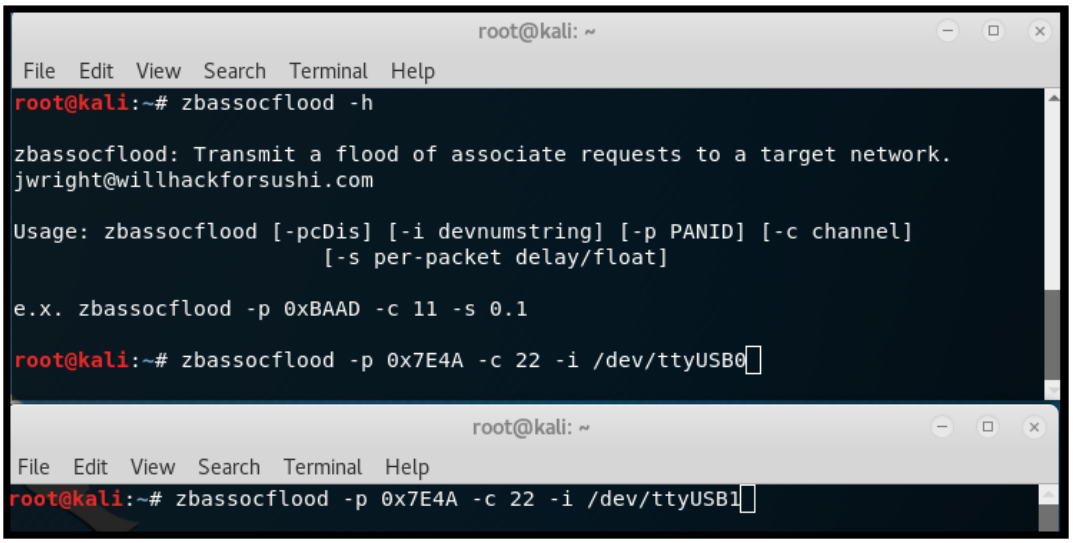

Figure 4.41. Executing ‘zbassocflood’ in Kali Linux.

4.6.1.2.2 Impact on Received Packets (Router Nodes)

Figure 4.42 shows the number of successfully received packets on router nodes sent from end devices over 15 minutes.

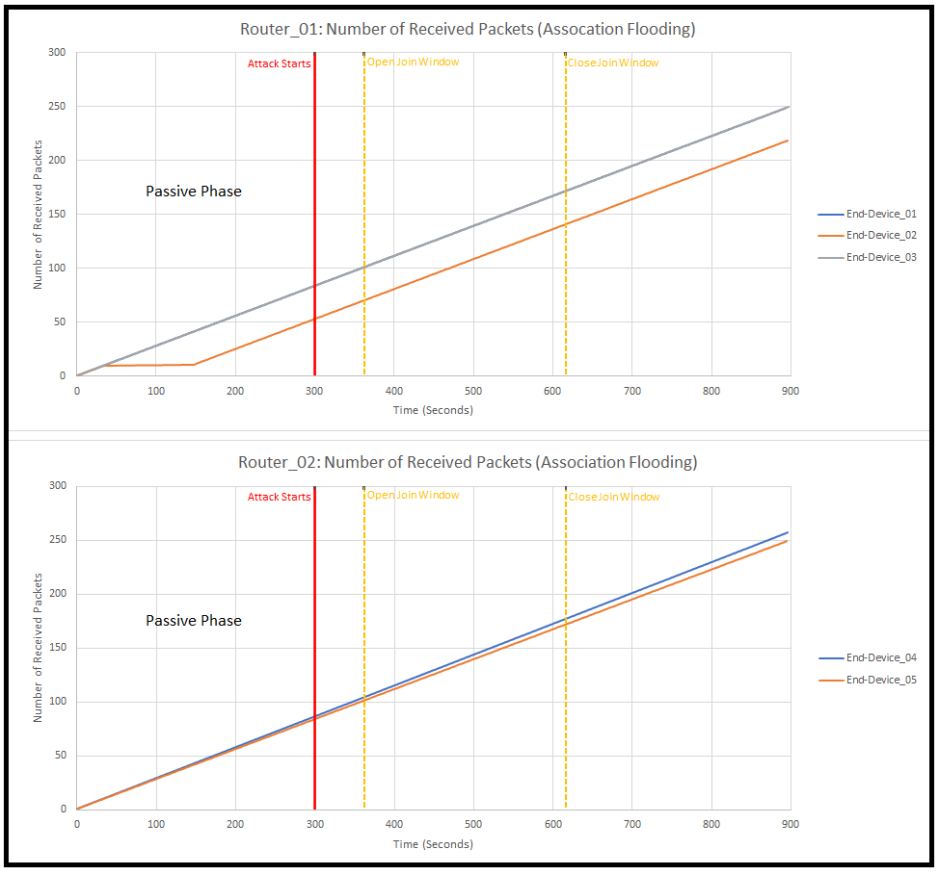

Figure 4.42. Number of received packets on router nodes (association flooding).

Throughout the experiment (900 seconds), the number of received packets grew at a linear rate on both router nodes. This result indicates that the association flooding attack had no impact on the rate of received packets, even after opening the join window.

4.6.1.2.3 Impact on XCTU Functionality and Device Authentication

After opening the join window during the attack, End-Device_01 could rejoin the network, and End-Device_02 could successfully authenticate and join the network through the coordinator node. Fake association requests appeared in the XCTU network scan from gateway nodes after opening the join window (see Figure 4.43). However, it did not affect the nodes' functionality. Once the join window closes, the fake association requests are dropped, and the gateway nodes do not respond to the requests.

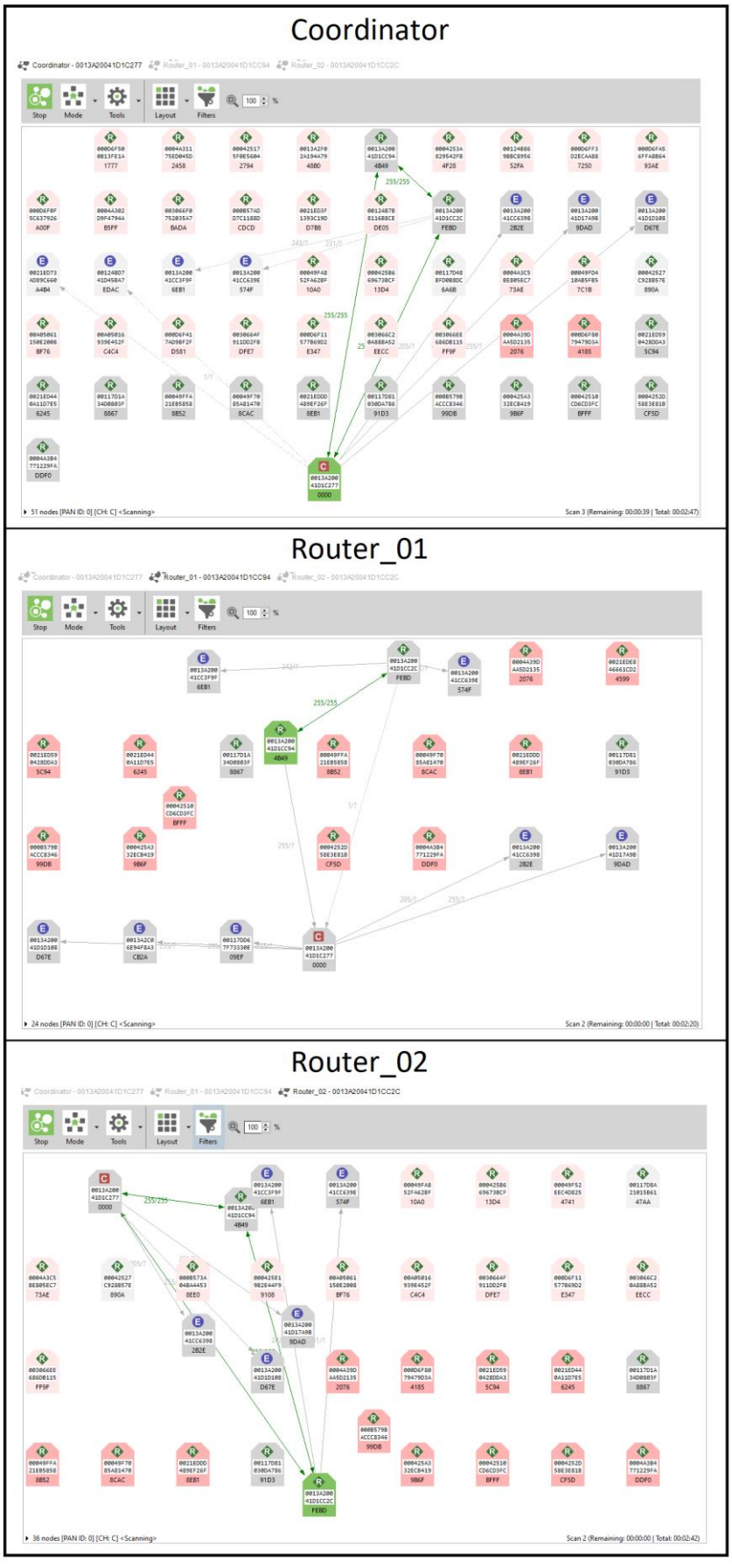

Figure 4.43. XCTU network scan from gateway nodes (association flooding).

4.6.1.3 Experiment 11: Network Realignment Attack

The network realignment attack targets a single node and realigns its operating parameters to isolate it from its network, resulting in DoS. This attack aims to realign the target node ‘Router_02’ to a new PAN-ID to prevent its ability to route/receive data and cause its connected child nodes to leave the network. Figure 4.44 shows the target node (Router_02) and its four connected end device nodes.

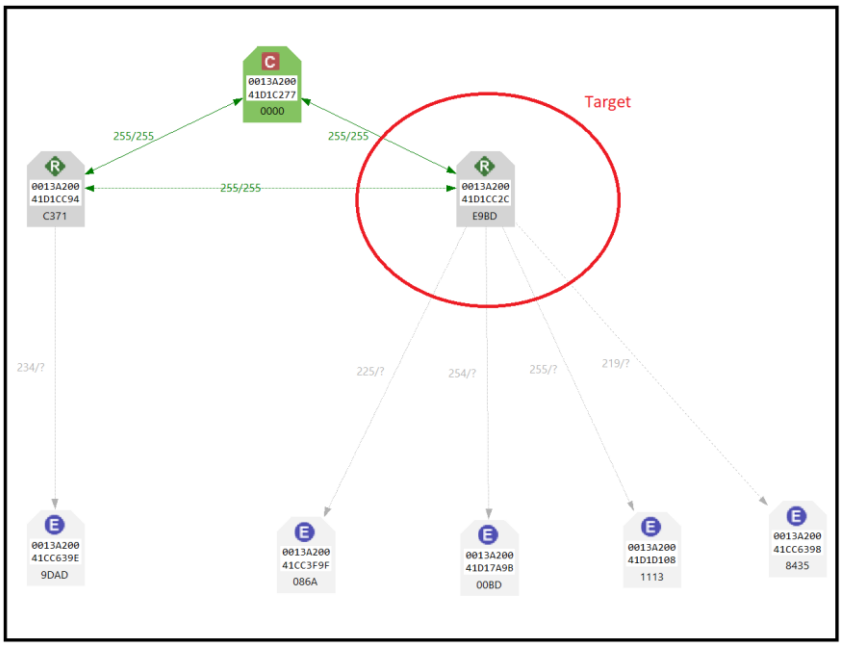

Figure 4.44. Pre-attack scan on XCTU (network realignment attack).

4.6.1.3.1 Attack Execution

Figure 4.45 shows the KillerBee tool ‘zbrealign’ used to create and transmit a spoofed coordinator realignment frame to the MAC address of Router_02. The spoofed frame is sent and captured in Wireshark, showing the realignment commands issued to Router_02.

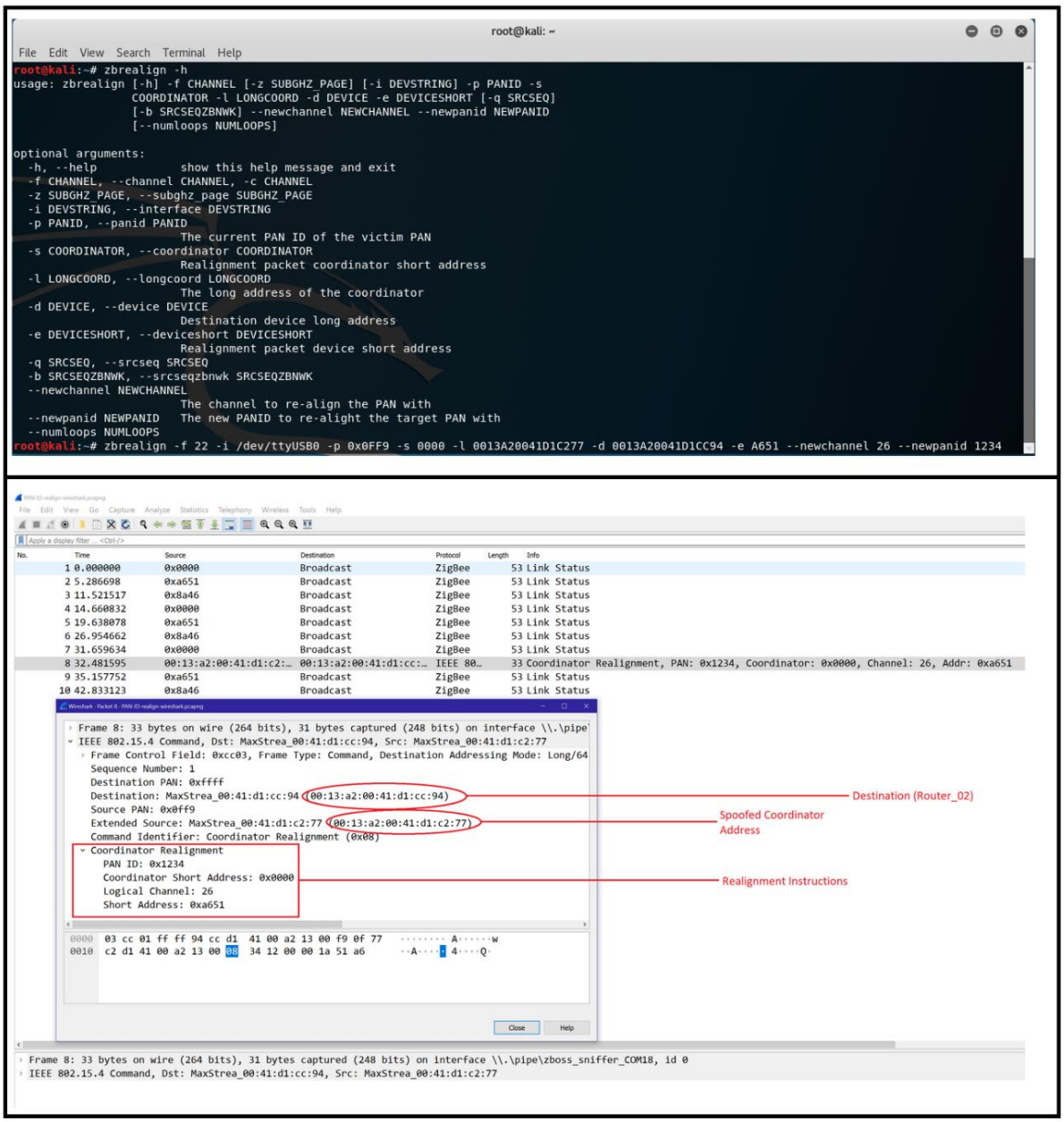

Figure 4.45. Executing ‘zbrealign’ script and frame capture.

4.6.1.3.2 Impact on ZigBee 3.0 Network

Immediately after launching the attack, Router_02 leaves the network and changes its PAN-ID to match the realignment instructions contained in the spoofed coordinator realignment frame (see Figure 4.46).

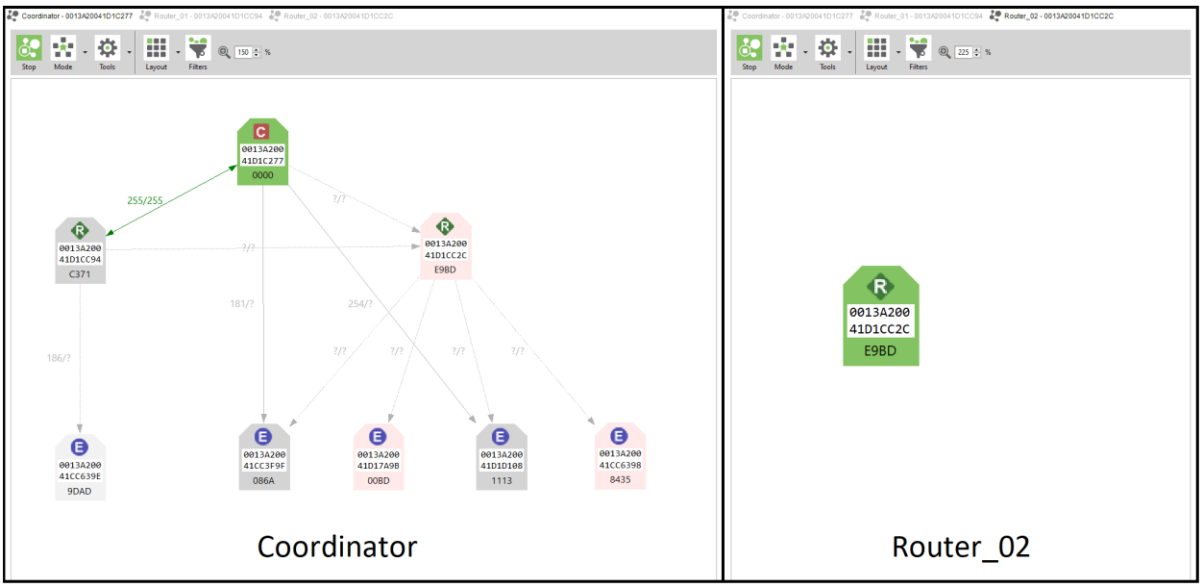

Figure 4.46. Network realignment attack on ZigBee 3.0 network.

End-Device_01 and End-Device_04 did not change parent nodes after launching the attack and only rejoined the network after opening the join window. Router_02 remained separated from the coordinator's network and could only rejoin after resetting its networking parameters and re-registering the device.

4.6.2 Internal DoS Attacks

In this section, internal DoS attacks are performed against the ZigBee 3.0 network using a compromised router node (Router_02) that is already part of the network. The first experiment (Experiment 12) is a flooding attack that spams a single node with legitimate messages containing the largest supported payload size. In the second experiment (Experiment 13), remote AT commands are exploited to cause DoS against the victim coordinator. These experiments evaluate internal attacks that target the availability of the network.

Table 4.9

Internal DoS Experiment Descriptions

Test ID	Network	Test Description
Experiment 12: Protocol Flooding (Internal)		
T23	CSM	Measuring the number of successfully received packets on Router_01 sent from the end device nodes
T24	CSM	Testing the impact of an extended protocol flood against a ZigBee 3.0 network
Experiment 13: Blackhole Attack Using Remote AT Commands (Internal)		
T25	CSM	Sending a remote AT network reset (NR) command to a victim device to halt its ability to route/receive data and leave the network

4.6.2.1 Experiment 12: Protocol Flooding

Protocol flooding is an internal attack that involves flooding a victim node with legitimate messages from inside the network (Chaitanya & Arindam, 2011). This attack attempts to hog network resources and disrupt the victim node's ability to send or receive data.

4.6.2.1.1 Packet Creation for Attack

A legitimate packet is created within XCTU's console, which contains the maximum payload size to be transmitted from Router_02 to Rouer_01. The frame is rapidly transmitted on an infinite loop from Router_02 with the transmit interval (ms) set to '0' (see Figure 4.47).

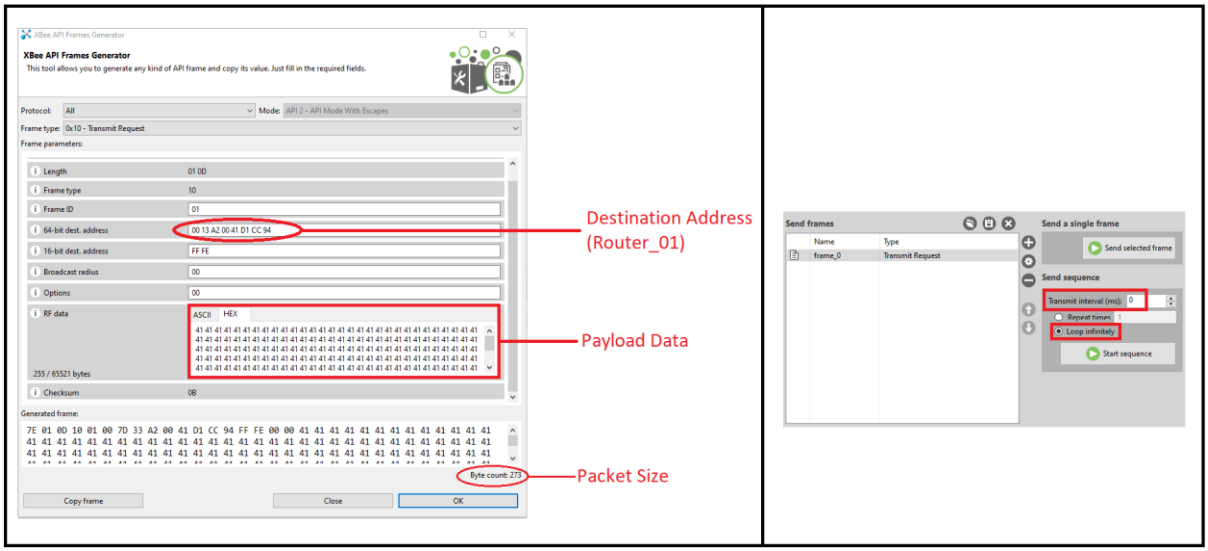

Figure 4.47. Creating and sending protocol flooding packets.

4.6.2.1.2 Impact of Received Packets (Router_01)

Figure 4.48 shows the number of successfully received packets sent from the end device nodes to Router_01 over 15 minutes:

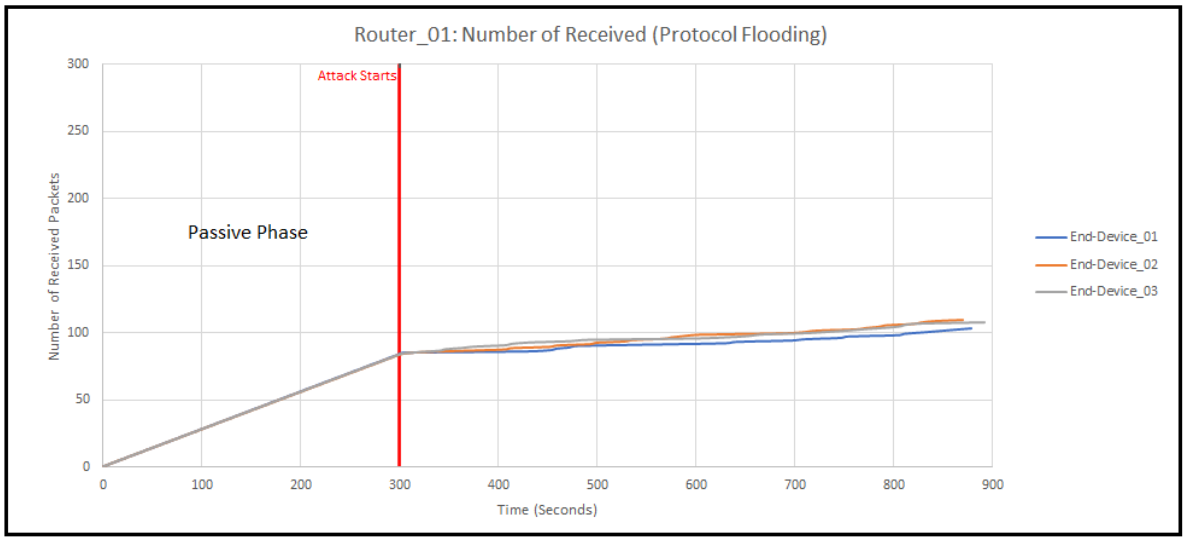

Figure 4.48. Number of received packets on Router_01 (protocol flooding)

Immediately after launching the attack, the rate of received packets sent from the end device nodes significantly dropped. Router_01 could not receive most packets while the attack was in motion.

4.6.2.1.3 Impact of an Extended (12 Hours) Attack

Figure 4.49 shows the XCTU network scans taken from the coordinator node before and after the protocol flooding attack.

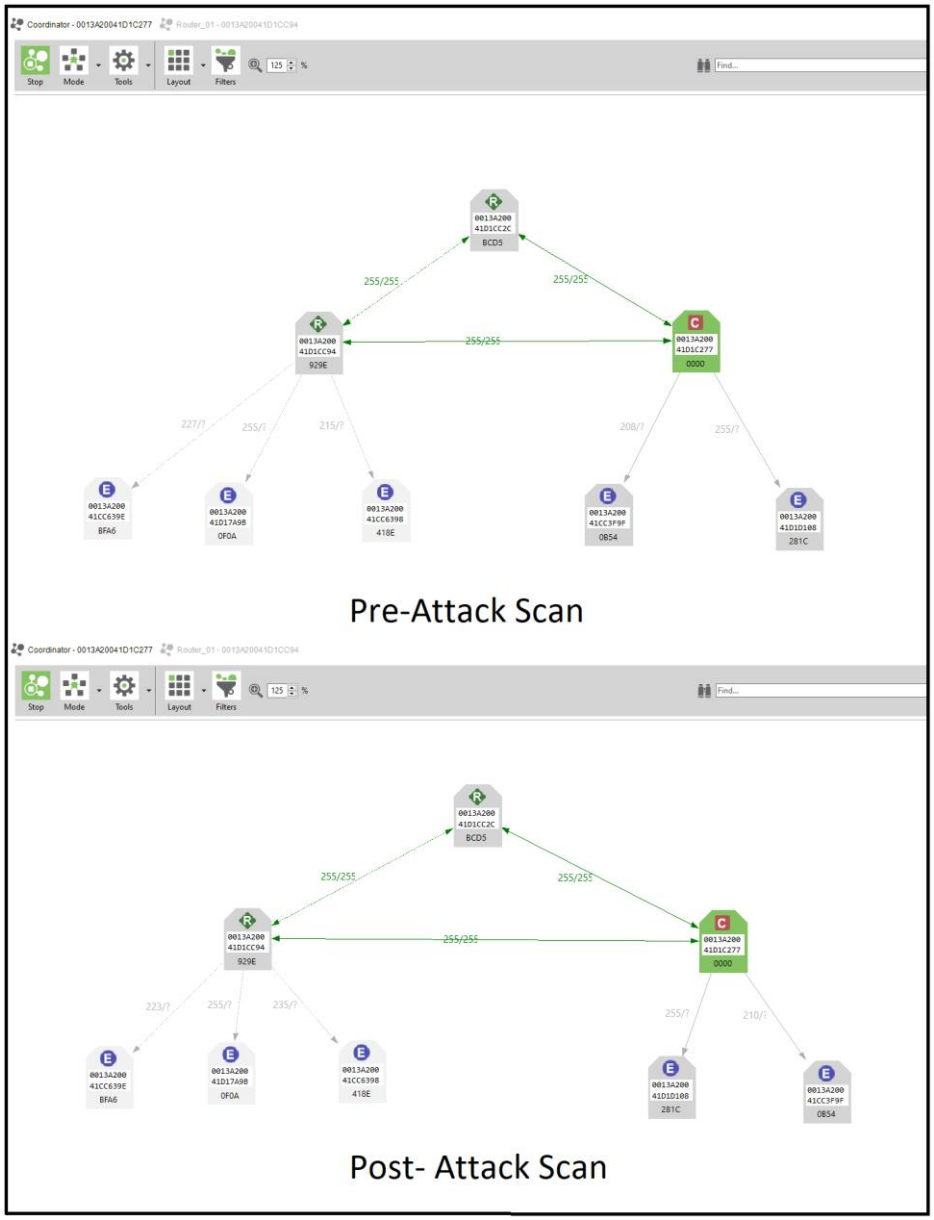

Figure 4.49. XCTU network scans before and after extended attack (protocol flooding).

The extended protocol attack caused no changes to the network's initial routing structure. Moreover, none of the nodes became disconnected from the network. Over 12 hours, the compromised router (Router_02) flooded the victim node (Router_01) with 151,374 packets.

4.6.2.2 Experiment 13: Blackhole Attack Using Remote AT Commands

Remote AT commands are used in the ZigBee protocol to configure devices remotely. In this experiment, the functionality of remote AT commands is exploited to cause DoS on a ZigBee 3.0 network. The objective of the attack is to reset the victim coordinator's networking parameters to eliminate the routing path of its child nodes and thus cause their sent packets to be discarded.

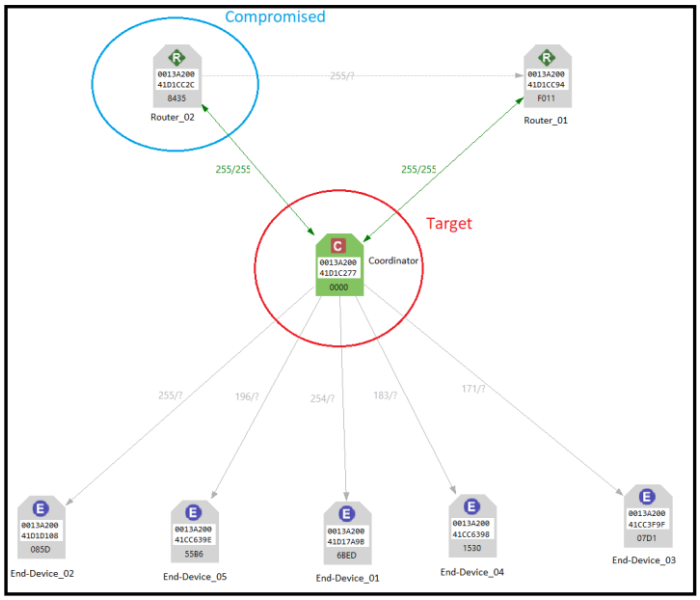

Figure 4.50. Pre-attack XCTU network scan (blackhole attack).

4.6.2.2.1 Packet Creation for Attack

A remote AT command frame is created in the XCTU console on the compromised router node (Router_02) containing the 'NR' (Network Reset) AT command. The coordinator's MAC address is inserted as the 64-bit destination address (see Figure 4.51).

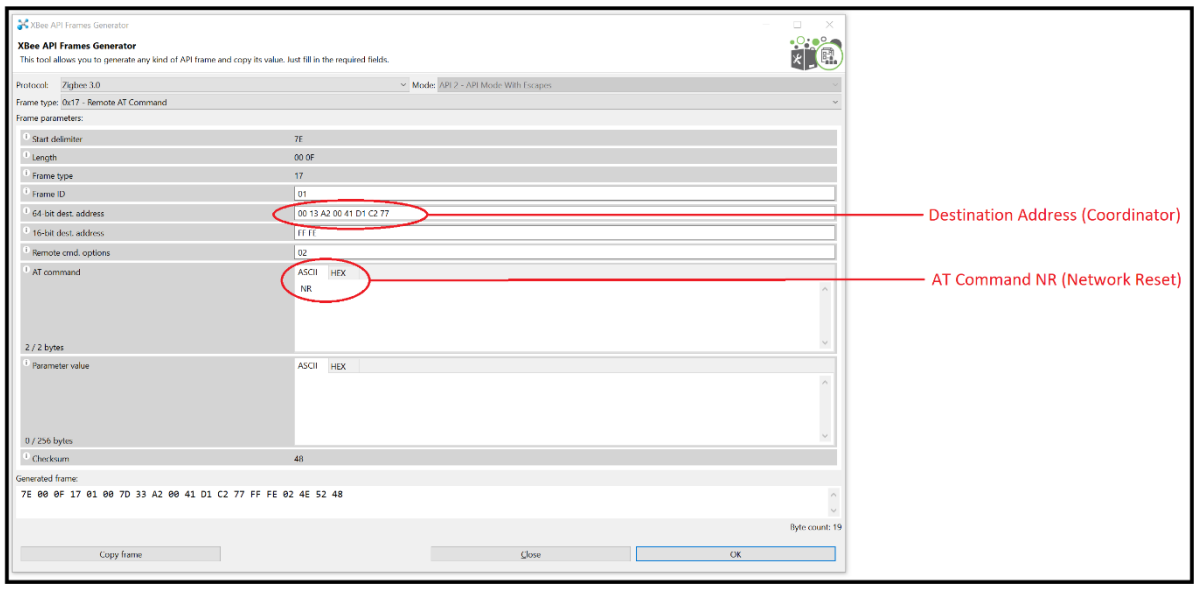

Figure 4.51. Creating a malicious remote AT command.

4.6.2.2.2 Impact on ZigBee 3.0 Network.

After transmitting the malicious remote AT command from the compromised router node to the victim coordinator node, its networking parameters, including its 16-bit PAN-ID, 64-bit PAN-ID and operating channel, are reset. This causes the coordinator to form a new network on its changed parameters. Furthermore, the end device nodes initially connected to the coordinator before the attack follow the coordinator to the new network. A blackhole effect is created as the packets generated from the end device nodes intended for Router_01 and Router_02 are discarded because there is no routing path to these nodes (see Figure 4.52).

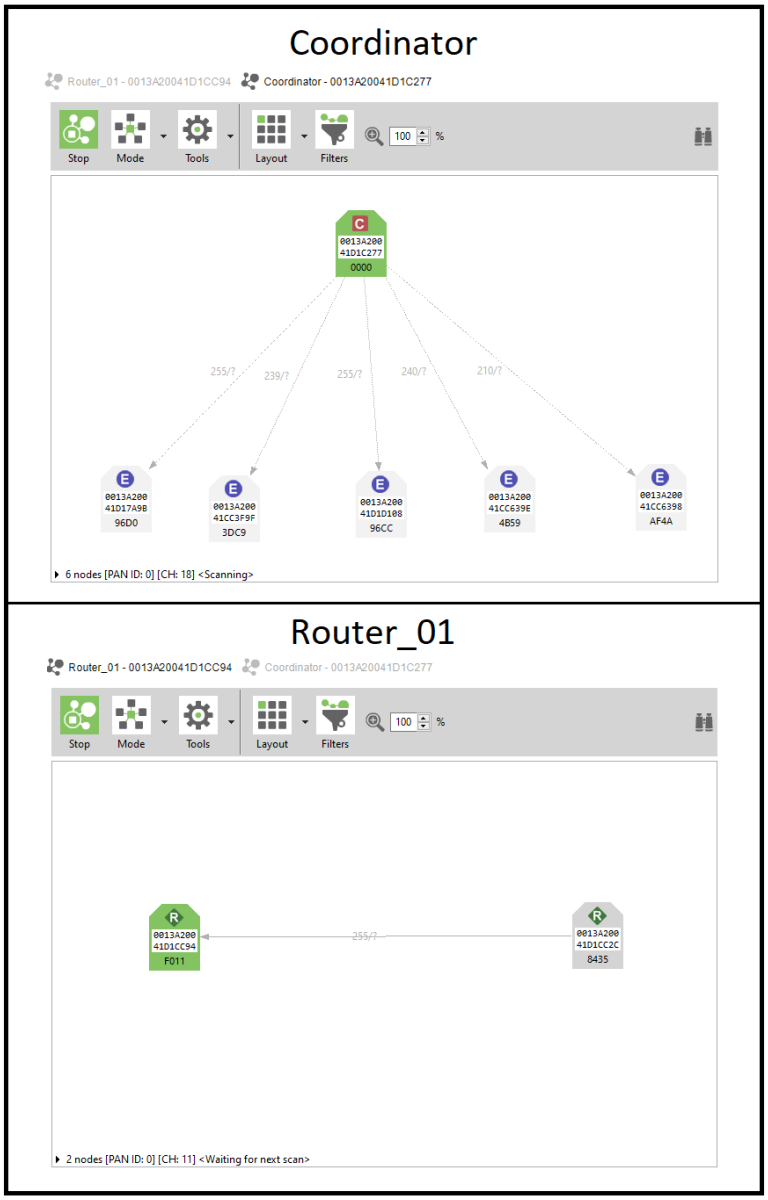

Figure 4.52. Post-attack XCTU network scan (blackhole attack).

4.6.2.2.3 Mitigation

XBee 3 ZigBee 3.0 supports Secure Remote Password (SRP) authenticated remote access (Secure Session), which can be applied to prevent the exploitation of remote AT commands on ZigBee 3.0 networks. In the security configuration on the XBee 3 modules, 'Secure Access Options' (SA) can be enabled to require SRP authentication against remote AT commands (see Appendix E for SRP configuration). The SRP must be individually configured with a unique password for secure session authentication.

When attempting to connect to an SRP-secured XBee 3 module, the user is prompted for the secure access password before a connection can be established to read and write AT commands remotely (see Figure 4.53).

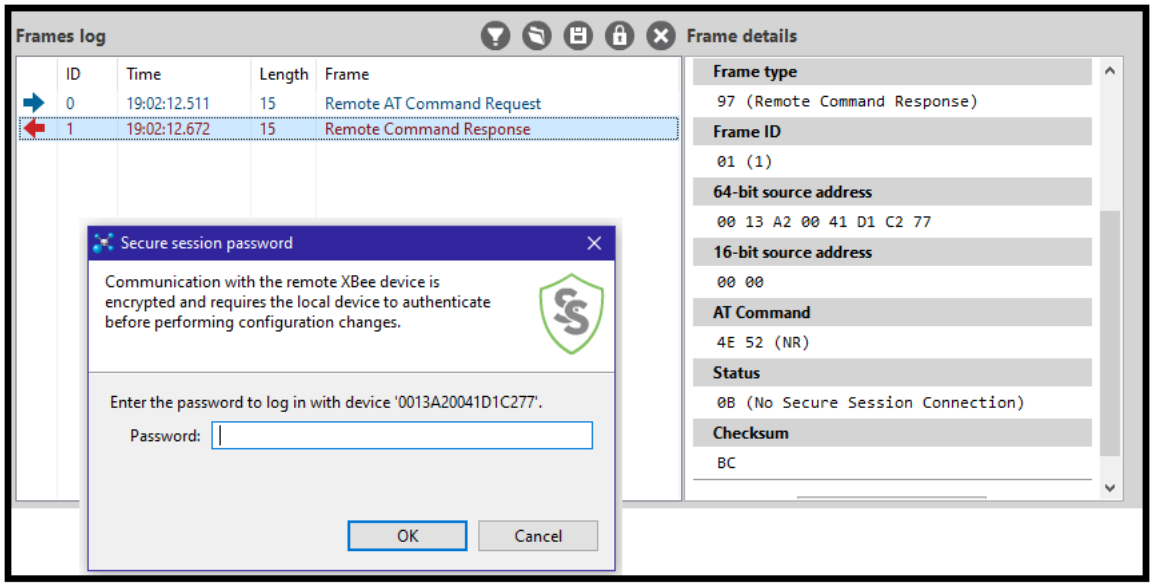

Figure 4.53. SRP-secured node on XBee 3.

4.6.3 Summary of Security Issue 3 Findings

4.6.3.1 External DoS Attacks

The findings of the external DoS attack experiments against a ZigBee 3.0 (CSM) network are summarised as follows:

4.6.3.1.1 PAN-ID Conflict Flooding

PAN-ID conflict flooding was performed, as described in Section 4.6.1.1, and was shown to have only a small to moderate impact against the network. T15 and T16 measured the number of successfully received packets on router nodes, and the results showed that only slight delays to the received rate occurred. Furthermore, packets were not dropped but were received at delayed intervals when a PAN-ID change occurred. When testing the attack against XCTU's functionality and device authentication in T17, an effect on the router node's ability to perform an XCTU network scan was detected. The router nodes could issue the scan only

after the PAN-ID aligned with the coordinators. Device authentication on the trust centre remained undisrupted. T18 showed that an extended (12 hours) attack caused no changes to the initial routing structure or node/network crash. The final test (T19) showed that the 'PAN Conflict Threshold' setting can be used to address or mitigate the attack. By adjusting this value to A (10), the coordinator changed PAN-ID only once after 5 minutes, which caused minimal disruption to the network and nodes. Alternately, the setting can be set to 0 to disable the PAN-ID change feature.

4.6.3.1.2 Association Flooding

Association flooding was performed, as described in Section 4.6.1.2, and was found to have no noticeable impact on the performance or availability of the network. Furthermore, the ZigBee 3.0 protocol mitigates this attack through its limited join window (default 254 seconds), because devices would respond to this attack only when the join window is open. T20 measured the number of successfully received packets on router nodes from end device nodes. However, the results showed that the number of received packets grew at a linear rate on both router nodes. In T21, XCTU's functionality and device authentication was tested against this attack. It was found that fake association requests appeared when issuing an XCTU network scan on gateway nodes, but these requests dropped once the join window closed. Furthermore, the attack had no noticeable impact on device authentication.

4.6.3.1.3 Network Realignment Attack

The network realignment attack discussed in Section 4.6.1.3 targeted a single node with a spoofed coordinator realignment frame containing realignment instructions causing significant DoS against the victim node. As shown in T22, when the victim router node received the spoofed frame, it immediately left the network and changed its 16-bit PAN-ID to match the realignment instructions. Furthermore, two out of four of its initially connected child end device nodes were forced to leave the network and could rejoin only after the join window

opened. The victim router node remained isolated from its network and could rejoin only after resetting its networking parameters and re-registering the device.

4.6.3.2 Internal DoS Attacks

The findings of the internal DoS attack experiments against a ZigBee 3.0 (CSM) network are summarised next.

4.6.3.2.1 Protocol Flooding

Protocol flooding was performed, as described in Section 4.6.2.1, in which a victim router node was bombarded with legitimate packets sent from a compromised router node containing ZigBee's maximum payload size. The attack was found to cause significant processing and routing interruptions against the victim node. T23 measured the number of successfully received packets from end device nodes on the router (victim) node. The results showed that the attack caused a significant delay in receiving the sent packets, with the majority of packets unable to be delivered while the attack was in motion. However, when performing an extended attack (12 hours) in T24, it was found that the attack caused no change to the network's initial routing structure or any node crash.

4.6.3.2.2 Blackhole Attack (Exploiting Remote AT Commands)

As described in Section 4.6.2.2, a remote AT 'network reset' command was used to perform a blackhole attack, which caused significant DoS against the ZigBee 3.0 CSM network. As demonstrated in T25, a network reset AT command was created on a compromised router node and transmitted to the victim network's coordinator node. When the coordinator received the malicious AT command, its networking parameters, including its 16-bit and 64-bit PAN-IDs, and operating channel were reset. This attack caused the coordinator to create an entirely new network and its initially connected child end device nodes to follow it. Consequently, the packets sent from the connected end device nodes intended for the router nodes had no routing path and were therefore discarded. It was found that the XBee 3 modules

can mitigate unauthorised remote AT commands through SRP. Nodes can be configured to enable SRP authentication to establish secure sessions for reading and writing AT commands remotely.

4.7 Conclusion

Chapter 4 presented the findings gathered through the security testing experiments that analysed ZigBee 3.0 against the three identified prevalent security issues. The experiments investigating the security of symmetric keys demonstrated a reduced impact against the ZigBee 3.0 protocol and XBee 3 modules in comparison to the earlier protocol revisions. The experiments investigating the second security issue, compromised symmetric keys, identified the significance of each key type and revealed the more complex device authentication mechanisms of a CSM network. The final set of experiments that exploited ZigBee's lack of DoS protection mechanisms showed that the ZigBee 3.0 protocol mitigates specific known DoS attacks. It was determined that the positioning and type of the DoS attack plays a significant role in the overall impact against the network availability.

The findings are further analysed and discussed in Chapter 5. Each of the sub-questions and the primary research question is answered based on these findings. A discussion of evidence is included for each research question answered.

Chapter 5: Discussion

5.1 Introduction

The findings and results of this research were presented in Chapter 4 through a series of security testing experiments performed against testbed ZigBee 3.0 networks. These experiments were conducted following the research design and methodology approach outlined in Chapter 3.

Chapter 5 provides a further discussion and analysis of these findings and is separated into two parts. Section 5.2 presents the answers to each of the research sub-questions posed in Chapter 3 using the results gathered from the experiments. The answers are followed by a discussion linking them to relevant Chapter 4 findings. Last, the primary research question is answered in Section 5.3 based on the compiled conclusions for the sub-questions. Section 5.3 includes a discussion that analyses the tested security issues and assesses their overall impact against ZigBee 3.0.

5.2 Research Sub-Questions

The five sub-questions established in Chapter 3 are answered in this section. These sub-questions were designed to evaluate the ZigBee 3.0 protocol against the identified symmetric key and DoS security issues prevalent in earlier revisions of ZigBee and assist in answering the primary research question.

5.2.1 Sub-Question 1

SQ1: *What impact do the exploitation of ZigBee and IEEE 802.15.4's known symmetric key vulnerabilities pose against the security of symmetric keys in ZigBee 3.0 networks?*

Answer

The results indicate that exploiting known symmetric key vulnerabilities can cause the symmetric keys to be exposed, but only on unsecured ZigBee 3.0 networks.

Discussion

One of ZigBee's security assumptions was that the security of symmetric keys is based on the assumption that the keys are securely stored, and devices are preconfigured with the keys to prevent unsecured transmission (ZigBee Alliance, 2017, pp. 407–408). It was found that the XBee 3 ZigBee 3.0 modules have default security measures that prevent the tested known symmetric key vulnerabilities from occurring. However, the vulnerabilities can still be present on ZigBee 3.0 networks if the trust centre enables them through its encryption options. Two known symmetric key vulnerabilities were tested against ZigBee 3.0 as follows:

- **Unencrypted Network Key Transport:**

The unencrypted transmission of the network key is a well-known vulnerability that can occur when ZigBee devices are not preloaded with symmetric keys (Zillner, 2015). In the experiment against ZigBee 3.0 described in Section 4.4.1.1, it was found that this vulnerability could only occur in DSM networks. Moreover, the EO bit 1 had to be set in the trust centre and joining nodes' security configuration to allow unencrypted network key transport.

On the CSM network, it was found that the network key was still encrypted with the preconfigured link key and sent OTA to the joining device, despite having the EO bit 1 set. It is assumed that this was the case because nodes must be preconfigured with the trust centre link key on a CSM network before joining the network. Otherwise, the trust centre would have to register nodes individually with an 0x24 registration frame (Digi International, 2018).

The nodes were not required to be preconfigured with a link key on the DSM network. It was found that when a joining node is not configured with the link key, a trust centre will send the network key unencrypted to this node. Therefore, an attacker could externally capture the unencrypted network key if forwarded to a joining device on a DSM network while EO bit 1 is enabled. This vulnerability can be addressed on a DSM network by ensuring all devices

are preconfigured with the link key before joining and optimising the trust centre's security configuration to enable encryption.

- **Default Link Key Values:**

Default link key values can be vulnerable when no link key is specified by the APS when a device joins the network. The global default trust centre link key can be the well-known key defined by the ZigBee Alliance or a preconfigured link key defined by the manufacturers of certain ZigBee devices (Rudresh, 2017c). When testing this vulnerability against XBee 3, as described in Section 4.4.1.2, it was found that default link key values can only be allowed when the trust centre's security configuration is adjusted from its default configuration to enable EO bit 4. It was demonstrated that an attacker would be able to maliciously capture and decrypt the network key with a default link key when a device joins the network. Moreover, an attacker could authenticate an unauthorised device onto the network when the join window opens. This vulnerability can be addressed on ZigBee 3.0 networks by configuring the trust centre and joining nodes with a unique preconfigured link key.

5.2.2 Sub-Question 2

SQ2: *What impact do the exploitation of ZigBee and IEEE 802.15.4's known denial of service vulnerabilities pose against the availability of ZigBee 3.0 networks?*

Answer

The results revealed that ZigBee 3.0 had mitigated a few of the tested DoS attacks; however, it is still susceptible to network realignment, protocol flooding and remote AT command exploitation (blackhole) attacks. The overall impact on the availability of the network of these attacks was data loss.

Discussion

ZigBee is known to be vulnerable to DoS attacks based on the lack of DoS protection mechanisms (Radmand et al., 2010). In the experiments described in Section 4.6, different DoS

attacks were performed against a ZigBee 3.0 CSM network externally and internally. The results measuring the impact on the availability of the network for each attack varied:

- **External DoS on ZigBee 3.0:**

A total of three external DoS experiments were performed against the ZigBee 3.0 network, as described in Section 4.6.1. The first two experiments were flooding attacks that spammed the network with spoofed packets to manipulate core functionalities of the ZigBee protocol. These attacks were overall found to cause only slight disruptions on the network. The third experiment exploited the ZigBee protocol with a single spoofed coordinator realignment frame, causing significant DoS to the victim node and the connected child nodes.

PAN-ID conflict flooding, described in Section 4.6.1.1, targeted the frequency agility mechanism that enables the network manager (Coordinator) to migrate the network to a new PAN-ID when it detects or receives a PAN-ID conflict report (Mukherji & Sadu, 2016). It was found that this attack was successful in triggering multiple PAN-ID changes against the default PAN Conflict Threshold (3) on the coordinator; however, there were only slight disruptions to network availability. The router nodes could continue to send/receive packets without significant delays, and network nodes could keep up with the frequent PAN-ID changes while the attack was in motion. The PAN Conflict Threshold (CR) can be adjusted on the coordinator to increase the number of PAN-ID conflict reports required to trigger a PAN-ID change.

The association flooding attack, described in Section 4.6.1.2, flooded the victim network with spoofed beacon request frames containing the victim network PAN-ID in an attempt to overwhelm gateway nodes from too many connecting stations (River Loop Security, n.d.-b). However, it was found that this attack caused no disruptions on the network, and the gateway nodes did not register the spoofed beacon requests. In addition, the join window was required to be open for gateway nodes to respond to the spoofed packets, but it did not increase the attack's impact while in motion.

A network realignment attack was performed against ZigBee 3.0 using a spoofed coordinator realignment frame, as described in Section 4.6.1.3, which caused significant DoS to a victim router and its connected child nodes. Immediately after sending the spoofed frame against the network, the victim router changed its PAN-ID, causing it to be isolated from its network. Two out of four of its initially connected child end device nodes were also disconnected. The end device nodes could not rejoin the network until the join window was opened, while the victim router could only rejoin after its network settings were reset. The findings of this experiment demonstrated that an attack of this nature could potentially cause a significant impact on the availability of network data. The attack entirely disrupted the victim router's ability to send and receive data and affected its connected child nodes.

Replaying previously captured data (replay attack) is an attack that can be crafted and executed externally to achieve DoS against ZigBee networks. For example, an attacker can capture packets at one point of the network and then retransmit the packets at another point to paralyse the network functionality (Rudresh, 2017c). However, the ZigBee 3.0 protocol implements a reinforced NWK frame counter preventing it from being reset from a standard factory or OTA reset (Texas Instruments, 2019). This additional protection mechanism makes it difficult to initiate replay attacks against ZigBee 3.0, and therefore, this attack to achieve DoS was deemed infeasible and not investigated in this study.

- **Internal DoS on ZigBee 3.0:**

Two internal DoS experiments against the ZigBee 3.0 network, which are discussed in Section 4.6.2, targeted ZigBee's APL and NWK layers (Radmand et al., 2010). Internal DoS attacks were found to inflict a greater level of impact on the network's availability than external DoS attacks. The attacks conducted were protocol flooding and a blackhole attack using remote AT commands.

In the protocol flooding experiment, described in Section 4.6.2.1, a victim node was flooded with legitimate packets containing ZigBee's largest payload size sent from a compromised router node. After initiating the attack, the victim node began to receive the flooded packets at a rate slower than the number of packets sent. Furthermore, the packets sent from the end device nodes intended for the victim node were received at significantly delayed intervals or otherwise dropped. The protocol flooding attack was shown to be effective in unfairly consuming the processing capabilities and networking resources of the victim node.

A blackhole attack, described in Section 4.6.2.2, was performed against ZigBee 3.0 by transmitting a remote AT command instruction from a compromised router node to initiate a 'network reset' (NR) on the victim coordinator node. It was found that when the coordinator received the malicious AT command, it created an entirely new network on its reset networking parameters. Furthermore, the end device nodes that were initially connected to the coordinator also migrated to the new network, causing their packets intended for its old network router nodes to be discarded. This singular example of remote AT command misuse demonstrated a method to achieve significant DoS against a ZigBee 3.0 network and the necessity to enforce internal security to prevent unauthorised use. On legacy XBee equipment with earlier revisions of ZigBee, remote AT commands do not have robust internal security mechanisms to prevent unauthorised use. Internal security is generally upheld with the symmetric keys, allowing only devices that are part of the network to send remote AT commands to other devices (Vaccari et al., 2017). However, SRP authentication, introduced on XBee 3 and supported by the ZigBee 3.0 protocol, can be configured on individual nodes to protect them against unauthorised remote AT commands. Alternatively, OTA updates, including remote AT commands, can be entirely disabled on nodes to prevent unauthorised use.

5.2.3 Sub-Question 3

SQ3: *What impact do compromised ZigBee and IEEE 802.15.4 symmetric keys pose against the confidentiality of ZigBee 3.0 networks?*

Answer

The results revealed that compromised ZigBee and IEEE 802.15.4 symmetric keys would significantly affect the confidentiality of a ZigBee 3.0 network and its data. Moreover, the overall level of impact on the network's confidentiality is determined by the type of symmetric key exposed.

Discussion

As discussed in the literature review in Section 2.5, ZigBee networks encrypt communications with the AES-128 bit encryption suite using two primary symmetric keys, namely, the 'Link Key' and the 'Network Key'. Furthermore, ZigBee 3.0's supported security models, CSM and DSM, differ in device authentication and message protection mechanisms (X. Fan et al., 2017). Section 4.5 presented experiments conducted based on the assumption that an attacker has acquired the symmetric keys. The experiments were performed against the ZigBee 3.0 security models configured with an appropriate security configuration that employed encryption and preconfigured link keys. The findings revealed that the overall impact on network confidentiality varied by the type of key exposed.

- **Compromised Link Key:**

The APS layer applies the link key to secure unicast communications on ZigBee networks, and this key is shared between only the trust centre and router/end device (Zillner, 2015). The eavesdropping experiments described in Section 4.5.1 revealed that if an attacker were to obtain a ZigBee 3.0 network's link key, then the security of the other symmetric keys would be at risk in addition to compromised unicast communications.

A compromised link key was first evaluated against a CSM network. It was demonstrated how a device receives two symmetric keys from the trust centre upon joining, that is, the network key and the updated trust centre link key. Both keys are encrypted with the initial (compromised) link key. The device first received a copy of the network key and then the updated link key to encrypt/decrypt all ongoing APS secured frames. In contrast, the DSM network only shared a single symmetric key (network key) to the joining device. The network key and all APS secured communications are encrypted with the initial (compromised) link key. From an attacker's perspective, it is shown that a compromised link key could enable them to capture and decrypt the symmetric keys of both ZigBee 3.0 security models. Therefore, an entire ZigBee 3.0 network could become compromised from an exposed link key.

- **Compromised Network Key**:

ZigBee secures transmitted frames and broadcast communications with a network key shared between all network devices and applied to the NWK and APL layers of the protocol stack (Rudresh, 2017b). As shown in the eavesdropping experiments discussed in Section 4.5.1, an exposed network key affects network data confidentiality and the security of future key rotations.

On ZigBee 3.0, it was found that a compromised network key can be used to decrypt all NWK layer communications, including broadcasts or any non-secured APS frames, while the key is valid. On CSM networks, the trust centre can regularly issue network key rotations to invalidate the old network key. The key rotation works by broadcasting the updated network key to nodes at defined intervals (in days). However, it was found that the updated network key was encrypted with the old (compromised) network key and could, therefore, be decrypted by an eavesdropping attacker. In contrast, the network key is always valid on DSM networks as these do not support key rotations.

- **Additional Attacks with Compromised Symmetric Keys:**

Radmand et al. (2010) claimed that impersonation and spoofing attacks are possible if an attacker compromises a ZigBee network's symmetric keys. In Section 4.5.2, it was investigated whether ZigBee 3.0 has this issue by launching an impersonation attack against the different security models, which found DSM networks more susceptible to the performed attack.

In this attack, a malicious device was configured to impersonate a legitimate coordinator node using the stolen symmetric keys and the network information of a victim network. While the impersonated coordinator was deployed with its join window open, a spoofed coordinator realignment frame was transmitted to a victim node in an attempt to cause it to realign and join the malicious network. This attack was unsuccessful on CSM networks, for it is determined that the victim node is already registered to a centralised trust centre and could, therefore, not join a new trust centre. However, the victim node was caused to join the impersonating coordinator's network when tested against a DSM network.

The techniques demonstrated in Section 4.5.2 are a first-hand example of how compromised symmetric keys can be used to initiate spoofing and impersonation attacks against ZigBee 3.0 networks. An additional impersonation attack could be performed by deploying a spoofed coordinator node with stolen symmetric keys close to a ZigBee application and waiting idly with the join window opened for legitimate nodes to join. This attack, in theory, would work against both ZigBee 3.0 security models as the nodes are likely not to be pre-registered to a trust centre.

5.2.4 Sub-Question 4

SQ4: *What methods can be applied to strengthen the security of symmetric keys on ZigBee 3.0 networks?*

Answer

The security of symmetric keys can significantly be strengthened on ZigBee 3.0 networks by registering devices in out-of-band methods, using securely generated keys and following best security practices to ensure the keys are safely exchanged and cannot be exposed. The primary methods incorporate:

- individually registering nodes to the trust centre with link keys derived from the joining device's install code (high-security applications);
- configuring nodes with a global preconfigured link key in out-of-band methods for device authentication (moderate security applications);
- employing the Centralized Security Model for Key Management.

Discussion

As discussed in the literature review in Section 2.6, the ZigBee 3.0 protocol contains additional mechanisms and improvements to its security features, ultimately providing additional layers of security to its symmetric keys. These additional mechanisms include both optional and mandatory security services, which can be applied depending on the security requirements and the ZigBee application in use. In addition, several security practices should be employed to ensure that the link key's safekeeping, loading and commissioning are secure, given that it was identified that this key could be used to expose the other symmetric keys. Moreover, establishing a high level of security in device registration with the link key is vital. The methods identified to strengthen the security of symmetric keys on ZigBee 3.0 networks are as follows:

- **Link Keys Derived from Install Code:**

In ZigBee 3.0, every device supporting the protocol contains a unique install code factory-programmed into the node (Digi International, 2018), which can be used to create the link key to authenticate nodes into a network and securely receive the network key (NXP

Semiconductors, 2017). This method is suitable for ZigBee applications requiring the highest security level. It ensures each node has a unique and random link key and is identified to the trust centre (Digi International, 2018).

Furthermore, it significantly reduces the chances of the symmetric keys being exposed. As demonstrated in Section 4.4.2, nodes can be registered to the trust centre(s) using an 0x24 registration frame containing the install code of a joining device. The trust centre uses the install code, inserted through out-of-band methods, with a hash function to create a random link key. Subsequently, the trust centre and node use this link key to join the network and securely exchange the network key (NXP Semiconductors, 2017). In a DSM network, devices can be registered with an install code. However, the 0x24 registration frame must be issued from the router adjacent to the joining device since registration information is not shared between nodes (Digi International, 2018). Registering nodes with install codes significantly increases the security of symmetric keys but requires devices to be individually and manually registered to the trust centre. Therefore, this authentication method may not be ideal on larger-scale networks that have high scalability requirements.

- **Preconfigured Link Keys:**

Authenticating nodes with a preconfigured global link key can be an effective solution to incorporate a moderate level of security into the symmetric keys in larger-scale networks and allow easy network deployment. This solution can be accomplished by configuring the joining nodes with a link key using out-of-band methods to match the key established on the trust centre, as shown in the CSM security configuration in Figure 4.16. When a device attempts to join the network with an association broadcast request, the preconfigured link key will be used to authenticate and receive the network key sent from the trust centre.

Using preconfigured global link keys eliminates the need for networks to authenticate with the well-known default link key, which is susceptible to exposure (NXP Semiconductors,

2017). However, since the key is of the global type, every node contains a copy of the key, making it less secure than link keys derived from install codes.

- **Centralised Security Model for Key Management:**

In ZigBee 3.0, CSM networks have distinct advantages in terms of key management and the security of symmetric keys, compared with DSM networks. The literature review in Section 2.6 discussed how CSM networks have an added security mechanism that mandates every device to be updated with a trust centre link key upon joining for all ongoing APS layer encryptions (Silicon Labs, n.d.). This security mechanism prevents the network key from being compromised when a device leaves and rejoins the network and provides additional security to APS layer communications. Furthermore, ZigBee 3.0 coordinators can reject legacy devices that do not initiate the trust centre link key update (Texas Instruments, 2019). Another advantage is the CSM trust centre's ability to initiate network key rotations at regular intervals.

CSM trust centres have more robust, secure device registration mechanisms. Device registration is only authorised through the trust centre and is transient. Therefore, registered devices are only authorised to join for a specific time interval separate from the join window (defined by the KT parameter on XBee 3). If a device fails to join within this time, it will need to be re-registered to the trust centre (Digi International, 2018). In addition, the key entry tables to authorise devices are stored in the trust centre's RAM and do not persevere across power cycles (Digi International, 2018).

Earlier revisions of the ZigBee protocol required networks to contain only a single trust centre, similar to ZigBee 3.0 CSM networks. A single trust centre has distinct advantages. It is the only node responsible for establishing and managing symmetric key distribution and for allowing other nodes to join the network based on its join policy. Therefore, the trust centre has an overview of all network nodes, symmetric keys and security policies. Accordingly, this thesis recommends that in ZigBee applications with high-security demands, the CSM model

should be implemented into the network's design and architecture. The additional mechanisms and advantages strengthen the security of its symmetric keys and provide an additional level of control over security policies and management.

5.2.5 Sub-Question 5

SQ5: *What are the security limitations regarding the security of symmetric keys for 'Distributed Security Model' networks compared with 'Centralised Security Model' networks in ZigBee 3.0?*

Answer

The security limitations of a DSM network result from the simpler mechanisms implemented for device authentication that favour simplicity over security. The primary security limitations to symmetric keys on a DSM network compared with a CSM network include the following:

- The network key is fixed and cannot be changed/rotated once the network is formed.
- Any router node can authorise and authenticate joining nodes and distribute the network key.
- Devices are not updated with an additional symmetric key for APS layer encryptions.
- Network joining options are limited. Individual registration with an 0x24 frame can only be performed on router nodes adjacent to the joining device.

Discussion

As discussed in the literature review in Section 2.5.2.2, DSM networks were introduced into ZigBee 3.0 and are a security model that enables networks to be formed without a coordinator or single trust centre (X. Fan et al., 2017). DSM networks employ simplified security mechanisms, allowing their networks to be easily deployable and scalable instead of the more complex and secure CSM model. The security limitations in DSM networks that reduce the security of its symmetric keys were identified as follows:

- **Network Keys in DSM Networks:**

The security of network keys is limited in DSM networks, and the keys are more susceptible to exposure. A single router node is responsible for forming the network and setting the key for all NWK layer encryptions between every device in a DSM network. Once the network key is established on the forming router, it cannot be changed or rotated unless the entire network is reset (Digi International, 2018). This feature poses a security concern for NWK layer communications are indefinitely compromised if an attacker obtains the network key.

- **Device Authentication in DSM Networks:**

The device authentication mechanisms are less secure and complex than those in CSM networks. In DSM networks, device registration is persistent because every registered device is authorised to join the network provided the join window is open. Every router node can authenticate and add nodes to the network and distribute the shared network key. Moreover, joining nodes preconfigured with the global link key will receive the network key from their adjacent router (Digi International, 2018). As demonstrated on XBee 3 in Section 4.4.1.1, on a DSM network, the network key transmitted OTA unencrypted to a joining device can be vulnerable when the device is not configured with a link key and EO bit 0 is set in the security configuration.

Unlike CSM routers, distributed routers act as trust centres, and each contains a key table stored in flash memory that remains persistent across power cycles (Digi International, 2018). Distributed trust centres can individually register nodes with an 0x24 registration frame. However, since registration information is not shared between nodes, registration must occur on the router adjacent to the joining node. Another distinct limitation to device authentication compared with CSM networks is the reduced level of security on APS/unicast communications after a device joins the network. Nodes in DSM networks are not updated with a trust centre

link key to encrypt all ongoing APS secured frames. Instead, nodes will continue to secure APS frames with the preconfigured link key.

5.3 Primary Research Question

The primary research question was the overall focus of this study and was created to analyse the ZigBee 3.0 protocol against the identified security issues. The answer and discussion to the following RQ1 are based on the findings gathered for the sub-questions:

RQ1: *What impact do symmetric key and denial of service security issues that are prevalent against earlier revisions of ZigBee pose against ZigBee 3.0 networks*?

Answer

Prevalent symmetric key and DoS security issues in earlier revisions of ZigBee affect the confidentiality and availability of ZigBee 3.0 networks. However, ZigBee 3.0's improved mechanisms and added security features have reduced the overall impact of these security issues compared with the earlier protocol revisions.

Discussion

Based on the Chapter 4 findings and answers gathered for each sub-questions, the overall impact of each identified prevalent security issue could be determined as follows:

- **Security Issue 1—Security of Symmetric Keys:**

The security of symmetric keys was a security issue identified to be prevalent in the earlier revisions of ZigBee, with existing literature widely covering this topic. This security issue concerns how an attack could obtain the symmetric keys and whether the implemented security mechanisms are sufficient to protect the keys. The ZigBee Alliance (2017, pp. 407-408) stated that the security of symmetric keys depends on their safekeeping, the protection mechanisms employed and the proper implementation of cryptographic mechanisms and associated security policies involved. The ZigBee 3.0 protocol stays true to this assumption.

Moreover, its advancements have enabled the protocol to maintain an overall increased level of security for the symmetric keys over previous versions of ZigBee.

In Section 5.2.1, SQ2 was answered, which evaluated the impact of well-known symmetric key vulnerabilities that could compromise the keys if exploited. The known vulnerabilities that were tested included unencrypted network key transport and default link key values. It was concluded that the XBee 3 modules with the ZigBee 3.0 protocol address these vulnerabilities through their default security configuration (with security enabled). In addition, XBee 3 employs the CSM model for key management through its default configuration, which has ultimately proven to uphold a moderate to high level of security for its symmetric keys.

Section 5.2.4 answered SQ4, which identified the methods that could be applied to strengthen the security of symmetric keys on ZigBee 3.0 networks. Three methods that are part of the ZigBee 3.0 protocol were identified and are dependent on the network's scalability requirements. The first method is registering nodes to the trust centre using install codes, a new mechanism introduced in ZigBee 3.0 for device authentication. This method was concluded to provide the highest level of security because each node is guaranteed a random link key, and it enables the network key to be securely passed to a joining node with a low risk of exposure. The second method is preconfiguring nodes with a global link key in out-of-band methods as an alternative to registration with install codes for networks with higher scalability requirements. The third method, employing the CSM model for key management, has distinct advantages, more complex security mechanisms and additional security features over the alternative DSM model, as identified by exploring SQ5 in Section 5.2.5.

The findings to SQ2 and SQ4 indicate that the ZigBee 3.0 protocol upholds security over its symmetric keys to a greater extent than the previous versions of ZigBee. The protocol

has addressed the known symmetric key vulnerabilities discussed in SQ1 and has additional security features and mechanisms as outlined through answering SQ4.

- **Security Issue 2—Compromised Symmetric Keys:**

Compromised symmetric keys is a security issue identified in previous revisions of ZigBee. This issue concerns the impact against a ZigBee network's confidentiality due to one or more of its symmetric keys being compromised by an attacker and the attacks that could be inflicted with them.

In Section 5.2.3, SQ3 was answered, which evaluated the different impacts that each type of symmetric key poses against the network's confidentiality. It was concluded that the link key is the most crucial for it can be used in eavesdropping attacks to expose the other symmetric keys and compromise unicast/APS secured communications. Second, a compromised network key can be used to decrypt all NWK layer communications, including broadcasts, non-APS secured packets and future network key rotations. Another result revealed that impersonation and spoofing attacks are possible if an attacker obtains both symmetric keys. An impersonation attack was performed, which confirmed that DSM networks are susceptible to joining an impersonating coordinator's network with stolen keys. Furthermore, an impersonation attack was suggested in Section 5.2.3, where a malicious coordinator configured with stolen keys is deployed in proximity to a ZigBee application while idly waiting for nodes to join. This attack, in theory, would work against both security models since it is likely that the victim nodes have not been registered to a trust centre.

The results for SQ3 determined that compromised keys significantly affect the confidentiality of the network and its data, but the impact is dependent on the type of key exposed.

- **Security Issue 3—Insufficient Denial of Service Protection Mechanisms:**

Insufficient DoS protection mechanisms were a security issue identified to be prevalent in earlier revisions of ZigBee, making the protocol susceptible to several DoS attacks. This issue was investigated by performing a series of DoS attacks externally and internally against ZigBee 3.0.

Section 5.2.2 provided the answer to SQ2, obtained through analysing the impact of each DoS attack performed against ZigBee 3.0. It was concluded that the protocol mitigated a few of the tested attacks but remains susceptible to specific attacks. Moreover, the positioning of the attack (external/internal) was the most prominent contributing factor to the overall level of impact against the network's availability.

Three external DoS attacks were tested against a ZigBee 3.0 (CSM) network. These attacks attempted to exploit the functionality of the ZigBee protocol through flooding and spoofing techniques. PAN-ID flooding and association flooding were among the external flooding attacks, and neither caused any significant DoS against the network. While the PAN-ID flooding attack was in motion, network nodes could keep up with the frequent PAN-ID changes, causing minor processing delays on router nodes. Furthermore, association flooding was recorded to cause no impact, and gateway nodes would only respond to the flooded broadcast request packets when the join window opened. The final external DoS attack used a spoofed coordinator realignment frame to isolate a victim router node from its network. This attack was confirmed to disconnect the victim node and two out of four of its connected end device nodes from the network, causing data loss.

Two internal DoS attacks that targeted ZigBee's APL and NWK layers were tested. The first attack, protocol flooding, was confirmed to unfairly consume a victim router node's processing capabilities and network resources. While the attack was in motion, the majority of packets sent from the end device nodes to the victim router node were received at significantly

delayed intervals or otherwise dropped, causing data loss. The second internal attack achieved significant data loss by exploiting the functionality of remote AT commands. The attack caused the victim coordinator node to create and migrate to a separate network and cause the packets sent from end device nodes to be discarded. It was identified that remote AT commands could be internally secured through SRP authentication supported by the ZigBee 3.0 protocol or by disabling the function entirely to prevent misuse.

The findings for SQ2 confirm that the ZigBee 3.0 protocol is vulnerable to network realignment attacks (external), protocol flooding (internal) and remote AT command misuse (internal) if not secured. The primary impact of these attacks was data loss.

5.4 Conclusion

Chapter 5 discussed and analysed the findings presented in Chapter 4. This chapter answered each of the sub-questions and the primary research question that were established as the focus of this study. The prevalent security issues from earlier revisions regarding symmetric key and DoS issues were analysed against the ZigBee 3.0 protocol. Overall, it was determined that these issues affect ZigBee 3.0 to a lesser extent than they did the previous versions because of its improved security mechanisms and added security features. This chapter discussed the impact of each of the performed attacks, addressed solutions where necessary to mitigate these attacks and suggested methods that can be incorporated to increase the security of symmetric keys in ZigBee 3.0 networks.

Chapter 6 concludes this thesis with a summary of the research, a discussion of the research limitations and recommendations for future studies as a continuation of this research.

Chapter 6: Conclusion

6.1 Introduction

Chapter 1 introduced and outlined the background and motivation behind the research topic. It provided an overview of the aims and objectives, and last, it outlined the structure of this thesis. In Chapter 2, a literature review was presented that built a body of knowledge on the ZigBee protocol and its security concepts. The review outlined each of the main security components implemented into the protocol and analysed its security issues that have remained prevalent across the revisions of ZigBee over the years.

The security issues found from surveying existing literature and related studies were used to formulate a research question and five supporting sub-questions to investigate the ZigBee 3.0 protocol. Chapter 3 devised an appropriate research methodology and design for testing ZigBee 3.0 against the identified prevalent security issues. The research phases were outlined, which entailed a physical approach to investigating the protocol and gathering and analysing the necessary data to answer the proposed research questions.

Chapter 4 presented the findings and results gathered through the security testing experiments. These findings outlined the impact inflicted against the testbed ZigBee 3.0 networks resulting from the attacks associated with each security issue. The findings and results were further discussed and analysed in Chapter 5. Each sub-question was answered based on the findings gathered in Chapter 4, and where necessary, the claims were supported by relevant literature from Chapter 2. Last, the five sub-questions were compiled to answer the primary research question.

This chapter presents a conclusion to this thesis over three parts. First, Section 6.1 summarises the research. Section 6.2 discusses the limitations identified in the adopted research approach. Last, Section 6.3 discusses research that could be conducted as a continuation of this research.

6.2 Summary of Research

This research was conducted to analyse the ZigBee 3.0 protocol and networks against the prevalent security issues found in the earlier revisions of the protocol. Three security issues were identified and investigated: 'Security of Symmetric Keys', 'Compromised Symmetric Keys' and 'Insufficient DoS Protection Mechanisms'. These issues were further analysed through reviewing existing literature and related studies to determine their associated attacks that can be performed against ZigBee networks.

The study used a practical security testing approach to investigate the ZigBee 3.0 protocol against the prevalent security issues related to symmetric keys and DoS. As part of this approach, ZigBee 3.0 networks were constructed in the laboratory using XBee 3 equipment to create an environment in which realistic attacks scenarios could be performed against the networks, and their impact verified. Furthermore, the networks were flexible and adjustable to suit each experiment, as regards the total number of nodes and the security configuration. The study utilised the KillerBee framework and ApiMote hardware along with a CC2531 USB dongle to perform external network attacks and a compromised XBee 3 module to perform internal network attacks.

The first prevalent security issue that underwent investigation concerned how an attacker could obtain ZigBee's symmetric keys by exploiting known vulnerabilities and whether the implemented security mechanisms are sufficient to protect the keys. Overall, it can be claimed that the ZigBee 3.0 protocol upholds a greater level of security for its symmetric keys than its previous versions based on the following findings:

- The well-known symmetric key vulnerabilities that were tested included the unencrypted network key transport and default link key values. These vulnerabilities were found to have been addressed on XBee 3 by default through the device's security configuration (with security enabled). They could exist only when deliberately enabled

through the device's encryptions options. In addition, the unencrypted network key vulnerability was found to only be possible on DSM networks when the encryption options were unsecured and no link key was set.

- Specific security features and mechanisms are included in the ZigBee 3.0 protocol that the previous versions do not offer. The security feature in ZigBee 3.0 allows devices to be registered individually to the network by a trust centre using a link key derived from the joining device's install code. This feature ensures that the generated link key is random and prevents the link key from being exposed and used to compromise the network key. Moreover, the protocol's updated security mechanism mandates that devices joining a CSM network are updated with an APS trust centre link key to be used for all ongoing APS layer encryptions.

The second prevalent security issue concerned the breach against a ZigBee network's confidentiality if one or more of its symmetric keys were exposed by an attacker. The study found that the type of key exposed was the most prominent factor determining the overall impact. A compromised link key was demonstrated to allow an attacker to acquire the other symmetric keys and compromise APS communications. In contrast, the network key could be used to expose broadcast and NWK layer secured communications and future network key rotations. Moreover, the compromised keys could be used to initiate impersonation attacks. This study demonstrated an attack that successfully hijacked a router node from a DSM network using stolen symmetric keys and network information configured onto an attacker node impersonating as a legitimate coordinator.

The last security issue was based on the ZigBee protocol's lack of DoS protection mechanisms. Ultimately, the study revealed that ZigBee 3.0 remains vulnerable to specific internal and external DoS attacks that were tested, resulting in data loss and disturbance to the network availability. The external flooding attacks, including PAN-ID and association

flooding, were shown to cause only minimal to unnoticeable disruptions on gateway nodes. However, the results for network realignment (external), protocol flooding (internal) and remote AT command misuse/exploitation (internal) demonstrated that these significantly affect the ZigBee 3.0 network, primarily resulting in data loss.

This study has presented ZigBee 3.0's stance against the security issues prevalent in the earlier revisions of ZigBee as a practical undertaking. It demonstrated attacks that can be performed against ZigBee 3.0 networks using easy-to-acquire hardware and software tools, along with a first-hand impersonation attack. Last, it recommended incorporating methods that are part of the protocol to strengthen the security of symmetric keys in ZigBee 3.0 networks.

6.3 Limitations of Research

Several possible limitations can be identified in the research design components through the course of this research. The experiments of this research were designed to be replicable in a laboratory using similar hardware and software. However, notably, changes to specific experimental design decisions could have influenced the collected findings and conclusions. The potential limitations that apply to this research are discussed as follows:

- **Researcher's Bias:**

The researcher was solely responsible for deciding each research design component and its adopted security testing approach to evaluate the ZigBee 3.0 protocol. Consequently, the researcher's bias could be factored in. Despite the processes followed to penetrate the protocol and design elements outlined, an argument may be presented that the testing could have been conducted differently or more effectively.

- **Scope:**

The scope for the security testing experiments outlined in Chapter 3 (see Section 3.4.1) defined the extent of testing to be done on ZigBee 3.0. However, the scope had to comply with the capabilities of the utilised hardware and software. As a result, specific attacks associated

with each security issue under assessment were excluded from this research. Moreover, the scope restricted the security testing experiments to only include attacks that target known vulnerabilities in the main security components of ZigBee. Therefore, unknown vulnerabilities for each security issue were not addressed in this research.

- **Selection of ZigBee 3.0 Hardware:**

The ZigBee 3.0 protocol is implemented into a wide range of products from different manufacturers; however, this research was limited to Digi International's XBee 3 modules. Although the security specification is consistent between ZigBee 3.0 enabled devices, the power and processing capabilities can differ. This does not affect the eavesdropping attacks performed in this research but does present the possibility of the device enduring different levels of DoS from specific attacks. As a result, the DoS attacks performed in Chapter 4 (see Section 4.6) could affect other manufacturers' devices differently.

- **Number of Nodes:**

The ZigBee protocol supports up to 65,000 nodes per network (Digi International, n.d.-c), and specific applications are built on large-scale networks. This factor could influence the DoS experiments performed in Chapter 4 (see Section 4.6) for the attacks could be investigated only against a small-scaled network consisting of eight nodes. Therefore, it is possible that if specific DoS attacks were performed against a larger-scaled network, the impact against the network's availability would be different.

- **Exploitation Hardware and Software:**

The external network attacks conducted in this research were limited to the CC231 USD dongle (packet sniffer) and two ApiMote transceivers with the KillerBee framework. Consequently, specific attacks could not be performed, including DDoS and jamming/interference attacks. In addition, most of the external attacks were limited to the attack scripts found in the KillerBee framework. Although this is the most widely recognised and

utilised framework for penetrating ZigBee (River Loop Security, 2019), there are other testing frameworks, including Z3sec and ZigDiggity, that potentially have additional testing capabilities.

6.4 Future Research

A few research areas have been identified that can be conducted as a continuation of this research. While this study has outlined ZigBee 3.0's stance against the prevalent security issues related to symmetric keys and DoS found in the earlier revisions, additional studies could be performed to expand the overall knowledge on ZigBee 3.0's security posture. The relevant future research areas are as follows:

- **Additional Testing:**

As mentioned in the limitations, certain attacks were excluded from the study primarily owing to hardware limitations. It would be relevant to explore additional attacks associated with the security issues to evaluate the ZigBee 3.0 protocol further. For example, DDoS attacks could be performed to exploit ZigBee's 'Insufficient DoS Protection Mechanisms'. In particular, volumetric and protocol-based flooding attacks, including the PAN-ID flooding and association flooding attacks that were performed in Chapter 4 (see Section 4.6.1). These and similar attacks could have a more significant impact if distributed from multiple external sources.

- **Integration of Legacy Equipment on ZigBee 3.0 Networks:**

The ZigBee 3.0 protocol is designed to allow for interoperability between ZigBee 3.0 devices and legacy equipment, predominantly in ZigBee Light Link and Home Automation devices (Silicon Labs, 2021). This design may expose ZigBee 3.0 networks to specific weaknesses that threaten the security of its symmetric keys. For example, the demonstrated well-known symmetric key vulnerabilities were shown to be addressed on the ZigBee 3.0 devices; however, adding legacy equipment onto a network may re-introduce these

vulnerabilities. Another weakness is against the updated APS link key mandated in ZigBee 3.0 CSM networks. ZigBee 3.0 devices must request and receive the updated link key, but legacy devices may not initiate the key update procedure (Moorthy, 2019). As a result, the integration of legacy equipment raises security concerns that require consideration before implementing them into a ZigBee 3.0 network. Research could further investigate the impact that introducing legacy ZigBee devices would pose against the security of symmetric keys in ZigBee 3.0 networks.

- **Future Revisions of ZigBee:**

The study of prevalent security issues found in the earlier revisions of ZigBee can continue beyond the ZigBee 3.0 protocol. Future protocol releases are likely to inherit specific issues related to symmetric keys or DoS, which would necessitate investigation.

References

Adams, J. T. (2006). An introduction to IEEE STD 802.15. 4. In *2006 IEEE Aerospace Conference* (pp. 8-pp). Big Sky, Montana: IEEE.

Ahmed, M. R., Huang, X., Sharma, D., & Cui, H. (2012). Wireless Sensor Network: Characteristics and Architectures. *International Journal of Information and Communication Engineering, 6*(12), 1398-1401. https://doi.org/10.5281/zenodo.1072589

Aju, O. G. (2015). A Survey of ZigBee Wireless Sensor Network Technology: Topology, Applications and Challenges. *International Journal of Computer Applications, 130*(9), 47-55. https://doi.org/10.5120/ijca2015907130

Azzi, C. (2016). *Vulnerability analysis and security framework for ZigBee communication in IoT* (Unpublished master's thesis). University of Nevada, Las Vegas.

Barbareschi, M., Battista, E., Mazzeo, A., & Venkatesan, S. (2014). Advancing WSN Physical Security Adopting TPM-based Architectures. In *Proceedings of the 2014 IEEE 15th International Conference on Information Reuse and Integration (IEEE IRI 2014)* (pp. 394-399). Redwood City, CA: IEEE. https://doi.org/10.1109/IRI.2014.7051916

Carlos-Mancilla, M., López-Mellado, E., & Siller, M. (2016). Wireless sensor Networks Formation: Approaches and techniques. *Journal of Sensors, 2016*, 1-18. https://doi.org/10.1155/2016/2081902

Carlsen, J. (2021). *Outfitting your smart home: ZigBee devices*. Retrieved from https://www.safewise.com/ZigBee-devices/?fbclid=IwAR17ofXdziPBIcGprErk_QltDfyYe2dTIW2CNlR-0EMJItK32liMFhTrOrg

Chaitanya, D. K., & Arindam, G. (2011). Analysis of Denial-of-Service attacks on Wireless Sensor Networks Using Simulation. In *IT Security for the Next Generation-European Cup 2011*. Erfurt, Germany: Kaspersky.

Creswell, J. W., & Creswell, J.D. (2018). *Research design: Qualitative, quantitative, and mixed methods approaches* (5th ed.). Los Angeles, CA: SAGE Publications.

Digi International. (2017, November). *ZigBee coordinator operation*. Retrieved from https://www.digi.com/resources/documentation/Digidocs/90002002/Content/Concepts/c_zb_coord_op.htm#:~:text=The%20coordinator%20is%20responsible%20for,network%20must%20have%20one%20coordinator

Digi International. (2018, November). *ZigBee 3.0 security*. Retrieved from https://www.digi.com/support/knowledge-base/ZigBee-3-0-security

Digi International. (2020). Digi XBee® 3 802.15.4 *Radio frequency (RF) module User Guide*. Retrieved from https://www.digi.com/resources/documentation/digidocs/PDFs/90002273.pdf

Digi International. (n.d.-a). *Digi XBee 3 ZigBee 3 RF module*. Retrieved from https://www.digi.com/products/embedded-systems/digi-xbee/rf-modules/2-4-ghz-rf-modules/xbee3-ZigBee-3#specifications

Digi International. (n.d.-b). *Map your Digi XBee IoT network with Digi XCTU*. Retrieved from https://www.digi.com/resources/examples-guides/map-your-digi-xbee-iot-network-with-digi-xctu

Digi International. (n.d.-c). *ZigBee wireless mesh networking*. Retrieved from https://www.digi.com/solutions/by-technology/ZigBee-wireless-standard

Dini, G., & Tiloca, M. (2010). Considerations on Security in ZigBee Networks. In *2010 IEEE International Conference on Sensor Networks, ubiquitous, and trustworthy computing* (pp. 58-65). Newport Beach, California: IEEE. https://doi.org/10.1109/SUTC.2010.15

Elahi, A., & Gschwender, A. (2009). *ZigBee wireless sensor and control network* (1st ed.) Boston, MA: Pearson Education.

Engmann, F., Katsriku, F. A., Abdulai, J., Adu-Manu, K. S., & Banaseka, F. K. (2018). Prolonging the lifetime of wireless sensor networks: A review of current techniques. *Wireless Communications and Mobile Computing, 2018,* 1-23. https://doi.org/10.1155/2018/8035065

Fan, B. (2017). Analysis on the Security Architecture of ZigBee Based on IEEE 802.15.4. In *IEEE 13th International Symposium on Autonomous Decentralized System (ISADS)* (pp. 241-246). Bangkok, Thailand: IEEE. https://doi.org/10.1109/ISADS.2017.23

Fan, X., Susan, F., Long, W., & Li, S. (2017). Security analysis of ZigBee. *MWR InfoSecurity, 2017,* 1–18.

Farahani, S. (2008a). The ZigBee development environment. In D. Gislason (Ed.), *ZigBee wireless networking* (pp. 77–78). Burlington, MA: Elsevier. https://doi.org/10.1016/B978-0-7506-8597-9.00003-3

Farahani, S. (2008b). ZigBee and IEEE 802.15.4 protocol layers. In S. Farahani (Ed.), *ZigBee wireless networks and transceivers* (pp. 126–129). Burlington, MA: Elsevier. https://doi.org/10.1016/B978-0-7506-8393-7.00003-0

Gascón, D. (2009). *Security in 802.15.4 and ZigBee networks*. Retrieved from https://www.libelium.com/libeliumworld/security-802-15-4-ZigBee/

Gislason, D. (2008). *ZigBee wireless networking*. Burlington, MA: Newnes. https://doi.org/10.1109/AERO.2006.1655947

Kandris, D., Nakas, C., Vomvas, D., & Koulouras, G. (2020). Applications of wireless sensor networks: An up-to-date survey. *Applied System Innovation, 3*(1), 1-24. https://doi.org/10.3390/asi3010014

Khanji, S., Iqbal, F., & Hung, P. (2019). ZigBee security vulnerabilities: Exploration and Evaluating. In *2019 10th International Conference on Information and Communication Systems (ICICS)* (pp. 52-57). Irbid, Jordan: IEEE. https://doi.org/10.1109/IACS.2019.8809115

Koubâa, A., Alves, M., & Tovar, E. (2007). Time sensitive IEEE 802.15.4 protocol. In N.P. Mahalik (Author), *Sensor Networks and Configurations: Fundamentals, standards, platforms, and applications* (pp. 19–49). Berlin, Germany: Springer. https://doi.org/10.1007/3-540-37366-7_2

Kumar, T., & Mane, P. B. (2016). ZigBee topology: A survey. In *2016 International Conference on Control, instrumentation, communication and Computational Technologies (ICCICCT)* (pp. 164-166). Kumaracoil, India: IEEE. https://doi.org/10.1109/ICCICCT.2016.7987937

Lea, P. (2018). *Internet of Things for architects: Architecting IoT solutions by implementing sensors, communication infrastructure, edge computing, analytics, and security*. Birmingham, England: Packt Publishing.

Libelium. (n.d.-a). *Interacting with Waspmote*. Retrieved from https://development.libelium.com/waspmote-technical-guide/interacting-with-waspmote

Libelium. (n.d.-b). *IoT Products Waspmote*. Retrieved from https://www.libelium.com/iot-products/waspmote/

Lu, G., Krishnamachari, B., & Raghavendra, C. S. (2004). Performance evaluation of the IEEE 802.15.4 MAC for low-rate low-power wireless networks. In *2004 IEEE International Conference on Performance, Computing, and Communications* (pp. 701-706). Phoenix, AZ: IEEE. https://doi.org/10.1109/PCCC.2004.1395158

Masica, K. (2007). *Recommended Practices Guide For Securing ZigBee Wireless Networks in Process Control System Environments* (pp. 1-22, Report No. UCRL-TR-xxyyzz). Berkeley, CA: Lawrence Livermore National Laboratory.

Matin, M. A., & Islam, M. M. (2012). Overview of Wireless Sensor Network. In *Wireless Sensor Networks-technology and Protocols* (pp. 1-3). London, England: InTechOpen. https://doi.org/10.5772/49376

Moorthy, K. (2019). The 'key' to security: ZigBee 3.0's security features. Retrieved from https://e2e.ti.com/blogs_/b/process/posts/the-key-to-security-ZigBee-3-0-s-security-features

Mordor Intelligence. (2020). ZigBee market. Retrieved from https://www.mordorintelligence.com/industry-reports/ZigBee-market

Morgan, D. L. (2014). Research Design and Research Methods. In *Integrating qualitative and quantitative methods: A pragmatic approach.* (pp. 45-62). Thousand Oaks, CA: SAGE Publications. https://doi.org/10.4135/9781544304533.n3

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* (pp. 230-240). Boston, MA: ACM. https://doi.org/10.1145/3098243.3098254

Mukherji, A., & Sadu, S. (2016). ZigBee performance analysis. In *2016 International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET)* (pp. 325-329). Chennai, India: IEEE. https://doi.org/10.1109/WiSPNET.2016.7566148

NXP Semiconductors. (2017). *Maximizing Security in ZigBee networks* [White Paper]. https://www.nxp.com/docs/en/supporting-information/MAXSECZBNETART.pdf

Ocenasek, P. (2009). Towards security issues in ZigBee architecture. In *Symposium on Human Interface* (pp. 587-593). Berlin, Heidelberg: Springer. https://doi.org/10.1007/978-3-642-02556-3_66

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* (pp. 199-206). Kuwait City, Kuwait: IEEE. https://doi.org/10.1109/HIS.2014.7086198

Radmand, P., Domingo, M., Singh, J., Arnedo, J., Talevski, A., Petersen, S., & Carlsen, S. (2010). ZigBee/ZigBee PRO Security Assessment Based on Compromised Cryptographic Keys. In *2010 International Conference on P2P, Parallel, Grid, Cloud and Internet Computing P2P, Parallel, Grid, Cloud and Internet Computing (3PGCIC)* (pp. 465-470). Fukuoka, Japan: IEEE. https://doi.org/10.1109/3PGCIC.2010.79

Ramya, C. M., Shanmugaraj, M., & Prabakaran, R. (2011). Study on ZigBee technology. In *2011 3rd International Conference on Electronics Computer Technology Electronics Computer Technology (ICECT)* (pp. 297-301). Kanyakumari, India: IEEE. https://doi.org/10.1109/ICECTECH.2011.5942102

Reddy, J. (2005, December). *ZigBee security layer technical overview*. [PowerPoint presentation]. ZigBee Alliance presentation. http://www.zigbee.org/en/events/documents/December2005_Open_House_Presentations/ZigBee_Security_Layer_Technical_Overview.pdf

River Loop Security. (2019, 25 June). River Loop Security presents interactive workshop at energy industry security event [Web log post]. Retrieved from

https://www.riverloopsecurity.com/blog/2019/06/credc-summer-symposium-ZigBee-19/

River Loop Security. (n.d.-a). *ApiMote IEEE 802.15.4/ZigBee sniffing hardware*. Retrieved from https://www.riverloopsecurity.com/projects/apimote/

River Loop Security. (n.d.-b). *GitHub – riverloopsec/killerbee: IEEE 802.15.4/ZigBee Security Research Toolkit*. Retrieved from https://github.com/riverloopsec/killerbee

Rudresh, V. (2017a). *ZigBee security: Basics (part 1)*. Retrieved from https://research.kudelskisecurity.com/2017/11/01/ZigBee-security-basics-part-1/

Rudresh, V. (2017b). *ZigBee security: Basics (part 2)*. Retrieved from https://research.kudelskisecurity.com/2017/11/08/ZigBee-security-basics-part-2/

Rudresh, V. (2017c). *ZigBee security: Basics (part 3)*. Retrieved from https://research.kudelskisecurity.com/2017/11/21/ZigBee-security-basics-part-3/

Sajjad, S. M., & Yousaf, M. (2014). Security analysis of IEEE 802.15.4 MAC in the context of Internet of Things (IoT). In *2014 Conference on Information Assurance and Cyber Security (CIACS)* (pp. 9-14). Rawalpindi, Pakistan: IEEE. https://doi.org/10.1109/CIACS.2014.6861324

Salman, N., Rasool, I., & Kemp, A. H. (2010). Overview of the IEEE 802.15.4 standards family for Low Rate Wireless Personal Area Networks. In *2010 7th International Symposium on Wireless Communication Systems* (pp. 701-705). York, UK: IEEE. https://doi.org/10.1109/ISWCS.2010.5624516

Sarijari, M. A., Abdullah, M. S., Lo, A., & Rashid, R. A. (2014). Experimental studies of the ZigBee frequency agility mechanism in home area networks. In *39th Annual IEEE Conference on Local Computer Networks Workshops* (pp. 711-717). Edmonton, Canada: IEEE. https://doi.org/10.1109/LCNW.2014.6927725

Scarfone, K., Souppaya, M., Cody, A., & Orebaugh, A. (2008). Technical Guide to Information Security Testing and Assessment. *National Institute of Standards & Technology Special Publication, 800*(115), 1-80. Gaithersburg, MD: National Institute of Standards & Technology.

Shanmugapriya, T., Kousalya, K., Rajeshkumar, J., & Nandhini, M. (2019). Wireless Sensor Networks Security Issues, Attacks and Challenges: A Survey. In *International conference on Computer Networks, Big data and IoT* (pp. 1-12). Madurai, Tamil Nadu: Springer, Cham. https://doi.org/10.1007/978-3-030-43192-1_1

Silicon Labs. (2021). *ZigBee 3.0 device interoperability with legacy ZigBee devices*. Retrieved from https://community.silabs.com/s/article/ZigBee-3-0-device-interoperability-with-legacy-ZigBee-devices?language=en_US&fbclid=IwAR0nWy8nMz1BaOa3cZRFVnwtNhRkGBrdUgs4v3ramv3wQsUdWottC_LlhNk

Silicon Labs. (n.d.). *AN1233: Zigbee Security* [Application Notes]. https://www.silabs.com/documents/public/application-notes/an1233-zigbee-security.pdf

Stelte, B., & Rodosek, G. D. (2013). Thwarting attacks on ZigBee - Removal of the KillerBee stinger. In *Proceedings of the 9th International Conference on Network and Service Management (CNSM 2013)* (pp. 219-226). Zurich, Switzerland: IEEE. https://doi.org/10.1109/CNSM.2013.6727840

Texas Instruments. (2019). *What's new in ZigBee 3.0* [White paper]. https://www.ti.com/lit/an/swra615a/swra615a.pdf?ts=1643525133543&ref_url=https%253A%252F%252Fwww.google.com%252F

Tomar, A. (2011). Introduction to ZigBee technology. *Global Technology Centre, 1*, 1–24.

Vaccari, I., Cambiaso, E., & Aiello, M. (2017). Remotely exploiting at command attacks on ZigBee networks. *Security and Communication Networks, 2017,* 1-9. https://doi.org/10.1155/2017/1723658

Vasseur, J.-P., & Dunkels, A. (2010). *Interconnecting smart objects with IP: The next internet*. Burlington, MA: Morgan Kaufmann.

Vidgren, N., Haataja, K., Patino-Andres, J. L., Ramirez-Sanchis, J. J., & Toivanen, P. (2013). Security Threats in ZigBee-Enabled Systems: Vulnerability Evaluation, Practical Experiments, Countermeasures, and Lessons Learned. In *2013 46th Hawaii International Conference on System Sciences* (pp. 5132-5138). Wailea, HI: IEEE. https://doi.org/10.1109/HICSS.2013.475

Wagh, S. S., More, A., & Kharote, P. R. (2015). Performance Evaluation of IEEE 802.15.4 Protocol under Coexistence of WiFi 802.11b. In *Procedia Computer Science, 3rd International Conference on Recent Trends in Computing, ICRTC 2015* (pp. 745-751). Delhi, India: Elsevier. https://doi.org/10.1016/j.procs.2015.07.467

Wheeler, A. (2007, April 16). Commercial Applications of Wireless Sensor Networks Using ZigBee. *IEEE Communications Magazine*, *45*(4), 70-77. https://doi.org/10.1109/MCOM.2007.343615

Wright, J. (2009). *KillerBee: Practical ZigBee Exploitation Framework*. [PowerPoint presentation]. 11th ToorCon conference, San Diego, CA. https://infocon.org/cons/QuahogCon/2010/Presentations/wright-killerbee.pdf

Yang, B. (2009). Study on Security of Wireless Sensor Network Based on ZigBee Standard. In *2009 International Conference on Computational Intelligence and Security* (pp. 426-430). Beijing, China: IEEE. https://doi.org/10.1109/CIS.2009.208

ZigBee Alliance. (2017). *ZigBee Specification* (Document No. 05-3474-22). Retrieved from https://csa-iot.org/wp-content/uploads/2022/01/docs-05-3474-22-0csg-zigbee-specification-1.pdf

ZigBee Alliance. (n.d.-a). *Utility – Connectivity Standards Alliance*. Retrieved from https://ZigBeealliance.org/market-uses/utility/

ZigBee Alliance. (n.d.-b). *ZigBee – Connectivity Standards Alliance*. Retrieved from https://ZigBeealliance.org/solution/ZigBee/

Zillner, T. (2015). ZigBee Exploited The good, the bad and the ugly (S. Schumacher & R. Pfeiffer, Eds.). In *In Depth Security – Proceedings of the DeepSec Conferences* (pp. 699-704). Magdeburg, Germany: Magdeburger Institut für Sicherheitsforschung.

Zillner, T., & Strobl, S. (2015). *ZigBee Exploited – The good, the bad and the ugly*. [PowerPoint presentation]. blackhat USA 2015. https://www.blackhat.com/docs/us-15/materials/us-15-Zillner-ZigBee-Exploited-The-Good-The-Bad-And-The-Ugly.pdf

Appendices

Appendix A: XBee 3 Base Configuration

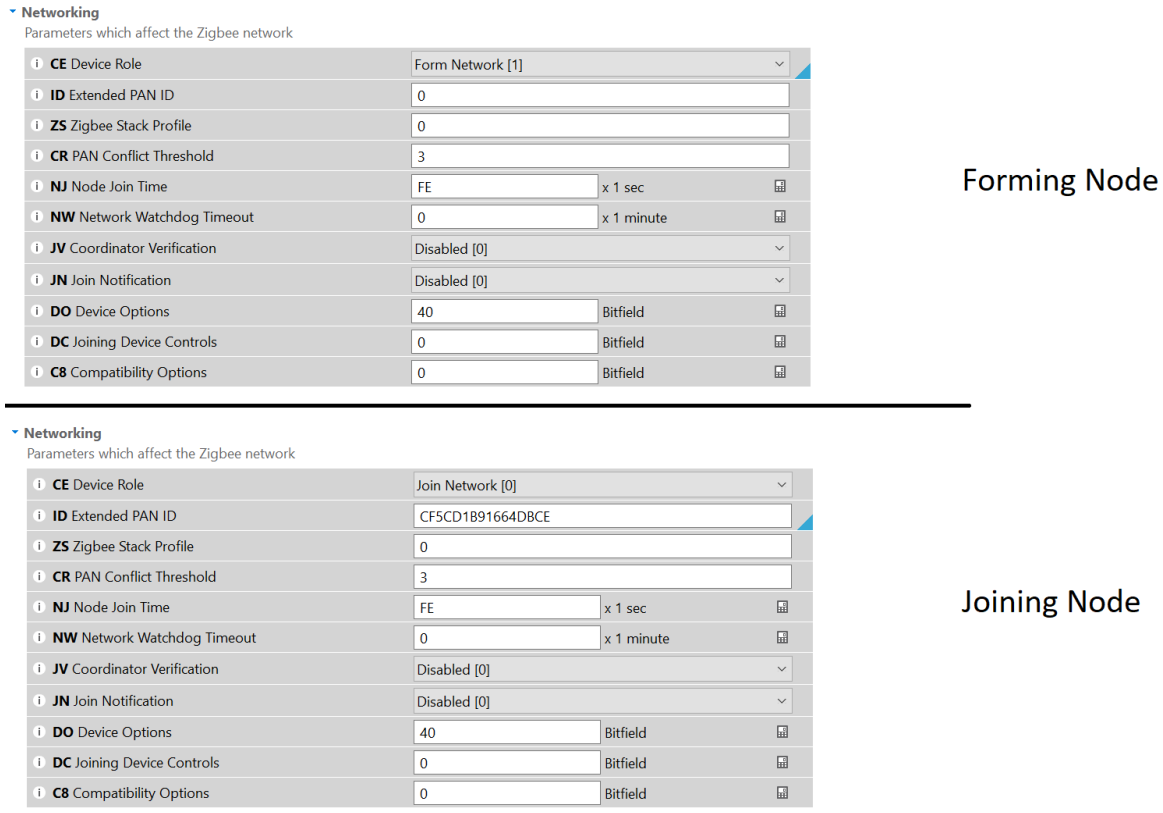

Figure A.1. XBee 3 Networking base configuration.

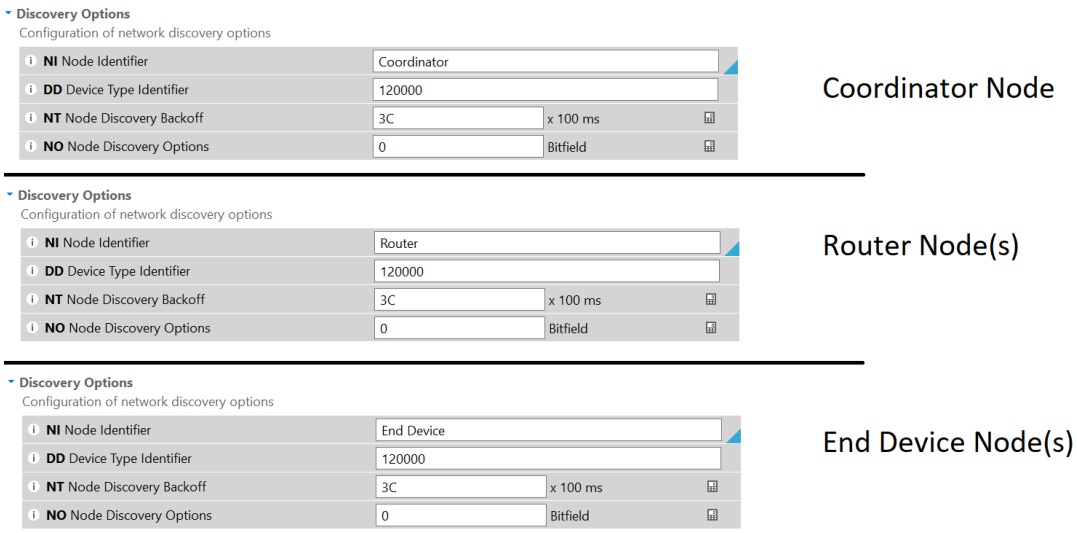

Figure A.2. XBee 3 Discovery options base configuration.

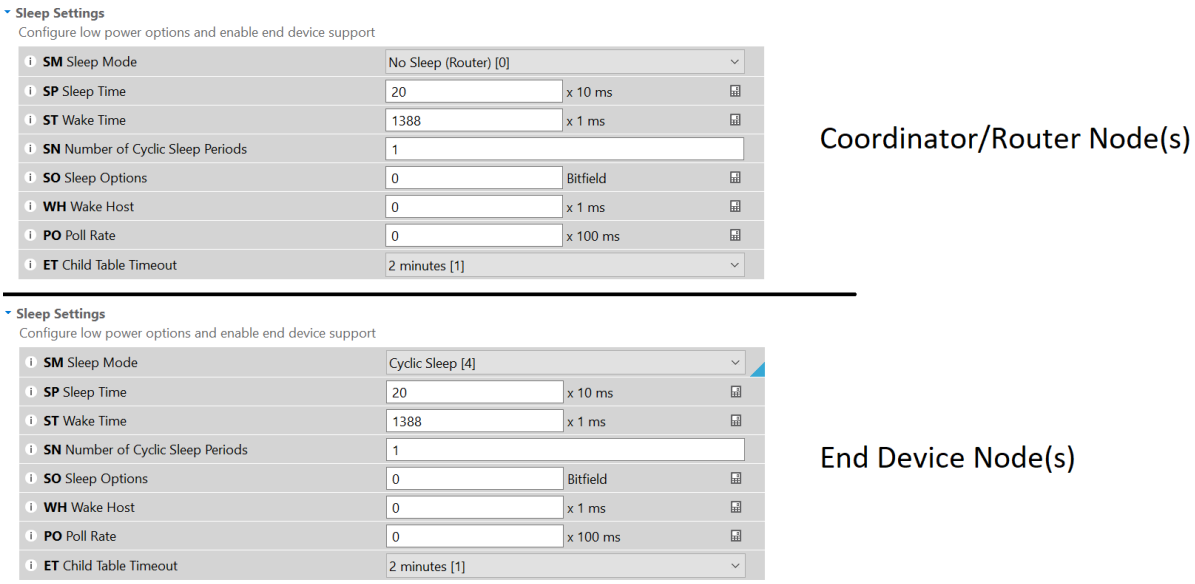

Figure A.3. XBee 3 sleep settings base configuration.

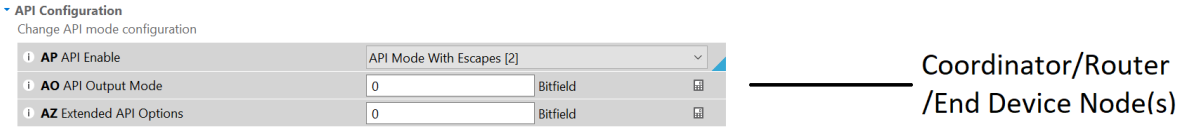

Figure A.4. XBee 3 API configuration base configuration.

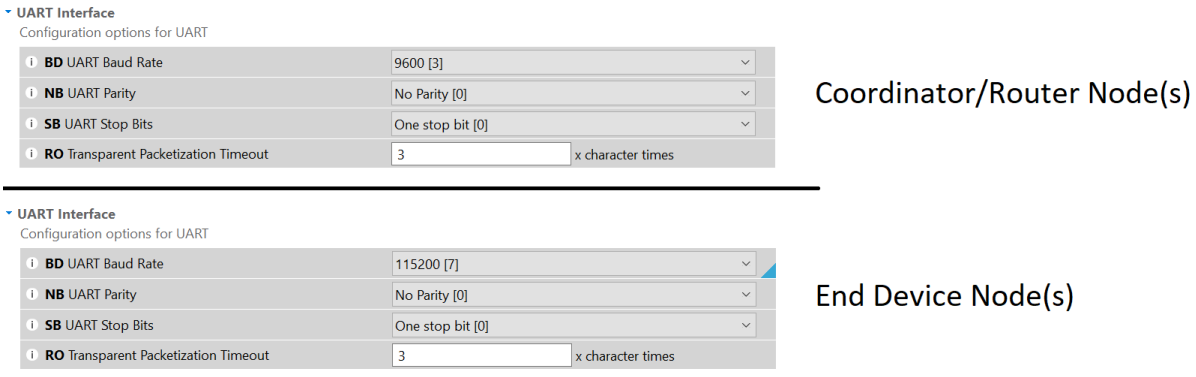

Figure A.5. XBee 3 UART interface base configuration.

Appendix B: Individual Security Test Processes

Table B.1

Test 01 Processes

T01: External Information Gathering	
Nodes	Test Processes
1x Coordinator 2x Router 1x End Device	1. Execute zbstumbler tool (Obtain Operating Channel, PAN-IDs and Source MAC addresses) 2. Perform Network Sniffing using Wireshark (Obtain MAC addresses, and PAN-IDs) 3. Execute zbstumbler tool (Monitor Join Window)

Table B.2

Test 02 Processes

T02: Physical/Internal Information Gathering	
Nodes	Test Processes
1x Coordinator 2x Router 1x End Device	1. Read AT Parameters (Compromised End Device) 2. Read AT Parameters (Compromised Router) 3. Remotely Connect to Coordinator Node (Compromised Router)

Table B.3

Test 03 Processes

T03: Capturing Unencrypted Network Key on CSM Network			
Nodes	Network Information		Test Processes
1x Coordinator 1x Router 1x Router (Joining)	***Operating Channel:***	25	1. Start Wireshark Capture 2. Open Join Window 3. Capture Network Key 4. End Wireshark Capture 5. Save Wireshark Capture
	PAN-ID:	0xFC17	
	Extended PAN-ID:	05:2D:ED:79:4F:84:28:BC	

Table B.4

Test 04 Processes

T04: Capturing Unencrypted Network Key on DSM Network			
Nodes	Network Information		Test Processes
2x Router 1x Router (Joining)	***Operating Channel:***	14	1. Start Wireshark Capture 2. Open Join Window 3. Capture Network Key 4. End Wireshark Capture 5. Save Wireshark Capture
	PAN-ID:	0x85A4	
	Extended PAN-ID:	8C:FC:CC:C8:9E:90:BC:69	

Table B.5

Test 05 Processes

T05: Capturing and Decrypting Network Key (Default Link Key)			
Nodes		Network Information	Test Processes
1x Coordinator 1x Router 1x Router (Joining)	***Operating Channel:***	14	1. Start Wireshark Capture 2. Open Join Window 3. Capture Network Key 4. End Wireshark Capture 5. Save Wireshark Capture
	PAN-ID:	0x85A4	
	Extended PAN-ID:	8C:FC:CC:C8:9E:90:BC:69	
	Link Key:	5A6967426565416C6C 69616E63653039	

Table B.6

Test 06 Processes

T06: Unauthorised Network Joining (Default Link Key)			
Nodes		Network Information	Test Processes
1x Coordinator 1x Router 1x End Device 1x Router (Unauthorised)	***Operating Channel:***	16	1. Configure the unauthorised device with captured network information and default link key 2. Monitor Join Window
	PAN-ID:	0x3D49	
	Extended PAN-ID:	DC:60:C1:54:CD:27:3F:FB	
	Link Key:	5A6967426565416C6C 69616E63653039	

Table B.7

Test 07 Processes

T07: Securely Registering Device to Trust Centre	
Nodes	Test Processes
1x Coordinator 1x Router (Joining)	1. Configure trust centre (Coordinator) and Joining Node for secure device registration. 2. Issue 0x24 frame from trust centre (Coordinator)

Table B.8

Test 08 Processes

T08: Capturing Symmetric Keys on a CSM Network			
Nodes	**Network Information**		**Test Processes**
1x Coordinator 1x Router 1x Router (Joining)	***Operating Channel:***	14	1. Start Wireshark Capture 2. Open Join Window 3. Capture Symmetric Keys 4. End Wireshark Capture 5. Save Wireshark Capture
	PAN-ID:	0xB139	
	Extended PAN-ID:	98:FC:6B:56:15:D8:D7:9A	
	Link Key:	bbf5820052d0a57173cc5cfd6237e3d1	

Table B.9

Test 09 Processes

T09: Capturing Network Key Rotation on CSM Network			
Nodes	Network Information		Test Processes
1x Coordinator 2x router	***Operating Channel:***	14	1. Start Wireshark Capture 2. Capture Network Key Rotation 3. End Wireshark Capture 4. Save Wireshark Capture
	PAN-ID:	0xB139	
	Extended PAN-ID:	98:FC:6B:56:15:D8:D7:9A	
	Link Key:	bbf5820052d0a57173cc5cfd6237e3d1	
	Network Key:	575d8968af279d9665028dfc717895d3	
	Trust Centre Link Key:	c4dda2b23ef3dd06ad6259a6bba49a92	

Table B.10

Test 10 Processes

T10: Capturing Symmetric Keys on a DSM Network			
Nodes	Network Information		Test Processes
2x Router 1x Router (Joining)	***Operating Channel:***	12	1. Start Wireshark Capture 2. Open Join Window 3. Capture Symmetric Keys 4. End Wireshark Capture 5. Save Wireshark Capture
	PAN-ID:	0xCAB0	
	Extended PAN-ID:	72:98:AF:48:D2:71:9E:D7	
	Link Key:	bbf5820052d0a57173cc5cfd6237e3d1	

Table B.11

Test 11 Processes

T11: Decrypting NWK Layer/Broadcast Communications on ZigBee 3.0 (CSM) Network			
Nodes	**Network Information**		**Test Processes**
1x Coordinator 2x router	***Operating Channel:***	11	1. Start Wireshark Capture 2. Capture NWK Layer/Broadcast Communications 3. Send Transmit Request frame from Coordinator to Router_01 4. End Wireshark Capture 5. Save Wireshark Capture
	PAN-ID:	0xEB0A	
	Extended PAN-ID:	5E:7B:03:F8:3E:25:0F:09	
	Link Key:	bbf5820052d0a57173cc5cfd6237e3d1	
	Network Key:	a2a70bfc9631223d81c55fbc5d6acd4f	
	Trust Centre Link Key	99f17ff50f37506744cf87b5c0ee442d	

Table B.12

Test 12 Processes

T12: Decrypting APS Layer (Unicast) Communications on a ZigBee 3.0 (CSM) Network			
Nodes	Network Information		Test Processes
1x Coordinator 2x router	***Operating Channel:***	11	1. Start Wireshark Capture 2. Send Transmit Request frame from Coordinator to Router_01 (With APS Encryption) 3. End Wireshark Capture 4. Save Wireshark Capture
	PAN-ID:	0xEB0A	
	Extended PAN-ID:	5E:7B:03:F8:3E:25:0F:09	
	Link Key:	bbf5820052d0a57173cc5cfd6237e3d1	
	Network Key:	a2a70bfc9631223d81c55fbc5d6acd4f	
	Trust Centre Link Key	99f17ff50f37506744cf87b5c0ee442d	

Table B.13

Test 13 Processes

T13: Node Impersonation Attack on CSM Network			
Nodes	Network Information		Test Processes
1x Coordinator 2x Router 1x Coordinator (Attacker)	***Operating Channel:***	13	1. Configure Coordinator-Attacker node with obtained victim network information. 2. Open Join Window (Coordinator-Attacker) 3. Execute PAN-ID realignment attack from Kali Linux 4. Test attack results in XCTU network scan from Coordinator (Attacker)
	PAN-ID:	0x7A27	
	Extended PAN-ID:	5E:7B:03:F8:3E:25:0F:09	
	Link Key:	bbf5820052d0a57173cc5cfd6237e3d1	
	Network Key:	5292a3edf70e3e42f35be2eedec0b074	
	Trust Centre Link Key	8b074b2a4f7475dc658f5e9d251e6074	

Table B.14

Test 14 Processes

T14: Node Impersonation Attack on DSM Network		
Nodes	**Network Information**	**Test Processes**
3x router	***Operating Channel:*** 13 ***PAN-ID:*** 0x27E9 ***Extended PAN-ID:*** A82CEAEE6189AEEB ***Link Key:*** bbf5820052d0a57173cc5cfd6237e3d1 ***Network Key:*** 0aa412f225e15e9be8eb5a9a6f74bccf	1. Configure Coordinator-Attacker node with obtained victim network information. 2. Open Join Window (Coordinator-Attacker) 3. Execute PAN-ID realignment attack from Kali Linux 4. Test attack results in XCTU network scan from Coordinator (Attacker)

Table B.15

Test 15 Processes

T15: Received Packets on Router Nodes from End Device Nodes (PAN-ID Flood)					
Network Information		Experiment Times		Measurement Criteria	Test Processes
Operating Channel:	16	***Passive Phase:***	5 Minutes	• Measure Number of Received Packets on Router_01 • Measure Number of Received Packets on Router_02 • Measure Number of PAN-ID Changes	1. Start Passive Phase (00:00) 2. Execute Attack (0:05:00) 3. End Attack (0:15:00) 4. Save Router_01 console 5. Save Router_02 console 6. Save Coordinator console
PAN-ID:	0x654A	***Attack Start Time:***	0:05:00		
Extended PAN-ID:	35:DF:33:9E:09:A0:F6:B2	***Attack Stop Time:***	0:15:00		
		Total Time:	15 Minutes		

Table B.16

Test 16 Processes

T16: Received Packets on Router Node (Router to Router PAN-ID Flood)					
Network Information		Experiment Times		Measurement Criteria	Test Processes
Operating Channel:	23	***Passive Phase***	5 Minutes	• Measure Number of Received Packets on Router_02 sent from Router_01 • Measure Number of PAN-ID Changes	1. Start Passive Phase (00:00) 2. Execute Attack (0:05:00) 3. End Attack (0:15:00) 4. Save Router_02 console 5. Save Coordinator console
PAN-ID:	0x31B3	***Attack Start Time:***	0:05:00		
Extended PAN-ID:	58:8E:F4:B6:00:53:C5:A3	***Attack Stop Time:***	0:15:00		
		Total Time:	15 Minutes		

Table B.17

Test 17 Processes

T17: Network Authentication and XCTU Functionality (PAN-ID Flooding)					
Network Information		Experiment Times		Testing Criteria	Test Processes
Operating Channel:	24	***Passive Phase***	5 Minutes	• Test End Device Network Rejoin (End-Device_01) • Test End Device Network Join (End-Device_02) • Test XCTU Network Scan from Gateway Nodes	1. Start Passive Phase (0:00:00) 2. Execute Attack (0:05:00) 3. Open Join Window 4. Power End-Device_01 5. Power End-Device_02 6. Perform XCTU Network scan from gateway nodes. 7. End Attack
PAN-ID:	0xE741	***Attack Start Time:***	0:05:00		
Extended PAN-ID:	EF:65:6F:2D:D8:00:D8:E5	***Attack Stop Time:***	0:21:00		
		Total Time:	21 Minutes		

Table B.18

Test 18 Processes

T18: Extended Attack (PAN-ID Flooding)					
Network Information		Experiment Times		Testing Criteria	Test Processes
Operating Channel:	11	***Passive Phase***	5 Minutes	• Test Network Crash/ Realignment • Test Changes to Routing Structure	1. Perform Pre-Attack XCTU Scan 2. Start Passive Phase (0:00:00) 3. Execute Attack (0:05:00) 4. End Attack (12.05:00) 5. Perform Post-Attack XCTU Network Scan 6. Save Coordinator console
PAN-ID:	0xC964	***Attack Start Time:***	0:05:00		
Extended PAN-ID:	62:AA:4A:76:1A:F2:64:4F	***Attack Stop Time:***	12:05:00		
		Total Time:	12 hours and 5 Minutes		

Table B.19

Test 19 Processes

T19: Mitigation Test (PAN-ID Flooding)					
Network Information		Experiment Times		Testing Criteria	Test Processes
Operating Channel:	12	***Passive Phase***	5 Minutes	• Adjust PAN Conflict Threshold (CR) • Measure Number of PAN-ID Changes	1. Start Passive Phase (0:00:00) 2. Execute Attack (0:05:00) 3. End Attack (0:15:00) 4. Save Coordinator console
PAN-ID:	0x2EE3	***Attack Start Time:***	0:05:00		
Extended PAN-ID:	36:C8:EE:22:8E:8E:7A:81	***Attack Stop Time:***	0:15:00		
		Total Time:	15 Minutes		

Table B.20

Test 20 Processes

T20: Received Packets on Router Nodes from End Device Nodes (Association Flooding)							
Network Information		Experiment Times		Join Window		Measurement Criteria	Test Processes
Operating Channel:	17	***Passive Phase:***	5 Minutes	***Open:***	0:06:01.8	• Measure Number of Received Packets on Router_01 • Measure Number of Received Packets on Router_02	1. Start Passive Phase (0:00:00) 2. Execute Attack (0:05:00) 3. Open Join Window 4. End Attack (0:15:00) 5. Save Router_01 console 6. Save Router_02 console 7. Save Coordinator console
PAN-ID:	0x3FB1	***Attack Start Time:***	0:05:00	***Close:***	0.10:15.8		
Extended PAN-ID:	48:8C:8E:F7:05:DC:65:E3	***Attack Stop Time:***	0:15:00	-	-		
		Total Time:	15 Minutes	***Total Join Window:***	4 Minutes and 14 Seconds		

Table B.21

Test 21 Processes

T21: Network Authentication and XCTU Functionality (Association Flooding)							
Network Information		Experiment Times		Join Window		Testing Criteria	Test Processes
Operating Channel:	12	***Passive Phase:***	5 Minutes	***Open:***	0:11:01.9	• Test End Device Network Rejoin (End-Device_01) • Test Network Join (End-Device_02) • Test XCTU Network Scan from Gateway Nodes	1. Start Passive Phase (0:00:00) 2. Execute Attack (0:05:00) 3. Open Join Window 4. Power End-Device_01 5. Power End-Device_02 6. Perform XCTU Network scan from gateway nodes. 7. End Attack
PAN-ID:	0x4C26	***Attack Start Time:***	0:05:00	***Close:***	0:15:15.7		
Extended PAN-ID:	5A:C0:9E:7C:6A:14:E5:AF	***Attack Stop Time:***	0:17:00	-	-		
		Total Time:	17 Minutes	***Total Join Window:***	4 Minutes and 14 Seconds		

Table B.22

Test 22 Processes

T22: Network Realignment Attack			
Network	Network Information		Test Processes
1x Coordinator 2x Router 3x End Device	***Operating Channel:***	24	1. Execute ‘zbrealign’ tool 2. Perform XCTU Network Scan from Gateway Nodes
	PAN-ID:	0x7385	
	Extended PAN-ID:	04:D9:82:E3:98:DA:03:4E	

Table B.23

Test 23 Processes

T23: Received Packets on Router_01 (Protocol Flooding)					
Network Information		Experiment Times		Measurement Criteria	Test Processes
Operating Channel:	21	***Passive Phase:***	5 Minutes	• Measure Number of Received Packets on Router_01	1. Start Passive Phase (0:00:00) 2. Execute Attack (0:05:00) 3. End Attack (0:15:00) 4. Save Router_01 console
PAN-ID:	0xCFBD	***Attack Start Time:***	0:05:00		
Extended PAN-ID:	78:88:9A:AD:CF:77:B1:A2	***Attack Stop Time:***	0:15:00		
		Total Time:	15 Minutes		

Table B.24

Test 24 Processes

T24: Extended Protocol Attack					
Network Information		Experiment Times		Testing Criteria	Test Processes
Operating Channel:	17	***Passive Phase***	5 Minutes	• Test Network Crash/ Realignment • Test Changes to Routing Structure	1. Perform Pre-Attack XCTU Scan 2. Execute Attack (0:05:00) 3. End Attack (12:05:00) 4. Perform Post-Attack XCTU Network Scan
PAN-ID:	0xF56F	***Attack Start Time:***	0:05:00		
Extended PAN-ID:	ED:BE:F8:6F:E6:3F:1E:55	***Attack Stop Time:***	12:05:00		
		Total Time:	12 hours and 5 Minutes		

Table B.25

Test 25 Processes

T25: Blackhole Attack Using AT Commands			
Network	**Network Information**		**Test Processes**
1x Coordinator 2x Router 3x End Device	***Operating Channel:***	17	1. Send Remote AT Command 2. Initiate network scan (XCTU) 3. Configure SRP Authentication (Mitigation)
	PAN-ID:	0x3B50	
	Extended PAN-ID:	16:09:12:6D:5B:0B:71:F8	

Appendix C: End Device Code

Table C.1

End Device Sender Code

End Device Code (End-Device_01)

```
/*
 *  ------ [ZB_03] - send packets to a gateway --------
 *
 *  Version:          0.2
 *  Design:           David Gascón
 *  Implementation:   Yuri Carmona
 */

#include <WaspXBeeZB.h>
#include <WaspFrame.h>

// Destination MAC address
//////////////////////////////////////////
char RX_ADDRESS[] = "0013A20041D1CC94";
//////////////////////////////////////////

// Define the Waspmote ID
char WASPMOTE_ID[] = "End-Device_01";


// define variable
uint8_t error;



void setup()
{
  // init USB port
  USB.ON();
  USB.println(F("Sending packets example"));

  // store Waspmote identifier in EEPROM memory
  frame.setID( WASPMOTE_ID );

  // init XBee
  xbeeZB.ON();

  delay(3000);

  ///////////////////////////
  // 2. check XBee's network parameters
  ///////////////////////////
  checkNetworkParams();

}
```

```
void loop()
{
  //////////////////////////////////////////
  // 1. Create ASCII frame
  //////////////////////////////////////////

  // create new frame
  frame.createFrame(ASCII);

  // add frame fields
  frame.addSensor(SENSOR_STR, "new_sensor_frame");
  frame.addSensor(SENSOR_BAT, PWR.getBatteryLevel());


  //////////////////////////////////////////
  // 2. Send packet
  //////////////////////////////////////////

  // send XBee packet
  error = xbeeZB.send( RX_ADDRESS, frame.buffer, frame.length );

  // check TX flag
  if( error == 0 )
  {
    USB.println(F("send ok"));

    // blink green LED
    Utils.blinkGreenLED();

  }
  else
  {
    USB.println(F("send error"));

    // blink red LED
    Utils.blinkRedLED();
  }

  // wait for three seconds
  delay(3000);
}
```

```
/*****************************************
 *
 *  checkNetworkParams - Check operating
 *  network parameters in the XBee module
 *
 *****************************************/
void checkNetworkParams()
{
  // 1. get operating 64-b PAN ID
  xbeeZB.getOperating64PAN();

  // 2. wait for association indication
  xbeeZB.getAssociationIndication();

  while( xbeeZB.associationIndication != 0 )
  {
    delay(2000);

    // get operating 64-b PAN ID
    xbeeZB.getOperating64PAN();

    USB.print(F("operating 64-b PAN ID: "));
    USB.printHex(xbeeZB.operating64PAN[0]);
    USB.printHex(xbeeZB.operating64PAN[1]);
    USB.printHex(xbeeZB.operating64PAN[2]);
    USB.printHex(xbeeZB.operating64PAN[3]);
    USB.printHex(xbeeZB.operating64PAN[4]);
    USB.printHex(xbeeZB.operating64PAN[5]);
    USB.printHex(xbeeZB.operating64PAN[6]);
    USB.printHex(xbeeZB.operating64PAN[7]);
    USB.println();

    xbeeZB.getAssociationIndication();
  }

  USB.println(F("\nJoined a network!"));

  // 3. get network parameters
  xbeeZB.getOperating16PAN();
  xbeeZB.getOperating64PAN();
  xbeeZB.getChannel();

  USB.print(F("operating 16-b PAN ID: "));
  USB.printHex(xbeeZB.operating16PAN[0]);
  USB.printHex(xbeeZB.operating16PAN[1]);
  USB.println();

  USB.print(F("operating 64-b PAN ID: "));
  USB.printHex(xbeeZB.operating64PAN[0]);
  USB.printHex(xbeeZB.operating64PAN[1]);
  USB.printHex(xbeeZB.operating64PAN[2]);
  USB.printHex(xbeeZB.operating64PAN[3]);
  USB.printHex(xbeeZB.operating64PAN[4]);
  USB.printHex(xbeeZB.operating64PAN[5]);
  USB.printHex(xbeeZB.operating64PAN[6]);
  USB.printHex(xbeeZB.operating64PAN[7]);
  USB.println();

  USB.print(F("channel: "));
  USB.printHex(xbeeZB.channel);
  USB.println();

}
```

Appendix D: XCTU Packet Creation for External DoS Experiments

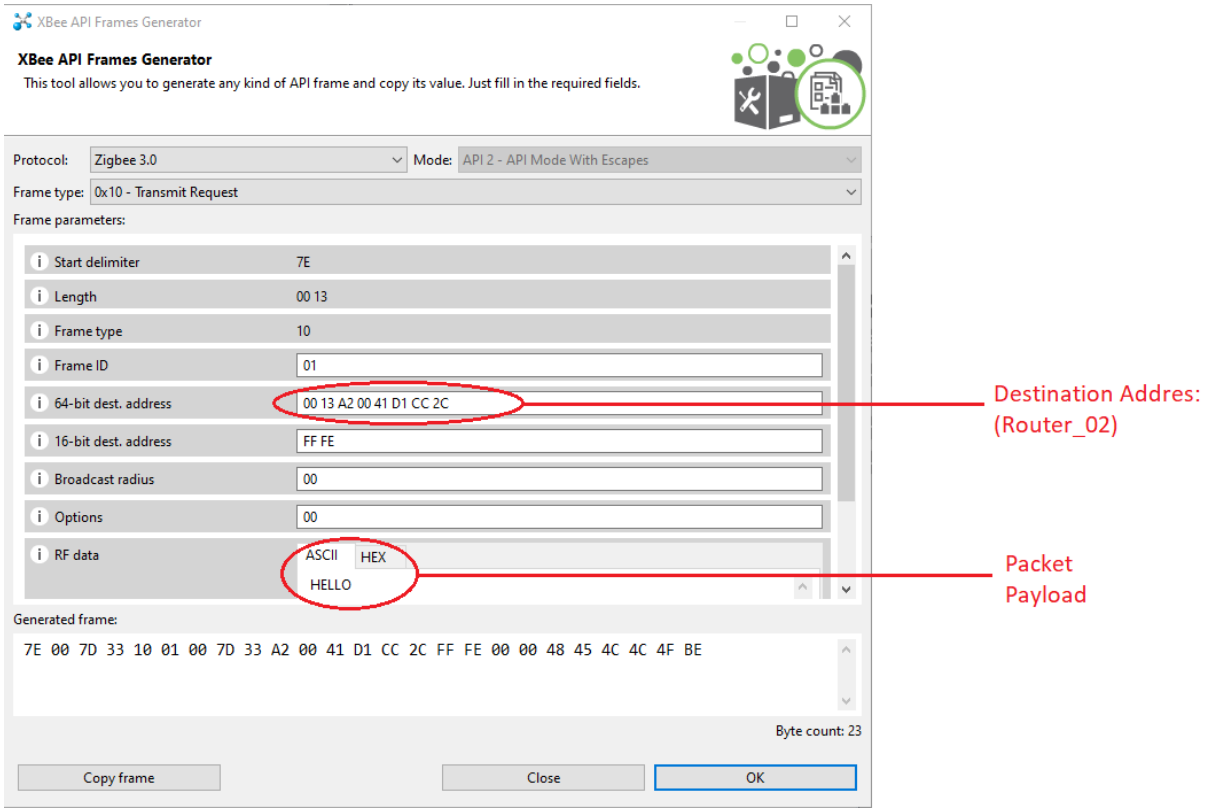

Figure D.1. Frame creation in XCTU for router functionality (external DoS).

Appendix E: SRP Configuration

XBee 3 Secure Access Options:

Bitfield 2 of the Secure Access Options (SA) can be enabled to require SRP authentication for Remote AT commands:

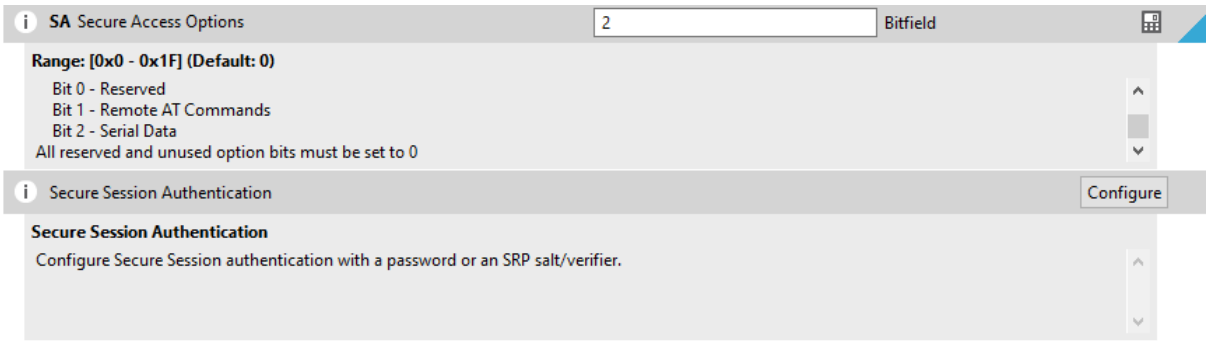

Figure E.1. XBee 3 secure access option for SRP authentication.

A password should also be configured in the Secure Session Authentication setting:

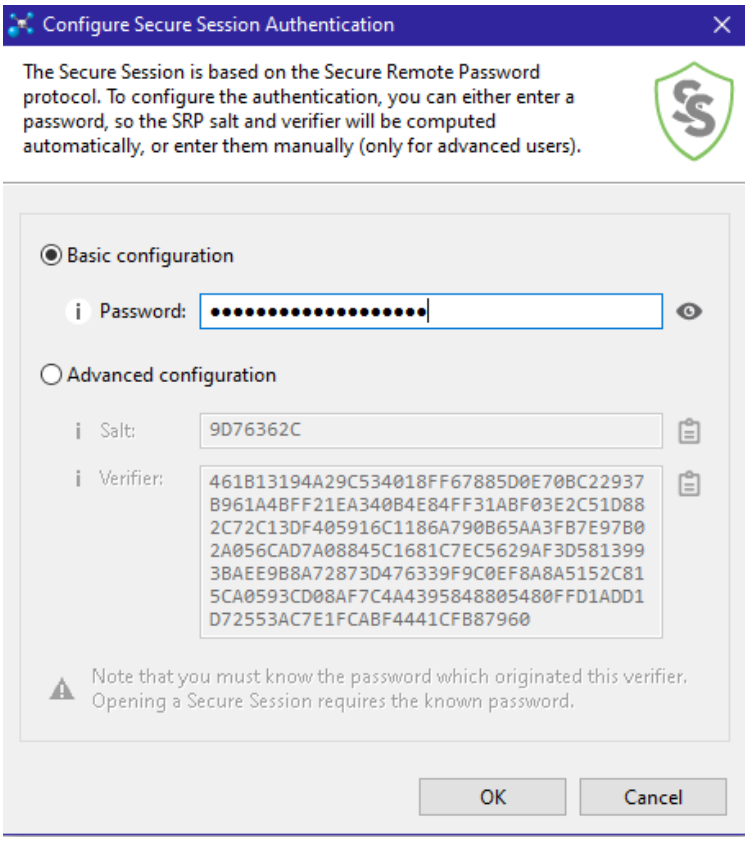

Figure E.2. XBee 3 secure remote password.