UNIVERSITY OF SOUTHAMPTON

Practical Adaptive Security for Resource-Constrained IoT Nodes

by

Sultan A. Alharby

Thesis for the degree of Doctor of Philosophy

in the
Faculty of Physical Sciences and Engineering
Electronics and Computer Science

March 2020

UNIVERSITY OF SOUTHAMPTON

ABSTRACT

FACULTY OF PHYSICAL SCIENCES AND ENGINEERING
Electronics and Computer Science

Thesis for the degree of Doctor of Philosophy

PRACTICAL ADAPTIVE SECURITY FOR RESOURCE-CONSTRAINED IOT NODES

by Sultan A. Alharby

Considered a key enabler on the Internet of Things (IoTs), Wireless Sensor Networks(WSNs) allow data to be collected from the surrounding environment. However, protecting the messages being transferred between these nodes is a complex issue. One of the reasons for security complexity in WSNs is resource limitations, such as energy and processing capability. Wireless nodes are usually powered by small batteries, which require reduced energy consumption so as to extend network lifetime. However, enabling security increases energy consumption due to the extra computation and transmission, required for encryption and authentication. Another reason is that the WSNs medium for transmission; and unattended operation make them vulnerable to various types of attack. The contradiction between the need for security and limited resources creates a trade-off between security and energy. There are several solutions to secure WSNs in the literature; however, few consider both efficiency and security. This research first evaluates the security overhead on IoT nodes. Then, it proposes a Practical Adaptive SEcurity architecture for Resource-constrained nodes (PASER). PASER uses the application preferences to select security level. Also, it contains a table for threat level, so that off the shelf threat detection systems can be easily used with PASER to feed that table and enhance the security decision.

The per-packet evaluation shows that security overhead in terms of energy consumption fluctuates between 31.5% at a minimum level over non-secure packets and 60.4% at the top security level of IEEE802.15.4 specification. The evaluation is carried out using simulation and then validated using actual hardware. According to the evaluated scenario, the proposed architecture shows an energy saving of 11% compared to the static-security model. The results may vary with different scenarios according to the application requirements and threat level in the surrounding environment. Also, the proposed architecture shows an improvement in network throughput, lifetime by 25% and delay by almost 141%.

Contents

List of Figures

List of Tables

Abbreviations

ACK	Acknowledgement frame
AES	Advanced Encryption Standard
APA	Application Preferences Agent
ASA	Adaptive Security Agent
ASH	Auxiliary Security Header
BDC	Battery Degradation Control
CBC	Cipher Block Chaining
CCA	Clear Channel Assessment
CCM	counter mode encryption and cipher block chaining message authentication code
CSMA	Carrier Sense Multiple Access
CTR	Counter
DES	Data Encryption Standard
DoS	Denial of Service
ECB	Electronic Code Book
FCS	Frame Check Sequences
FFD	Full-Function Device
IoT	Internet of Things
IV	Initialization Vector
MAC	Medium Access Control
MCU	Micro-controller unit
MIC	Message Integrity Code
PASER	Practical Adaptive SEcurity model for Resource-constrained nodes
PDR	Packet Delivery Rate
PHY	Physical Layer
QoS	Quality of Service
RAM	Random Access Memory
RDC	Radio Duty Cycle
RFD	Reduced-Function Device
RSA	Rivest-Shamir-Adleman
RSSI	Receive Signal Strength Indicator
RX	Receive
TDS	Threats Detection System
TEA	Tiny Encryption Algorithm
TX	Transmit
WSN	Wireless Sensor Network

UNIVERSITY OF Southampton

Academic Thesis: Declaration Of Authorship

I, Sultan Awad Alharby declare that this thesis and the work presented in it are my own and has been generated by me as the result of my own original research.

Practical Adaptive Security for Resource-Constrained IoT Devices

I confirm that:

1. This work was done wholly or mainly while in candidature for a research degree at this University;
2. Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated;
3. Where I have consulted the published work of others, this is always clearly attributed;
4. Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work;
5. I have acknowledged all main sources of help;
6. Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself;
7. Parts of this work have been published as:
    - Alharby, S., Harris, N., Weddell, A. and Reeve, J., 2018, April. Impact of duty cycle protocols on security cost of IoT. In *2018 9th International Conference on Information and Communication Systems (ICICS)* (pp. 25-30). IEEE.
    - Alharby, S., Weddell, A., Reeve, J. and Harris, N., 2018. The Cost of Link Layer Security in IoT Embedded Devices. *IFAC-PapersOnLine*, *51*(6), pp.72-77.
    - Alharby, S., Harris, N., Weddell, A. and Reeve, J., 2018. The security trade-offs in resource constrained nodes for iot application. *International Journal of Electrical, Computer, Energetic, Electronic and Communication Engineering*, *12*(1), pp.52-59.
8. Parts of this work have been submitted but still under review:
    - Practical Adaptive Security for Resource-Constrained IoT Nodes (IEEE Transactions on Communications).

Signed: .....

Date: .....

Acknowledgements

Many people have supported me throughout the journey of my PHD journey, and without them it would have been very difficult to complete this thesis. Firstly, I would like to thank my first supervisor, Dr. Nick Harris, for his invaluable contributions and advice throughout my PhD. Also, my sincere thanks go to my second supervisor, Dr. Alex Weddell, who has always been there to support and advise me throughout my study. Thanks to my third supervisor, Dr. Jeff Reeve, who has provided considerable support for the early development of my ideas, and for his insights too.

Also, my sincere thanks to Dr. Gary Wills for his helpful input as internal examiner over the course, and Dr. Thanassis Tiropanis for his valuable feedback also as internal examiner. Many thanks to my colleagues and the staff of the EEE group for all the useful discussions and comments throughout my research. I want to thank my fellow lab mates for the stimulating discussions, including, but not limited to, Dr. Singh for many late-night discussions on embedded systems, and Dr. Graeme Bragg for the few discussions we had about Contiki OS at the early stage of my research.

Thanks to my sponsor Majmaah University for supporting me financially and allowing me to dedicate my whole time to my research without being hampered and distracted by financial constraints. Thanks to my father and mother who are always there to provide encouragement. Finally, my special thanks to my wife Mashael for understanding that weekend working and late-night finishes were necessary to finish this work, my kids: Yazeed, Layan and Tameem who were welcome distractions and lightened the mood during days of hardship.

Chapter 1

Introduction

In recent years, smart homes, smart grids, environmental monitoring, smart irrigation, and other similar applications have helped to make the world more connected than ever before. The common vision associated with such distributed systems is linked to the paradigm of the Internet of Things (IoT). The term "IoT" is not associated with a particular technology, however, the Wireless Sensor Network (WSN) is a foundational technology for the IoT[1][2]. Sensors are the main element for reporting events in "things" such as cars and home appliances. They enable physical things to be part of the global infrastructure. WSNs have recently grabbed the attention of the academic and industrial communities[3]. They consist of distributed sensor nodes which are used to monitor physical and environmental conditions and send data to a sink node. A sink node are used to collect and pre-process data before send them to a server or a cloud. Data which are generated by sensor nodes can be, for example, pressure, humidity, temperature, pollution, or other physical phenomena. A typical WSN architecture is shown in Figure 1.1.

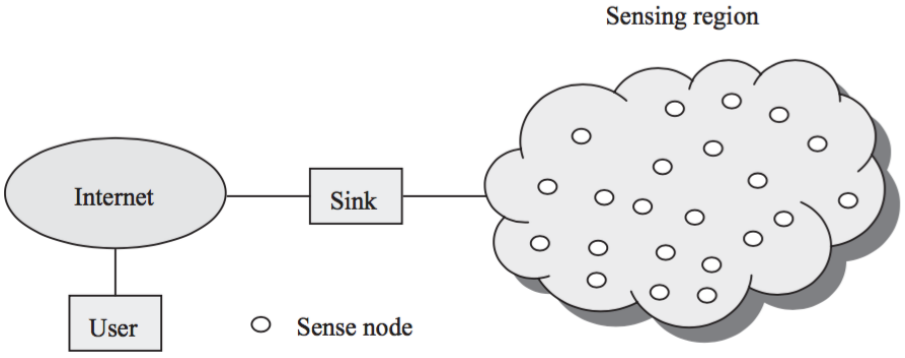


Figure 1.1: Prospective WSN architecture, taken from [4].

The use of WSNs is increasing and is becoming popular in many areas. In healthcare, WSNs are expected to change the way that hospitals work[5] by allowing real-time patient monitoring in hospitals [6]; and facilitating telemonitoring of people [7]. In smart

home applications, WSNs can be used to control lighting systems and the automation of daily chores. As for environmental applications, WSNs can be used to monitor atmospheric humidity, wind speed, pollution, and other environmental parameters. WSNs can also be used by the military to monitor battlefield resources and detect biological and chemical materials [8]. WSNs are an upcoming technology that could change the way we collect data, similar to how electronics and messaging has changed the way we communicate[9].

However, there are some significant concerns regarding WSNs. As already mentioned, wireless sensor nodes are limited in their resources. This limitation affects network operation. For example, WSN systems should work as efficiently as possible to save energy. However, enabling security does not support this requirement, as security increases energy consumption. Hence, the extra overhead added by security threatens the availability of the network by decreasing battery lifetime. Yet, ignoring security needs renders the network vulnerable to many types of attack. The security techniques which are used in wired and traditional wireless networks may not work or suit WSNs, where resources are limited. Security solutions should therefore take resource-constraints into consideration to extend network lifetime. Although many studies address the trade-off between security and energy in WSNs, security is still an unresolved issue. Some of the previous approaches which separate the interaction between network layers at the design stage neglect the existing trade-off between energy and security. The impact of security on QoS parameters should be studied and taken into consideration when developing efficient security solutions. There is therefore a need for a smart system which could manage this conflict.

This thesis discusses the security issue and its impact on WSNs. It investigates the relationship between security, energy consumption and QoS. The QoS parameters used in this thesis are packet latency, throughput, and network lifetime. These parameters are chosen as they are affected by enabling security. More information about these parameters are discussed in Chapter Three. Physical security is beyond the scope of this thesis. The security is discussed from the perspective of resource-constrained nodes, precisely energy resource. The outcome of this analysis will be used to develop an adaptive security system which can manage the trade-off between opposite parameters at run time. The security which are dealt with in this research are at the data link layer. Recent research pushes the idea of adaptive solutions. Adaptive solutions are used in wired networks for security and performance purposes, yet they are a hot topic in the area of WSNs. Adaptivity in this research refers to the ability of a system to switch between IEEE802.15.4 security levels at run-time. Detailed information about the design steps of the proposed adaptive security solution is presented throughout this thesis.

1.1 Research Questions

The main goal of this thesis is to develop an adaptive architecture which allows a continual trade-off between security, QoS and energy consumption in IoT embedded devices. This task is achieved by answering the following research questions:

1. **What is the impact on performance of current security implementations at the media access control(MAC) layer?**

   There are ambiguities regarding the overhead of security in WSN. The current implementation of security at the link layer is mostly IEEE 802.15.4 security specification. IEEE 802.15.4 supports eight different security levels optionally. The literature regarding security requirements and security services will be reviewed. Existing security techniques at the link layer will be investigated to identify the factors that contribute to the security overhead. Determining the impact of security is vital to the progression of this research. The intended evaluation includes the effect of enabling security on energy consumption and QoS parameters.

2. **How should we adaptively secure messages in IoT nodes?**

   The answer to question one will be used to develop an adaptive security architecture. The proposed architecture considers the trade-off between security, energy and QoS at the data-link layer. It should work autonomously at the run-time.

3. **How can these nodes be integrated into networks?**

   The objective of this question is to investigate whether the proposed concept can be implemented practically in WSNs, and also clarify whether the architecture is affordable to resource-constrained nodes. The challenges of running the proposed architecture on simulated and real hardware will be investigated and solved at this stage. The outcome of this question will show the feasibility of the proposed architecture.

4. **What is the impact of the proposed architecture on network performance?**

   The proposed architecture will be evaluated in terms of energy and QoS. Then, a comparison to static-security and no-security models will be made to determine the improvements which are achieved by the proposed architecture. The evaluation will be carried out first using simulation, then validated using real hardware.

This research investigates security at the data link layer for those *applications which are seeking a trade-off between security and energy consumption or network QoS.* It uses the widely used IEEE802.15.4 security specification at the data link layer of the IoT stack.

1.2 Research Justifications

Security has become essential to the acceptance of IoT embedded devices by numerous applications [10][11]. Any advantage which is brought by IoT may be rejected if the data is not protected. Unfortunately, applying security in IoT embedded devices is not an easy task, because "*standard security mechanisms, which are heavy in weight and resource consumption, are unsuitable*", as stated by[12]. Here 'heavy in weight' is meant in terms of memory requirement and processing time, and 'heavy on resources' in terms of energy consumption. Although there are ambiguities in the exact security overhead, a noticeable overhead on resources is evident. Batteries are the primary source of energy for WSNs, and also the indicator of node lifetime [13]. Thus, more overhead also means more energy consumption. The harsh environments in which WSNs are often deployed [14] make the option of changing batteries impractical[15]. Another issue with static security is that it does not consider other vital parameters, such as QoS. Static approaches may also not suit wireless sensor nodes which are inhibited by limited resources, and therefore leads to an improvement of one parameter at the expense of other parameters. For example, there is a correlation between security and energy consumption, so increasing security may shorten network lifetime.

Security and QoS in resource constrained nodes are opposite parameters[16]. Hence, there is going to be a trade-off between resource consumption minimisation and security maximisation[17]. "*Protocols must focus primarily on power conservation. They must have inbuilt trade-off mechanisms that give the end-user the option of prolonging network lifetime at the cost of lower throughput or higher transmission delay*", as stated by [18]. The option of extending the network lifetime applies to security as well, since it increases energy consumption. Applications should be given different options to manage the trade-off between security and energy. However, choosing the proper security trade-off is difficult at the deployment stage, because the conditions of the device and the environment change over time. A possible solution in such a scenario is to use an adaptation approach. An adaptive solution can play a vital role in managing the trade-off autonomously. Adaptation can enable system flexibility to allow changes to security settings at a run-time, if necessary. Adaptation can manage the trade-off between security and energy more efficiently than a static system. Also, it can enhance decision-making by involving other parameters which reflect changes in device condition and the surrounding environment. A few adaptive security architectures are presented

in the literature, however, they either focus on one specific security service, such as encryption, or provide an architecture without detailed practical information such as [19]. Most of the existing solutions do not consider threat level in the security decision. They use energy level to control security. Using energy to select the security level is not an appropriate approach, because it makes sensitive data vulnerable to an attack when the battery level is low. *Developing an adaptive security architecture which properly considers an input for threat level, available energy and application preferences in security solutions is the main focus of this research.*

1.3 Thesis Methodology and Organization

This thesis consists of six chapters which represent the stages taken to accomplish this research. The following are a brief description of each stage.

1.3.1 Literature Review

The first stage is to review the relevant literature on WSNs. It commences with a general discussion about WSN limitations and the IEEE 802.15.4 standard. The review includes identifying security requirements, security challenges and security services in resource-constrained nodes. Security challenges due to the resource limitations of sensor nodes are also explored, and the security services which are used at the MAC layer are investigated. *The investigation of security services has led to the selection of IEEE802.15.4 security specification in this research because it is the dominant MAC protocol in WSN.* The literature regarding existing security solutions has been studied and analysed. The relationship between security and quality of services is discussed. The literature review shows that energy is a significant concern in security solutions, and reveals a necessary trade-off between security and energy consumption. However, the exact overhead induced by the IEEE802.15.4 security specification is not clear. This chapter also discusses and critiques the current security solutions at the MAC layer and categorises them based on their functionality. The outcome of this stage has helped to identify gaps in this area and led to the selection of which parameters should be considered in security selection. In the next step, the results of this stage are considered in an initial review of a detailed assessment of security overhead. This chapter partially answers research question 1.

1.3.2 Evaluation of Security Overhead

The second stage presents a detailed evaluation of the overhead introduced by the different security levels of the IEEE802.15.4. This evaluation is accomplished through

simulation, the results of which are validated using real hardware. The assessment includes the impact of security on the following parameters: energy consumption, data throughput, latency and memory requirements. The effect of data length on security overhead is investigated as well. The Powertrace tool is used to measure the energy consumption in the simulation, while a power analyser is used to record the power in the testbed evaluation. The evaluation investigates the impact of each security level on the previously mentioned parameters. It shows that each security level is associated with a certain degree of energy consumption. The overall results show that security levels add significant overhead to node resources. The results depict that not only is energy affected, but other network parameters such as latency and throughput are affected by security as well. In general, as security level is increased, the more overhead is added to the network. The results of this evaluation encourage further progress in this research to answer the remaining research questions. The results of this chapter have been published in the IFAC/IEEE Conference on Programmable Devices and Embedded Systems (2018), the International Conference on Information and Communication Systems (2018), and the International Journal of Electronics and Communication Engineering (2018). This chapter answers research question 1.

1.3.3 Developing a Security Architecture

The third stage is the development of a solution which trades-off security with energy consumption. The difference in the overhead, which is introduced by IEEE802.15.4 security levels has led to the solution of using of adaptive security at the data-link layer. Chapter Four introduces Practical Adaptive Security for Resource-constrained IoT nodes (PASER). PASER is developed to balance the trade-off between security and energy consumption in resource-constrained nodes. It uses the application requirements and utilises input for threat level to select the security level. It also uses an energy-aware mechanism to control battery degradation, and consequently extend network lifetime. The design and functionalities of PASER and its sub-modules are discussed theoretically in Chapter Four, and use cases of the proposed architecture are given. Chapter Four answers research question 2.

1.3.4 Evaluation of the Proposed Architecture

The fourth stage is an evaluation of the proposed architecture. The aim of this evaluation is to test the practicality of the concept in terms of operation, and investigate improvements attributable to the PASER model. The Contiki OS and Cooja simulations are used. The evaluation shows an energy saving and an improvement in network performance. The simulated evaluation is then validated using CM-TMT-5000 hardware. Many experiments have been carried out to assess the PASER architecture in terms of

various network parameters. The results show a significant improvement using PASER compared to static security. This chapter answers research questions 3 and 4.

1.3.5 Conclusion and Future Work

Chapter Six summarises the work and contribution of this thesis, and presents suggestions for how this work can be expanded for further improvement. It assists future research in identifying research gaps in the security of resource-constrained nodes.

Chapter 2

Literature Review

2.1 Introduction

The previous chapter has concluded that security presents a challenge to resource constrained nodes due to its overhead and has demonstrated that a trade-off between security and energy is necessary to achieve an efficient security solution. This chapter gives an overview of the IoTs and WSNs and their associated security technologies, and addresses the state of the art in research on security solutions, QoS and the types of attack at the data-link layer of IoTs infrastructure. This chapter begins by providing an overview of the architecture of the IoT 2.2, the main components of a wireless sensor node, and the software which manages the hardware components. Section 2.3 discusses IoT security at the link layer. This section covers the requirements to secure the communication between wireless sensor nodes and the challenges of applying security in IoT embedded nodes. It then studies the security specification of the IEEE 802.15.4 standard and clarifies its security levels. This section continues with a discussion about the Advanced Encryption Standard (AES) algorithm. In Section 2.4, QoS and the effect of security on QoS is described. The impact of processing security algorithms on energy consumption is explored in 2.5. The concept of context-aware solutions is discussed in Section 2.6. Section 2.7 presents a review and analysis of security solutions at the link layer in the literature.

2.2 Overview of The IoT and WSNs

2.2.1 The Internet of Things

The number of Internet-connected objects is increasing rapidly. Expectations from different organizations are that the world is going to witness a large growth in the number

of connected objects in the coming years. This growth in the number of connected nodes is associated with the term "Internet of Things". This term was first used by Ashton in 1999 as a title of a presentation [20], and has received much attention in the past few years. The IoT is a term used to cover various areas related to the extension of the internet into the physical domain [21]. This extension takes place through the widespread use of distributed nodes which have identities in a network and a capability to sense or actuate [21]. There are many definitions of the IoT, but at a fundamental level it refers to the capability of things or objects which exist around us to communicate and interact with each other and to cooperate with their neighbours through a unique system to achieve common goals [22]. One of the biggest characteristics of the IoT is that it bridges the gap between the physical world and the virtual world. There is not yet any standardization definition for the IoT and its security, as the concept is still under development by research and industrial communities. The word things or objects in IoT means "*active participants in business, information and social processes where they are enabled to interact and communicate among themselves and with the environment by exchanging data and information "sensed" about the environment, while reacting autonomously to the "real/physical world" events*"[23]. In this context, WSNs are an essential component of the IoT because they represent the digital interface for physical objects [24]. There is a decline in the number of research which use the term WSN, and this because of that "*researchers are beginning to treat WSN as a technology integrated into the IoT ecosystem*" [24]. According to [24], the main technical difference between the IoT and WSNs is that a WSN can feature the internet as an option, however, in the IoT the Internet is an essential component. Table 2.1 shows the architecture of IoT communication protocols. This thesis proposes a security solution at the link layer of the IoT architecture. Hence, it uses the terms WSN and IoT to indicate a smart environment where things can sense and react to a change in the physical world and communicate autonomously with each other through a unique system of address.

Table 2.1: Typical IoT communication protocols

<table>
<tr><td colspan="2">Application Protocol</td><td>CoAP</td><td>MQTT</td></tr>
<tr><td rowspan="4">Infrastructure Protocols</td><td>Routing Protocol</td><td colspan="2">RPL</td></tr>
<tr><td>Network Protocol</td><td>6LoWPAN</td><td>IPV4/IPV6</td></tr>
<tr><td>Link Layer</td><td colspan="2" rowspan="2">IEEE802.15.4</td></tr>
<tr><td>Physical Layer</td></tr>
</table>

Many challenges slow the growth of IoT, one of which is the security issue. IoT applications may not be used and accepted by the public until reliable security solutions are put in place [25]. If these challenges are not addressed, the IoT will remain limited to a few areas, such as manufacturing and logistics [23].

2.2.2 Overview of Wireless Sensor Networks

A WSN consists of many battery-driven sensor [26] nodes. These nodes have the capability of sensing, computing and communicating [27] with each other over a specific location for a specific purpose, such as environmental monitoring [28]. Wireless sensor nodes usually operate unattended in a hostile environment [28]. These circumstances make nodes vulnerable to different types of attack[27]. WSNs are usually deployed to cover an area ranging from ten metres to several hundred kilometres [29]. A typical method of deploying these nodes is dropping them from a aircraft [30]. Under basic operation, WSN sensors detect physical phenomena such as temperature, vibration, etc., and then send their readings to a base station [31][27]. The base station, in its turn, sends all or part of the received data to a workstation or the Internet.

WSNs have been used in different applications for different purposes. Figure 2.1 shows some of these applications. For example, WSNs are used in environment surveillance applications to increase safety and security (such as monitoring forest fire and earthquakes). They are also used in military applications where they can be rapidly deployed to monitor suspicious motion in a friendly zone, and can help in detecting a chemical and biological attack on the battlefield. There are many other instances where WSNs can be utilised to collect data about changes in the physical world. Use examples can be found in [32] [33] [34].

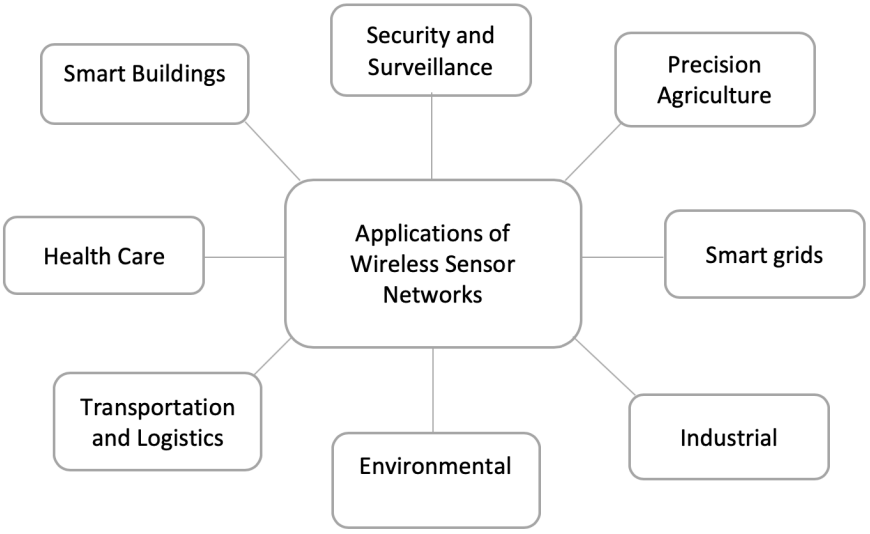


Figure 2.1: WSN applications.

2.2.3 Wireless Sensor Node Components

A typical wireless sensor node, as stated in [4], is comprised of four components, as shown in Figure 2.2:

- Sensing unit: this consists of one or a group of sensors and an Analog-to-Digital Converter (ADC). The sensor is the hardware which detects the change in physical phenomenon [35] and produces an analogue signal. The ADC then converts it to digital and sends it to the processing unit. There are many types of sensors available, such as accelerometer, temperature, light, humidity, and pressure etc.

- Processing unit: : this consists of a microcontroller that is responsible for processing data, performing tasks such as encryption/decryption, and controlling other components of a node. It usually employs onboard memory [36].

- Transceiver: this allows a wireless sensor node to communicate with other nodes over an electromagnetic spectrum [35]. This communication can be either transmission or reception. The radio chosen varies based on the frequency range and compliance with IEEE 802.15.4 [37].

- Power unit: this is used to power each component of a wireless sensor node. Usually, it runs on a limited battery. The battery is the most critical part of a node, and its limitation poses a problem for network designers. Hence, it requires each component to perform efficiently [36]. A typical nodes spend almost 99% of its time in sleep mode.

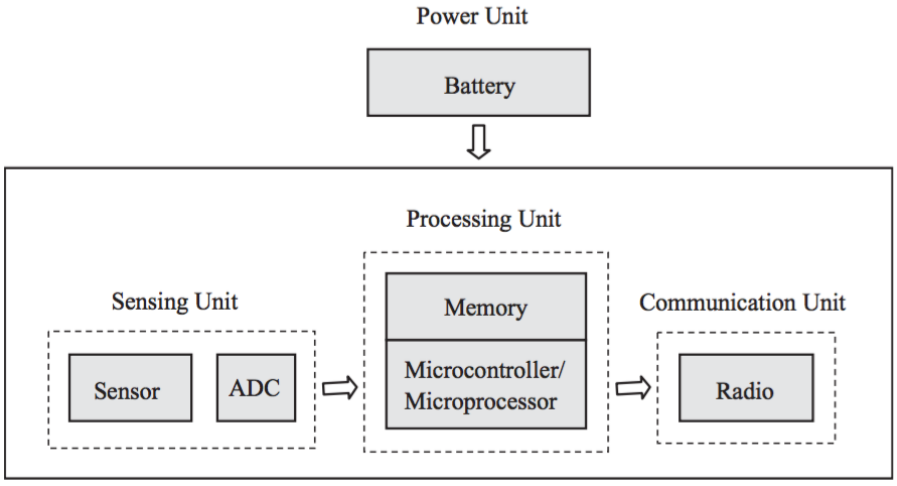


Figure 2.2: Wireless sensor node components, reproduced from [4].

There are many types of wireless sensor hardware of various capabilities and components. Tables2.2 and 2.3 present the differences between some popular platforms. They differ in MCU (ARM, Atmel ATmega or TI MSP430, etc.), supported operating system(Contiki, TinyOS and /or RIOT, etc.), connectivity(IEEE 802.15.4, Bluetooth and/or Wi-Fi, etc.), memory size(4, 48 or 128kb, etc.), and on-board sensors (temperature, humidity, and/or pressure, etc.).

A typical example of wireless sensor hardware is Tmote sky [39], which is a low-power node. This hardware is popular, and the results obtained from it can be compared

Table 2.2: Microcontroller and memory comparison

Mote platform	MCU	RAM	Flash	Radio
Teleos B/ Tmote sky	TI MSP430F149	10k	48k	CC2420
Mica2	Atmel ATmega128L	4k	128k	CC1000
Zolertia RE-Mote	ARM Cortex-M3	32k	512k	dual radio (2.4GHz and 868/915MHz)

to other research for credibility. It complies with the IEEE 802.15.4 standard and is powered by two AA batteries [39]. It has a USB port for configuration and also works as an alternative source of power. It runs on a Texas Instruments MSP430 micro-controller. It has 10 kilobytes of RAM, and 48 kilobytes of flash memory [39]. A CC2420 antenna is integrated on-board and supports a range of 50m indoors, and 125m outdoors. It features three integrated sensors for temperature, humidity, and light. Figure 2.3 shows Tmote sky hardware components.

Deciding which hardware brand to use can be a challenge, as no single platform suits all solutions. Nodes can be chosen based on many criteria such as their supported features, cost or popularity. The choice will be application dependent. However, the majority of WSN applications are and will continue to be application dependent [38].

2.2.4 IoT Operating Systems

The software which operates wireless sensor nodes plays a vital role in the development of WSNs [40]. The main role of an OS is to enable reliable operation and management of system resources such as processors, timers and memory in a well-ordered and controlled fashion[41]. Also, an OS implements a communication protocol and controls the hardware energy consumption [40]. IoT OSs which are designed to run on wireless sensor nodes should be small in size and use less energy [42]. These OSs are different from traditional OSs designed for powerful devices such as laptops and PCs. The main difference is the limited hardware of a typical wireless sensor node [43]. These constraints require a new approach to developing WSN OSs [41] that emphasises minimal power consumption. Wireless sensor nodes are designed to work for a long time on a limited battery. There are many lightweight operating systems available for the IoT, and new ones are emerging. It might be difficult to determine the most suitable OSs, as different applications may require different OSs. IoT OSs provide similar services in general, but there are some differences in the way they function. Older IoT OSs tend to use an event-driven programming model, while recent OSs support a threading based model [41]. Characteristics of these operating systems may affect the decision of the developers. Examples of the most widely used operating systems are Contiki and TinyOS [40][44]. Table 2.4 shows the most important differences between Contiki and TinyOS.

Table 2.3: Sensor platforms comparision, (reproduced partly from [38])

Parameter	**TelosB/Tmote Sky**	**MICA2/MICAZ**	**SHIMMER**	**IRIS**	**Waspmotes**
Size(inches)	2.55 x 1.24 x 0.24	2.25 x 1.25 x 0.25	1.75 x .8 x 0.5	2.25 x 1.25 x0.25	2.9 x 2.01 x 0.51
Weight(gms)	63.05	63.82	10.36	69.40	20.00
Battery	2xAA	2xAA	250 mAh Li-Ion	2xAA	1150 mAh Li-on
Year	2005	2002/2004	2006	2007	2013
Manufacturer	UC Berkeley	Crossbow	Intel	Crossbow	Libenium
Cost(US$)	99/139	99	269	115	173.77
Strong Points	Low power microcontroller and RF module	Expansion connectors for attaching external sensors	-Supports Bluetooth, Real time capability -For long term wearable use	3 times radio range compared to MICA nodes at half the power consumption	-Eight types of radio support

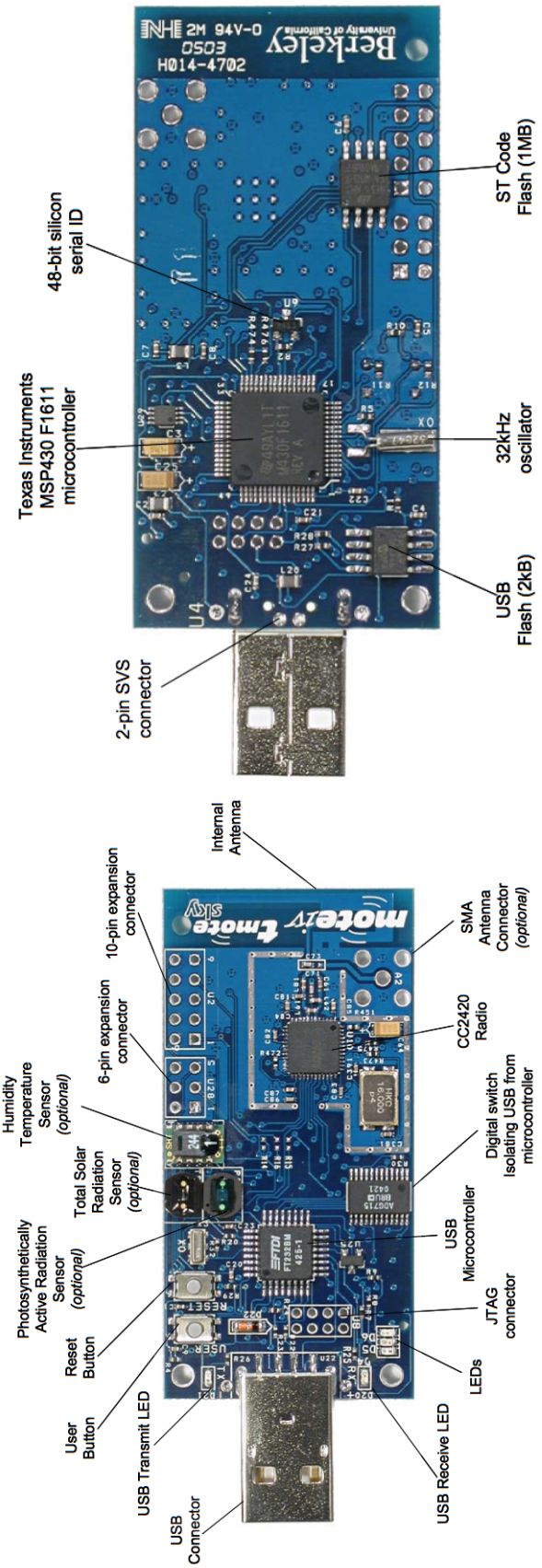


Figure 2.3: Tmote sky platform, (taken from [39])

Table 2.4: Comparison of two typical IoT operating systems

Operating system	Contiki	TinyOS
Supported platforms	For old and recent sensors, Mica family, Telosb, RE-Mote,etc	For old sensors, Mica family, Telosb, ATMega,etc
Scheduling	Event-driven with optional pre-emptive multitasking	Event-driven OS with non-preemptive multitasking
Programming language	C language	nesC
Memory management	Dynamic	Static
Communication protocol	μIP and Rime	Active message
Simulator	Cooja	Tossim

The authors of [43] indicate that TinyOS is preferable when resources are scarce and every little bit of memory must be used efficiently. Contiki is better when flexibility is most important. An example of a flexibility requirement is the need to frequently update node software for many nodes. Contiki can dynamically update the required program, while TinyOS requires a change of software and OS. Using this feature, programmers can reconfigure and download firmware through broadcast rather than the typical method of connecting a cable to nodes. This feature would save time and effort for the network administration. More information about IoT operating systems is discussed in [42], [45], [41] and [43]. This research has used the Contiki OS which is developed by a worldwide community of experts [46]. Contiki is a highly portable open-source OS and runs on many different wireless sensor platforms [46].

2.2.5 Cooja Simulator

This section discusses the Cooja simulator, used in this thesis. Cooja is a WSN simulator supplied with Contiki OS [47]. Figure 2.4 shows a snapshot of the Cooja graphical interface. This simulator allows developers to debug and simulate their applications on large-scale networks [46] before they run it on real hardware, and is also used for performance evaluation. This simulator is written in Java and can emulate real hardware. Cooja has two emulators for the MCU: MSPSim and Avrora. MSPSim is used to emulate TI MSP430 based nodes, while Avrora is used to emulate Atmel AVR-based nodes [48]. Cooja supports many hardware, such as TelosB/SkyMote, MicaZ, Wismot, and can be run by downloading the Instant Contiki 3.0 development environment, which has all the necessary tools to run and emulate a WSN.

Cooja supports a tool called Powertrace which can be used to estimate the power consumption of a radio and MCU. The Powertrace tool [49] is used to provide detailed information about where power is being consumed (transmission, receiving, etc.). It calculates the time each component spends in a particular mode, as shown in Figure

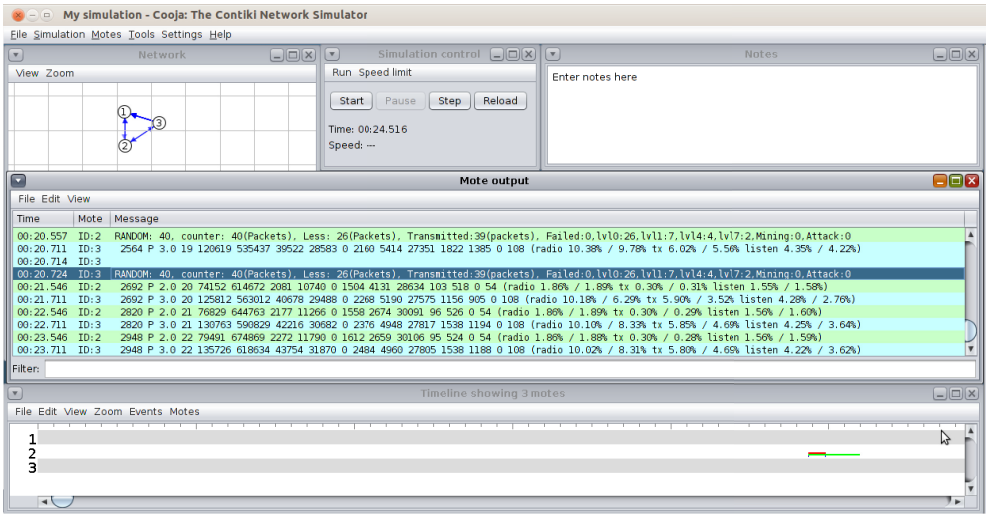


Figure 2.4: Cooja Simulator

2.5. This tool is claimed to be 94% accurate in measuring the power consumed by a real device [49].

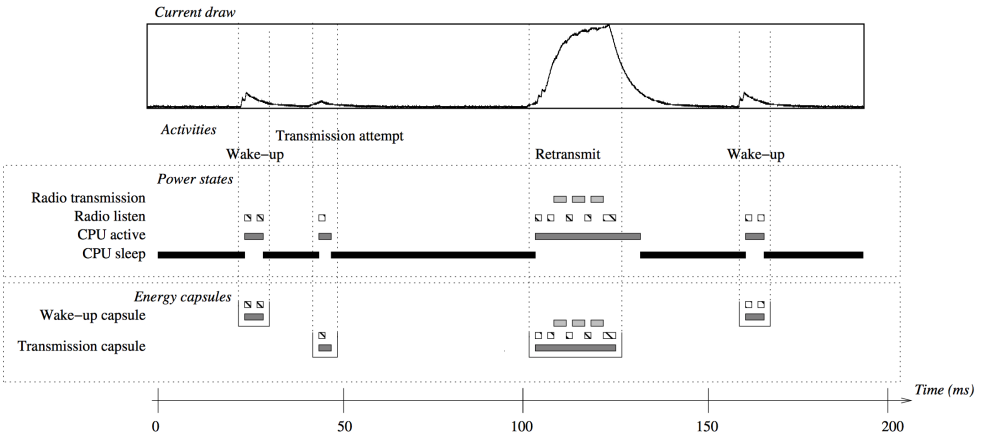


Figure 2.5: Powertrace tool mechanism , taken from [49].

2.2.5.1 Radio Duty Cycle Protocol

Low power nodes have limited batteries which usually cannot be replaced [41]. Hence, in these instances it is necessary to keep the radio turned off as much as possible [50] while still allowing low power nodes to communicate with each other at the minimum requirement. This technique of controlling the radio state to save energy is called duty cycling[51]. Several protocols have been proposed in the literature to control the radio state to extend the network lifetime. These protocols can be classified into synchronous, asynchronous and semi-synchronous duty cycles[52], depending on the mechanism utilised to control the node schedule. In the synchronous scheme, nodes are time synchronised [52], hence, all nodes wake up and sleep at a set time. Contrary to this, asynchronous nodes do not need to work simultaneously, and they have no agreed

wakeup/sleep schedule. A semi-synchronous duty cycle combines the two methods by grouping neighbour nodes into a synchronised cluster where different clusters communicate with each other asynchronously. A typical example of a duty cycle protocol is Contiki-MAC [50], which uses an asynchronous mechanism. Contiki-MAC supports two methods of transmission: unicast and broadcast.

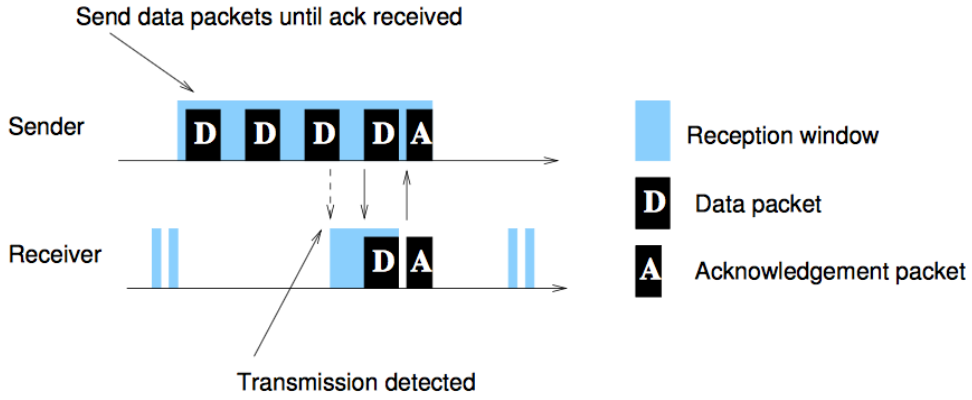


Figure 2.6: Unicast transmission of Contiki-MAC (taken from [50]).

In a unicast transmission, as shown in Figure 2.6, every node wakes up regularly to check the channel for incoming communications. If a transmission is detected, the radio remains in an "on" state to receive the next packet and transmit a link-layer acknowledgement back to the sender. In the broadcast transmission, the radio keeps sending the packet repeatedly for a full interval wake-up time in order to make sure that all nodes have received the packet. There is no acknowledgement in the broadcast transmission, as illustrated in Figure 2.7.

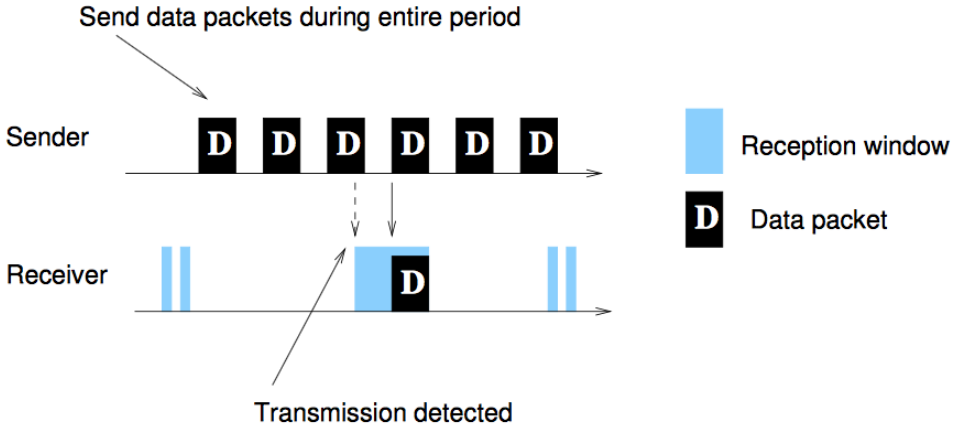


Figure 2.7: Broadcast transmission using ContikiMAC, taken from [50].

Contiki-MAC uses Clear Channel Assessment (CCA) to check channel activities. The RSSI value is then checked during the radio wake-up time. [50]. CCA returns a positive value only if the RSSI is below a pre-defined threshold, which means the channel is clear [50]. In the unicast transmission, the whole packet is sent repeatedly until an

acknowledgement is received from the receiver. Contiki-MAC uses a mechanism called Phase-Lock to reduce the number of transmitted packets during the transmission process [50]. In this mechanism, a sender can predict the wake-up time of a receiver by recording the time of the received acknowledgement for future transmission. This information is then used to anticipate the wake-up time of a node. Contiki-MAC is used in this research because it is supported in Contiki OS. It is also chosen due to its ability, like other MAC protocols, to keep the radio switched off for 99% of the time.

2.3 IoT Security at the PHY and MAC layer

Many researches have been carried out to advance the widespread deployment of the IoT, however, as security issues have not yet received significant attention [53], there is still no existing uniform security standard. The issue of power consumption is one of the main problems that face security solutions development. Security solutions should not only be secure but efficient as well. The IEEE802.15.4 security specification is widely used at the IoT data-link layer. This section discusses the major security requirements of IoT applications, then presents the IEEE802.15.4 standard and its security specification. The section concludes with a general classification of attack types at the data-link layer.

2.3.1 Security Services and Requirements

There are specific requirements which should be met to secure the communication between wireless sensor nodes. The following are the most important security services:

- Data confidentiality: Wireless sensor nodes should not reveal information to non-member nodes. An adversary should not have the ability to obtain any information from the data exchanged between network nodes, particularly in the case of critical mission applications. Only the intended receiver should interpret the message [54]. Confidentiality can be maintained using cryptography[55] [56].
- Data integrity: Data confidentiality can only assure that no entity except the intended receiver can interpret the message. However, it cannot detect whether a message has been altered when data are exchanged between wireless sensor nodes. An adversary can inject packets into nodes or alter the content of packets to confuse the intended receiver. Integrity ensures that any received message is as it was sent from the source [27]. Packet integrity can be checked by including a Message Integrity Code (MIC) in every packet. MIC is computed by calling a hash function which can check whether or not the data is altered during the communication. Changes in packets can occur either due to an attack or an error. The checksum can only detect the corrupted packet [57].

- Data availability: is the mechanism of ensuring that a network is available for data to communicate [58][59]. Network availability is targeted by attacks such as Denial of Service (DoS). A battery provides an indication of node lifetime, thus, the software overhead on nodes should be reduced to extend network availability.

- Data freshness: is a strategy for assuring that data is recent, and old data has not been re-used [59]. An adversary may re-send a copy of a valid packet to all or particular nodes to confuse them. Hence, there should be a mechanism to assure that data contents are recent. This requirement is important in cases where a shared key is used in a network. To achieve freshness of data, a nonce or any kind of timestamp should be added to each packet [60][59].

- Data authenticity: is the ability to verify that a message has been sent from the claimed sender [58]. An adversary can easily inject data in the packet through the transmission, so the receiver should ensure that the message has arrived from a legitimate party. This can be achieved by appending a message authentication code header to every packet. The message authentication code is also named the message integrity code (MIC). This research will use MIC to indicate to message authentication code, so we can differentiate between media access control (MAC) and message authentication code.

In practice, there is no single solution that would assure all security requirements when building a WSN. Some applications may require that specific security goals are guaranteed, and other may need to enable all security services[17]. Hence, security services should utilise minimum authentication and computation overheads to meet the efficiency requirement in resource-constrained nodes. Security could be adjustable and multi-level based on the available resources [61]. Employing minimum security improves the trade-off between resource consumption and security requirements. The following section discusses the challenges to enabling security in WSNs.

2.3.2 Challenges to Security in Resource-constrained Nodes

Due to multiple challenges, enabling security in a WSN is not a simple task. The following subsections discuss some of these difficulties.

2.3.2.1 Hardware Limited Resource

A wireless sensor network is different from a traditional network. It uses nodes which are limited in processing capability, small in memory capacity, and low in energy capacity. For example, tmote sky hardware which uses MSP430 micro-controller has only 10k RAM and 48k Flash memory [39]. Such hardware limitations mean that traditional

security solutions which demand high processing are difficult to employ [62]. Thus, optimised lightweight techniques should be used with this special network [58]. Below are the significant constraints associated with WSN nodes:

- **Energy limitations**: A typical wireless sensor node runs on two AA batteries with a limited lifetime [63]. Each hardware component of a node consumes energy, such as the microcontroller and transceiver. Unfortunately, ensuring security requires both extra computation by the MCU and the transmission of extra bytes by the transceiver. This extra overhead is translated to more energy consumption. Hence, there will always be a trade-off between security and energy consumption in WSNs [17].
- **Memory constraints**: Wireless sensor nodes are limited to a few kilobytes of storage capability. The use of complex algorithms, such as heavy traditional encryption, is not efficient in these limited nodes [62]. Normally, wireless sensor node storage consists of flash memory for storing the operating system code and RAM for storing variables. Security algorithms should be small enough to be accommodated by such limited memory. With such limitation, developers should avoid using complex algorithms such as public keys.

2.3.2.2 Unattended Operations

One of the features of a WSN is its capability of operating in a hostile unattended environment. However, this feature brings new security challenges. For example, it is difficult to respond to an attack manually in unattended operation, so security services should be in place to protect network operation. If security services are not established, sensitive data might be exposed to the risk of being read by an adversary. Wireless sensor nodes could be located in an unfriendly environment [27], such as behind enemy lines or in harsh environmental conditions, which increases the possibility of network attack. This type of network is usually managed remotely and is inaccessible physically, and as such is in need of protection.

2.3.2.3 Unreliable Communication

WSNs are connectionless and this make them vulnerable to security issues and other communication issues such as packets loss and collision. If an attacker is situated in the network coverage area, they can listen to the network traffic and may use the collected information to launch an attack.

2.3.3 Attack Types at the Data-link Layer

There are various types of attack which target WSNs. Attacks can be classified depend on factors such as: performer, goals or layer-wise [64]. In this research, the attacks are categorised into two main types: passive attacks and active attacks [65]. In a passive attack, an intruder monitors the traffic attempting to learn or understand some information from the exchanged messages. The communication occurs over the air in WSNs, so the medium is open to everyone, including intruders. Anyone with the right transceiver can listen to the traffic in the network. In a passive attack, intruders do not affect the system resources or functionality. However, the information they obtain through the passive attack can be used later to launch other types of attack [64]. A passive attack violates the confidentiality of data. An attacker can easily read sensitive data which are transmitted unencrypted or with weak encryption. Hence, the passive attack occurs without the consent or knowledge of the system user because no active actions are yet taken. Traffic analysis and eavesdropping are examples of passive attacks [65]

On the contrary, in an active attack, attackers take active action to control the system. They may modify the content of a message to mislead the network users. The active attack affects the integrity, authenticity and availability of data. Such attack is not limited to packet modification, it can also inject new packets into the network. Examples of active attacks are: replay, sinkhole, spoofing and man-in-the-middle attacks. A detailed discussion of attack types is beyond the scope of this research, however, they are mentioned briefly to understand the suitable security settings to defend against such attacks. The reader can find more information about attack types in [65] [64].

2.3.4 IEEE 802.15.4 Security Overview

This section provides an overview of the IEEE 802.15.4 standard and then discusses its security specification in detail. IEEE 802.15.4 is used in many applications such as healthcare, home automation and industrial. This standard is designed to support communication with low power consumption and low data rate [15]. It could run on a battery from several days with heavily duty-cycled nodes to several years with a low duty-cycle. It supports three frequency bands: 868 MHz, 915 MHz, and 2.4 GHz [66]. The maximum frame size in this standard is 127 bytes with a data rate of 250 kbit/s [67]. The frame size includes the frame headers used for packet delivery and the packet payload size. Figure2.8 shows the frame format of IEEE 802.15.4. The standard defines two types of nodes based on their functionality: Reduced-Function Device (RFD) and Full-Function Device (FFD) [15]. FFD can run as a coordinator or ordinary device, while RFD only works as a device. This standard defines only the physical and medium access control (MAC) sub-layer [15], therefore different protocols can work on top of that for the network and application layer.

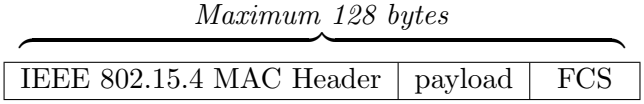


Figure 2.8: IEEE 802.15.4 MAC frame format

IEEE 802.15.4 offers two operational modes: a beacon-enabled mode and a beacon-disabled mode [15]. In the former case, network communication is managed by a coordinator [66]. The coordinator sends regular beacons to synchronise nodes and manage the whole communication. With beacon-disabled mode, every node can access the channel through a CSMA/CA protocol. IEEE802.15.4 standard is adopted in a range of applications, including military surveillance, environmental monitoring, and industrial automation. These applications require secure communication to protect the exchanged data. IEEE 802.15.4 can function in a secure or non-secure mode. If security is not enabled, then no extra header are added.

IEEE 802.15.4 security specification supports eight different optional security levels. The supported security services at each level are different in strength and type of protection. The security services protect data confidentiality, integrity, authenticity and replay protection on a per-frame basis [68]. Security levels generally offer from no security, encryption only (AES-CTR), authentication and integrity only (AES-CBC-MAC), to all three security services: encryption, integrity and authentication (AES-CCM). AESCBC-MAC has three different MIC lengths: 4, 8 and 16 bytes [69]. CCM supports a high level of security and has three options: 4, 8 and 16 bytes. MIC is used to guarantee that data has not been changed and also guarantee that data originates from a legitimate source. The length of the MIC represents the strength of integrity and authentication. Figures 2.9, 2.10 and 2.11 show the difference between security levels of IEEE 802.15.4 security suites. The name of each level consists of two to three parts. The first part indicates the cryptography scheme (AES if security parameters are enabled). The second part indicates the mode of operation used in the cryptography scheme. The last part, if applicable, indicates the message integrity code (MIC), which could be of varying length. The security services are enabled at the MAC layer [66]. An application can operate at the preferred security level. The security level is specified in the application layer. The IEEE 802.15.4 security specification provides hop-by-hop security, where every node in the network should be trusted [69].

Advanced Encryption Standard (AES) cipher is used in the standard with a fixed block size of 128 bits. The key length is variable and can be 128, 192 or 256 bits [15]. However, the key length recommended by the standard is 128 bits. An unsecured frame consists of three fields: a MAC header with 7 to 23 bytes, data payloads with 0 to 115 bytes, and Frame Check Sequences (FCS) with 2 bytes [68]. A secure frame has one more header named the Auxiliary Security Header (ASH), with 5 to 14 bytes. Also, if authentication is enabled, the secure frame has an additional header for the Message Integrity Code

(MIC) . One of the contents of ASH is the Frame Counter, with 4 bytes for replay attack detection. The frame counter is set by the outgoing frame at the transmitter side. The frame counter field is then checked at the receiving node, and is accepted only if the value is higher than the previously received value.

Figure 2.9: CTR security level

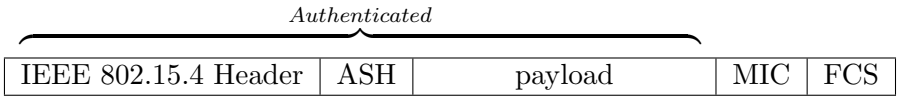


Figure 2.10: CBC-MAC security level

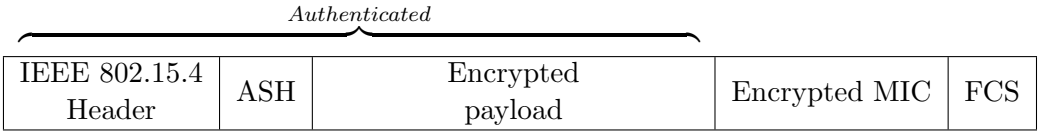


Figure 2.11: CCM security level

CTR mode encrypts only the payload, and ASH is sent in clear text, as shown in Figure 2.9. At the CBC-MAC security level, MAC header, ASH and payload are authenticated, but not encrypted, as shown in Figure 2.10. CCM, as shown in Figure 2.11, is considered the highest security level. It combines the two mentioned security services: encryption for the payload and integrity and authentication for the MAC header, ASH and payload. Only a single 128 bit key is required for this mode [70]. In the security enabled mode, security services are added for the outgoing frame at the transmitter side according to the security level. At the CCM security level, the transmitter node starts by authenticating a packet and then encrypting the computed MIC and payloads in CTR mode. The receiving node decrypts and computes the MIC. If the received MIC and the generated MIC are equal, the payload is then decrypted and passed to the upper layer; otherwise, the receiving process fails.

2.3.5 Encryption Algorithms

Many techniques for using security algorithms have been presented in the literature. However, the cryptographic algorithms themselves in terms of complexity and efficiency make a difference in resource-constrained nodes. The difference can be observed in computation time, allocated memory and energy consumption. Security algorithms are an important part of security solutions [71]. In general, cryptography can be classified into a symmetric key and asymmetric key [72]. The first and most popular is symmetric encryption. The symmetric key is preferred by most of the research community in the

constrained-resource nodes due to its efficiency [71]. The main challenge which faces symmetric encryption is the use of a single key [73] for the entire network. Another challenge is the distribution of the key to network member nodes. The second type is asymmetric encryption, which uses different keys for encryption and decryption processes. There have been numerous attempts to use asymmetric encryption, but the impact on wireless sensor nodes and network performance requires further research. Asymmetric cryptography is expensive in terms of power consumption, bandwidth, memory requirements and computational latency [71][74]. A popular example of public-key encryption is RSA (Rivest–Shamir–Adleman) algorithm. However, attention has recently been given to more efficient forms of asymmetric encryption such as elliptic curve cryptography [75]. Another technique which can be used is the combination of symmetric and asymmetric encryption, known as hybrid-protocol. This type of encryption uses asymmetric encryption to generate keys, and symmetric encryption to encrypt the actual data. The hybrid approach utilises the advantages of the high security of asymmetric encryption to generate keys, and the efficiency and low power consumption of symmetric encryption. Although it solves the problem of key distribution, it is still an additional overhead on the network and does not solve the issue of single key usage. This research uses AES symmetric encryption because it is faster and introduces less overhead compared to asymmetric encryption [76].

2.3.5.1 Symmetric Encryption

Symmetric encryption is a method of cryptography whereby the sender and receiver use the same key for encryption and decryption, as shown in Figure 2.12. There are two types of symmetric ciphers: stream ciphers and block ciphers[77]. According to [78], block cipher is more energy-efficient than other methods. The size of encryption differs for stream cipher and block cipher. Block cipher usually operates as a block of 64 or 128 bits, while stream cipher encrypts per bit by XORing the bits stream with a pseudorandom sequence [78]. A significant difference between the two is that stream cipher can be variable in length based on the plain-text, while with a block cipher the length of an encrypted plain-text has a fixed size [79]. If the length of a plain-text is less than the block cipher length, padding will be added to the plain-text at the source, and this padding is removed at the destination. Symmetric encryption is the standard used in WSNs due to its efficiency in terms of power consumption and memory usage. It provides a faster service with low resource consumption. Several algorithms have been developed using symmetric key cryptography. Examples of symmetric encryption include Data Encryption Standard (DES), Advanced Encryption Standard (AES) [80], Tiny Encryption Algorithm (TEA)[81] and other symmetric algorithms. Most of the proposed security protocols for WSNs in the literature are based on symmetric encryption. The problem with symmetric encryption is the key distribution, as there should be a supportive method to deliver the keys through a secure channel. Another problem

is the usage of a single key for the entire network [77]. Hence, compromising one sensor can lead to compromising the entire network. Figure 2.12 shows the mechanism of encryption in symmetric cryptography.

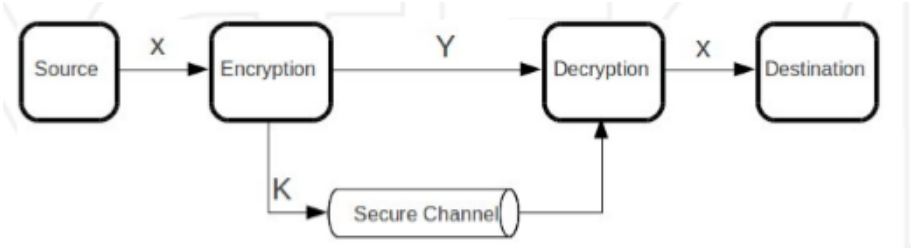


Figure 2.12: Symmetric cryptography (taken from[82]).

2.3.5.2 Mode of Operation

Alongside the selection of encryption cipher, it is also essential to choose the most appropriate operation mode. Mode of operation specifies how a block cipher handles data when a message length is longer than the employed block cipher. Each mode of operation has its own characteristics. However, some modes do not protect identical packets, so using the same plain-text under the same password produces the same cipher block. Some operation modes use a randomly generated number known as an initialisation vector, which produces a different cipher when re-encrypting the same plain-text. There are five standard modes recommended by the National Institute of Standards and Technology (NIST). These are: Electronic Code Book (ECB), Cipher Block Chaining (CBC), Output Feedback (OFB), Cipher Feedback (CFB), and Counter (CTR)[83]. CBC is the most popular block cipher mode and is therefore the one used in this research. CBC uses an initialisation vector to generate a random block each time.

2.3.5.3 Asymmetric Encryption

Asymmetric cryptography uses two different keys: private, and public [72], as shown in Figure 2.13. These keys are mathematically related to each other. The public key is broadcast to the other nodes, but the private key is kept secret. Asymmetric cryptography is not preferred for use in WSNs because it requires a significant amount of resources to perform [53].

Recently, there have been many attempts to use asymmetric encryption, especially with lightweight asymmetric encryption such as Elliptic Curve. However, it still introduces significant overhead compared to symmetric encryption. Figure 2.13 shows the mechanism of encryption using asymmetric cryptography. According to [84], the Elliptic Curve Integrated Encryption Scheme"*consumed 1,230 times and 250 times more energy than*

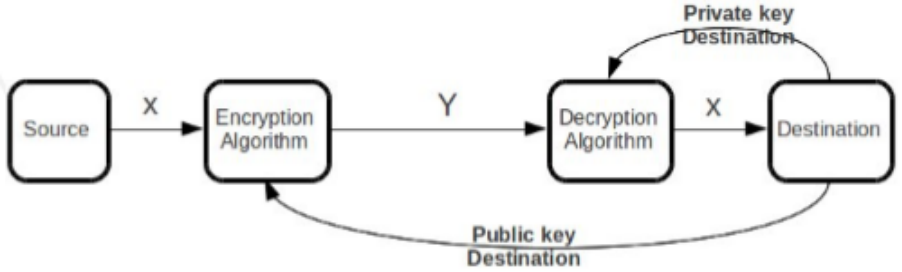


Figure 2.13: Asymmetric cryptography (taken from [82]).

AES-128 during encryption and decryption, respectively". Table 2.5 shows the difference in energy consumption between AES-128 and ECIES encryption.

Table 2.5: Energy consumption of each cryptosystem algorithm for a packet of 60 bytes in size (table is reproduced from [84]).

Algorithm	**Action**	**Energy**
AES-128	Encryption	0.078 mJ
	Decryption	0.19 mJ
ECIES	Encryption	96 mJ
	Decryption	48 mJ

AES encryption is the most broadly used encryption [83]. It is efficient in terms of energy as well[85]. Besides, it is recommended and used by the IEEE 802.15.4 security specification. This research uses AES symmetric encryption because it is faster and more efficient compared to asymmetric encryption [76].

2.3.5.4 Key Length and Encryption Strength

The strength of cryptography depends on both the key length and the cipher algorithm [86]. Key length denotes the number of bits contained in the encryption key [87]. Key length is generally a power of 2 [87], because data is usually stored in chunks of 8 bits. Hence a longer key provides stronger encryption. For instance, AES-256 bits provides more security than 128-bits with the same encryption. It is important to note that different ciphers may require different length keys to achieve the same security level. For example, 128 bits AES encryption is more secure than 128 bits RSA encryption [88]. It is recommended to choose a key of 2048 bits RSA or more to achieve a level that matches 128 bits AES [87]. The difference is due to the nature of the mathematical problem in these protocols [89]. In general, a successful attack against AES may try every possible key, while RSA relies on the assumption that factoring a relatively large number is difficult [90]. Key length is often used to measure the strength of cryptography. Key length can be used to provide different levels of security if the key length has a different overhead on the resources.

2.4 Quality of Services in WSNs

2.4.1 Notion of QoS

Quality of Service can be defined as a set of service requirements which must be fulfilled throughout the transmission from source to destination[91]. QoS has an essential role in all types of network, wired or wireless. Conventional QoS parameters are not sufficient for WSNs due to the dynamic topology, resource limitations [91] and the nature of their applications domains. Network lifetime, latency, and throughput are examples of QoS parameters. Energy-aware protocols are one of the options used in the trade-off for QoS improvement in WSNs. The available resources and application requirements ought to be considered when designing a security solution. Some applications may require high QoS; others may not. However, the cost of energy is of great importance when considering any solution for WSNs. The right trade-off between power and QoS can help to extend the network lifetime to the maximum it can reach.

2.4.2 Effects of Security on QoS

Security services must ensure the confidentiality, integrity, authentication [12], and freshness of the data being shared between wireless sensor nodes. However, these security services come at a cost to QoS parameters. Major proposed solutions for securing the communication in WSNs consider QoS and security separately [89]. Security has a substantial impact on network QoS [12]. It degrades the performance of WSNs. Security and QoS should thus both be considered when designing a security solution for WSNs [86]. In some applications, security and efficiency are both needed to meet the application requirements. However, achieving all application requirements could be difficult due to the trade-off between performance and other parameters. For instance, the latency introduced by authentication decreases the throughput due to the extra bytes required for security processes [12]. Also, the complexity of an encryption algorithm may affect the processing time. QoS and security are opposite parameters [89], so their coexistence should be studied and taken into consideration. High security is necessary for some applications, such as those used in the healthcare system. However, other applications may only require a minimum security level, such as environmental applications. Hence, a suitable trade-off between security and performance is required [92][27]. This trade-off is approached through adaptation. Adaptation can be used to control the system operation to meet certain requirements in terms of performance, security and energy consumption at the run-time. Different approaches of adaptation are discussed in Section 2.7.2.

2.5 Energy Consumption

Energy use is one of the most significant issues when dealing with wireless sensor nodes, as they are restricted in terms of energy resources, and this energy limitation affects node operation. Usually, they run on batteries, which need to be either replaced or recharged [93]. Unfortunately, this may not be possible for many wireless sensor nodes, as the deployment area may be in a harsh environment or in a difficult to reach area [94]. Hence, batteries are used until their energy is depleted and then discarded [54]. The inconvenience of replacing batteries due to difficulties of accessibility or cost of maintenance demands the careful use of energy. Energy is used to power the hardware of a wireless sensor node. Energy consumption at a node begins at the sensing unit where a physical phenomena changes, such as temperature or humidity, that can be captured in the form of analogue signal [95]. Then, at the processing unit where energy is consumed by the MCU for data processing, aggregation, and security algorithm computation. The processing unit can manage all other components, and it has a memory for data storage. It consumes far more energy than a sensing unit [95]. Finally, a communication unit requires power to transmit and receive data. The communication unit consumes more energy than the other parts of a node, and is the most significant contributor to battery depletion. According to [96] and [97], data transmission consumes much more energy than data processing. Figure 2.14 shows the current drawn by tmote sky hardware.

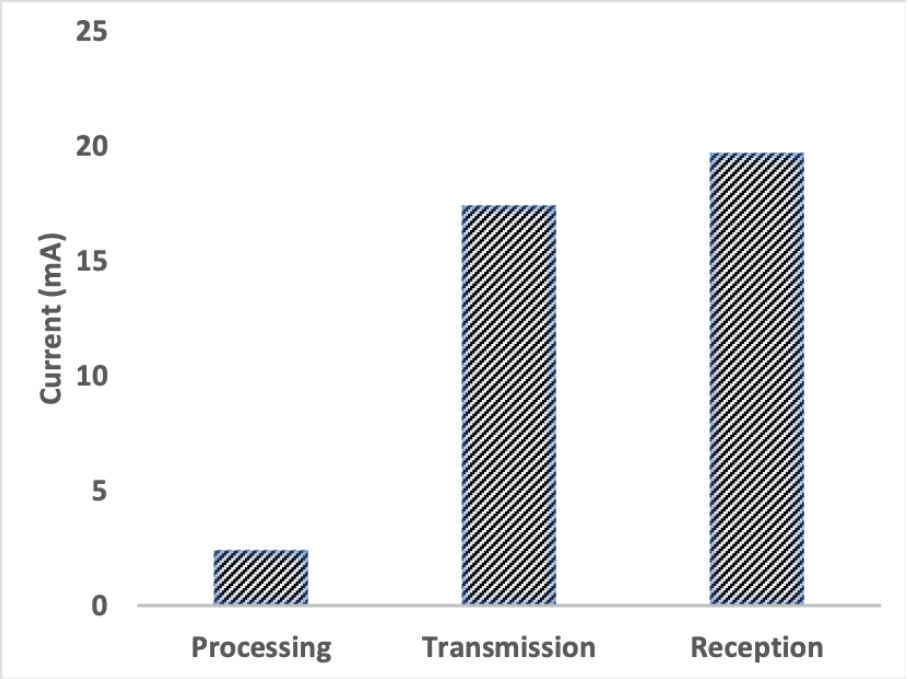


Figure 2.14: Amount of current drawn by tmote Sky [39].

Longer packets have more bits; hence, they take longer to send, and as a result, consume more energy. For example, the energy cost to send one bit by the radio module is the same as processing several thousand instructions through the processing unit [95].

According to [98], the energy needed to transmit 1 KB is 3 J; with the same amount of energy, a typical processor can process 300 million instructions. Radio can run in several different operational modes: transmit, receive, listen and sleep. These operational modes consume energy differently as sown in Figure 2.14. The current drawn in every mode is different from hardware to another. However, it can be taken from the hardware datasheet or measured using a power analyser. Hence, maintaining a transceiver at the lowest level of energy consumption whenever possible is crucial. Duty cycle protocols have been developed to keep the transceiver in the off state as much as possible. Calculating the overall energy consumption in a WSN depends on the sum of energy consumed from sender to destination, including the energy consumed by relay nodes.

The lifetime of a wireless sensor node can be maximised using different techniques. According to [95], optimising energy consumption does not depend solely on the operational modes of a node, it is also affected by the employed security techniques.

2.5.1 Security and Energy Consumption

Energy consumption is a crucial factor when it comes to WSNs, as the entire system is dependent on a limited battery. Many researches in the literature have considered energy when developing a WSN protocol. Recently, more attention has been paid to the security issue in WSNs, an essential factor for many applications [10]. However, most security approaches require a certain amount of node resources for their implementation. Energy consumption has always been a critical issue when designing security solutions. Resource-constrained nodes cannot afford the overhead introduced by some complex security algorithms. There are two main components which are responsible for drawing current from a battery in terms of security [61]. One is a transceiver, which is used to sense the channel and transmit/receive packets. Authentication increases energy consumption by adding extra bytes. The second is the MCU, which is used for computation work. The MCU requires more time to compute operations related to encryption, decryption and MIC calculation. Different security services, encryption algorithms and key lengths consume resources at varying rates [61]. The more complex the encryption algorithm is, the more energy consumed by the MCU.

Security services affect the two components (MCU and Radio) differently. Cryptography mainly affects the MCU, while authentication impacts the transceiver. The extra overhead on the MCU and transceiver is translated to more energy consumption. Accordingly, security and energy cannot be treated separately [89], as each one influences the other. Traditional wired and wireless security solutions, for example, do not take energy limitations into account. Thus, they do not suit wireless sensor nodes [99] [62]. Traditional security solutions are designed with the assumption that the operating environment is known and static [82]. Also, they consider the resources to be relatively rich. However, these assumptions do not apply to WSNs, where the battery and processing

capability are limited. The resource limitations make complex encryption infeasible for use in WSNs [17]. Efficient security algorithms are necessary for resource-constrained nodes [4]. Also, energy-aware solutions are helpful to deactivate unnecessary services in favour of extending network lifetime. In WSNs, it is impractical to assess the strength of security solutions only in terms of security [100]. Other aspects, such as efficiency are important as well.

The relationship between security and energy consumption in WSNs is still an open issue in WSNs. It needs further investigation. However, evaluating the trade-off between security and energy is not straightforward [101]. One reason is that the relationship between QoS, security and energy in WSNs is still ambiguous. Another reason is that WSN is an application-dependent network, so different applications may require different requirements in terms of performance and security.

2.5.2 Security Overhead Calculation

The energy consumption of security processes can be evaluated using the following formula:

$$E_s = I * T * V \tag{2.1}$$

Where E_s represents energy consumption of security overhead in $Joules$, I denotes the current drawn for security in $Amperes$; and T is the time needed for the MCU to compute security operations or the transceiver to transmit the extra bytes in $seconds$. V is the supply voltage in $Volts$. This formula is the abstract for overall security consumption. The energy consumption of both the MCU and transceiver can be measured separately. The energy consumption of the MCU can be calculated as follows:

$$E_{mcu} = I_{mcu} * T_{mcu} * V \tag{2.2}$$

Where E_{mcu} is the energy consumption required by the MCU to compute encryption algorithms; I_{mcu} is the current drawn by the MCU, and T_{mcu} is the time spent by the MCU to perform security tasks. I can be obtained from the hardware data sheet for simulation or can be measured by the power analyser in the real hardware experiments.

Security algorithms keep the MCU active longer to compute the utilised cryptography. The communication overhead can be calculated by observing transceiver states. The transceiver can be active in three modes: transmitting, listening, and receiving. It should be noted that the power consumption in listening mode is equal to the power consumption in receive mode. Hence, it is better to turn the transceiver off as much as possible. Security does not affect radio in listening mode. The current drawn by radio in different modes differs depending on the hardware type. The formula to calculate

transceiver energy consumption is as follows:

$$E_r = E_{rtx} + E_{rrx} \tag{2.3}$$

Where E_r is the energy consumed by the transceiver; E_{rtx} is the energy consumed by the transceiver in transmit mode; and E_{rrx} in receive mode. The energy consumption of the transceiver in each mode can be calculated as follows:

$$E_{r[state]} = I_{r[state]} * T_{r[state]} * V \tag{2.4}$$

Where $E_{r[state]}$ is the energy consumed by transceiver in a specific state [transmit or receive], $I_{r[state]}$ is the current drawn from a battery in specific mode, and $T_{r[state]}$ is the time taken for the transceiver in specific mode. The current withdrawn by the transceiver in each mode can be obtained from the hardware data-sheet. Security affects the transceiver by adding the extra bytes required for authentication. Authentication is provided by appending a MIC to the end of the payloads. A one-way hash function computes the MIC. Packets are valid only if the computed MIC at the destination matches the appended MIC. In this research, the IEEE 802.15.4 security specification is used. It supports eight security levels. Each level is associated with a corresponding length of MIC. Higher security levels use more energy as they are associated with longer MIC length. MIC can be 4, 8 or 16 bytes. The total energy consumption for transmitting one packet between two wireless sensor nodes can be calculated as follows:

$$E_{s-total} = I_{mcu} * T_{mcu} * V + I_{rtx} * T_{rtx} * V + I_{rrx} * T_{rrx} * V \tag{2.5}$$

Where: $E_{s-total}$ is the total energy consumption by security overhead, I_{rtx} the current drawn for transmitting extra bytes for security, I_{rrx} is the current drawn for receiving, T_{rtx} is the time needed for transmitting the extra bytes, and $Trrx$ is time taken to receive them. The time mentioned in this section is that needed for a component to perform a security task.

Communication overheads are associated with message length, and the security cost can be calculated by knowing how many extra bytes have been added to the frame for security. Table 2.6 presents the extra bytes required by each security level.

There are many types of authentication which vary in their energy consumption. CBC(Cipher Block Chaining) and MAC, which we consider MIC in this research, can provide authentication of different lengths. The greater the length is, the higher the assurance of authenticity [102]. The length can be a multiple of 16 and should be between 32 and 128 characters. The National Institute of Standards and Technology[102] recommends using a minimum MIC length of 64 bits, which provides an adequate security level at fair budget [103]. According to [104], the estimated cost of transmitting and receiving one bit on TelosB is 0.72μJ-0.81μJ consecutively; and 0.60μJ-0.67μJ on MICAz. The overall

Table 2.6: Security suites available in 802.15.4 (reproduced from [82]).

Security Suites			
Security Levels	Description	Service	Overhead [bytes]
0	Null	No Security	0
1	AES-CBC-MAC-32	Authentication	4
2	AES-CBC-MAC-64		8
3	AES-CBC-MAC-128		16
4	AES-CTR	Encryption (only)	5
5	AES-CCM-32	Authentication and Encryption	9
6	AES-CCM-64		13
7	AES-CCM-128		21

authentication overhead can be calculated by computing transceiver energy consumption at the source node, the destination node and through relay nodes. AES-CCM-128 adds 128 bits overhead, and according to the cost of 1 bit in [104], the energy consumption would be 92μJ at the source node and 103μJ at the destination node on a TelosB platform. The maximum frame size of IEEE802.15.4 standard is 127 bytes[67]. Thus using 16 bytes for authentication out of 127 represents almost 13% of the frame's total size. This percentage is the cost of authentication using level 7 of the IEEE802.15.4 when transmitting one packet. It is noted that the frame length in many applications such as temperature and humidity is almost 32 bytes, so 16 bytes for authentication is a significant overhead in this case. Also, it is noted that energy consumption differs from one node to another based on hardware components [104]. However, the energy consumption required for security is quite high regardless of the difference in energy consumption between these different platforms. In summary, it is difficult to calculate the overall security cost [101] because it depends on many factors, such as frame length, hardware type [105], utilised encryption, number of relay nodes, and distance between nodes.

2.5.3 Energy-aware Security Solutions

The battery is the primary source of power in resource-constrained nodes. It is the indicator of the node's lifetime. The characteristics of WSN necessities to use the power efficiently, while meeting specific requirements in terms of security and performance. Many solutions have been proposed to optimise the energy consumption in WSNs. One possible technique to extend network lifetime is to use energy-aware solutions. Network lifetime can be increased significantly if the software is developed to operate in an energy-aware fashion [97]. The battery level is used in energy-aware solutions as a criterion to control the node operation. A node can operate differently when the battery level is critical to extending the node's lifetime. For example, the data rate can be reduced if necessary. If a node has sufficient energy, it operates normally, but if the battery

level is critical, it then tends to spend the remaining energy carefully. Hence, trading-off the nodes operational quality for power savings. This technique may prolong the node's lifetime, especially if it is used with the transceiver which consumes high energy compared to the microcontroller. In [106], the author proposes an energy-aware protocol which prioritises data based on a set of system rules. It uses the battery level and the payload content as criterion to make the transmission/forwarding decision. A node with high energy can route and generate data to other nodes. However, when the energy level is critical, the nodes tend to behave selfishly by only routing or creating packets with high-priority in order to extend the node's lifetime.

This optimisation is beneficial in resource-constrained nodes. Energy-aware designs are used in security solutions as well. Recent researches, such as [84] and [89], have used the energy-aware technique to control security. For example, the use of a battery level to switch between encryption algorithms, so it can perform robustly when the battery level is high, and in a limited fashion when the battery level is critical. In [84], the authors propose an energy-aware protocol which can switch between symmetric encryption and asymmetric encryption according to the battery level. They support their solution with an energy harvesting system. Energy-aware security solutions are more efficient than static security; however, this efficiency is achieved at the expense of data security. There should be a way to protect sensitive data when the battery level is low.

Discussion

Energy-aware protocols can be beneficial in reducing the traffic of less critical data and can provide a significant improvement in energy reduction. However, this technique cannot be applied directly to security solutions. Using an energy-aware design to control security according to the battery level is not recommended. The reason being that the system ends up sending sensitive data either that is weakly protected or plain-text when the battery level is low. However, this does not mean that security solutions can neglect the battery. Battery level can be used in security solutions as an auxiliary method to trade-off less important services for energy saving ends, but battery level cannot be used as the core criteria for security decisions. Other techniques such as network performance, threat level and application preference would be a better input to control security. Security should not be overestimated, and it should be appropriate for the data value being transmitted between parties. If security is a concern for an application, then the security level of sensitive data should not be changed according to the battery level. The data protection level should remain the same and be reduced only according to the application preferences and the environmental threat level.

2.6 Context-based Security

This section highlights the role of context-aware solutions and how they could be beneficial in terms of adaptive security. Most existing security solutions cannot detect ever-changing threats. They cannot differentiate the situation where security is prioritized over other parameters, and other cases where energy conservation should be the highest priority [107]. There are many definitions for the term 'context' in the literature, and its usage might drive the meaning. However, a general definition given by [108] is "*any information that can be used to characterize the situation of an entity*". Synonyms for context term are environment and situation [109], which have been used in this research to indicate the same meaning. Context-aware methods could be centralized or distributed. A server is used in the centralized method to collect data from different sensor nodes. This method imposes several challenges regarding the performance and might not be the best choice in the case of the constrained nodes. This is due to the extra overhead generated for management which may affects nodes resource consumption. The distributed approach, which is used in this research, is where an intelligent computation technique is performed on nodes to overcome the performance degrade. The distributed context metrics in sensor nodes can be local or link context. Battery level is an example for the local parameters, and the link-quality between two nodes is an example for link parameters. The context-aware in terms of security means that nodes are sensitive to suspicious activities. Nodes adapt their configuration according to the changing conditions in their situations at run-time.

The dynamic change in the environment makes contextual information a beneficial input for adaptive security. Contextual parameters such as packet loss, latency and other network parameters could be used to enhance the security decision. However, incorporating other parameters which are more indicative to attacks attempts are required. For example, receiving replay packets or packets with the wrong key can be more beneficial for the security decision-maker than parameters such as latency and RSSI. However, the choice of what context parameters to use is depending on the application requirements. Context-based security is defined by [110] as a situation where security solutions use a set of information when making security decision to adapt security services. The context-aware mechanism encompasses gathering data about the node situation and then processing these data to get a deep understanding of the situation. The contextual information is vital for WSN reconfiguration, because it minimizes human interaction [111]. Some recent research has tackled the context at middle-ware layers rather than benefiting from the context information at all layers to enhance the security decision. For example, the context metrics at the application layer are essential to identify the application requirements, and at the network layer to identify the network condition, and at the physical layer to monitor the device's condition. Energy consumption should be reduced as much as possible to extend network availability. Hence, adaptive security

could enhance network lifetime. The following section discusses some of the security solutions presented in the literature.

2.7 Security Solutions

Several studies have been carried out to address the security issue in IoT at the data link layer. This section reviews the existing solutions in the literature. The presented solutions mainly target the trade-off between security, QoS and energy consumption. The approaches in the literature vary from one solution to another. The solutions can be classified broadly into two main groups. The first group separates security from other parameters such as QoS and energy resource. The second group, which is the focus of this research, considers energy and other necessary parameters when designing security solutions. The following is a discussion of both groups.

2.7.1 Static Security

Static security protocols cannot change security during run-time. A fixed level of security is used for transmission regardless of the available resources or the context condition. The following are examples of these protocols: SPINS[112], TinySec [113], SenSec [114], MiniSec [115], IEEE802.15.4 security specification and ContikiSec [57]. Table 2.7 presents the security services provided by some of these solutions.

Table 2.7: Security services provided by different security protocols

Security Services\Security Algorithms	ContikiSec	MiniSec	TinySec	SPINS	IEEE 802.15.4
Confidentiality service	✓	✓	✓	✓	✓
Authentication service	✓	✓	✓	✓	✓
Integrity services	✓	-	✓	✓	✓
Replay protection Service	-	✓	-	✓	✓

The traditional method of providing security considers security as assuring only the privacy, authenticity and integrity of data; whereas business line and operational organizations may consider availability of most important than integrity and confidentiality. This means that we should look at security from the use-case perspective that the application requires. The availability of a sensor network usually depends on the battery, so security solutions should take energy into account. An issue with static security protocols is that they do not adapt their function to changes in resource and threat levels. As a result, security might be overestimated in some cases at the expense of resources and QoS. Static security leads to a situation where the overhead of security becomes higher than potential threats. They use pessimistic hard security techniques during the entire system lifetime, and this degrades system performance and drains the battery quickly,

hence, static security may threaten the availability of a system by exhausting its resources. Although static protocols can provide different levels of security, the required level should be chosen before the network commences, as it cannot be changed during run-time. Consequently, this method may not be the best option to handle security in resource-constrained nodes.

2.7.2 Adaptive Security

Adaptive security refers to the ability of a system to reconfigure itself autonomously. The re-configuration occurs in response to changes in its operatinal environment. Self-adaptation techniques have been used to enhance the efficiency of real-time system operation and optimise its resources. Today's systems should be advanced to include the ability to self-manage, such as self-protection [116]. Resources-constraints demand that wireless sensor nodes perform self-adaptation to solve the trade-off between the conflicting parameters of security and energy. An adaptation technique can be used to handle unpredictable changes either to system resources or in the surrounding environment. Self-protecting systems should have the ability to anticipate, detect, identify and protect a network from attacks [116].

Adaptive optimisation is a complicated task [101]. The reason for this is that the system should satisfy opposing requirements such as security and efficiency. In [84], the authors introduce a scheme which switches between symmetric and asymmetric encryption based on the available energy. Also, they support their scheme with a solar harvesting system to supply nodes with power. The concept of this solution is to reduce data protection according to battery level. Other security countermeasures such as authentication are not considered in this solution. In [82], the authors use a technique where packets are prioritised based on their importance, so the most-important packets are prioritised over other packets when the residual energy is low. Hence, the security decision is made based on the remaining energy. In [89], the authors introduce a routing protocol called QwS-ADOV. This solution controls security according to battery level, and uses three QoS parameters, throughput, latency, and jitter, to choose the appropriate path for network traffic. It is applied to a cluster head. QwS-ADOV does not consider the application preference when making a security decision, which is vital in this type of application-driven network. It reduces security to reflect the available resources. The relationship between security and QoS parameters should be defined and used in the trade-off. In [19], the authors provide a theoretical dynamic security solution which controls security based on the available resources. This solution reduces the security level as soon as the resources are degraded. Prior to developing an adaptive security solution, it is necessary to evaluate the security overhead and understand how this impacts network performance.

Discussion

The security solutions discussed in this section provide a significant contribution to the literature. They use various different techniques to solve the trade-off between security and energy. However, further improvements can be achieved. Some of the security solutions in the literature consider application preferences when utilising security at the expense of available resources and the impact this may have. Hence, they have employed security exclusively from the *application perspective*. Examples of application-driven solutions are static security and adaptive security solutions which use application preference as a criterion to control security. On the contrary, other solutions use resources to control security regardless of the application preferences. Examples for this are security solutions which modify security level based solely on battery level. In this case, sensitive information is vulnerable to attacks when the battery level is low. The solutions which use this type of trade-off employ security from a *resources perspective*; they preserve the resources at the expense of application requirements. Security of specific data should not be reduced, in some applications, even in the presence of scarce resources. According to [117], QoS should not be achieved at the expense of security, and a minimum specified security level should be maintained at all times. If the system cannot transmit at the required minimum security level, then it should not transmit. Security solutions should consider all application requirements, and have the ability to differentiate between a situation where security takes precedence over other elements and cases where QoS and resources are prioritised over security. QoS, resources and security are opposite parameters. Thus, a trade-off between these parameters is necessary for efficient security solutions.

Another essential factor, and a current gap in the literature, is that threat level is not included in the security decision with the other factors. Most of the proposed solutions use battery level to modify security level; however, a more intelligent approach is necessary for security decisions. Battery level can be used in adaptive security to control battery degradation adaptively. This helps to conserve energy for important data to be sent securely in a critical situation. Finally, most of the solutions in the literature use either theoretical or simulated evaluation, therefore, a more practical evaluation using real hardware is needed. Practical evaluation helps to obtain accurate results and overcome some of the implicit issues which can be revealed only using a real network. The first step in developing an adaptive security solution for resource-constrained nodes is to analyse the security overhead and to clearly understand its impact on performance. Such understanding will allow the designer to customise security services based on their impact on network operation.

2.8 Conclusion

This chapter has given an overview of security in WSNs. It discusses the security requirements for WSNs applications; and challenges which limit the use of traditional

wireless security techniques in WSNs. It also discusses the importance of reducing power consumption in resource-constrained nodes to extend network lifetime. Finally, the relationship between security and energy consumption is reviewed. The literature review shows that energy and security are opposite parameters. Thus, increasing security is associated with an increase in energy consumption, and there is a trade-off between security and energy which should be considered in security solutions. Besides, it clarifies how enabling security adds overhead to the microcontroller and transceiver. However, much uncertainty still exists about the relation between security, energy and QoS parameters. What is not yet clear is the exact impact of different security services on energy and performance. This chapter has discussed the existing security solutions at the data-link layer given in the literature, and has presented an overview of static and adaptive security solutions. Most of the proposed studies in adaptive security either use application-driven or resource-driven designs. Each enhancing one parameter at the expense of other parameters. There is ambiguity in some research about the criteria used to adapt security. Hence, more attention should be paid to enhancing security decisions made at run-time. Many of the proposed solutions use an energy-aware design to control security. However, this chapter argues that this technique is inappropriate in the case of security, as it leaves sensitive data vulnerable to attack when battery level is critical. *Until recently, there has been no reliable evidence that adaptive security is practical, since most of the proposed solutions are either theoretical or simulated work.* The relationship between security, energy and QoS should be analysed first to understand the exact security overhead. Hence, the next chapter evaluates the impact of security on resource-constrained nodes using simulation, and then validates the results using real hardware.

Chapter 3

The Security Trade-offs in Resource Constrained Nodes

3.1 Introduction

The previous chapter gives an overview of security in WSNs, and shows how security services impact the microcontroller and transceiver. This impact translates into more energy consumption and a reduction in network performance. However, identifying the exact overhead of each security level of IEEE802.15.4 is essential in this research. The evaluation in this chapter is a stepping-stone for developing an adaptive security solution. It clarifies how security services relate to other parameters such as energy consumption and network performance. This chapter evaluates security services at the data-link layer. It examines each security level of IEEE802.15.4 and demonstrates the most suitable security levels for adaptive security according to their overhead and strength. The evaluation is first executed using simulation, then validated using real hardware.

3.2 Performance Evaluation

Several studies have evaluated the cost of security at the IoT link layer, but the relationship between security levels of IEEE802.15.4 and QoS parameters is still not clear. For instance, [118] have analysed the energy consumption of the following cryptography algorithms: AES, RC5 and RC6. The authors have evaluated the memory requirements for utilising these cipher algorithms. Another study [119] has assessed the cost of AES, RC5 and RC6. It investigated the impact of key size on energy consumption and concluded that RC5 is the most energy-efficient for limited-resource devices. However, none of these studies discusses authentication overhead, which is crucial to security evaluation. The cost of using different encryption block ciphers is evaluated in [83]. The

authors used AES and RC5 on two popular hardware platforms: MicaZ, and TelosB, and assessed the effects of different key sizes on energy cost, as well as the energy cost of different MICs algorithms based on AES-128. IEEE 802.15.4 security specification and its implications on other parameters remains ambiguous, therefore the security levels need to be examined separately. Furthermore, the cost of security over non-secure transmissions is undefined. Most studies present a comparison between different cipher algorithms rather than security over non-secure communication.

This chapter evaluates the security levels of IEEE 802.15.4. It identifies the impact of each security level on energy consumption and QoS parameters. The previous chapter has established that security systems demand that the microcontroller and transceiver run for a longer duration. As a result, the following parameters are expected to be affected: energy, latency, and throughput. The relationship between security cost and packet length is also investigated. Below are the definitions of the parameters which are used in the evaluation:

1. Per-packet energy consumption E: The total energy needed to deliver one packet from source to destination at each security level. This includes the energy consumed by transmission mode E_{tx} and receiving mode E_{rx}. Hence, the total energy consumption of transmitting one packet E can be represented as follows:

$$E = E_{tx} + E_{rx} \tag{3.1}$$

2. Latency (L): The required time to generate and process a packet until it is received by the sink node. This includes the transmission time.

3. Throughput (Thr): The number of packets which are received successfully at the sink during a specific period.

At the end of this chapter the most significant security levels will be selected based on their impact on network performance, particularly in terms of energy consumption. The chosen security levels will be used on the proposed adaptive security solution in the next chapter.

3.2.1 Experiment Setup and Parameters

There are many lightweight operating systems which can be used in wireless sensor nodes. These operating systems provide similar services, but certain characteristics of these operating systems may affect the choice of the developers. Examples of these operating systems are Contiki, RIOT and TinyOS. The Contiki operating system is selected for this evaluation because it is supported by a simulator and radio duty cycle protocols. The Cooja simulator, which comes with Contiki OS, is used to obtain the

results in the initial stage. Then, the evaluation is carried out using real hardware to validate the results obtained by the simulator. CM5000 hardware is chosen because it is supported by Cooja and also because of its popularity in academic research. CM5000 is built based on the open-source Tmote Sky hardware [39]. Table 3.1 shows the parameters used in the evaluation.

Table 3.1: Simulation parameters

Parameter	**Value**
Platform	Tmote Sky
MAC protocol	CSMA
Radio Duty Cycle	ContikiMAC
Payload	24 and 80 byte
TX/RX success ratio	100%
Radio	CC2420
MCU	MSP430

The simulation uses single-hop communication to deliver packets from source to destination. The evaluated scenario consists of two sky nodes; The first is acting as a source node, and the second as the sink.

3.2.2 Security Services Utilised in the Evaluation

The security evaluation in this research uses a link-layer security protocol which supports eight levels, as defined by the IEEE 802.15.4 security specification. Table3.2 shows the security levels of the IEEE802.15.4 standard. The minimum security level is 0, whereby no security mechanism is used, and the highest level is 7, which includes encryption, replay protection, integrity and authentication using AES-128. The security headers added at each security level are shown in Figure 3.1. AES-CTR mode only provides encryption for the payload, hence it supports confidentiality. The length of the key used is 128 bits, as recommended by the IEEE 802.15.4 security specification. This length is fixed at all levels. Authentication can be of various lengths based on the required security strength [4, 8 or 16 bytes].

Auxiliary Security Header(ASH), as shown in Table 3.3, consists of three fields: security control, frame counter, and key identifier. ASH is added to the frame only when the frame control bitfield is set to one[120]. The security control field specifies the utilised security level for a frame, the frame counter is used to provide protection against replay attack, and the key identifier provides information about the key identifier mode.

3.2.3 Accuracy of the Security Overhead Results

There are several factors which affect the accuracy of the evaluation results. For example, the padding mechanism and MAC protocol. ContikiMAC protocol is used as an

Table 3.2: Security suites, reproduced partly from [82]

Security Suites			
SuiteID	Description	Security Services	MIC size (byte)
0	No Security	Null	0
1	AES-CBC-MIC-32	Authentication	4
2	AES-CBC-MIC-64		8
3	AES-CBC-MIC-128		16
4	AES-CTR	Encryption only	0
5	AES-CCM-32	Authentication and encryption	4
6	AES-CCM-64		8
7	AES-CCM-128		16

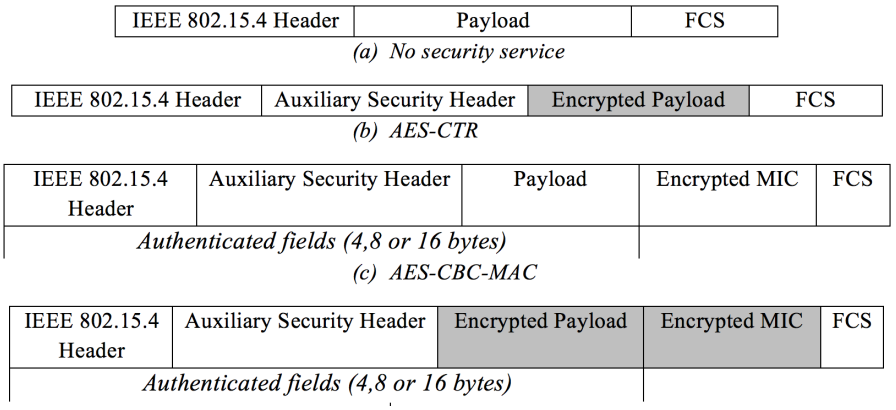


Figure 3.1: Security services frame format

Table 3.3: Auxiliary security header

1 byte	4 byte	0 -9 byte
Security Control	Frame Counter	Key Identifier

RDC protocol. Energy consumption is affected by the utilised RDC. In ContikiMAC protocol, the source node checks the channel before each transmission. If there is no radio transmission in the medium, then it starts to send a full data packet. It continues transmitting the same packet until either a receiver wakes up and acknowledges the message or the maximum number of transmission attempts is reached. The multiple transmission affects security evaluation because the number of AES invocation varies. At the receiver side, a node checks the medium channel periodically for any activity[50]. Figure 3.2 shows the working mechanism of ContikiMAC in unicast transmission mode. Both nodes work as a transmitter and receiver in this figure. Another observation is that ContikiMAC requires a minimum length packet size. The reason for that is to guarantee that the packet does not fall between two Clear Channel Assessment (CCA) [50]. Using the minimum packet size is critical in the broadcast case because there is no

acknowledgement from the receiver. If the packet size is smaller than the minimum size, then a padding mechanism is used to increase the packet size to the minimum. In order to avoid the padding mechanism impacting the experiment results, the packet size will always be larger than the minimum packet size.

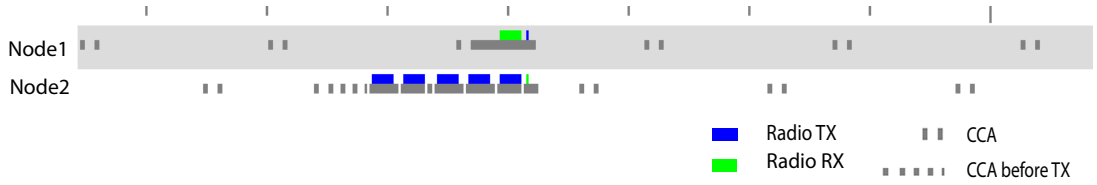


Figure 3.2: ContikiMAC mechanism

As can be seen in Fig 3.2, the radio is turned on and off regularly to save power. This is determined by a parameter in Contiki OS named $NETSTACK\_CONF\_RDC\_CHANNEL\_CHECK\_RATE$. An optimisation phase for ContikiMAC which reduces the number of re-transmissions by keeping track of the receiver wake-up period. The optimisation phase can help the sender transmit just before the receiver wakes up. In order to avoid the impact of re-transmitting the packet and to obtain an accurate result for security overhead, the sink node is kept in an 'on' state at all times, as shown in Figure 3.3. Node number 1 is the transmitter, and node number 2 is the receiver. The dark grey colour indicates that the radio is in an 'on' state. The blue colour represents the transmitting state, while the green colour shows the receiving state.

Figure 3.3: The radio state for both sender and receiver

Figure 3.3 depicts the CCA mechanism. At every transmission, the radio checks the channel to make sure it is clear. To eliminate the impact of CCA on the obtained energy consumption results, CCA is disabled before transmission, as can be seen in Figure 3.4.

Figure 3.4: CCA is disabled before transmission

3.2.4 Simulation Results

3.2.4.1 Energy Consumption Evaluation

In order to obtain the total energy consumption related to security, all components which are affected by security overhead are investigated. Two factors which contribute to the

energy consumed by security are computation and communication overhead. Security computation is the overhead added to the MCU to compute cryptography algorithms. Security algorithms make the MCU run longer to compute them either in the transmission or receiving state. Security communication is the overhead added to the transceiver to transmit the extra byte for authentication. Hence, the total energy consumption of security for transmitting a single packet can be represented as follows:

$$E_{sec-total} = \sum_{k=1}^{n} (E_{sec-compu} + E_{sec-comm}) \tag{3.2}$$

Where n indicates the number of nodes involved in the transmission. $E_{sec-total}$ is the total energy consumption for security, $E_{sec-compu}$ is the energy required for computation overhead, which includes processing the actual packet and cryptography algorithm, and $E_{sec-comm}$ is the energy required for transmitting a packet, which includes transmitting the actual frame and the extra bytes needed for authentication. In the following subsections, energy consumption is investigated for each security level of the IEEE 802.15.4 standard. The results represent the energy consumption of delivering a single packet. It is expressed in μJoule units. The experiment is repeated for each security level. Required security services are added/removed for each frame according to the security level. The cost of transmission without security services will be used as a baseline for comparison since security overhead increases when selecting a higher security level. The Powertrace tool, which is supported by Contiki, is used to record the power consumption in the simulation part. Powertrace [49] calculates the time each component (Radio or MCU) spends in a particular mode(active, transmit, receive, etc). The current drawn by the MCU and radio in different modes should be known in order to estimate the energy consumption. Tmote sky uses CC2420 as a radio driver and MSP430 as a microcontroller. According to the Sky mote datasheet [39], the current drawn by the radio and the micro-controller is shown in Table 3.4

Table 3.4: Typical current consumption for Tmote sky

Component	**Current drawn**
MCU- active state	2400 μA
Radio - transmitting mode	17.4mA
Radio - receiving mode	19.7mA

The objectives of this experiment are as follows:

1. Measure the energy consumed in delivering a single packet at all security levels of IEEE802.15.4 standard.

2. Investigate the impact of frame length on the security overhead.

3. Identify the most suitable IEEE802.15.4 security levels to be used in adaptive security architecture according to their energy consumption and strength.

4. Investigate the impact of power transmission on security overhead.

Scenario 1: Evaluation with a payload length of 24 bytes in transmit mode
First, the energy consumption of transmitting a single packet with 24 bytes without security is measured. The following formula is used to calculate energy consumption in the simulation:

$$E = Energest\_Value * Voltage * Current/RTIMER\_SECOND * runtime \quad (3.3)$$

Where E is the energy consumption of a node's component at a specific mode, $Energest\_Value$ is the difference between two interval times, and $RTIMER\_SECOND$ is the number of ticks per second, which in the current simulation is 32768 ticks/second.

Table 3.5: Energy consumption of transmitting one packet with a payload of 24 bytes at different security levels

Security level	MCU energy consumption (μJ)	Radio energy consumption (μJ)	Total energy consumption (μJ)	Percentage of increased security overhead over non-secure packet (%)
0	9.53	73.28	82.81	-
1	24.01	84.91	108.926	31.54%
2	24.15	92.39	116.54	40.72%
3	24.32	103.546	127.87	54.4%
4	28.95	81.24	110.19	33%
5	28.5	83.8	112.3	35.6%
6	29.11	90.80	119.91	44.8%
7	29.33	103.55	132.88	60.46%

Table 3.5 shows the energy consumed in transmitting a single packet with a 24-byte payload. It includes the energy consumed by both the MCU and transceiver. As can be seen from Table 3.5, the transceiver is the main contributor to energy consumption. The MCU consumption at level 0 constitutes 11.5% of the total energy consumption, and it grows as the code increases in complexity with higher security services. However, at the top security level, it constitutes only 22% of the total consumption. The increased consumption by the MCU at higher security levels is due to AES operation and the processing of extra bytes added by progressive levels of authentication.

On the contrary, the transceiver is responsible for the majority of energy consumption, as shown in Figure 3.5. It can be noted that transceiver consumption at all levels fluctuates between 73.7% and 88.5% of overall packet consumption, which is a significant percentage when compared to MCU consumption. The transceiver is responsible for transmitting packets, and it remains in use longer when dealing with a longer packet length. The impact of the extra bytes is reflected in energy consumption when enabling authentication.

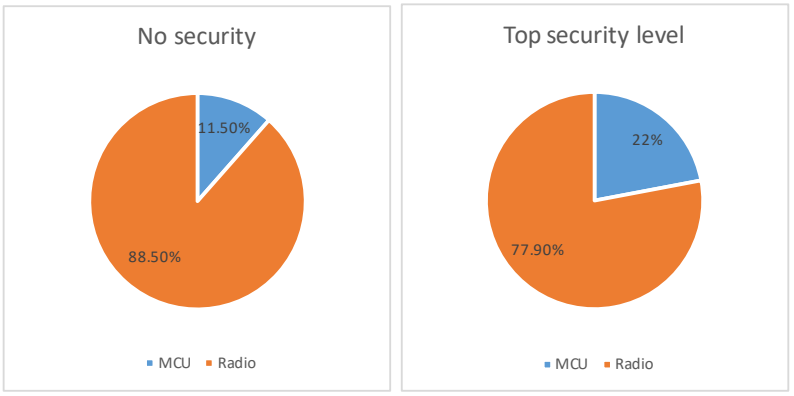


Figure 3.5: Energy consumption: Radio vs MCU, at security level 0 and 7.

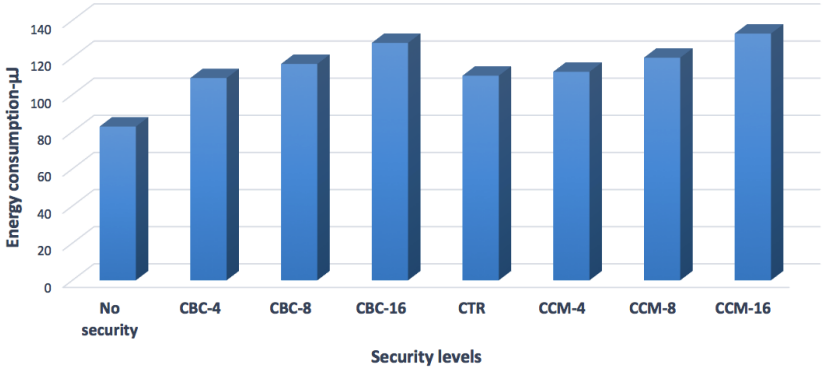


Figure 3.6: Total energy consumed in transmitting one packet with a payload of 24 byte in different security levels

As can be shown in Figure 3.6, the total energy consumption increases gradually from security level 0 to level 3, and from 5 to level 7. The increase in energy is due to the MIC, as every level employs a different MIC length. Security level 4 employs encryption only; therefore, the transceiver consumes less energy compared to the authenticated levels which require extra bytes. There is a slight difference in MCU energy consumption between security levels 1, 2 and 3. Similarly, security levels 4, 5, 6 and 7 show only small changes in MCU energy consumption. However, the increased energy consumption for the MCU at levels 5, 6 and 7 is almost four times the energy consumed by level 0. According to Table 3.5, the percentage increase in security overhead over non-secure communication is high. It can be observed that the minimum security level, level 1, adds a 31% overhead, and the highest security level adds 60%. This significant overhead affects the network lifetime and may shorten it significantly, depending on the security level employed.

Scenario 2: Evaluation with a payload length of 80 bytes in transmit mode
The previous experiment is repeated but with a longer payload. The purpose of this experiment is to investigate the effect of frame length on security overhead. Table 3.6

depicts the security overhead of all security levels using an 80-byte payload. The overall security overhead decreases at all security levels compared to the previous scenario, which uses a 24-byte payload. The reduction in energy consumption is due to the security services remaining the same in both scenarios at all security levels. However, the security overhead becomes more significant when the overall energy consumption is small, and less obvious when the overall energy consumption is large. Both scenarios show significant energy consumption when security is enabled. The results show that the network lifetime may be shortened by security systems.

Table 3.6: Energy consumed in transmitting one packet with a payload of 80 bytes at different security levels

Security level	MCU energy consumption (μJ)	Radio energy consumption (μJ)	Total energy consumption (μJ)	Percentage of increased security overhead over non-secure packet (%)
0	11.93	165.67	177.6	-
1	35.92	176.82	212.74	20%
2	36.2	184.79	220.99	24%
3	36.49	197.5	233.99	31.7%
4	51.26	173.63	224.89	26.62%
5	51.35	176.82	228.17	28.47%
6	51.65	184.8	236.45	33%
7	51.68	197.53	249.21	40.32%

The following security levels are chosen for an adaptive security solution: 0, 4, 6 and 7. The reason these levels have been selected is that the energy consumption at levels 2 and 3, which provides authentication only, is similar to 5 and 6, which provide encryption, integrity and authentication. Hence, the latter are chosen since they provide more protection for the same energy consumption. Level 4 is selected as it provides encryption only at less cost if authentication is not required.

Scenario 3: Evaluating the effect of the transmission power on security cost
The energy consumption of each security level is measured with the maximum and minimum transmission power. Then the security overhead is evaluated. It can be observed in Table 3.7 and Figure 3.7 that transmission power affects security overhead in terms of energy consumption. The overall security cost is higher with low transmission power. The reason for this is that the MCU runs independently and is unaffected by the transmission power change. As a result, the MCU overhead becomes more significant when the transmission power is reduced.

3.2.4.2 Latency Evaluation

In this sub-section, the trade-off between security and latency is evaluated. It is assumed that cryptography would increase the computation time when adding/removing security services. This assumption also applies for communication overhead, as authentication adds extra bytes to the frame. Consequently, a longer frame requires more time for

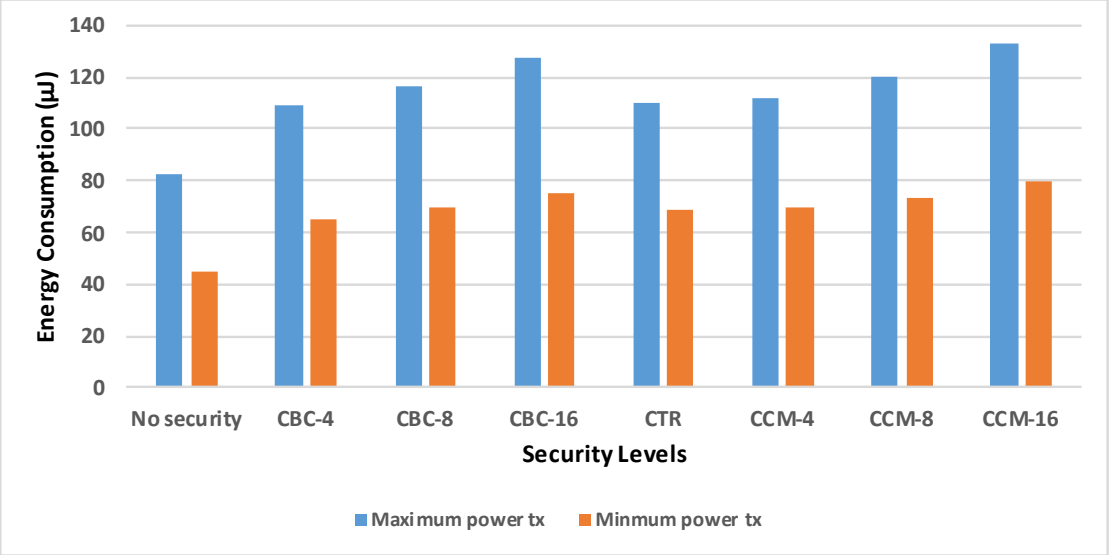


Figure 3.7: Impact of transmission power on energy consumption for all security levels

Table 3.7: Percentage of security cost over non-secure packet transmission with minimum and maximum transmission power

Sec_LVL\TX Power	Overhead at maximum transmission power	Overhead at minimum transmission power
No Security	-	-
CBC-4	31.54%	44.47%
CBC-8	40.72%	52.77%
CBC-16	54.40%	65.05%
CTR	33%	51.40%
CCM-4	35.60%	53.32%
CCM-8	44.80%	62.05%
CCM-16	60.46%	76.10%

transmission. There are many factors which affect the time required to deliver a single packet. Figure 3.8 depicts the process of transmitting a frame with and without security services. It is demonstrated based on the functionality of ContikiMac. ContikiMac waits for an acknowledgement after each transmission to guarantee that a packet has been received at the next hop.

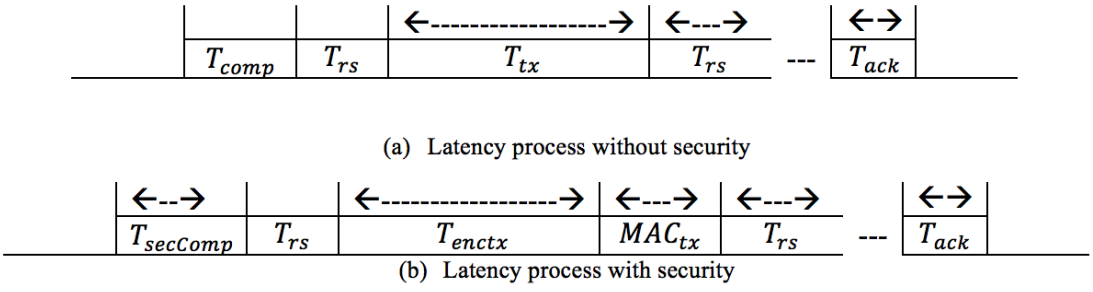


Figure 3.8: Latency process

Latency without security services can be calculated analytically, as follows:

$$Latency = T_{comp} + T_{tx} + T_{rs} + T_{wait} + T_{ack} \tag{3.4}$$

Where, T_{comp} is the time required to process a frame format by the MCU, T_{tx} is the time required to transmit the frame, T_{rs} is the time required for a transceiver to switch from transmit mode to receive mode or from idle to transmission mode, T_{wait} the time needed to receive an acknowledgement from the destination, and T_{ack} the time required to process an acknowledgement frame. Figure 3.8 (b) shows the required overhead when security services are added to the communication. It is assumed that cryptography, integrity and authentication are enabled. The latency with security can be demonstrated mathematically in the following formula:

$$Latency_{sec\_enabled} = T_{seccomp} + T_{enctx} + MIC_{tx} + T_{rs} + T_{wait} + T_{ack} \tag{3.5}$$

Where, $T_{seccomp}$ is similar to T_{comp} but with one or more security services such as cryptography, T_{enctx} is the required time to transmit an encrypted frame. MIC_{tx} is the time it takes to transmit the extra bytes needed for authentication. The time needed for the extra bytes depends on the length of the MIC header(4, 8 or 16 bytes).

Experiment results An experiment is conducted to evaluate the impact of enabling security on packet latency. The latency is obtained by calculating the required time to transmit a packet from one node to another. The time taken to place data into a buffer is included. The utilised RDC protocol in the sensor network affects the latency. Hence, the transceiver at the sink node is kept in an 'on' state to obtain accurate results for latency incurred by security. The experiment is first run without security services, at security level 0, then repeated for each progressive security level. Level 0 is used as a baseline to evaluate the extra time added by each security level. The timestamp at the transmitter is recorded, and then transmitted as a payload to the sink.

Table 3.8 shows the latency performance, in ms, of each security level of the IEEE 802.15.4 standard. As can be seen, latency increases sharply when security services are enabled. For example, the latency is almost $14ms$ without security; this rises dramatically by 200% when authentication is enabled (Level 1).

The results show that all authentication levels [*CBC-4, 8 and 16*] have similar latency once authentication is enabled. It is observed that ContikiMac, which is the RDC protocol used at the source node, is not accurate enough to observe the small differences in packet length. Consequently, the three different MIC lengths, which are used in the IEEE802.15.4 standard, add almost equal latency. CTR encryption increases the latency by almost 250% over level 0. The encryption has more latency than authentication. The

reason for this is that the complex algorithms at CTR mode make the microcontroller run longer. CTR does not affect the transceiver as it does not add extra bytes. Overall, latency increases sharply with authentication, to twice that of the baseline, but it increases with encryption by almost 250% over the baseline. The results indicate that latency is more affected by processing the security overhead than transmitting the extra bytes for authentication.

Table 3.8: Packet delivery latency using simulation

Security Level	Payload	Latency (ms)
No security	24	14
AES-CBC-MAC-32	24	42
AES-CBC-MAC-64	24	42
AES-CBC-MAC-128	24	43
AES-CTR	24	49
AES-CCM-32	24	50
AES-CCM-64	24	50
AES-CCM-128	24	51

3.2.4.3 Throughput Evaluation

The objective of this experiment is to assess whether security services affect the network throughput. As mentioned earlier, network throughput refers to the number of packets received successfully at the destination node over a specific period. Throughput is calculated between two nodes with different security levels for 300 *seconds*. A payload of 24 bytes is used at all levels. ContikiMac is used in both the transmitter and the sink node. Theoretically, security services are expected to affect the number of received packets for two reasons. One reason is that the transceiver remains in an 'on' state longer to transmit the extra bytes required for authentication. Another reason is that the MCU takes more time to process security algorithms. Throughput is evaluated using different data rates, as follows: 2, 8, 16, 32 packets/s. The reason for choosing different data rates is to investigate the impact on throughput under low and high rates.

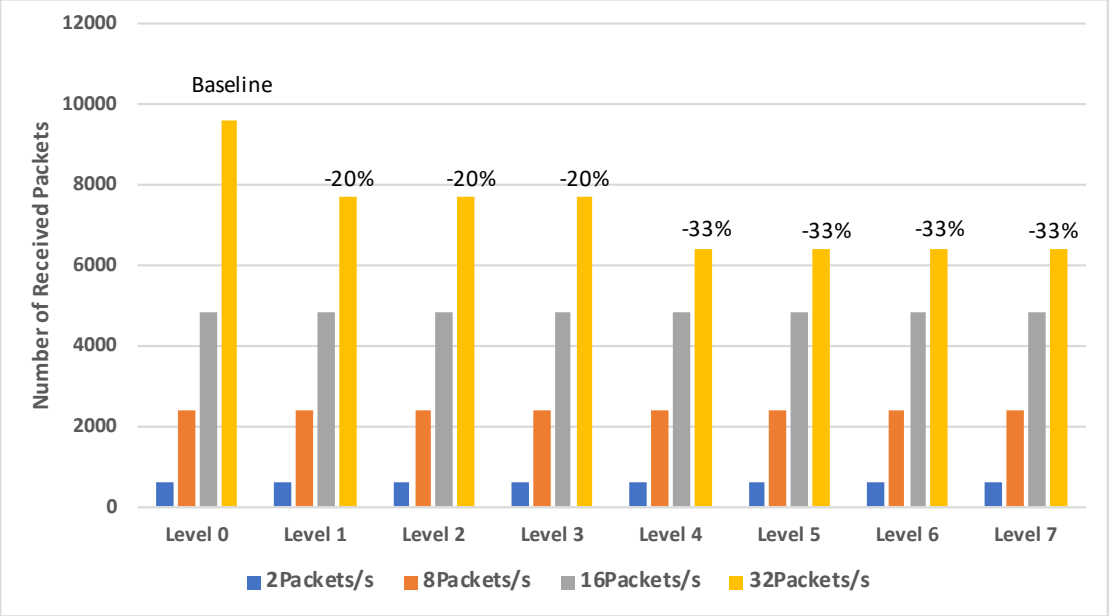


Figure 3.9: Number of packets received over different data rates using two nodes for 300 seconds

Figure 3.9 shows the number of received packets at the sink for all security levels of IEEE802.15.4 using different data rates. The results show that security does not affect throughput when using the following data rates: 2, 8, 16 packets/s. This is because only two nodes are using the bandwidth. However, throughput is affected when the data rate is increased to 32 packets/s. It is reduced by 20% when authentication is enabled. All authentication levels have a similar impact on throughput, with only small variations. It is logically true if we link it to the results of the latency in the previous subsection, where the impact on latency is similar for all authentication levels. However, the impact on throughput is greater when encryption is enabled, yet it is reduced by almost 33% compared to the baseline. It seems that network throughput is not affected by security with a low data rate in a small network where two nodes are used. Network throughput with a larger network is investigated as well. Figure 3.10 shows throughput evaluation for a network with 5 nodes. The data-rate is set to 8 packets/s, which shows no impact on throughput by security in the previous experiment. The results depict a 23% reduction in network throughput when authentication is enabled. Also, it shows that encryption reduces network throughput by almost 31% compared to the baseline. The results show that network size should be taken into consideration when evaluating security impact on network throughput. In general, throughput is reduced by enabling authentication, and even more so when enabling encryption. The impact percentage is affected by the following factors: data rate and network size.

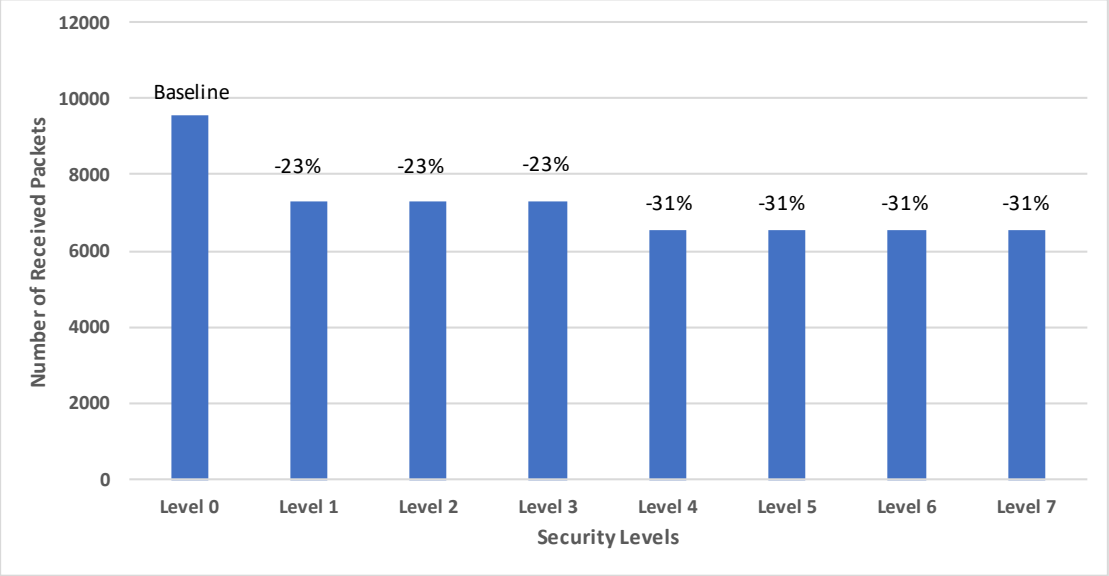


Figure 3.10: Number of packets received using five nodes with eight packets/s data rate over 300 seconds

3.2.5 Hardware Results

3.2.5.1 Energy Consumption

The purpose of this experiment is to evaluate energy consumption at each security level of IEEE 802.15.4 for both the MCU and transceiver using real hardware (MTM-CM5000 hardware in this evaluation). Figure 3.11 shows the energy consumed by transmitting one packet with a 24-byte payload. The evaluation includes the energy needed for receiving an acknowledgement from the sink.

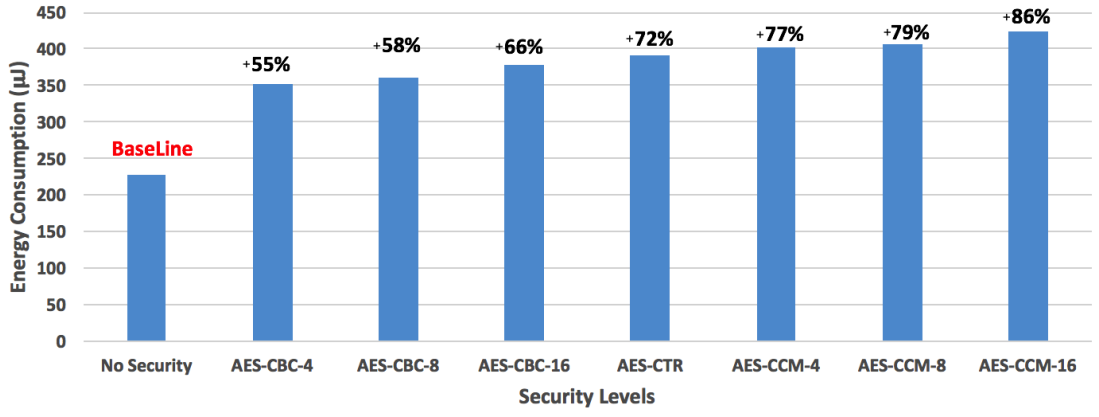


Figure 3.11: Energy consumed in transmitting one packet with a 24-byte payload, including acknowledgement

The results, as depicted in figure 3.11, show that energy consumption varies from level to level, according to the utilised security services. Security level 0 has no protection, and hence will be used as a *baseline* to calculate the extra energy required by the other IEEE802.15.4 security levels. The energy consumption increases progressively from security level 1 to level 7. This increase in energy consumption is due to more complex security services being enabled at higher levels. Energy consumption increases from security level 1 to 3 and from 5 to 7 as a result of increasing authentication length. Security level 4 adds almost 73% over the baseline level. This is due to it enabling encryption which keeps the MCU in *active* mode for longer in order to compute AES encryption. Figure 3.11 shows a significant increase in energy consumption when enabling security. It can be observed that the minimum security level, level 1, adds almost 55%, and the maximum security level, level 7, adds almost 86% over the baseline. This high overhead may significantly affect network lifetime according to the utilised security services.

3.2.5.2 Latency Evaluation

In this experiment, latency is evaluated using MTM-CM5000 hardware. The latency is calculated starting from the time a packet is prepared at the source until an acknowledgement is received from the sink. The experiment is first carried out without security to establish a *baseline* for comparison with other security levels. It is then repeated for all security levels.

Table 3.9: Packet delivery latency using real hardware

Security Level	Payload	Latency (ms)
No security	24	13.1
AES-CBC-MAC-32	24	30
AES-CBC-MAC-64	24	31
AES-CBC-MAC-128	24	32
AES-CTR	24	37
AES-CCM-32	24	38.7
AES-CCM-64	24	39
AES-CCM-128	24	39

The results, as depicted in Table 3.9, show that latency significantly increases when enabling security. Security level 1 increases latency by 129% over the baseline. Level 2 and 3 add almost the same latency as level 1. Hence, all authentication levels add almost the same latency. However, latency increases by almost 184% over the baseline when

encryption is enabled. The last three security levels, which enable both encryption and authentication, increase latency by almost 197%. It is obvious that enabling security increases latency significantly, and this can be noticed at all security levels.

3.2.5.3 Throughput Evaluation

The objective of this experiment is to assess the impact of security on network throughput using MTM-CM5000 hardware. Similar to the simulation, the throughput in this experiment refers to the number of packets received successfully at the destination node over a specific time. Throughput is calculated between two nodes with different security levels for 300 seconds. A payload of 24 bytes is used at all levels. Figure 3.12 shows the number of packets received at the sink for all IEEE802.15.4 security levels using a different data rate.

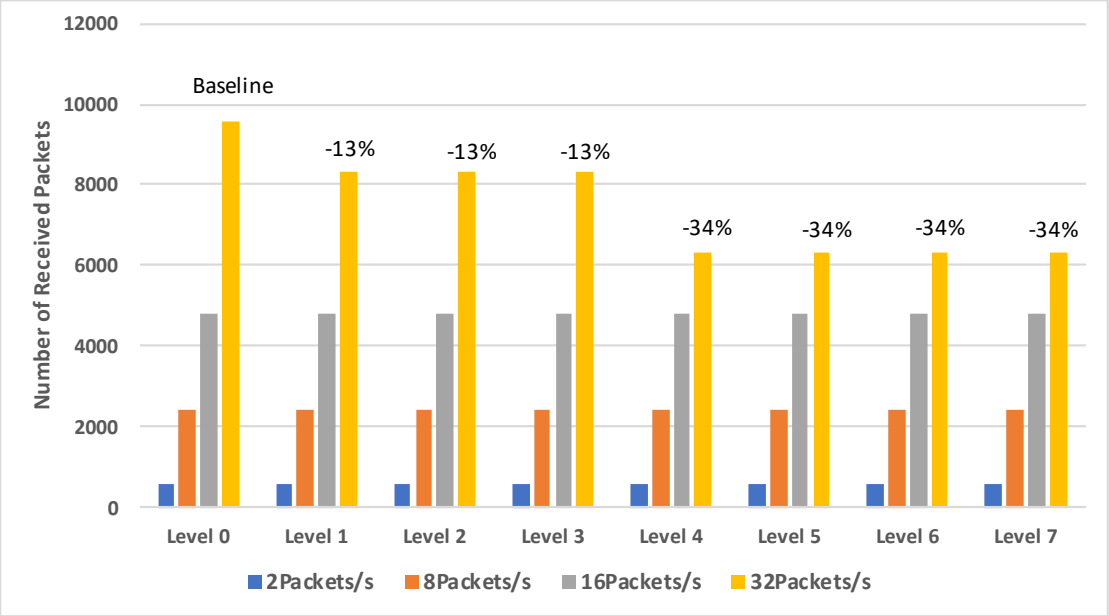


Figure 3.12: Number of packets received using different data rates over 300 seconds

The results show that security does not affect throughput when using the following data rates: 2, 8, 16 packets/s. As explained in the simulation part, this is because only two nodes are using the bandwidth. However, throughput is affected when the data rate is increased to 32 packets/s. It is reduced by 13% when authentication is enabled. All authentication levels have a similar impact on throughput, with only small variations. However, throughput is further reduced when encryption is enabled. It is reduced by almost 34% compared to the baseline. The results show that throughput is not affected by security with a low data rate in a small network where two nodes are used. Network throughput with a larger network is investigated using real hardware as

well. Figure 3.13 shows the throughput results for a network with 5 nodes. The data-rate is set to 8 packets/s, which shows no impact on network throughput by security in the previous experiment where only two nodes are used. The results depict a reduction in network throughput of almost 21% when authentication is enabled. Also, it shows that encryption reduces the network throughput by almost 30% compared to the baseline. In general, throughput is reduced by enabling authentication, and reduced further by enabling encryption. The impact percentage is affected by the following factors: data rate and network size.

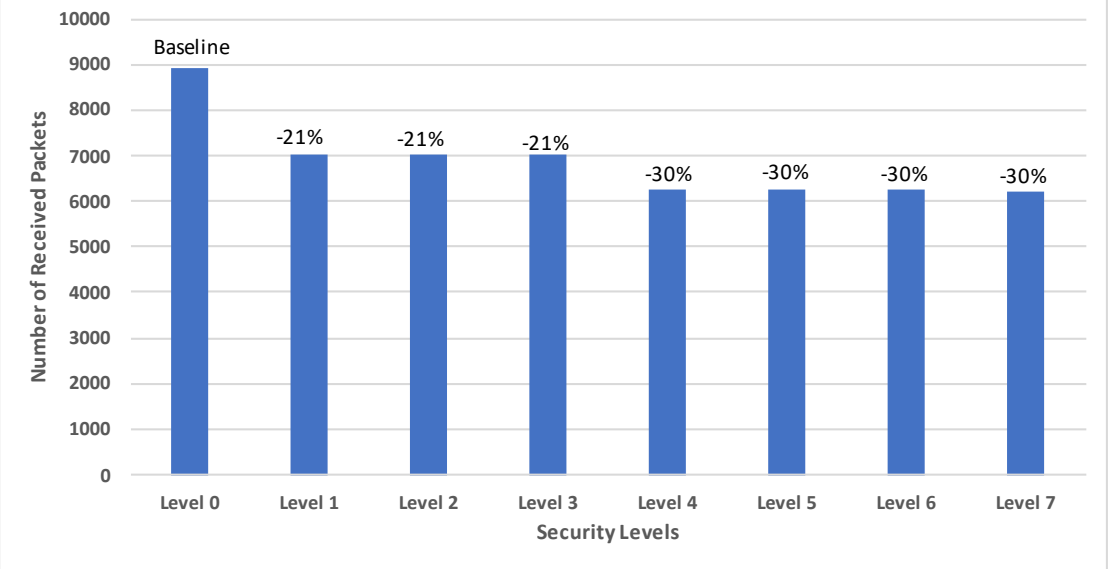


Figure 3.13: Number of packets received using five nodes with eight packets/s data rate over 300 seconds

3.2.5.4 Impact of the Transmission Power on Security Overhead

It can be seen in Table 3.10 that changing the transmission power affects the security overhead. The security cost is higher when using the minimum transmission power by almost 3% except for level 4. There are no extra bytes for security at level 4, and hence there is no difference in energy consumption. The MCU works independently and is therefore unaffected by the transmission power change. The MCU energy consumption becomes significant compared to the overall energy consumption when reducing the transmission power. Hence, security overhead becomes higher with the minimum transmission power and less when we use maximum transmission power.

Table 3.10: Percentage of security overhead over the baseline with minimum and maximum transmission power

Sec_LVL\TX Power	With minimum transmission power	With maximum transmission power
No security	-	-
AES-CBC-MAC-32	57%	53%
AES-CBC-MAC-64	59%	56%
AES-CBC-MAC-128	66%	63%
AES-CTR	74%	74%
AES-CCM-32	77%	74%
AES-CCM-64	80%	76%
AES-CCM-128	87%	85%

3.3 Conclusion

This chapter has discussed the trade-off between security and energy consumption. It has evaluated the security impact at the data-link layer on the following parameters: energy consumption, latency and throughput. Furthermore, it has investigated the effect of transmission power and packet length on the overall security overhead. In general, the results of both simulation and actual hardware show an increase in the overhead when security is enabled. It shows that energy consumption increases progressively as the security level increases. The per-packet evaluation shows that security overhead in terms of energy consumption fluctuates between 31.5% at a minimum level over non-secure packets and 60.4% at the top security level of IEEE802.15.4 specification. The results of this chapter have been encouraging for this thesis to continue and develop adaptive security architecture for IoT embedded nodes. The difference between security levels in terms of energy consumption will be used to minimise the energy consumption of IoT nodes in the next chapter. Also, the results in this chapter benefit network designers and researchers in terms of security overhead, and allow them to choose the level which suits their application requirements.

Chapter 4

Practical Adaptive Security for Resource-Constrained IoT Nodes

4.1 Introduction

The previous chapter established that increasing security also increases energy consumption. Security and energy factors are both critical to wireless sensor nodes. However, they are opposite parameters, as discussed in Chapter Three. It is difficult to identify optimum security settings which can be generalised to all WSNs. This is because a complete understanding of the environment state at the deployment time is challenging and operation factors may change over time. Hence, it may not be a good practice to assume that the application requirements would remain static. This chapter proposes an adaptive security solution for the IoT embedded devices to improve security efficiency. Also, it discusses some possible scenarios where this solution can be utilised.

4.2 Adaptive Security Architecture

This section presents the Practical Adaptive SEcurity architecture for Resource-constrained IoT devices (PASER), as shown in Fig 4.1. PASER is suitable for battery-powered applications which require a trade-off between security and energy consumption. It is motivated by the idea that data may require different security levels and the threat level in the operation location and the device's condition change over time. Enabling adaptation in security solutions will increase the efficiency of node functionality in terms of energy and performance. Some data may not require high security level, such as a regular sensor reading or when the threat level is low. These cases do not indicate incidents which require further action. However, critical sensor data which does require additional work or when an attack attempts are detected should be transmitted at the

appropriate security level. Adaptation is the solution in such a scenario. Adaptation refers to the ability of a device to monitor and optimise its resource usage adaptively at run-time. Adaptation can be more economical in terms of energy and therefore extend network lifetime. This assumption will be tested and validated in this chapter and the following two chapters.

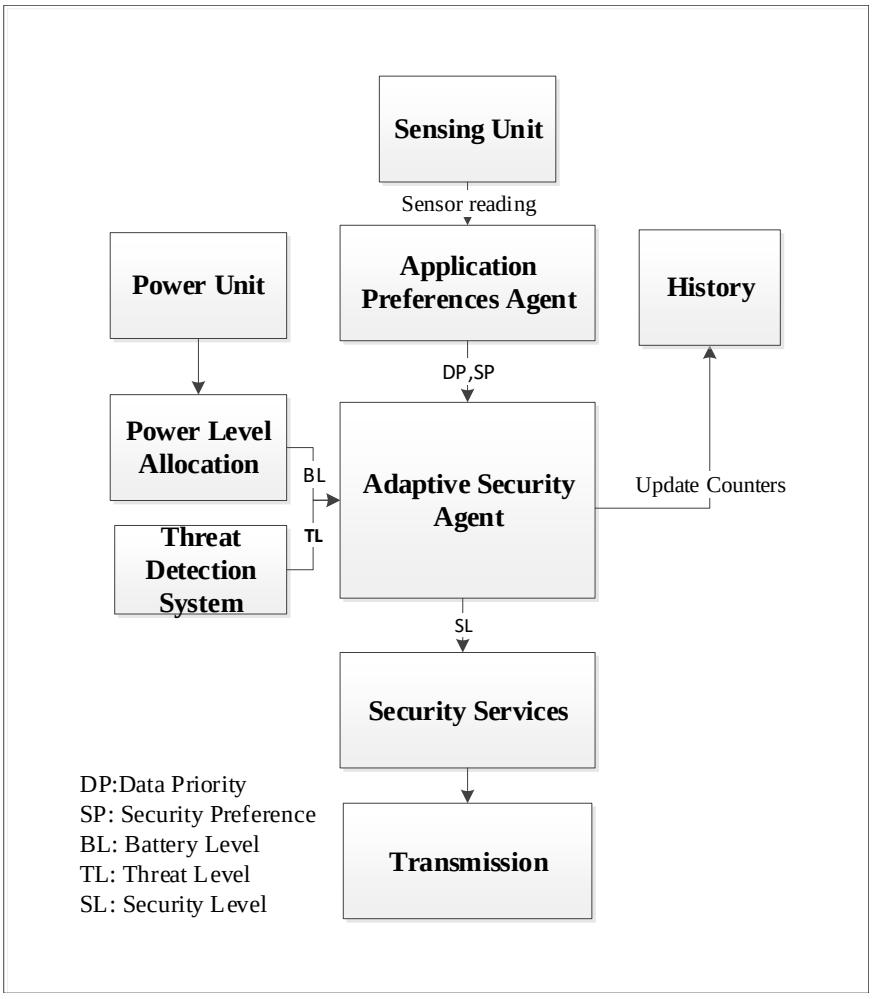


Figure 4.1: PASER architecture

The proposed architecture avoids unnecessary security overhead even if the energy level is high at initial deployment. Unnecessary cost is avoided by switching between the chosen four security levels of IEEE802.15.4 at Chapter Three. If some data does not need a high security or any security at all, then PASER switches to the required security level to extend the network availability, even if the battery level is high. If, however, data require high security then PASER set the security level to the fourth security level which is the highest level. Table 4.1 shows the four supported security levels in PASER mapped to the IEEE802.15.4 security specification.

The security change is carried out according to the application requirements and network conditions at run-time. Each security level is associated with a certain amount of overhead. An increase in security means an increase in resource consumption. The

Table 4.1: Security levels of the IEEE802.15.4 which supported by PASER

Security Levels	Description	PASER Levels
0	No Security	✓
1	AES-CBC-MIC-32	✓
2	AES-CBC-MIC-64	-
3	AES-CBC-MIC-128	-
4	AES-CTR	✓
5	AES-CCM-32	-
6	AES-CCM-64	-
7	AES-CCM-128	✓

PASER uses the variance between security levels energy consumption to extend network lifetime. Some attacks are difficult to detect, such as passive attacks, therefore PASER uses the trade-off with less-critical data. Security of critical data is always maintained, and the the trade-off occurs only with less-critical data. Less-critical data refers to regular data which indicates no incidents. Luckily, most data are less-critical, and hence, the energy-saving would be significant. The proposed approach provides security services on a per-packet basis and as needed. Each node makes security decisions autonomously, which means different nodes may use different security levels according to their threat level and condition. The PASER architecture, as shown in Fig 4.1, consists of three main blocks, and each block performs a specific task, as follows:

1. **Application Preferences Agent(APA)**: One of the essential factors to consider when making the security decision in WSNs is the application requirements [17]. Hence, this block specifies the application preferences in terms of data security and priority. Various classes are offered based on data's security and mission importance to the application. This classification can be used to structure a set of particular security services. It uses four types of security class based on IEEE802.15.4, as shown in Table 4.1. These security levels are: no security, confidentiality, authentication-integrity, or all security services with reply protection. The data is categorised in terms of their priority into two types: non-critical, and critical. Each packet will be prioritised and given an initial security level according to the application requirements. Priority means that non-critical data may be filtered when the battery level is low. In general, the relationship between data priority and its level of security is a direct relationship. The more critical the data is, the higher the level of protection. However, there are cases where data is critical but does not require encryption. It still needs to be authenticated and validated to assure its authenticity and integrity, but confidentiality is not essential. Therefore, a security level value is used alongside the priority value to provide an accurate classification. The data classification is chosen based on the application's preferences.

2. **Adaptive Security Agent (ASA)**: This block determines the security level for each packet. Each security level is associated with a set of security services such as encryption and authentication. ASA is responsible for assigning the security level adaptively in accordance with the following factors:

   (a) **Security Preference**: this value represents the application preference in terms of security and is obtained from the APA. Every application should initially determine data importance and data security level. The security preference is essential in the case of a passive attack. A passive attack is difficult to detect from network behaviour in some cases, as the attacker only listens to the wireless network broadcast. Hence, the security preference is employed to help protect the confidentiality of sensitive data. *The trade-off occurs with less-critical data.* In contrast, an active attack can be detected by observing network behaviour, as to be explained below. The combination of both protection methods is an effective solution that will strengthen the security layers against any attack.

   (b) **Threats Detection System**: The ASA also utilises a Threats Detection System (TDS) input which can react to situational changes and then adapt security level accordingly. A situation, sometimes named context or environment [109], can be "*any information that can be used to characterize the situation of an entity*" [108]. Most existing security solutions does not consider threat level input in their design. Hence, they cannot differentiate the situation where security is prioritised over other parameters, and other cases where energy conservation should be prioritised [107]. It is necessary to use parameters which are effective in terms of security. For example, making security solutions aware of when a node receives a replay packet. Understanding such situations may be more beneficial for the security decision-maker than parameters such as delay or throughput, which may not always be related to security. The dynamic change in the environment makes situational information a beneficial input for the PASER architecture.

   The TDS component contains a table which has two values: 0 and 1. The '0' value indicates that the environment is safe and no threat is detected. The '1' value indicates that a threat is detected. ASA then uses the TDS value to make conditional decisions which help to minimise human interaction. The development of TDS is in its initial stage. Currently, PASER has only the table, so that existing threat detection systems in the literature can be easily adapted with PASER to feed that table. It is difficult to design a secure system that an attacker absolutely cannot infiltrate [121], hence the DTS works as a second defensive line for the system.

ASA also uses a technique called Battery Degradation Control(BDC) to control battery degradation before passing the message to the security services component. ASA accomplishes this using the battery level. The BDC functions by utilising a set of

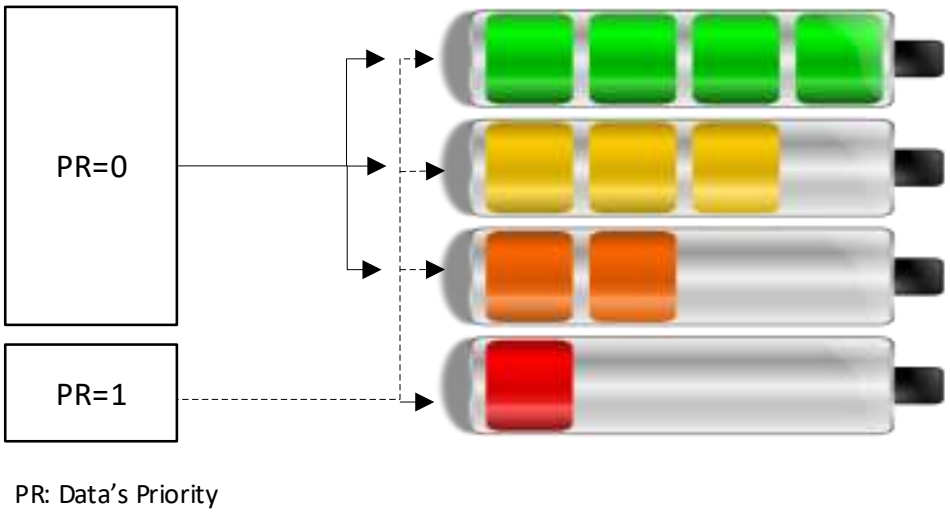


Figure 4.2: Battery degradation control.

rules to control the degradation of sensor node batteries. ***This technique does not reduce security according to the battery level as proposed in the literature, but instead, it filters less critical data when the battery level is low***. The reason this method used is to protect critical data from being exposed to an attack when the battery level is low. The non-critical data is traded-off with the remaining energy. Critical data is the only type which is allowed to be transmitted when the energy level drops below a pre-defined threshold. This technique extends network availability. Fig 4.2 demonstrates the BDC technique, where 0 means non-critical data and 1 is critical data. Non-critical data is discarded to preserve energy for processing data of a higher priority when the battery level is low. Then, a history counters are updated to be used by the application if required. Data will be sent only if its priority is high enough to justify the resources it requires. Algorithm 4.3 is a pseudo-code for the BDC technique.

3. **Security Services**: This component is responsible for adding/removing security services adaptively in accordance with the security decision made at the ASA agent.

4.3 Application Assumptions and Possible Scenario

This research is not looking to provide the best level for the trade-off between security and energy, as the best trade-off may not be achieved at the low layers. ***Instead, this research proposes a way for this trade-off to be controlled dynamically from different layers during the run-time***. Other protocols which deal with routing

```
while (1) do
  Sensor_Reading ← Get_Sensor_Reading()
  Priority_Non−critical = 0
  Priority_Critical = 1
  if Sensor_Reading ≥ 30 then
    Packet_pr = Priority_Critical
  else
    Packet_pr = Priority_Non−critical
  end if
  Energy_res ← Get_Residual_Energy()
  if Packet_pr = Priority_Critical then
    TX
  else
    if Energy_res ≥ 25% then
      TX
    else
      Discard message
      Update_history()
    end if
  end if
end while
```

Figure 4.3: The BDC technique pseudo-code

issues can be employed easily with this solution. The solution is employed at the data-link layer, so routing protocols can work on top of it. Fig 4.4 shows the network layers of IEEE 802.15.4.

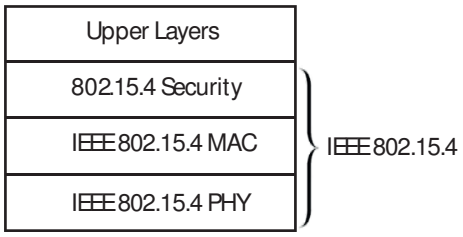


Figure 4.4: IEEE 802.15.4 Layers

Environmental surveillance applications are one area where the PASER architecture can be utilised. One example is a forest-fire detection application, as shown in Fig 4.5. The focus of this application is not the confidentiality, but the authentication of data. These requirements are due to the importance of preventing false alarm reporting from exhausting the resources of a fire control centre. The false alarm may be initiated by an adversary to mislead the control centre and consume their resources. High security protection may be needed only in the occurrence of an incident.

The proposed solution can satisfy these requirements. Any incident which requires action should be reported securely, and less-critical data can be sent unsecured or with

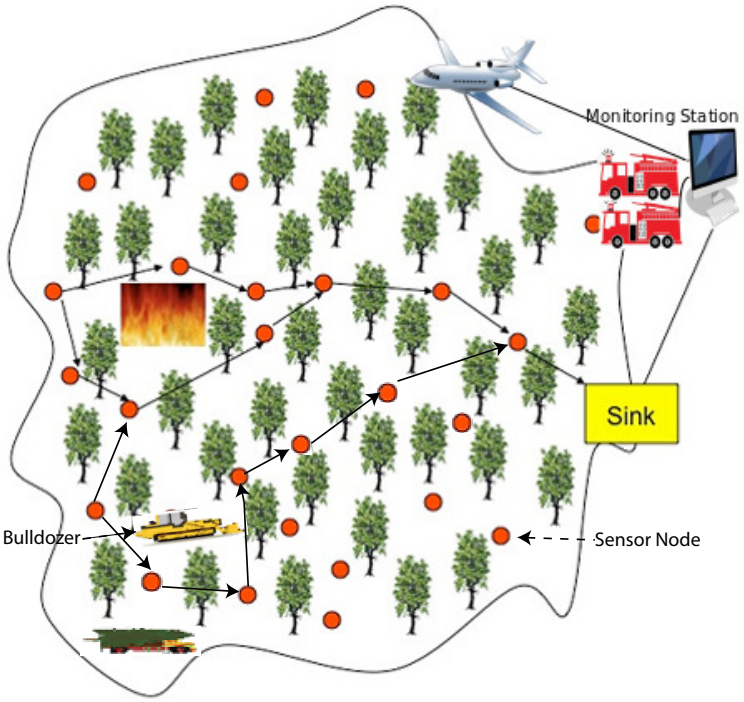


Figure 4.5: Fire detection and deforestation application.

minimum security. Any incident report which received not in accordance with the application policy should be treated as a possible attack. For example, receiving critical data, which is supposed to be highly secure, with no security should be treated as a possible attack. Table4.2 presents an example of data classification at the application layer for the fire detection application. In this scenario, the process starts with the detection of an incident within a forest. Data is then classified based on the two previously mentioned criteria: data importance, and data security. Regular events may be sent with no security services, whereas alerts which indicate an incident would be sent securely to the sink.

Another example of where this solution could be utilised, as shown in Fig 4.5, is a deforestation application. In this example, sensors are used to detect unusual movement of trucks or bulldozers. The network lifetime, in this example, maybe more important than the security of data. However, at the same time, the system should be secure enough to prevent adversaries from reporting false incidents. Any incorrect information may cause the monitoring centre to send vehicles and equipment to a location where no real incident has taken place. If the system does not protect against such an attack, then the whole system may become useless. In the deforestation application, vibration sensors are used to monitor movement in a particular area. Sensors are scattered over a target area, as shown in Fig 4.5. Sensor reading is checked against pre-defined application rules, then the application preferences in terms security and priority are assigned.

Data is then passed to ASA, where a security level is selected according to the application

Table 4.2: Data's security level.

Values	Security Levels	Rules
00	No Security	Temp. Less than 25°C
01	Lightweight Authentication & Integrity[4 byte]	Temp between 26 & 30°C
10	Encryption	Temp. between 31& 36°C
11	Encryption, Authentication, Integrity [16 byte] and Replay Protection	Over 36°C

preference and the TDS input value. The battery level is then checked. If the data's priority is permitted at the current battery level, the required security services are added, and the data is passed for transmission. If the energy policy does not allow the transmission of non-critical data, then data will be discarded. The history will be updated as well, so the application can be aware of filtered packets. An agile degradation method helps save energy for critical data in cases where the battery level is low. As a result, it helps to extend the network lifetime. The mechanism of ASA transmission is demonstrated in Algorithm 4.6. At the receiver side, the security level value is first checked and then decapsulation is accomplished accordingly. If the packet is unsecured, then it is decapsulated directly; otherwise, the security services, such as decryption and authentication, take place first. If, however, a suspicious activity is detected, such as receiving an unauthenticated packet, then the packet is neglected. The history is updated so that the application can use this information in the TDS. Algorithm 4.7 shows the PASER reception mechanism.

The environments in which these sensors are deployed are usually volatile, therefore replacing a sensor battery in such an environment might be costly or even impossible. PASER can provide a solution to this situation.

4.4 Security Assumption

This research considers the security of WSNs at the data-link layer from one sensor to another. A scenario where nodes send packets periodically to the sink is considered. It is assumed that the key is pre-configured on the member nodes. There are four security levels, as shown in Fig4.8, and every security level has different security services. Security level is based on the IEEE 802.15.4 security specification. The security services include cryptography, authentication, the integrity of data and replay protection. Fig4.9 shows the frame format when security is enabled. The security decision utilised by ASA can be extended to include more parameters according to the application requirements. For example, if the location of a sensor node is crucial, then it is added to the security conditions. A friendly environment where physical access to space is controlled demands less security than a deployment in a public area where the potential for network attack is greater. Each factor can be given different weight if necessary when making the

```
while (1) do
  Sensor_Reading ← Get_Sensor_Reading()
  if Sensor_Reading < 25 then
    SecurityPreference = Level0
  else
    if 26 ≤ Sensor_Reading ≤ 30 then
      SecurityPreference = Level1
    else
      if 31 ≤ Sensor_Reading ≤ 36 then
        SecurityPreference = Level4
      else
        SecurityPreference = Level7
      end if
    end if
  end if
  Threat_level ← {0, 1}
  if Threat_level = 1 then
    SecurityLevel = Level7
  else
    SecurityLevel = SecurityPreference
  end if
  switch (SecurityLevel)
  case 0
    Send()
  case 1
    Authenticate_[4byte]()
    send()
  case 4
    Encrypt_AES()
    send()
  default:
    Encrypt_AES()
    authenticate_[16byte]()
    send()
  end switch
end while
```

Figure 4.6: PASER: data transmission mechanism

security decision. For example, the application preference and threat level inputs may be given more weight, in the security decision, than the location factor. The threat level may increase the utilised security level up to three levels in the presence of suspicious activities. However, the weight values should be given according to the application circumstances.

```
SecurityLevel = ReceivedSecurityLevel
switch (SecurityLevel)
case 0
  DecapsulateFrame()
case 1
  MIC ← Deauthenticate[4byte]()
  if (compare_authentication(MIC)) then
    DecapsulateFrame()
  else
    Discard Packet
    UpdateHistory()
  end if
case 4
  Decrypt_AES()
  DecapsulateFrame()
default:
  MIC ← Deauthenticate[16byte]()
  if (compare_authentication(MIC)) then
    Decrypt_AES()
    DecapsulateFrame()
  else
    Discard Packet
    UpdateHistory()
  end if
end switch
```

Figure 4.7: PASER: data reception mechanism

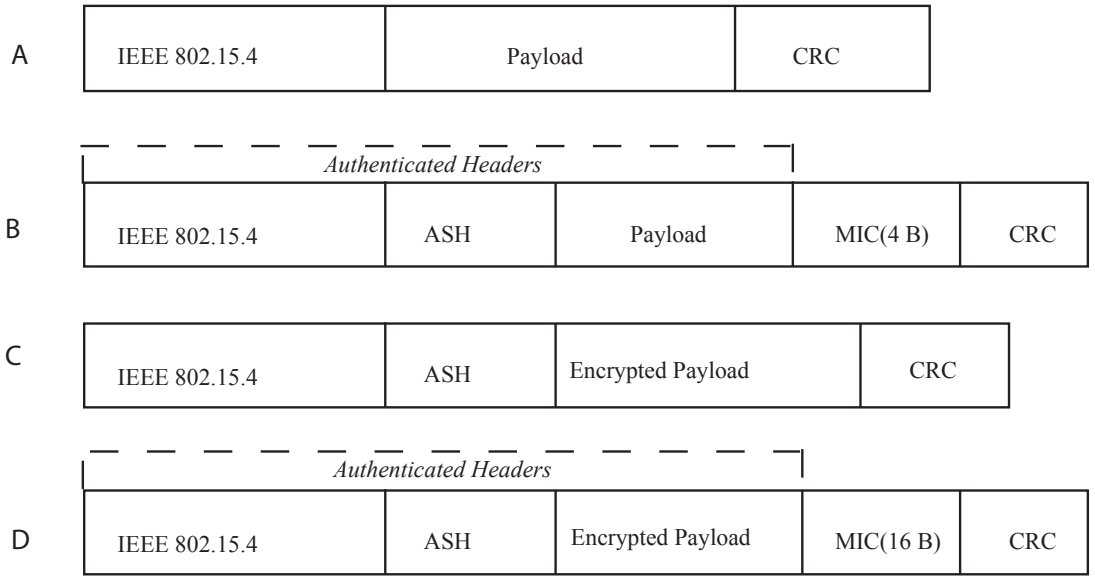


Figure 4.8: Frame format of security levels

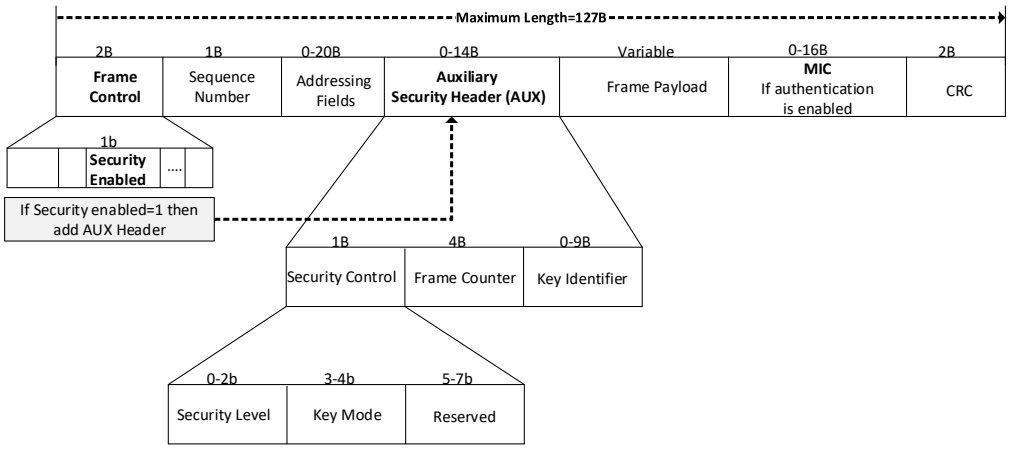


Figure 4.9: The 802.15.4 frame format

4.5 Conclusion

The aim of this chapter is to provide a conceptual theoretical architecture for adaptive security in WSNs. This architecture uses adaptivity to manage the existing trade-off between security and energy. PASER enables only the required security services. It trades-off security with the energy needed for less critical data in resource-constrained nodes. PASER supports four different security levels. Each security level provides a different degree of protection. Higher security levels are associated with an increase in energy consumption. The security decision is taken at ASA. ASA adaptively switches from one security level to another level at run-time. The selected security level is chosen based on criteria which justify the resources it requires. The decision criteria include the application preferences and the environmental threat level input. One major issue in early security research concerned the security decision. In contrast to solutions in the literature, PASER does not reduce the security of sensitive data according to the battery level. It uses a BDC to extend network availability. The PASER architecture can be used when energy consumption is a challenge to security, which is the case in some IoT applications. The energy saving which can be achieved by PASER is evaluated in the following chapter. This chapter answers research question 2.

Chapter 5

Experimental Design and setup

This chapter presents the experiments setup which are used in the evaluation of PASER architecture. These settings are used on both simulated and real hardware experiments. Five MTM-CM5000 nodes running Contiki OS have been used in the experiments. MTM-CM5000 is built based on the open source sky layout. Sky hardware [39] features a Texas Instruments MSP430 microcontroller and CC2420 transceiver. Node 1 is used as a sink node, and the other nodes work as end nodes. All nodes use ContikiMac for their duty cycle. Figure 5.1 shows the experiments layout.

Figure 5.1: Experiments layout(all nodes are in one collision domain)

The evaluation starts by clicking the start button in Cooja simulation. Hence, all nodes start at the same. However, this is not straightforward in real hardware because each node has its own reset button. Pressing the reset button separately for each node may affect the results, as there will be some delay. Therefore, all nodes are connected to one reset button to guarantee that they start at the same time. The button is connected to pin 6 (reset pin) of the 6-pin expansion connector, and pin 9 (ground pin) of the

10-pin expansion connector for each MTM-CM5000. Hence, the evaluation starts when the button is pushed.

A security level is assigned to each packet according to the ASA module, and this value should be part of the header for the receiver to decapsulate the packet. The security levels used in PASER architecture are 0, 1, 4 and 7 of the IEEE802.15.4 security specification. These levels are chosen based on the evaluation results on Chapter Three. The security mechanism works as follows: when protection is required, the security-enabled bit in the header is set to 1 at the source node. Then the necessary security headers are added according to the security level. If the security-enabled bit is set to 1, the ASH header is then added to the frame. The payload is encrypted first if required, and then the packet is authenticated if needed. The destination node first checks the security-enabled bit, and then decapsulate the packet accordingly. If packet authenticity verification is required, then a comparison is made between the generated authentication code at the destination and the received code. If the two codes are matched, the packet is then decrypted if required, and the database is updated. If the received packet is unauthenticated, replayed or received with the wrong security level, then the packet is discarded, and the database is updated. The evaluation metrics are discussed in the following subsections:

The evaluation has used the following architectures: static-security using level 7 of IEEE802.15.4, no-security, and PASER. The no-security architecture is used as a *baseline* for comparison in this evaluation. PASER changes between four different security levels at run-time. The application preferences regarding security in this evaluation are shown in Table5.1. The values used in the conditions in this scenario are arbitrary values representing a fire detection system as an example.

Table 5.1: Application preferences

Security Level	**Rules**
No security	Reading Less than 30°C
Security Level 1	Reading between or equal to 30 & 40°C
Security Level 4	Reading greater than 40 and less than or equal to 45°C
Security Level 7	Reading over 45°C

A random function is used to generate the payload in the evaluation. The random function is initialised with seeds to guarantee that it generates the same sequence numbers as the other architectures for comparison purposes. Figure 5.2 shows the standard normal distribution of the generated payloads using one of the end nodes in this evaluation. It shows that data is scattered in a wider shape between two predefined values. Most of the generated data in this scenario require different security levels, while less than half of the data requires no protection.

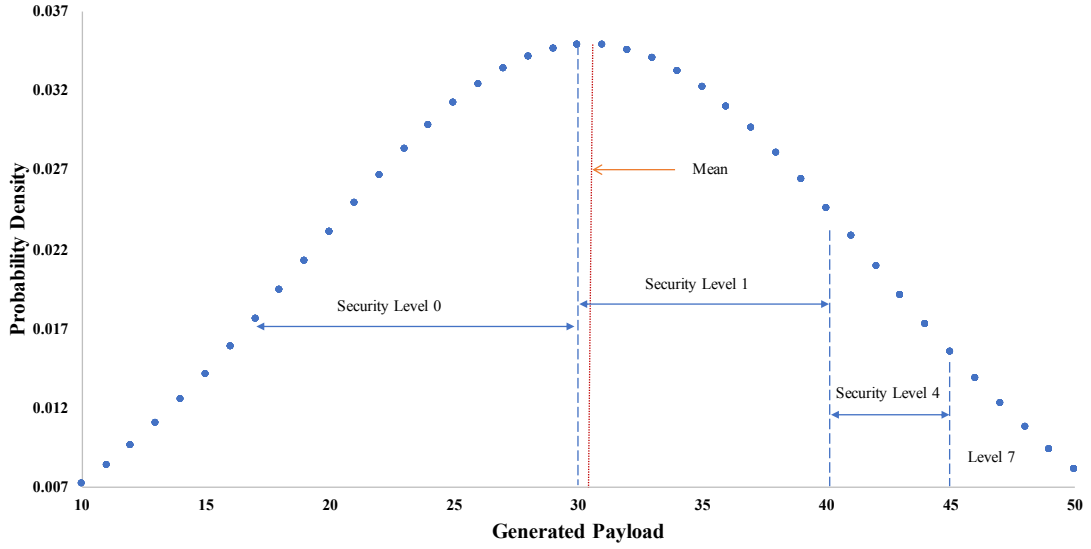


Figure 5.2: The probability density of the generated payloads.

The evaluation focuses on the following five metrics: energy consumption, network lifetime, throughput, latency and memory requirements. These metrics are affected directly by enabling security, as shown in Chapter Three. The following subsections are description for the used metric in the evaluation.

5.0.1 Energy Consumption

In the energy evaluation the experiment runs for five minutes with a data rate of two packets per second, and a channel check rate of $64Hz$. The reason for choosing five minutes is the need for comparision between simulation and real hardware, as the power analyser produces quite large data size when run longer than 5 minutes. The power consumption is recorded on all nodes, including the sink node. In the simulation part, the Powertrace tool, which is supported in Contiki OS, is used to record the power consumption. Powertrace [49] calculates the time each component spends in a particular mode(active, transmit, receive, etc). In real hardware experiments the power analyzer is used to record the power the consumption.

5.0.2 Network Lifetime

The network lifetime, in this evaluation, is defined according to [122] and [123], as the amount of time elapsed since the start of the network until the first node runs out of power. The purpose of network lifetime investigation is to to determine whether PASER increases network lifetime or not. The experiment is carried out using the PASER architecture. It is repeated five times, and the average is calculated. The same evaluation is repeated as well, with *no-security* and *static-security* architectures

for comparison purposes. All nodes start with the same battery capacity of 500 mAh. The lifetime of a network finishes as soon as the first node runs out of battery regardless of the remaining energy in the other nodes. The total remaining battery in the other nodes is reported as well.

5.0.2.1 Battery Degradation Control

Battery degradation control, which is part of the PASER architecture, enables wireless sensor nodes to last longer through the employment of selective transmission when the battery level is critical. Transmitted packets are selected according to their data-importance. The degradation control in this evaluation starts when the battery level drops below 25% of the battery capacity. The value of the predefined threshold can be different according to the application requirements and the network purpose.

5.0.3 Network Throughput

Security increases the time the microcontroller needs to process packets, and the time required by the transceiver to transmit and receive packets. Hence, PASER may increase network throughput due to the different security levels which are used in the transmission. Network throughput in this evaluation refers to the number of packets which are received successfully at the sink node. The experiments is executed using different data rates, as follows: 2,4,6,8 packets per second. The reason various data rates are used is to explore the impact of PASER on network throughput under different conditions. The experiments are repeated using no-security and static-security architectures for comparison purposes.

5.0.4 Latency

This section discusses the latency of delivering packets to the sink using the following three architectures: static-security, PASER and no-security architecture. The per-packet latency between two nodes is recorded using the following formula:

$$Packet\ Latency = Data\ receiving\ time - data\ transmission\ time. \tag{5.1}$$

Another experiment is executed to determine the total latency for the PASER architecture and compare it with the other architectures. The transmission timestamp at the source is recorded and forwarded to the sink as a payload. Then, at the destination, the latency is calculated for all nodes by subtracting the transmitted time at the source from

the receiving time at the sink for each node. The test runs for five minutes. The total latency for the whole network is the sum latency for all nodes as the following formula:

$$Total\ Latency = Total\ Latency + current\ Packet\ Latency. \tag{5.2}$$

5.0.5 Memory Footprint Evaluation

The memory required by the security modules might become an issue in constrained resource devices such as wireless sensor nodes. Evaluating memory usage is crucial, as it provides information on whether PASER can be allocated on the MTM-CM5000 memory. MTM-CM5000 has $48k$ ROM. The size of the PASER architecture can be obtained using the command '*size*' followed by the compiled file name on Linux OS.

Chapter 6

Evaluation and Validation Results

6.1 Introduction

The previous chapter has presented the settings which are used to obtain the results in this chapter. Until recently, there has been no reliable evidence that adaptive security is practical or energy-efficient. The reason being that most of the proposed architectures are either theoretical or have been examined through simulation only. This chapter extends the thesis to answer research questions number 3 and 4. It investigates whether the PASER architecture can be implemented in wireless sensor nodes, and if so, what degree of energy-saving does it achieve? PASER will first be implemented and evaluated through simulation, and then validated using real hardware.

6.2 Simulation Results

6.2.1 Energy Consumption

In this arbitrary scenario, where the payload is changed in a pseudo-random manner, the results show that PASER saves nearly 10% of energy compared to the fixed security architecture, as can be seen in Figure 6.1. The saving percentage can be higher in a safer environment and less in an unfriendly environment. We predict that the saving would be greater in the real scenario. The reason being that the probability of incidents occurring in a real scenario is low compared to the random sensor reading used in this evaluation. In this case study, the change in the sensor reading does not occur in a matter of seconds. Hence, less secure data are transmitted, and as a result, less energy is consumed. The percentage of energy-saving may be greater with other duty cycle protocols. The reason being that ContikiMac is an asynchronous protocol which sends the whole packet repeatedly until the sink wakes up and replies with an acknowledgement. The impact of

the duty cycle on the results can be avoided by keeping the sink node in the 'on' state, so the packet is received at the first attempt. In this case, the sink is not included in the evaluation as it in the 'on' state all the time. The energy saved in this situation increases to almost 12% using the same evaluated scenario. However, it is expected to be even more if the sink is included due to the cost of decryption.

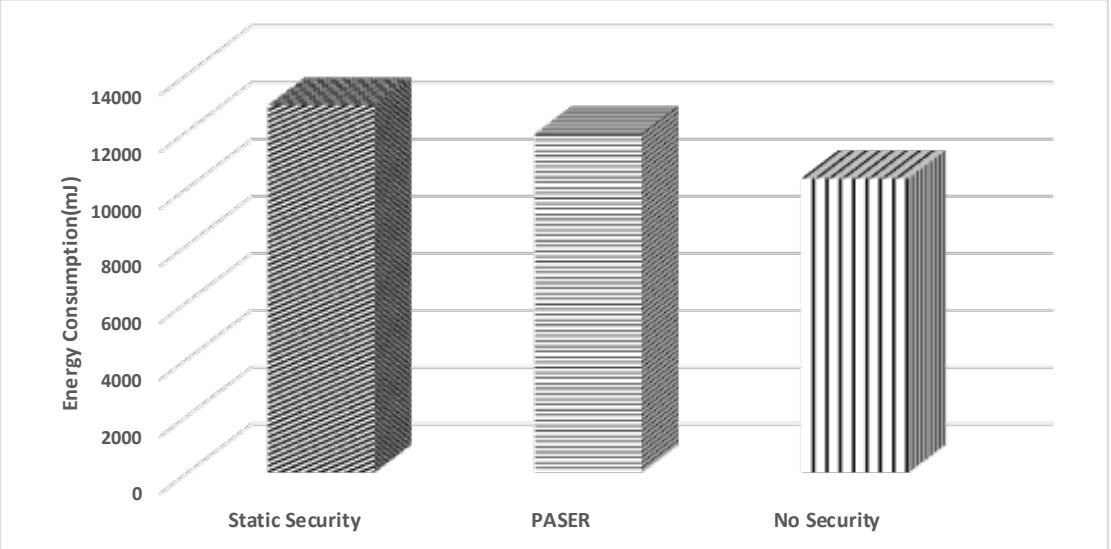


Figure 6.1: Energy consumption of three architectures: no-security, static-security and PASER

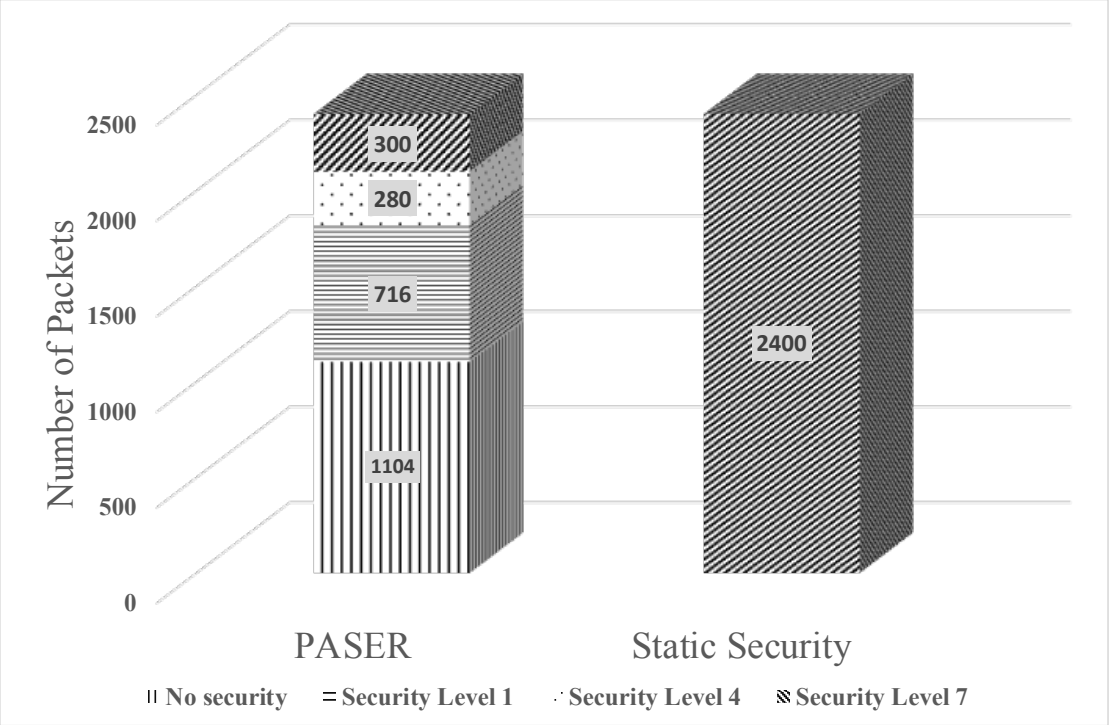


Figure 6.2: Number of packets transmitted at each security level

The number of packets used at each security level in the PASER architecture in this evaluation is shown in Figure 6.2. The energy-saving is achieved through the packets which are sent with low-security or no security.

6.2.2 Network Lifetime

The results, as shown in Figure 6.3, depict the total battery capacity of five nodes, which represent all nodes in the network. The results show that the network lifetime is longer when using PASER compared to the *static-security* architecture. The total battery capacity for at the beginning of the experiment is $2500mAh$, and it decreases gradually over time due to node activities such as transmission, security operations, and duty cycle activities. In this arbitrary scenario, the network lifetime using PASER is 416 *seconds*, while static-security runs for only 294 *seconds*. No-security architecture has been taken as a *baseline* for comparison. PASER, in this scenario, outperforms static-security by almost 28% in terms of network lifetime. The results show that there is a trade-off between security and network lifetime. The saving in energy consumption in the previous sub-section is 12%, whereas the extension in network lifetime is 28%. The reason being that is, in the previous sub-section, the energy consumption of all nodes is recorded for only five minutes; while, in the network lifetime evaluation, the experiment nodes start with specific battery capacity and the experiments run until the first node runs out of battery in each architecture.

The remaining energy of the five nodes in the PASER architecture is depicted in Figure 6.4. As can be seen from the figure, all nodes run out of battery at approximately the same time. The slight difference between them is due to the duty cycle. ContikiMac repeats the transmission for the whole packet until the sink wakes up and transmits an acknowledgement. Therefore, some of the nodes need to send a packet three times for it to be received by the sink; other nodes may need more or less transmission to deliver one packet. The evaluation results do not consider the saving which can be obtained by enabling the degradation control technique in PASER. The following subsection is an evaluation of the impact of enabling the degradation control on network lifetime.

6.2.2.1 Battery Degradation Control

The experiment in the previous sub-section is repeated, but this time with the Battery Degradation Control(BDC) enabled. As can be seen, the results in this arbitrary scenario, as depicted in Figure 6.5, show that enabling the degradation technique extends the network lifetime by almost 12%. Hence, PASER, in the evaluated scenario, saves in total nearly 22% of energy when the degradation control technique is enabled.

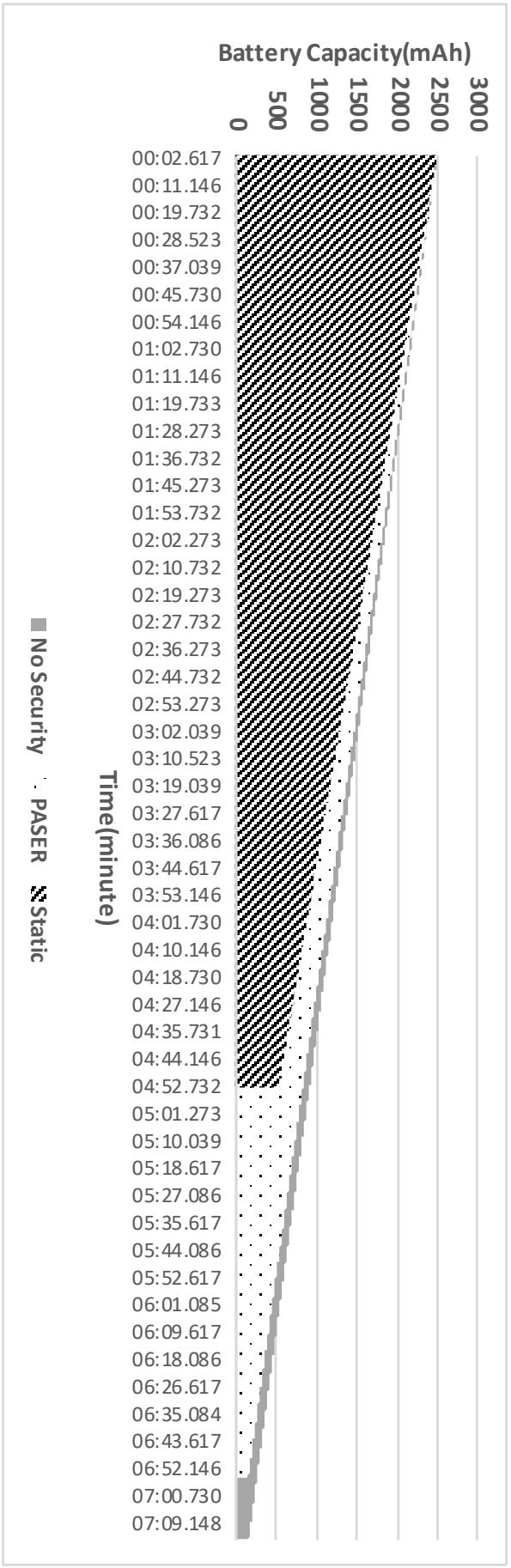


Figure 6.3: Network lifetime using PASER, static and no-security architectures

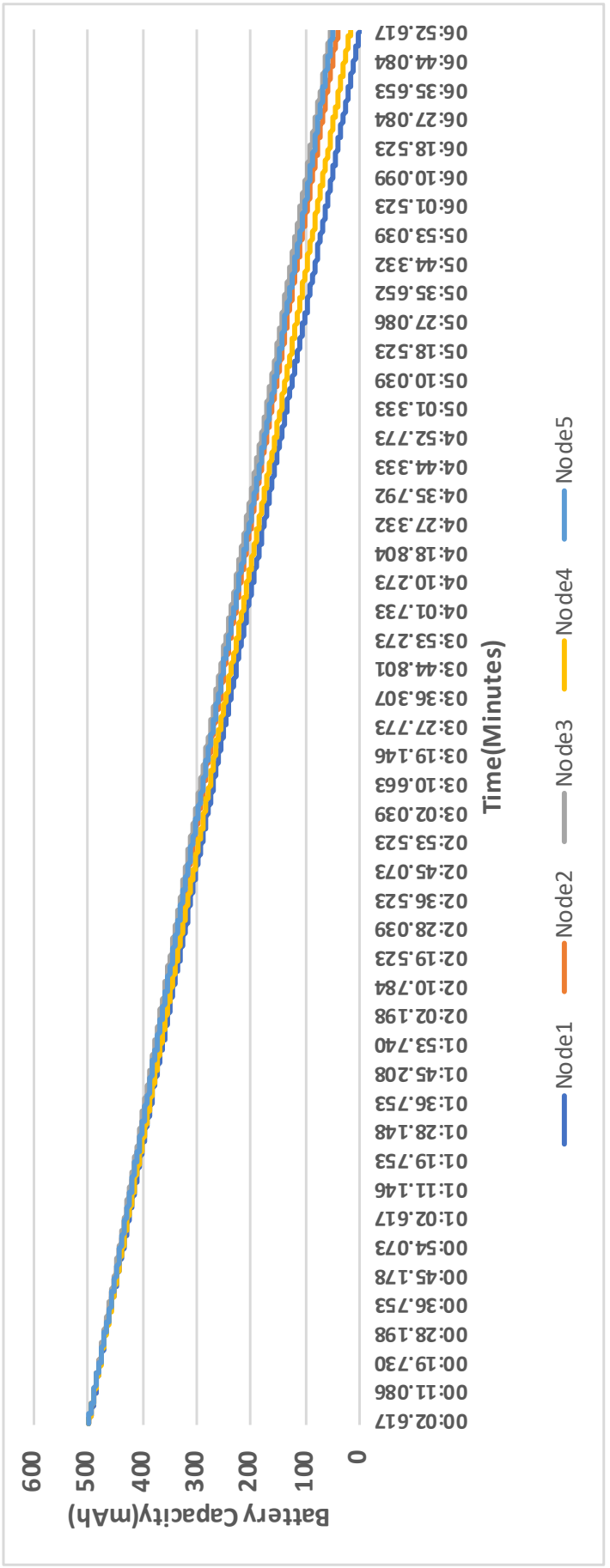


Figure 6.4: Lifetime of the nodes using PASER

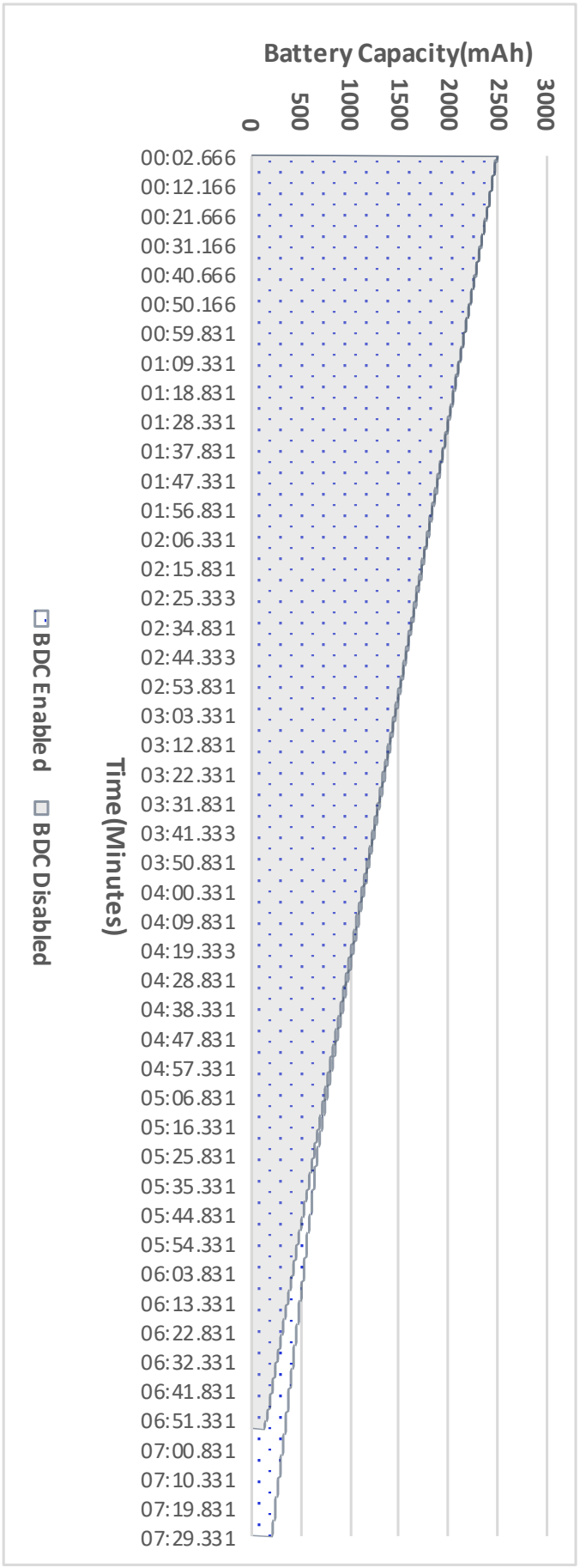


Figure 6.5: Impact of battery degradation control on network lifetime

6.2.3 Network Throughput

The results, as shown in Figure 6.6, depict that with low data rate: 2 and 4 packets/second, there is no difference between the three architectures in terms of network throughput. However, with a higher data rate, the throughput is significantly lower when compared to the *baseline*. PASER with a data rate of 6 and 8 packets/s achieves higher throughput, by 7.6% and 5% over static-security, consecutively. The lower percentage with 8 packets/s compared to 6 packets/s is due to the collision increasing with a higher data rate. The no-security experiment has higher data rate in this evaluation because plain packet takes less time to be prepared and transmitted than secured packet, and hence allowing more packets to be transmitted. It is observed that with the higher data rate, PASER outperforms static-security in terms of throughput.

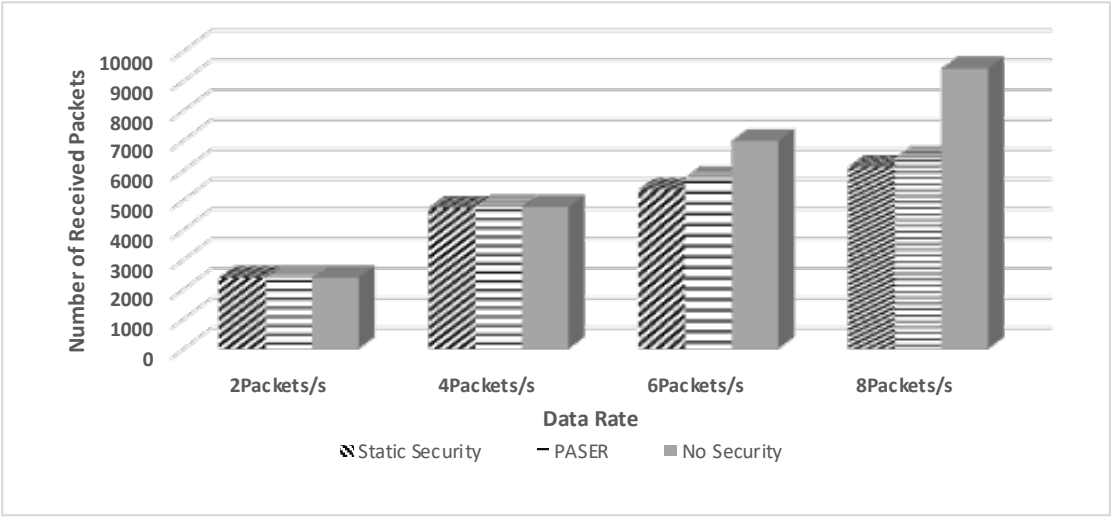


Figure 6.6: Number of received packets at the sink over different data rate using simulation

6.2.4 Latency

The results, as shown in Figure 6.7, depict the per-packet latency of the three architectures. According to the figure, a packet takes almost 11 ms to be delivered using the no-security architecture. However, this increases significantly to 64 ms when the static-security architecture is utilised. The PASER architecture uses four different security levels; hence the latency time fluctuates between the latency of the no-security architecture and the latency of the static-architecture. This fluctuation in security levels occurs according to the ASA decision regarding security at transmission time, as shown in Figure 6.7.

The results, which are shown in Figure 6.8 present the total latency for PASER and the other two architectures. The results indicate that there is a significant difference in the latency between static-security and the two other architectures. No-security architecture

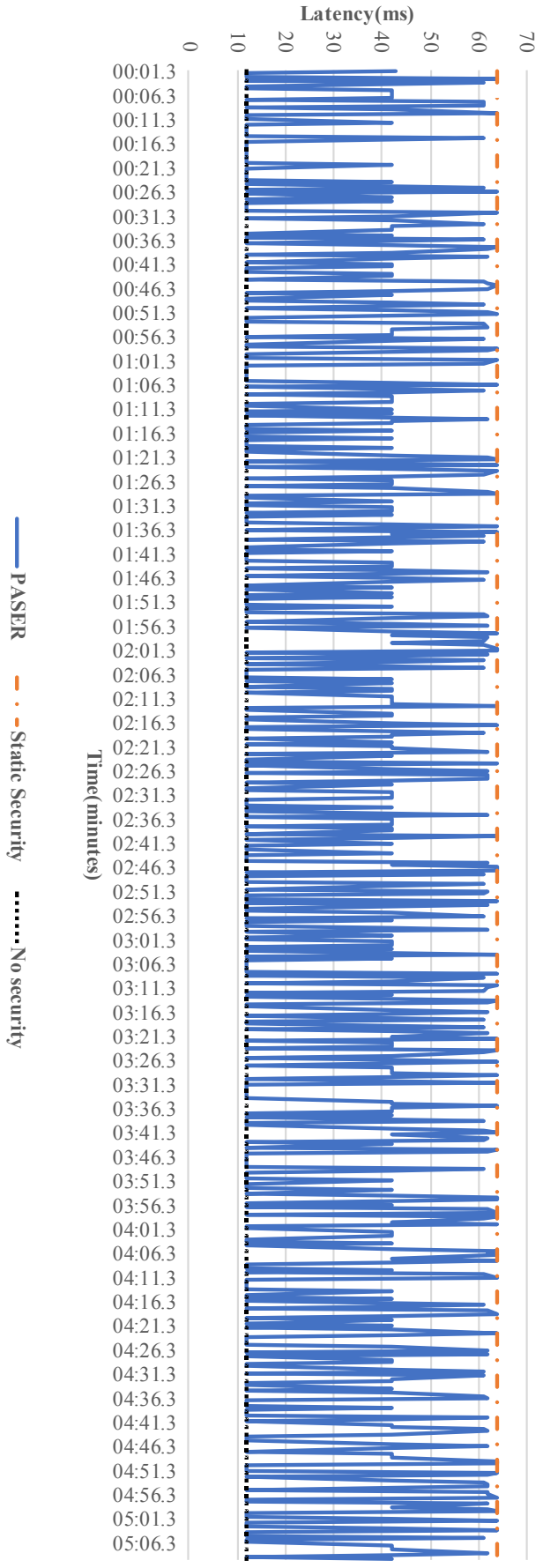


Figure 6.7: Per-packet latency

is used as a *baseline* in this evaluation. The results show that static-security increases the latency by 189% over the *baseline*, while PASER increases latency by only 48% over the *baseline*. Therefore, PASER outperforms static-security by almost 141%. The PASER graph is situated between the static-security and no-security architecture charts. The lower latency achieved by PASER is due to PASER using different security levels at run time. Each security level is associated with a certain amount of latency. The variation in latency between the employed security level enables PASER to be more efficient than static-security. Hence the microcontroller and transceiver take less time to process and transmit some packets due to low or no security services.

6.2.5 Discussion

One of the main objectives of this chapter is to investigate whether the proposed concept of PASER can be implemented in wireless sensor networks. The answer to this question clarifies whether or not the hardware or software restrictions limit the proposed architecture. The simulated work obtained using Cooja simulator show that PASER can be implemented in wireless sensor networks. The results show that data are exchanged between nodes using different security levels according to the ASA architecture. It shows that PASER can work as expected and as proposed in the previous chapter. Another aim of this chapter is to evaluate the performance of the proposed architecture in terms of energy consumption and QoS parameters. As shown in the previous sections, PASER significantly improves the performance of WSNs regarding energy consumption and QoS parameters. PASER is 9.7% more efficient in terms of energy compared to static-security in the evaluated scenario. This percentage can increase or decrease according to the threat level and application requirements. The efficiency of PASER has also increased network lifetime compared to static-security. The evaluation shows that PASER throughput with a higher data rate outperforms static-security. Besides, the total network latency, in the evaluated scenario, has fallen by 141% using PASER compared to static-security. It is observed that when degradation control is enabled the network lifespan increases by almost 12% compared to a sudden die out. *The overall simulated results show a significant improvement in efficiency when using PASER for applications which are seeking a trade-off between security and energy consumption or network performance.*

6.3 Testbed Experiments

Most existing security solutions for wireless sensor nodes have only been evaluated by simulation [124]. However, with the spread of real nodes and the availability of online testbeds, more accurate results based on real hardware are needed. Due to the challenges introduced by wireless sensor nodes such as using a shared medium for radio

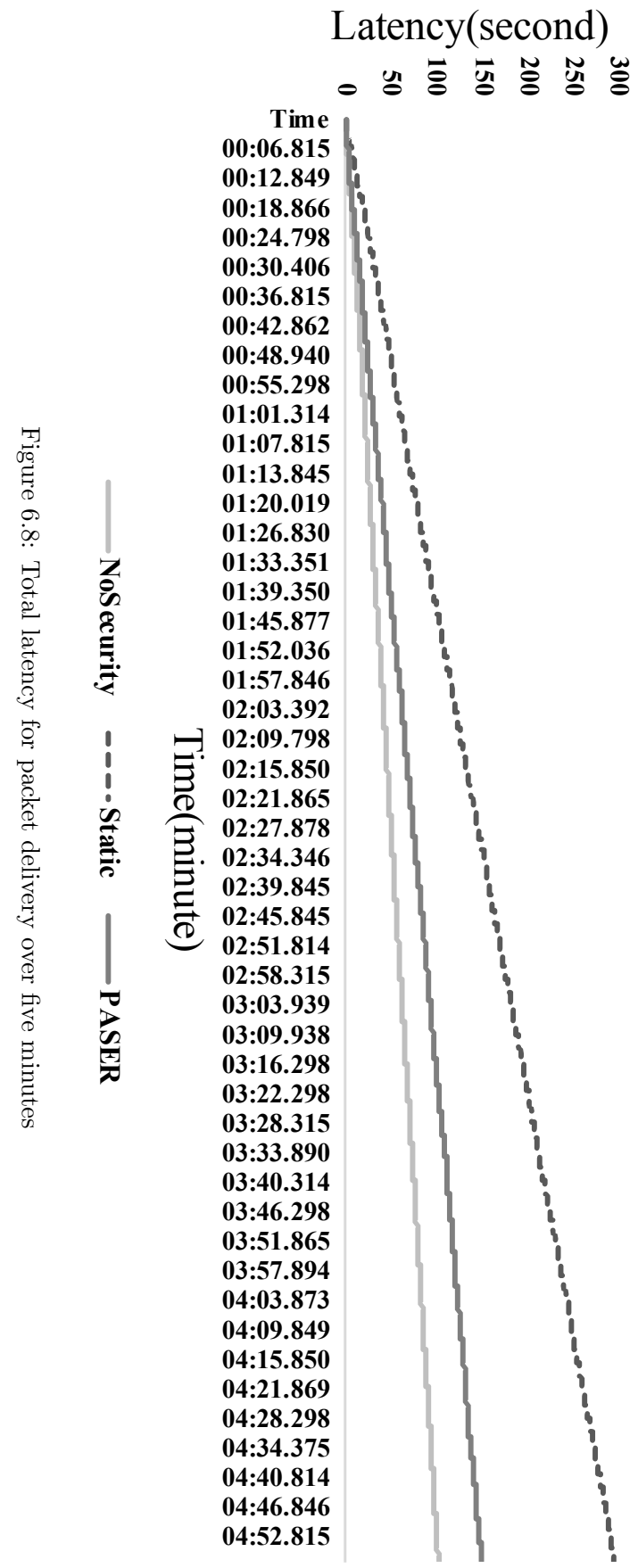


Figure 6.8: Total latency for packet delivery over five minutes

communication, results obtained from the simulative and theoretical evaluation can only be considered approximate [125]. Even perfect approximations need to be confirmed by real hardware measurements [125]. This does not mean that simulations are not useful. Indeed, simulation and emulation are beneficial when testing a new concept and evaluating a network at large-scale. They are essential steps ahead of real experimentation that can limit issues related to solution design [126]. Implicit issues exist in some simulators which are revealed when using real hardware. Hence, to be able to use PASER in the real world, it should not be restricted to the simulation environment. This section presents the evaluation results using real hardware. MTM-CM5000 hardware is used in this evaluation.

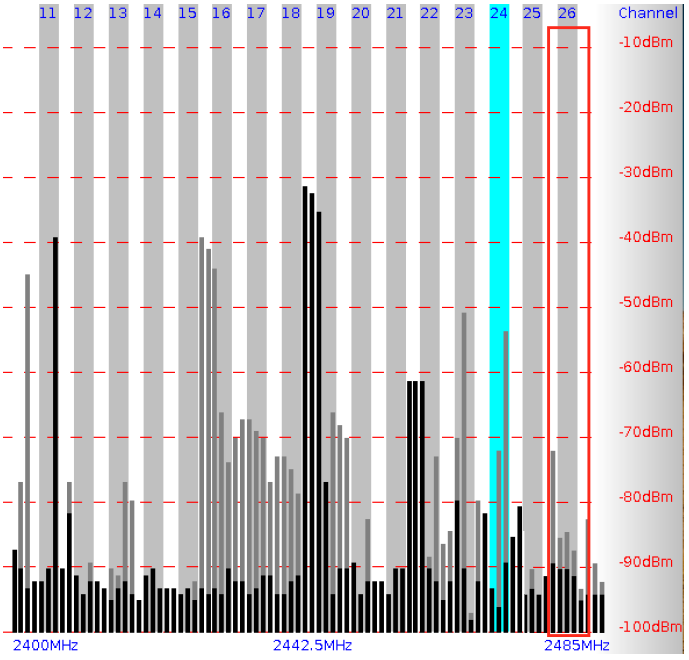


Figure 6.9: Floor noise level

In the actual hardware deployment, the floor noise should be sampled, and the CCA threshold should be set above the noise level to achieve reliable communication. This means that before each transmission, CCA measures the energy level in the channel and compares it with a pre-defined value. The transmission takes place only if the sampled Received Signal Strength Indicator (RSSI) is less than the pre-defined value. Figure 6.9 shows a sample for the floor noise in our lab. Contiki uses a default value of $-90dBm$ for CCA threshold, which is the sum of two parameters: $CC2420\_CONF\_CCA\_THRESH$ and $RSSI\_OFFSET$, each is assigned a default value of $-45dBm$. Hence, the default value for the CCA threshold is updated according to the floor noise test, -50 in this case. The node IDs are burned on the hardware. The experiment is implemented using the same parameters which are used in the simulation experiment in the previous sections. The CC2420 radio chip supports 16 channels in the 2.4GHz band, so the nodes are set to use channel 26, which does not interfere with Bluetooth and Wi-Fi.

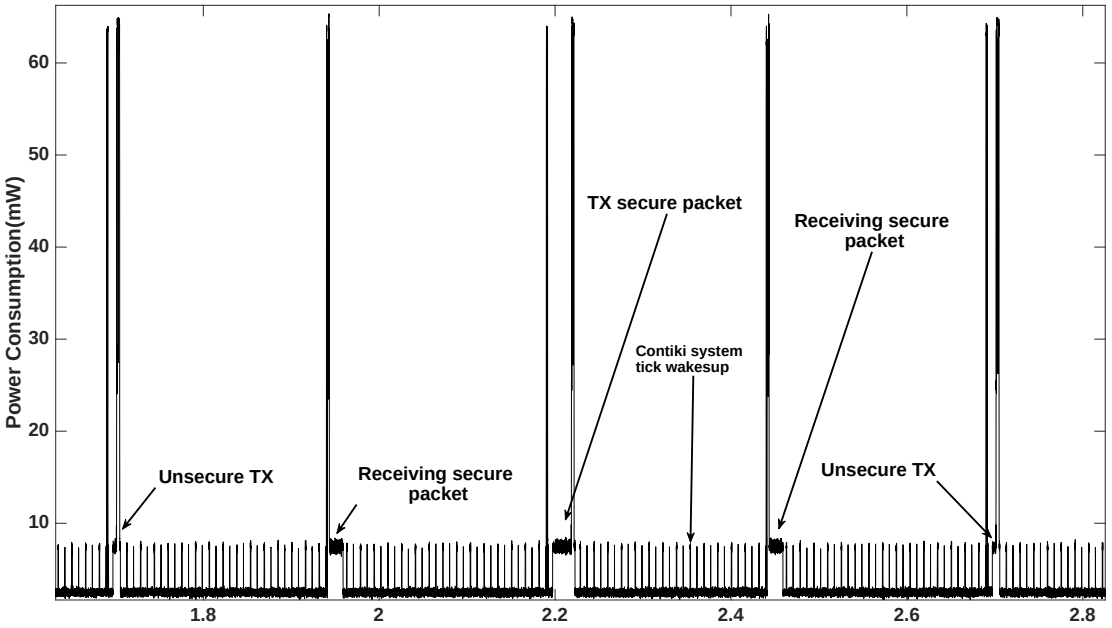


Figure 6.10: Adaptive security using PASER

Figure 6.10 shows a snapshot of PASER running on real hardware. It demonstrates PASER's practicality and shows that packets can take different security levels at the runtime using real network. Hence, it answers research question number 3. Figure 6.10 shows that some packets take a longer to transmit than others, according to the utilised security level. The following subsections present hardware results of the PASER architecture. The assessment covers the following parameters: energy consumption, network lifetime, latency, throughput, and size of PASER on constrained device memory.

6.3.1 Energy Consumption

This subsection presents the evaluation results of energy consumption using MTM-CM5000 hardware.

In this arbitrary scenario, where the payload is changed in a pseudo-random manner,

The results show that a saving of nearly 11% was achieved by PASER when compared with the use of a fixed level of security, as shown in Figure 6.11. Savings will vary from situation to situation. PASER consumes less energy through the packets which are sent with low-security or no security. The 11% energy saving does not include the energy saving which can be achieved by enabling the degradation control mechanism.

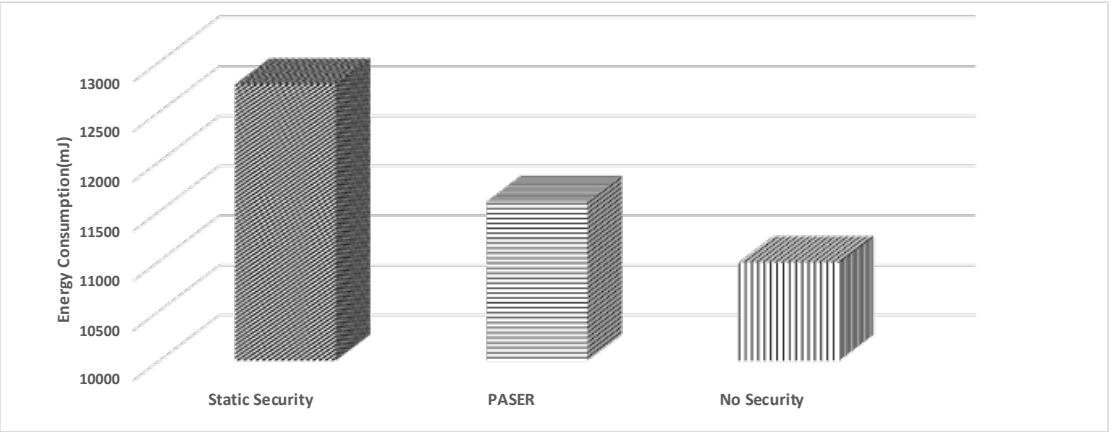


Figure 6.11: Energy consumption of three architectures: no-security, static-security and PASER using real hardware

6.3.2 Network Lifetime

The total battery capacity of the five nodes for each architecture is presented in Figure 6.12. The results show that PASER outperforms static-security by nearly 25%. The evaluation depicts that using PASER can increase the overall network lifetime on a real network.

6.3.3 Packet Latency

This section evaluates the latency caused by using security services on MTMCM5000 hardware. It is difficult to measure exactly the total latency for the whole network. One challenge is that nodes are often not synchronised, and if they are, the accuracy of the results do not meet the requirements of a few milliseconds in this evaluation. Also, connecting the nodes to one reset button does not guarantee that the nodes boot up at the exact same time, based on the experiments conducted on MTM-CM5000 hardware and Contiki OS. Contiki uses a random number to boot up identical nodes and avoid simultaneous transmission. Hence, a per-packet evaluation is executed using the power analyser to show the latency caused by security services at the source node and sink.

Figure 6.13 shows the difference between generating and transmitting a packet using static and no-security architectures. The results show that significant extra time is needed for the microcontroller using static-security to create and send a packet compared to the no-security architecture. Static-security adds $25ms$ over the no-security architecture. The computation of the encryption algorithm is mainly responsible for this extra time. Also, it is observed that the radio runs longer to transmit the extra packet for authentication. The transceiver requires $1.7ms$ to send a packet using the no-security architecture, whereas it requires $2.2ms$ in the static-security architecture. Figure 6.14 depicts the time required to receive packets using static and no-security

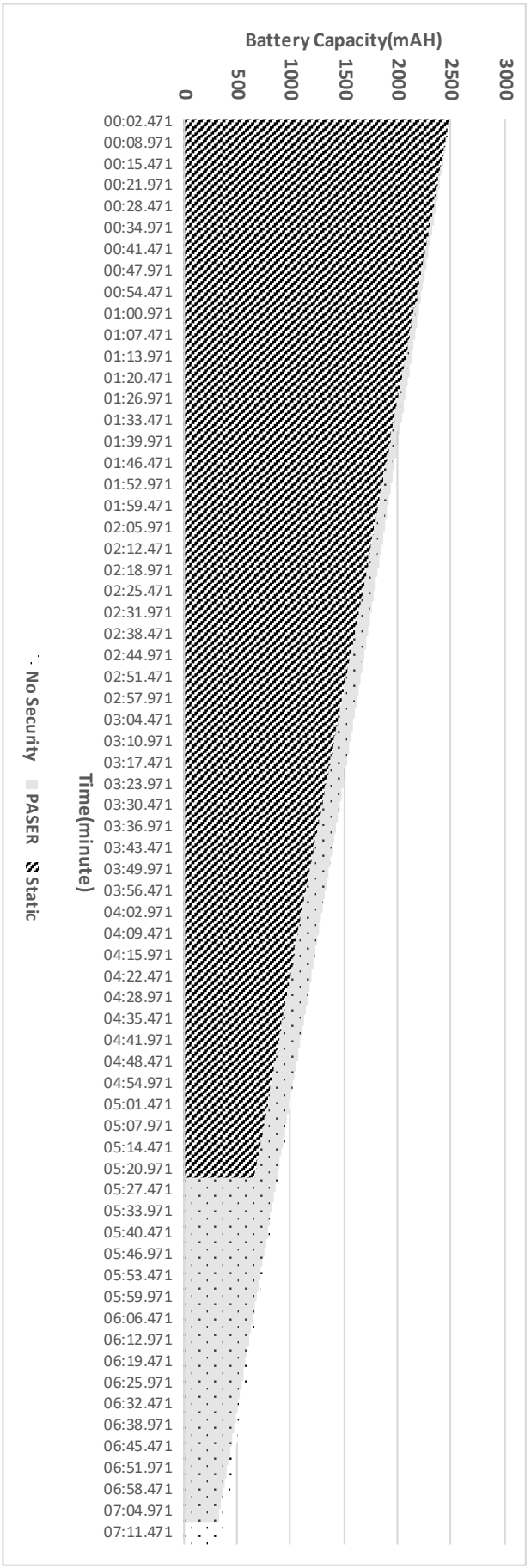


Figure 6.12: Network lifetime for PASER

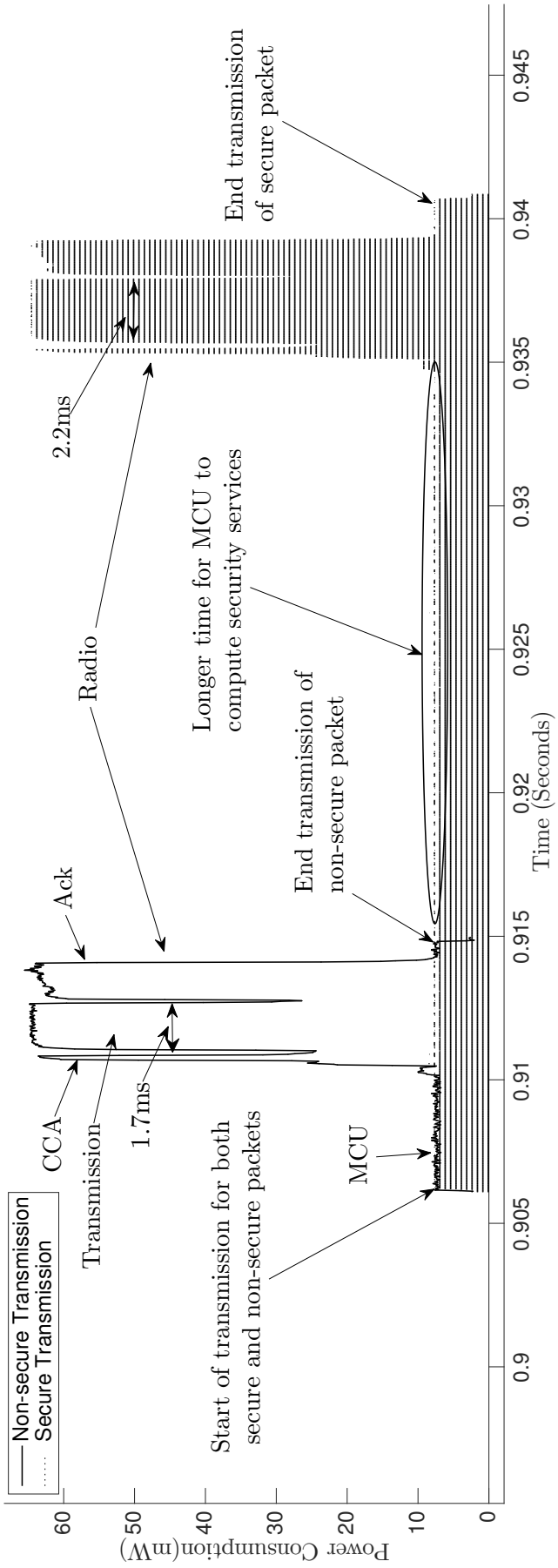


Figure 6.13: Packet generation latency: static vs no-security architectures

architectures. The results show that static-security architecture adds 29ms over the no-security architecture. This extra time is caused by decryption and de-authentication operations. It is observed that 16% more time is needed for packet decryption than packet encryption. PASER uses different security levels, which vary in their latency according to the utilised security level. Each security level is associated with a certain amount of latency. Hence, PASER reduces latency time when compared to static-security through the packets which are sent with low security or no security.

Another evaluation is carried out to calculate the whole latency from the source to the sink using the logic analyser. The assessment includes the time required to generate the packet, time to transmit, and time to receive and decrypt a packet if needed. The latency time in this evaluation is affected by the duty cycle mechanism. The ContikiMac protocol sends the same packet many times until the sink node receives it. Hence, the results vary according to the sink wake-up time. However, this shows the latency which occurs in real network deployment. Figure 6.15 shows the latency which is needed to transmit one packet using static-security and no-security architectures. The results show that the static-security architecture requires 53ms, while the no-security architecture requires only 13ms.

The results show that static-security requires more time to process, transmit and receive compared to the no-security architecture. The latency incurred by PASER varies from one scenario to another based on the utilised security level.

6.3.4 Network Throughput

This section presents the evaluation results of the hardware network throughput. Figure 6.16 shows that PASER has slightly higher throughput by 0.69% compared to the static-security architecture when using a data rate of 2 packets/s. However, when the data rate is increased to 4 and 6 packets/s, PASER achieves almost 2.8% higher throughput. There is a considerable difference between PASER and static-security architectures when the data rate is increased to 8 packets/s, PASER achieves higher throughput by 11%. Packets require more time to process and transmit when security is enabled, and this extra time decreases throughput. The results indicate that with low data-rate, there is a slight difference between PASER and the static-security architecture. However, when the data-rate increases, PASER outperforms the static-security architecture significantly.

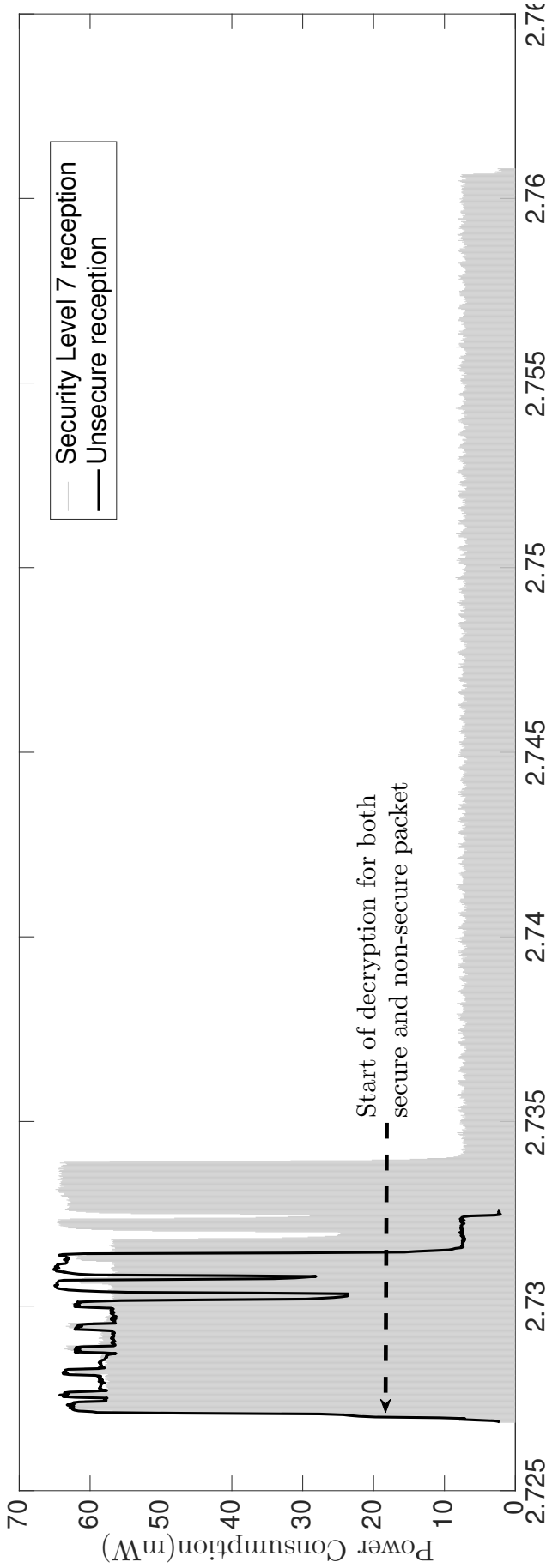


Figure 6.14: Packet receiving latency: static-security vs no-security architectures

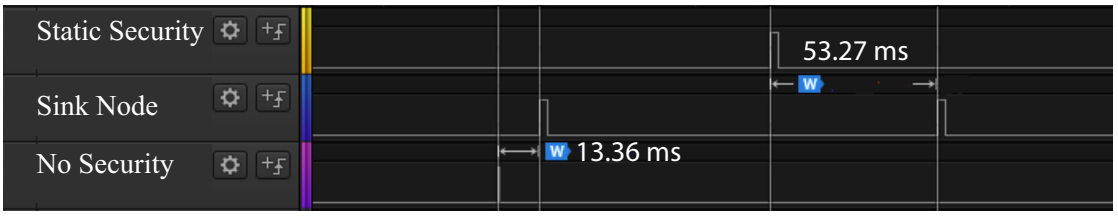


Figure 6.15: Packet latency for delivering one packet: static vs no-security architectures

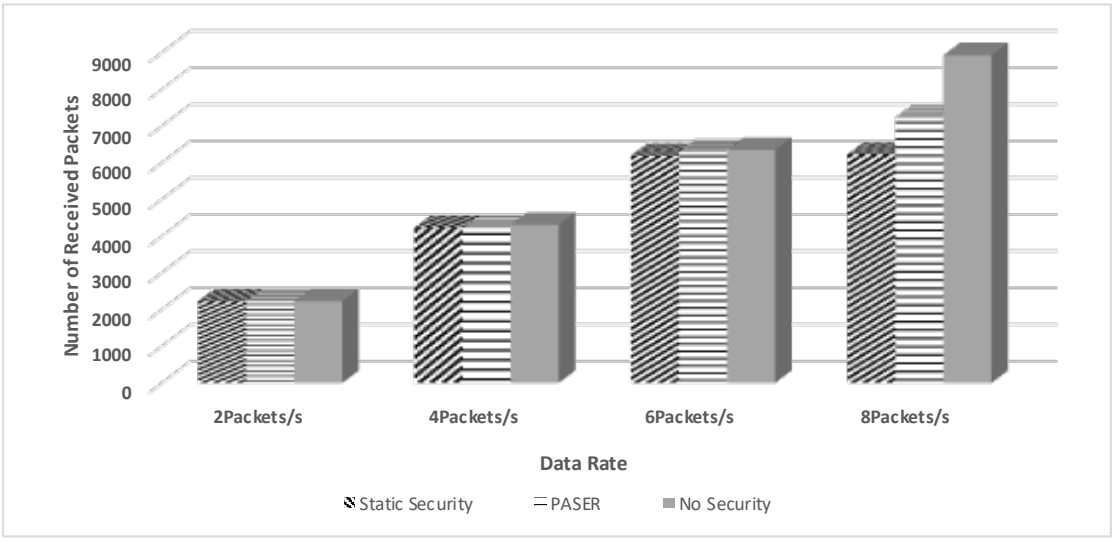


Figure 6.16: Number of packets received at the sink over different data rates using real hardware

6.3.5 Battery Degradation Control

The results, as shown in Figure 6.17, depict the network lifetime using PASER with and without BDC. PASER will be used as a *baseline* in this evaluation. The hardware results show an improvement in the network lifetime of nearly 16% using the battery degradation technique when compared to PASER. This technique causes the batteries in the nodes to run out at almost the same time. The PASER architecture is designed to be flexible so that a network manager can enable the BDC when required.

6.3.6 Memory Footprint Evaluation

Table6.1 shows the flash memory used by PASER on MTM-CM5000 hardware. The PASER scheme imposed a $3kb$ overhead on flash memory beyond the standard Contiki.

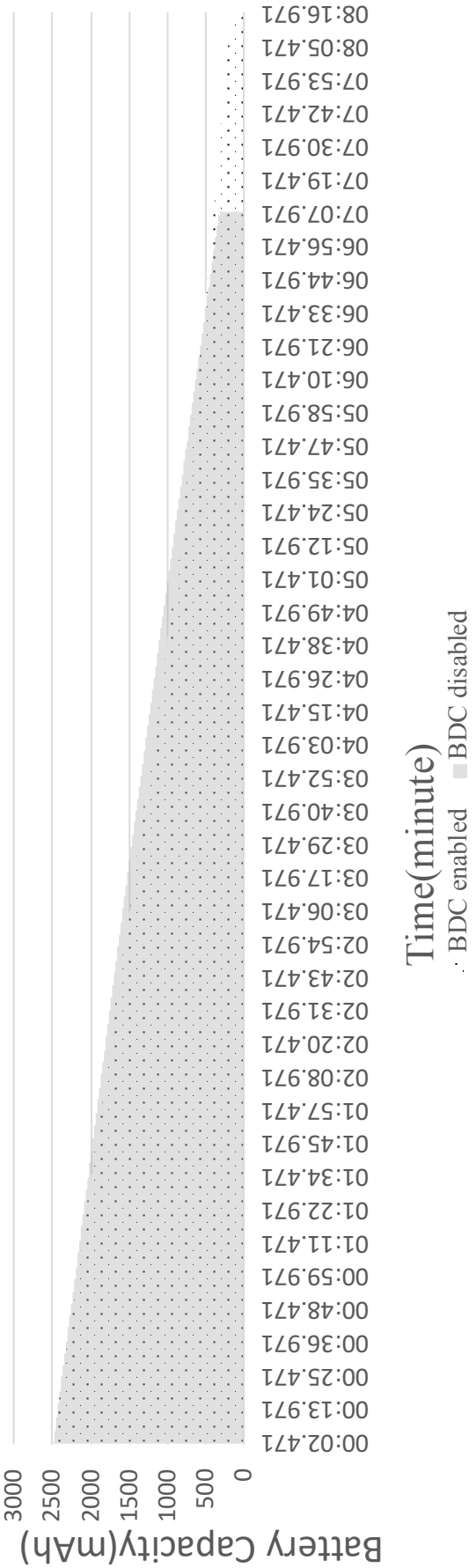


Figure 6.17: Impact of battery degradation technique on network lifetime using real hardware

Table 6.1: Flash memory requirement: static, PASER and no-security architectures

Architecture Name	Contiki	Contiki+PASER	Contiki+Static
ROM Size(KB)	22	25	25

6.3.7 Test-bed Discussion

The previous sections show an evaluation of three different architectures using real hardware. The actual hardware evaluation aims to confirm the results obtained by the simulation evaluation. The results show the trade-off between security, energy and network performance in wireless sensor networks. The results have shown that the PASER architecture is more efficient than static-security in terms of network performance. Energy consumption is reduced by nearly 11% using PASER when compared to the static-security architecture. Network lifetime is increased by 25% using PASER compared to the static-security architecture. Network latency and the throughput are also improved using PASER. The throughput evaluation shows that PASER increases network throughput by 11% with the higher data-rate, 8 packets/s in this evaluation, compared to the static-security architecture. The real hardware evaluation has proven that PASER provides a significant improvement to network performance.

6.4 Simulation vs Testbed

Network simulators produce a proximate imitation of real network deployment. Some simulators use the binary code in simulation as in real hardware which returns results that are as close a representation as possible of the actual implementation of WSNs. Cooja uses the same firmware used in the real hardware. Hence, binary-level emulation makes the results obtained by Cooja relatively accurate. However, it is difficult to obtain identical results from the simulation and real hardware due to the accuracy of the hardware calibration in the simulator and the different environments in which the nodes run. Cooja assumes perfect conditions as long as the nodes are within the same transmission range. Therefore, packets will be successfully delivered to the sink. In contrast, in a real deployment, the delivery ratio is affected by other factors such as radio interference. For example, it is observed that the results obtained by simulation overestimate throughput by 5% to 8% compared to the results obtained using real hardware.

Cooja uses the PowerTrace tool to record the time needed by hardware components: MCU and radio transceiver in different modes(active, transmission, receiving, etc.). An evaluation is carried out to compare the simulation results with those obtained from the MTM-CM5000 hardware. The experiment parameters in both simulation and actual hardware are set to match each other. For example, the Cooja-simulator randomizes the boot time for every node by a few milliseconds to a few seconds for identical nodes,

as shown in Figure6.18, by default it is set to 1000 ms. As can be seen in Figure 6.18, the nodes are booted at different times. Node boot-time can be controlled by the parameter 'Mote startup delay'.

Figure 6.18: Cooja start-up delay

Other parameters have been taken into account, such as the latency before initializing the operating system in Cooja, which is equal to 249 ms and used to emulate the time needed by actual hardware. There is a slight difference between the data sheet and actual hardware in terms of the current withdrawn by the MCU and transceiver. For accuracy, the current withdrawn by the MTM-CM5000 hardware is used in the simulation to calculate total energy consumption. Figure 6.19 shows an error of 8% when using the simulation.

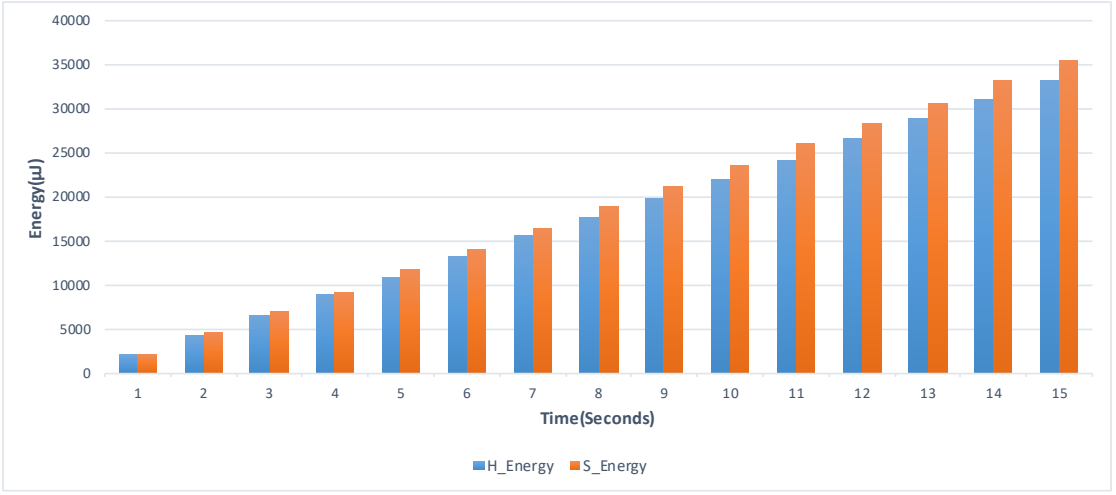


Figure 6.19: Energy consumption: simulation vs real hardware.

An experiment is carried out to explore the cause of the difference between the results obtained by simulation and those by actual hardware. The results show that Cooja simulator has accurate results for emulating the CC2420 transceiver. However, *MSPsim, which is used by Cooja to emulate the MSP430 microcontroller, overestimates the latency caused by the SPI bus, which links the MSP430 to the CC2420 transceiver.* The overestimation matches the findings in [127]. Although they used Z1 hardware, Cooja uses MSPSim module to emulate MSP430, which is used by both MTM-CM5000 and Z1 devices. Finally, the experiments carried out are sufficient to show a comparison between simulation and real hardware evaluation. The purpose of this chapter is to evaluate the PASER architecture rather than evaluate Cooja; however, it has clarified

the difference between them as well. The methodology used in this chapter can also be used to assess adaptive security for similar work.

6.5 Discussion

This chapter has evaluated the PASER architecture and compared its performance with static and no-security architectures. The simulation and real hardware experiments show that PASER outperforms static-security in many network parameters, especially energy consumption. *The results show that PASER can be a promising solution for applications which require a trade-off between security and energy consumption.* The BDC utilised by PASER has significantly extended network lifetime. This chapter proves the applicability of PASER to run on wireless sensor nodes. Besides, it shows the impact of the proposed architecture on network performance. This chapter provides practical evidence that PASER saves energy and improves network performance using a real network. It answers research questions number 3 and 4. In the future, an evaluation will be conducted using the PASER architecture in a real-life scenario.

Chapter 7

Conclusions and Future Work

7.1 Thesis Contributions

The main contribution of this thesis is the proposed PASER architecture which allows a flexible trade-off between security, QoS, and energy consumption in IoT embedded devices. The work which has been carried out in this thesis can be assessed against the research questions in Section 1.1.

The first contribution is the analysis and evaluation of the impact of current security implementation at the MAC layer. This analysis is executed first using a theoretical literature review, then followed by experimental evaluation. The results of the literature review have shown that both the microcontroller and transceiver are affected by security. The microcontroller is affected because the encryption algorithm requires more computation time, and the transceiver is affected because of the extra bytes needed for authentication. However, it was difficult to quantify the impact of security from the literature and obtain a solid understanding of the effects of security on energy and the relation between security and QoS parameters, such as latency, throughput and network lifetime. Hence, an experimental evaluation of the impact of security on resource-constrained nodes has been carried out in Chapter Three to elaborate the answer to research question number 1. The experimental evaluation has been conducted using simulation and then confirmed by testbed evaluation. The results of chapters two and three benefit both academic research and technical engineers when considering security overhead in WSNs.

The second contribution is the development of PASER architecture which changes the security level dynamically at the run time. The PASER architecture does not reduce security according to the battery level, which what is proposed in the literature. In contrast, it uses the battery level to filter less critical data without reducing security when the battery level is low. It uses the application requirements and utilises an

input for threat level to select the security level. Hence, off-the-shelf TDS solutions can easily be incorporated with PASER. Also, it uses an energy-aware mechanism to control battery degradation, and consequently extend network lifetime. This contribution is the answer for research question 2.

The third contribution is the proven feasibility of PASER on both simulated and real hardware implementation. practical implementation and evaluation of PASER architecture. The challenges and technical issues of running PASER on simulated and real hardware is solved. This answers research question number 3.

The fourth contribution is the analysis and evaluation of PASER architecture to determine the improvements which are achieved by using PASER. The evaluation is executed first using the simulation, and then the results validated using real hardware experiments. Much of the research and evaluations presented in the literature regarding security use either theoretical analysis or simulation. In contrast, the assessment in this thesis includes real hardware. A comparison is made to static and no security architecture. The results show that PASER, in the evaluated scenario, outperforms static security. Additional experiments are conducted to compare simulation to real hardware results, which show overestimation in the simulation results regarding power consumption. This contribution answers research question number 4.

7.2 Conclusion

Energy consumption is a crucial factor when designing a security architecture for wireless sensor nodes, due to their limitations in terms of the energy resources, computation capability, and memory size. The widespread use of IoT applications makes security an essential factor for numerous applications. However, providing security for a WSN is a challenging task due to restricted resources, unattended operation, and the nature of communication within a WSN. Security imposes a significant overhead on WSN communication. This overhead is due to the extra time needed for the microcontroller to compute the encryption algorithm and additional time required by the radio to transmit the extra bytes for authentication. This thesis has studied the security requirements and challenges facing WSN. It has demonstrated the trade-off between security, energy consumption, and QoS. The per-packet evaluation has shown an increase in the overhead on both the microcontroller and radio module when enabling security. It depicts that security overhead in terms of energy consumption fluctuates between 31.5% at a minimum level over non-secure packets and 60.4% at the top security level of the IEEE802.15.4 specification. Other network parameters such as latency and throughput are affected as well. Furthermore, the evaluation results have shown that determining the cost of security in WSN is not straightforward. Many factors such as payload length and the type

of encryption and authentication code can affect the overall security overhead. However, in all cases, security cost in terms of energy is still considerable. The evaluation is carried out using simulation and then validated using actual hardware. This research has investigated and discussed the current security solutions in the literature, and then identified gaps in the field.

Furthermore, this thesis proposes a generic adaptive security architecture to manage the trade-off between security and energy. The proposed solution is the PASER trade-off between security of less-critical data and energy consumption. PASER switches between security levels adaptively at run-time. It takes advantage of the overhead difference between security levels and uses it to save energy at run-time. It works in a decentralised manner, so there is no extra traffic generated by security services between nodes. Also, decentralised operation helps in avoiding single points of failure, since every node works independently. PASER uses the application preferences, remaining energy and the TDS properly without reducing security according to the battery level. It adapts the applied security settings to better match the network requirements, given the changing security environment. An example scenario is evaluated. The simulated and experimented results, in the evaluation scenario, both show energy savings of nearly 10% and 11% when compared to the static-security architecture, which validates the concept. Also, the proposed solution shows an improvement in network throughput, lifetime by 25% and latency by almost 141%.

7.3 Future Work

The work which has been carried out in this thesis has demonstrated the benefits of using an adaptive security properly to trade-off the opposite parameters of security and energy consumption. Potential future research which is highlighted by this thesis is suggested in the following points:

- **Threat-aware System**

  Threat-awareness in terms of security refers to nodes that are sensitive to their context by adapting their security configuration according to changing environmental conditions. The change in the environment state makes the threat level a beneficial input for adaptive security. This thesis has incorporated threat level input in the security decision. A threat-aware system monitors suspicious activity during the network operation to predict the threat level. An example for suspicious activity is when a node receives a replay packet or when it receives a message with

the wrong secret key. However, developing other methods which improve security decision-making at run time would be a useful input to the creation of adaptive security solutions. Parameters such as data rate, for instance, can be used to predict a DoS attack. The developed mechanisms may require the gathering of data about neighbour node behaviour and the processing of these data to assess the threat level.

- **Adaptive Battery Degradation Techniques** This thesis utilised an adaptive battery degradation technique to extend the network lifetime when battery level drops below a predefined threshold. The results show an extension to network lifetime. Further techniques can be used to extend network lifetime, such as changing the data rate dynamically when the energy level is critical. Another parameter which can be used for battery degradation control is transmission power, where coverage can be reduced gradually instead of stopped immediately. The mechanism utilised in this context should be evaluated to make sure it extends network lifetime.

- **Utilising Energy Harvesting System** Energy harvesting is not a new concept; however, utilising it with adaptive security will extend network availability. The harvested energy might not be predictable or reliable, so deploying both techniques would increase the benefits to network lifetime. Also, the harvested energy will help in avoiding the filtration of data with low priority in favour of transmitting data with high priority. An energy harvesting system can easily be added without changing the PASER design, because PASER checks the battery level before each transmission and behaves accordingly.

- **Multi-hop Evaluation**: The experiments in this evaluation have been conducted using start topology. The impact of security would be higher using multi-hop topology. For example, the time it takes each packet to travel from the source to the sink through multi nodes would be higher. All the nodes in-between will decrypt and encrypt the packet, which will lead to more overhead. According to the evaluation results, the difference between the static-security and PASER architectures in term of latency is significant. Hence, it is believed that PASER will save more energy and time when it is used in a multi-hop topology.

- **Generated Packets** The environmental change in a real-life scenario is usually stable and does not occur frequently. For example, environmental temperature or vibration readings do not normally change significantly within a matter of seconds. PASER evaluation can be extended in the future using real-life sensor readings. PASER is expected to save more energy in a real-life scenario. The reason being that the probability of incidents occurring in a real scenario is low compared to the random sensor reading which used in this thesis.

Network requirements is a key factor to consider when building security solutions. However, it is difficult to know these requirements at the deployment time. Hence, extending adaptive security solutions by customising other parameters such as data rate and transmission power, or enhancing the threat level assessment at the run-time would make the solution more flexible and allow network administrators to extend the network lifetime when required.

References

[1] M. Turkanović, B. Brumen, and M. Hölbl, "A novel user authentication and key agreement scheme for heterogeneous ad hoc wireless sensor networks, based on the Internet of Things notion," *Ad Hoc Networks*, vol. 20, pp. 96–112, sep 2014. [Online]. Available: http://linkinghub.elsevier.com/retrieve/pii/S157087051400064X

[2] L. D. Xu, W. He, and S. Li, "Internet of things in industries: A survey," *IEEE Transactions on Industrial Informatics*, vol. 10, no. 4, pp. 2233–2243, Nov 2014.

[3] Y. Zhong, L. Cheng, L. Zhang, Y. Song, and H. R. Karimi, "Energy-efficient routing control algorithm in large-scale wsn for water environment monitoring with application to three gorges reservoir area," *The Scientific World Journal*, vol. 2014, 2014.

[4] J. Zheng and A. Jamalipour, *Wireless Sensor Networks: A Networking Perspective*. Wiley, 2009. [Online]. Available: https://books.google.co.uk/books?id=qOPk-NWkgiMC

[5] A. Tsitsigkos, F. Entezami, T. A. Ramrekha, C. Politis, and E. A. Panaousis, "A case study of internet of things using wireless sensor networks and smartphones," in *Proceedings of the Wireless World Research Forum (WWRF) Meeting: Technologies and Visions for a Sustainable Wireless Internet, Athens, Greece*, vol. 2325, 2012.

[6] O. Chipara, C. Lu, T. C. Bailey, and G.-C. Roman, "Reliable clinical monitoring using wireless sensor networks: Experiences in a step-down hospital unit," in *Proceedings of the 8th ACM Conference on Embedded Networked Sensor Systems*, ser. SenSys '10. New York, NY, USA: ACM, 2010, pp. 155–168. [Online]. Available: http://doi.acm.org/10.1145/1869983.1869999

[7] G. Virone, A. Wood, L. Selavo, Q. Cao, L. Fang, T. Doan, Z. He, and J. Stankovic, "An advanced wireless sensor network for health monitoring," in *Transdisciplinary Conference on Distributed Diagnosis and Home Healthcare (D2H2)*, 2006, pp. 2–4.

[8] A. Darwish and A. E. Hassanien, "Wearable and implantable wireless sensor network solutions for healthcare monitoring," *Sensors*, vol. 11, no. 6, pp. 5561–5595, 2011.

[9] Y. Charfi, N. Wakamiya, and M. Murata, "Challenging issues in visual sensor networks," *IEEE Wireless Communications*, vol. 16, no. 2, pp. 44–49, 2009.

[10] H. Modares, R. Salleh, and A. Moravejosharieh, "Overview of security issues in wireless sensor networks," in *2011 Third International Conference on Computational Intelligence, Modelling Simulation*, Sep. 2011, pp. 308–311.

[11] S. Sciancalepore, G. Piro, E. Vogli, G. Boggia, and L. A. Grieco, "On securing ieee 802.15.4 networks through a standard compliant framework," in *2014 Euro Med Telco Conference (EMTC)*, Nov 2014, pp. 1–6.

[12] T. Pazynyuk, J. Li, G. S. Oreku, and L. Pan, "Qos as means of providing wsns security," in *Networking, 2008. ICN 2008. Seventh International Conference on.* IEEE, 2008, pp. 66–71.

[13] M. Elshrkawey, S. M. Elsherif, and M. E. Wahed, "An enhancement approach for reducing the energy consumption in wireless sensor networks," *Journal of King Saud University-Computer and Information Sciences*, 2017.

[14] Z. Jiang and Y. Pan, *From Problem to Solution: Wireless Sensor Networks Security.* Commack, NY, USA: Nova Science Publishers, Inc., 2009.

[15] I. Standard and I. C. Society, "IEEE Standard for Local and metropolitan area networks — Part 15 . 4 : Low-Rate Wireless Personal Area Networks (LR-WPANs) IEEE Computer Society S ponsored by the," *IEEE Std 802.15.4-2011*, vol. 2011, no. September, pp. 1–294, 2011. [Online]. Available: http://ieeexplore.ieee.org/servlet/opac?punumber=6012485

[16] A. Rachedi and A. Benslimane, "Multi-objective optimization for security and qos adaptation in wireless sensor networks," in *IEEE International Conference on Communications (ICC), Kuala Lumpur, Malaysia*, 2016.

[17] D. K. G., M. K. Singh, and M. Jayanthi, Eds., *Network Security Attacks and Countermeasures.* IGI Global, 2016. [Online]. Available: http://services.igi-global.com/resolvedoi/resolve.aspx?doi=10.4018/978-1-4666-8761-5

[18] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless sensor networks: a survey," *Computer networks*, vol. 38, no. 4, pp. 393–422, 2002.

[19] J. M. Parenreng and A. Kitagawa, "A Model of Security Adaptation for Limited Resources in Wireless Sensor Network," *Journal of Computer and Communications*, vol. 05, no. 03, pp. 10–23, 2017. [Online]. Available: http://www.scirp.org/journal/doi.aspx?DOI=10.4236/jcc.2017.53002

[20] K. Ashton *et al.*, "That 'internet of things' thing," *RFID journal*, vol. 22, no. 7, pp. 97–114, 2009.

[21] D. Miorandi, S. Sicari, F. D. Pellegrini, and I. Chlamtac, "Internet of things: Vision, applications and research challenges," *Ad Hoc Networks*, vol. 10, no. 7, pp. 1497 – 1516, 2012. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1570870512000674

[22] D. Giusto, A. Iera, G. Morabito, and L. Atzori, *The Internet of Things: 20th Tyrrhenian Workshop on Digital Communications*. Springer New York, 2010. [Online]. Available: https://books.google.co.uk/books?id=vUpiSRc0b7AC

[23] H. Sundmaeker, P. Guillemin, P. Friess, and S. Woelfflé, "Vision and challenges for realising the internet of things," *Cluster of European Research Projects on the Internet of Things, European Commision*, vol. 3, no. 3, pp. 34–36, 2010.

[24] J. A. Manrique, J. S. Rueda-Rueda, and J. M. T. Portocarrero, "Contrasting internet of things and wireless sensor network from a conceptual overview," in *2016 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData)*, Dec 2016, pp. 252–257.

[25] Lu Tan and Neng Wang, "Future internet: The internet of things," in *2010 3rd International Conference on Advanced Computer Theory and Engineering(ICACTE)*, vol. 5, Aug 2010, pp. V5–376–V5–380.

[26] L. Doherty, J. Simon, and T. Watteyne, "Wireless sensor network challenges and solutions," *Microwave Journal*, vol. 55, no. 8, pp. 22–34, 2012.

[27] S. B. Othman, A. A. Bahattab, A. Trad, and H. Youssef, "Confidentiality and integrity for data aggregation in wsn using homomorphic encryption," *Wireless Personal Communications*, vol. 80, no. 2, pp. 867–889, 2015.

[28] S. Alam and D. De, "Analysis of Security Threats in Wireless Sensor Network," *International Journal of Wireless & Mobile Networks*, vol. 6, no. 2, pp. 35–46, apr 2014. [Online]. Available: http://www.airccse.org/journal/jwmn/6214ijwmn04.pdf

[29] A. A. Alkhatib, G. S. Baicher, and W. K. Darwish, "Wireless sensor network-an advanced survey," *International Journal of Engineering and Innovative Technology (IJEIT)*, vol. 2, no. 7, pp. 355–369, 2013.

[30] T. Hara, V. Zadorozhny, and E. Buchmann, *Wireless Sensor Network Technologies for the Information Explosion Era*, ser. Studies in Computational Intelligence. Springer Berlin Heidelberg, 2010. [Online]. Available: https://books.google.co.uk/books?id=lNpsCQAAQBAJ

[31] S. Peter, D. Westhoff, and C. Castelluccia, "A survey on the encryption of convergecast traffic with in-network processing," *IEEE Transactions on Dependable and Secure Computing*, vol. 7, no. 1, pp. 20–34, Jan 2010.

[32] H. Wang, A. O. Fapojuwo, and R. J. Davies, "A wireless sensor network for feedlot animal health monitoring," *IEEE Sensors Journal*, vol. 16, no. 16, pp. 6433–6446, Aug 2016.

[33] T. Ojha, S. Misra, and N. S. Raghuwanshi, "Wireless sensor networks for agriculture: The state-of-the-art in practice and future challenges," *Computers and Electronics in Agriculture*, vol. 118, pp. 66–84, 2015.

[34] L. Wan, G. Han, L. Shu, N. Feng, C. Zhu, and J. Lloret, "Distributed parameter estimation for mobile wireless sensor network based on cloud computing in battlefield surveillance system," *IEEE Access*, vol. 3, pp. 1729–1739, 2015.

[35] V. Singhal and S. Suri, "Comparative study of hierarchical routing protocols in wireless sensor networks," *International Journal of Computer Science and Engineering*, vol. 2, no. 5, pp. 142–147, 2014.

[36] I. Akyildiz and M. Vuran, *Wireless Sensor Networks*, ser. Advanced Texts in Communications and Networking. Wiley, 2010. [Online]. Available: https://books.google.co.uk/books?id=7YBHYJsSmS8C

[37] L. S. Volpe, "Energy-efficient memories for wireless sensor networks," Ph.D. dissertation, Universidad de la República Montevideo, 2013.

[38] S. Gajjar, N. Choksi, M. Sarkar, and K. Dasgupta, "Comparative analysis of wireless sensor network motes," in *2014 International Conference on Signal Processing and Integrated Networks (SPIN)*, Feb 2014, pp. 426–431.

[39] ADVANTICSYS, "Ultra low power ieee 802.15. 4 compliant wireless sensor module," [Online] Available: https://www.advanticsys.com-/shop/mtmcm5000msp-p-14.html.

[40] R. Lajara, J. Pelegrí-Sebastiá, and J. J. P. Solano, "Power consumption analysis of operating systems for wireless sensor networks," *Sensors*, vol. 10, no. 6, pp. 5809–5826, 2010.

[41] M. O. Farooq and T. Kunz, "Operating Systems for Wireless Sensor Networks: A Survey," *Sensors*, vol. 11, no. 12, pp. 5900–5930, may 2011. [Online]. Available: http://www.mdpi.com/1424-8220/11/6/5900/

[42] A. Musaddiq, Y. B. Zikria, O. Hahm, H. Yu, A. K. Bashir, and S. W. Kim, "A survey on resource management in iot operating systems," *IEEE Access*, vol. 6, pp. 8459–8482, 2018.

[43] T. Reusing, "Comparison of operating systems tinyos and contiki," *Sens. Nodes-Oper. Netw. Appl.(SN)*, vol. 7, pp. 7–13, 2012.

[44] G. C. Pereira, R. C. Alves, F. L. d. Silva, R. M. Azevedo, B. C. Albertini, and C. B. Margi, "Performance evaluation of cryptographic algorithms over iot platforms and operating systems," *Security and Communication Networks*, vol. 2017, 2017.

[45] P. Gaur and M. P. Tahiliani, "Operating systems for iot devices: A critical survey," in *2015 IEEE Region 10 Symposium*, May 2015, pp. 33–36.

[46] Contiki, "Contiki: The Open Source OS for the Internet of Things kernel description," http://www.contiki-os.org, accessed: 2019-11-03.

[47] F. Osterlind, A. Dunkels, J. Eriksson, N. Finne, and T. Voigt, "Cross-level sensor network simulation with cooja," in *Proceedings. 2006 31st IEEE Conference on Local Computer Networks*, Nov 2006, pp. 641–648.

[48] A. Velinov and A. Mileva, "Running and testing applications for contiki os using cooja simulator," in *Information Technology and Education Development - ITRO 2016*, 2016.

[49] A. Dunkels, J. Eriksson, N. Finne, and N. Tsiftes, "Powertrace: Network-level power profiling for low-power wireless networks," SICS, Tech. Rep., 2011. [Online]. Available: http://www.diva-portal.org/smash/get/diva2:1042895/FULLTEXT01.pdf

[50] A. Dunkels, "The ContikiMAC Radio Duty Cycling Protocol," SICS, Tech. Rep., 2011. [Online]. Available: http://soda.swedish-ict.se/5128/1/contikimac-report.pdf

[51] J. Saraswat and P. P. Bhattacharya, "Effect of duty cycle on energy consumption in wireless sensor networks," *International Journal of Computer Networks Communications*, vol. 5, no. 1, p. 125, 2013.

[52] R. C. Carrano, D. Passos, L. C. S. Magalhaes, and C. V. N. Albuquerque, "Survey and taxonomy of duty cycling mechanisms in wireless sensor networks," *IEEE Communications Surveys Tutorials*, vol. 16, no. 1, pp. 181–194, First 2014.

[53] S. Bartariya and A. Rastogi, "Security in wireless sensor networks: Attacks and solutions," *International Journal of Advanced Research in Computer and Communication Engineering*, vol. 5, no. 3, 2016.

[54] W. Dargie and C. Poellabauer, *Fundamentals of wireless sensor networks: theory and practice*. John Wiley & Sons, 2010.

[55] M. Marwaha, R. Bedi, A. Singh, and T. Singh, "Comparative analysis of cryptographic algorithms," *International Journal of Advanced Engineering Technology*, vol. 16, p. 18, 2013.

[56] R. Bhanot and R. Hans, "A review and comparative analysis of various encryption algorithms," *International Journal of Security and Its Applications*, vol. 9, no. 4, pp. 289–306, 2015.

[57] L. Casado and P. Tsigas, "Contikisec: A secure network layer for wireless sensor networks under the contiki operating system," in *Nordic Conference on Secure IT Systems*. Springer, 2009, pp. 133–147.

[58] G. Padmavathi and D. Shanmugapriya, "A survey of attacks, security mechanisms and challenges in wireless sensor networks," *International Journal of Computer Science and Information Security(IJCSIS)*, vol. 4, no. 1, 2009.

[59] V. Kumar, A. Jain, and P. Barwal, "Wireless sensor networks: security issues, challenges and solutions," *International Journal of Information and Computation Technology (IJICT)*, vol. 4, no. 8, pp. 859–868, 2014.

[60] S. Sharma and S. K. Jena, "A survey on secure hierarchical routing protocols in wireless sensor networks," in *Proceedings of the 2011 International Conference on Communication, Computing & Security*. ACM, 2011, pp. 146–151.

[61] R. Beyah, J. McNair, and C. Corbett, *Security in Ad Hoc and Sensor Networks*, ser. Computer and network security. World Scientific, 2010. [Online]. Available: https://books.google.co.uk/books?id=zsBpDQAAQBAJ

[62] S. Khan, A. Pathan, and N. Alrajeh, *Wireless Sensor Networks: Current Status and Future Trends*. CRC Press, 2016. [Online]. Available: https://books.google.co.uk/books?id=A1DOBQAAQBAJ

[63] S. K. Singh, M. Singh, and D. Singh, "A survey of energy-efficient hierarchical cluster-based routing in wireless sensor networks," *International Journal of Advanced Networking and Application (IJANA)*, vol. 2, no. 02, pp. 570–580, 2010.

[64] K. Chelli, "Security issues in wireless sensor networks: Attacks and countermeasures," in *Proceedings of the World Congress on Engineering*, vol. 1, no. 20, 2015.

[65] T.-G. Lupu, I. Rudas, M. Demiralp, and N. Mastorakis, "Main types of attacks in wireless sensor networks," in *WSEAS international conference. proceedings. recent advances in computer engineering*, no. 9. WSEAS, 2009.

[66] S. Saleem, S. Ullah, and K. S. Kwak, "A study of ieee 802.15. 4 security framework for wireless body area networks," *Sensors*, vol. 11, no. 2, pp. 1383–1395, 2011.

[67] N. Mukherjee, S. Neogy, and S. Roy, *Building Wireless Sensor Networks: Theoretical and Practical Perspectives*. CRC Press, 2015.

[68] R. Daidone, G. Dini, and G. Anastasi, "On evaluating the performance impact of the ieee 802.15.4 security sub-layer," *Computer Communications*, vol. 47, pp. 65 – 76, 2014.

[69] S. Raza, S. Duquennoy, J. Höglund, U. Roedig, and T. Voigt, "Secure communication for the internet of things-a comparison of link-layer security and ipsec for 6lowpan," *Security and Communication Networks*, vol. 7, no. 12, pp. 2654–2668, 2014.

[70] M. J. Dworkin, "Sp 800-38c. recommendation for block cipher modes of operation: The ccm mode for authentication and confidentiality," Gaithersburg, MD, United States, Tech. Rep., 2004.

[71] S. Roy and M. J. Nene, "Methodology for Deploying a Security Framework in Mission Critical Infrastructure Based Wireless Sensor Networks," *International Journal of Computer Science and Information Technologies(IJCSIT)*, vol. 6, no. 6, pp. 5478–5486, 2015.

[72] R. Tripathi and S. Agrawal, "Comparative study of symmetric and asymmetric cryptography techniques," *International Journal of Advance Foundation and Research in Computer (IJAFRC), ISSN*, pp. 2348–4853, 2014.

[73] M. Chowdhury, M. F. Kader *et al.*, "Security issues in wireless sensor networks: A survey," *International Journal of Future Generation Communication and Networking*, vol. 6, no. 5, pp. 97–116, 2013.

[74] P. A. Jatoi, A. A. Memon, B. S. Chowdhry, M. G. Ullah, and S. Latif, "An efficient hybrid cryptographic algorithm, consuming less time for exchanging information in wireless sensor networks," *Wireless Personal Communications*, vol. 85, no. 2, pp. 449–462, 2015.

[75] V. Kapoor, V. S. Abraham, and R. Singh, "Elliptic curve cryptography," *Ubiquity*, vol. 2008, no. May, pp. 7:1–7:8, May 2008.

[76] S. B. Sasi, D. Dixon, J. Wilson, and P. No, "A general comparison of symmetric and asymmetric cryptosystems for wsns and an overview of location based encryption technique for improving security," *IOSR Journal of Engineering*, vol. 4, no. 3, p. 1, 2014.

[77] N. Singhal and J. Raina, "Comparative analysis of aes and rc4 algorithms for better utilization," *International Journal of Computer Trends and Technology*, vol. 2, no. 6, pp. 177–181, 2011.

[78] X. Zhang, H. M. Heys, and C. Li, "Energy efficiency of symmetric key cryptographic algorithms in wireless sensor networks," in *2010 25th Biennial Symposium on Communications*, May 2010, pp. 168–172.

[79] S. Ahmad, M. R. Beg, Q. Abbas, J. Ahmad, and S. Atif, "Comparative study between stream cipher and block cipher using rc4 and hill cipher," *International Journal of Computer Applications*, vol. 1, no. 25, pp. 0975–8887, 2010.

[80] F. P. Miller, A. F. Vandome, and J. McBrewster, *Advanced Encryption Standard*. Alpha Press, 2009.

[81] S. J. Shepherd, "The tiny encryption algorithm," *Cryptologia*, vol. 31, no. 3, pp. 233–245, 2007.

[82] A. V. Taddeo, M. Mura, and A. Ferrante, "Qos and security in energy-harvesting wireless sensor networks," in *2010 International Conference on Security and Cryptography (SECRYPT)*, July 2010, pp. 1–10.

[83] "The price of security in wireless sensor networks," *Computer Networks*, vol. 54, no. 17, pp. 2967 – 2978, 2010.

[84] J. M. Kim, H. S. Lee, J. Yi, and M. Park, "Power adaptive data encryption for energy-efficient and secure communication in solar-powered wireless sensor networks," *Journal of Sensors*, vol. 2016, 2016.

[85] M. Sharafi, F. Fotouhi-Ghazvini, M. Shirali, and M. Ghassemian, "A low power cryptography solution based on chaos theory in wireless sensor nodes," *IEEE Access*, vol. 7, pp. 8737–8753, 2019.

[86] E. TOMUR, "Security and quality of service for wireless sensor networks," Ph.D. dissertation, Citeseer, 2008.

[87] A. K. Lenstra, "Key length. contribution to the handbook of information security," http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.694.8206, 2004.

[88] A. k. Lenstra, "Unbelievable security matching aes security using public key systems," in *International Conference on the Theory and Application of Cryptology and Information Security*. Springer, 2001, pp. 67–86.

[89] A. Rachedi and A. Hasnaoui, "Advanced quality of services with security integration in wireless sensor networks," *Wireless Communications and Mobile Computing*, vol. 15, no. 6, pp. 1106–1116, apr 2015. [Online]. Available: http://doi.wiley.com/10.1002/wcm.2562

[90] I. R. M. Association, *IT Policy and Ethics: Concepts, Methodologies, Tools, and Applications*, 1st ed. Hershey, PA, USA: IGI Global, 2013.

[91] J. E. Mbowe and G. S. Oreku, "Quality of service in wireless sensor networks," *Wireless Sensor Network*, vol. 6, no. 02, pp. 19–26, 2014.

[92] D. Rusinek, B. Ksiezopolski, and A. Wierzbicki, "Security trade-off and energy efficiency analysis in wireless sensor networks," *International Journal of Distributed Sensor Networks*, 2015.

[93] N. Zaman, *Wireless Sensor Networks and Energy Efficiency: Protocols, Routing and Management: Protocols, Routing and Management*. IGI Global, 2012.

[94] M. T. Lazarescu, "Design of a wsn platform for long-term environmental monitoring for iot applications," *IEEE Journal on emerging and selected topics in circuits and systems*, vol. 3, no. 1, pp. 45–54, 2013.

[95] Y. Touati, B. Daachi, and A. Arab, *Energy Management in Wireless Sensor Networks*. Elsevier Science, 2017. [Online]. Available: https://books.google.co.uk/books?id=4p0RDQAAQBAJ

[96] O. K. Sahingoz, "Large scale wireless sensor networks with multi-level dynamic key management scheme," *Journal of Systems Architecture*, vol. 59, no. 9, pp. 801–807, 2013.

[97] V. Raghunathan, C. Schurgers, S. Park, and M. B. Srivastava, "Energy-aware wireless microsensor networks," *IEEE Signal processing magazine*, vol. 19, no. 2, pp. 40–50, 2002.

[98] K. Sohrabi, J. Gao, V. Ailawadhi, and G. J. Pottie, "Protocols for self-organization of a wireless sensor network," *IEEE personal communications*, vol. 7, no. 5, pp. 16–27, 2000.

[99] S. Sicari, A. Rizzardi, L. A. Grieco, and A. Coen-Porisini, "Security, privacy and trust in internet of things: The road ahead," *Computer Networks*, vol. 76, pp. 146–164, 2015.

[100] X. Zhang, H. M. Heys, and C. Li, "Energy efficiency of encryption schemes applied to wireless sensor networks," *Security and Communication Networks*, vol. 5, no. 7, pp. 789–808, 2012.

[101] C. Chigan, L. Li, and Y. Ye, "Resource-aware self-adaptive security provisioning in mobile ad hoc networks," in *IEEE Wireless Communications and Networking Conference, 2005*, vol. 4. IEEE, 2005, pp. 2118–2124.

[102] M. J. Dworkin, *Recommendation for block cipher modes of operation: The CCM mode for authentication and confidentiality*, 2007.

[103] P. Demeester, I. Moerman, and A. Terzis, *Wireless Sensor Networks: 10th European Conference, EWSN 2013, Ghent, Belgium, February 13-15, 2013, Proceedings*. Springer, 2013, vol. 7772.

[104] G. De Meulenaer, F. Gosset, F.-X. Standaert, and O. Pereira, "On the energy cost of communication and cryptography in wireless sensor networks," in *2008 IEEE International Conference on Wireless and Mobile Computing, Networking and Communications*. IEEE, 2008, pp. 580–585.

[105] A. Rachedi and A. Hasnaoui, "Advanced quality of services with security integration in wireless sensor networks," *Wireless Communications and Mobile Computing*, vol. 15, no. 6, pp. 1106–1116, 2015.

[106] G. V. Merrett, N. R. Harris, B. M. Al-Hashimi, and N. M. White, "Energy managed reporting for wireless sensor networks," *Sensors and Actuators A: Physical*, vol. 142, no. 1, pp. 379–389, 2008.

[107] M. Hamdi and H. Abie, "Game-based adaptive security in the internet of things for ehealth," in *2014 IEEE International Conference on Communications (ICC)*, June 2014, pp. 920–925.

[108] G. D. Abowd, A. K. Dey, P. J. Brown, N. Davies, M. Smith, and P. Steggles, "Towards a better understanding of context and context-awareness," in *International symposium on handheld and ubiquitous computing*. Springer, 1999, pp. 304–307.

[109] A. K. Dey, G. D. Abowd, and D. Salber, "A conceptual framework and a toolkit for supporting the rapid prototyping of context-aware applications," *Human–Computer Interaction*, vol. 16, no. 2-4, pp. 97–166, 2001.

[110] P. Brezillon and G. K. Mostefaoui, "Context-based security policies: a new modeling approach," in *IEEE Annual Conference on Pervasive Computing and Communications Workshops, 2004. Proceedings of the Second*, March 2004, pp. 154–158.

[111] N. Gámez, J. Cubo, L. Fuentes, and E. Pimentel, "Configuring a context-aware middleware for wireless sensor networks," *Sensors*, vol. 12, no. 7, pp. 8544–8570, 2012.

[112] A. Perrig, R. Szewczyk, J. D. Tygar, V. Wen, and D. E. Culler, "Spins: Security protocols for sensor networks," *Wireless networks*, vol. 8, no. 5, pp. 521–534, 2002.

[113] C. Karlof, N. Sastry, and D. Wagner, "Tinysec: a link layer security architecture for wireless sensor networks," in *Proceedings of the 2nd international conference on Embedded networked sensor systems*. ACM, 2004, pp. 162–175.

[114] Y.-T. Wang and R. Bagrodia, "Sensec: A scalable and accurate framework for wireless sensor network security evaluation," in *Distributed Computing Systems Workshops (ICDCSW), 2011 31st International Conference on*. IEEE, 2011, pp. 230–239.

[115] M. Luk, G. Mezzour, A. Perrig, and V. Gligor, "Minisec: a secure sensor network communication architecture," in *Proceedings of the 6th international conference on Information processing in sensor networks*. ACM, 2007, pp. 479–488.

[116] A. G. Ganek and T. A. Corbi, "The dawning of the autonomic computing era," *IBM systems Journal*, vol. 42, no. 1, pp. 5–18, 2003.

[117] S. Misra and S. Goswami, *Network Routing: Fundamentals, Applications, and Emerging Technologies*. John Wiley & Sons, 2017.

[118] S. B. Othman, A. Trad, and H. Youssef, "Performance evaluation of encryption algorithm for wireless sensor networks," in *Information Technology and e-Services (ICITeS), 2012 International Conference on.* IEEE, 2012, pp. 1–8.

[119] A. Trad, A. A. Bahattab, and S. Ben Othman, "Performance trade-offs of encryption algorithms for wireless sensor networks," in *2014 World Congress on Computer Applications and Information Systems (WCCAIS)*, Jan 2014, pp. 1–6.

[120] J. Misic and V. Misic, *Wireless personal area networks: Performance, interconnection, and security with IEEE 802.15. 4.* John Wiley & Sons, 2008, vol. 1.

[121] A. Abduvaliyev, A.-S. K. Pathan, J. Zhou, R. Roman, and W.-C. Wong, "On the vital areas of intrusion detection systems in wireless sensor networks," *IEEE Communications Surveys & Tutorials*, vol. 15, no. 3, pp. 1223–1237, 2013.

[122] G. Raju, D. Ghosh, T. S. Kumar, S. Kavyashree, and V. Nagaveni, "Wireless sensor network lifetime optimization," 2011.

[123] Y. Wu, S. Fahmy, and N. B. Shroff, "On the construction of a maximum-lifetime data gathering tree in sensor networks: Np-completeness and approximation algorithm," in *IEEE INFOCOM 2008-The 27th Conference on Computer Communications.* IEEE, 2008, pp. 356–360.

[124] M. Dener, "Security analysis in wireless sensor networks," *International Journal of Distributed Sensor Networks*, vol. 10, no. 10, p. 303501, 2014.

[125] J. Horneber and A. Hergenröder, "A survey on testbeds and experimentation environments for wireless sensor networks," *IEEE Communications Surveys Tutorials*, vol. 16, no. 4, pp. 1820–1838, Fourthquarter 2014.

[126] A.-S. Tonneau, N. Mitton, and J. Vandaele, "How to choose an experimentation platform for wireless sensor networks? a survey on static and mobile wireless sensor network experimentation facilities," *Ad Hoc Networks*, vol. 30, pp. 115–127, 2015.

[127] K. Roussel, Y.-Q. Song, and O. Zendra, "Lessons Learned through Implementation and Performance Comparison of Two MAC/RDC Protocols on Different WSN OS," INRIA Nancy, Research Report RR-8777, Mar. 2015.

Appendix A

Publications

The following are some of the publications which have been contributed during the PHD study:

2018 9th International Conference on Information and Communication Systems (ICICS)

Impact of Duty Cycle Protocols on Security Cost of IoT

Sultan Alharby, Nick Harris, Alex Weddell, Jeff Reeve
Electronics and Computer Science Department
Southampton University
Southampton, UK
sa1c15, nrh, asw, jsr@ecs.soton.ac.uk

***Abstract*—With the evolution of IoT embedded devices and their broad application, security has become crucial. Security comes at a cost to these limited devices in terms of energy. However, evaluating security cost is not straightforward, as there are many factors involved, such as the employed security services and type of hardware. This research studies the impact of duty cycle protocols on security cost in IoT embedded devices. It begins by evaluating the cost of security on a per-packet basis, and then evaluates how duty cycle protocols could affect security cost. The research demonstrates the relationship between duty cycle protocols and security cost, which could be a source of confusion when measuring the actual security cost in a real scenario.**

***Index Terms*—Security Cost, Duty Cycle Protocol, Energy Consumption**

I. Introduction

The concept of the Internet of Things (IoT) has received much attention over the last five years. It is predicted that the IoT will influence every aspect of our lifestyles in the near future [1] . The IoT embedded devices are one of the key enablers of the operation of IoTs, allowing data to be collected from the surrounding environment. However, due to their limited resources, nature of deployment and unattended operation, IoT embedded devices are vulnerable to various types of attack. Security is paramount and essential for reliable and safe communication between IoT embedded devices [2] [3]. Therefore, utilising the limited resources available to protect the communication between these embedded devices is a complex task. These embedded devices are usually equipped with small batteries [4], which makes energy conservation crucial. Security cost is not straightforward [5], as many factors affect the obtained result such as the employed security services and type of hardware.

In this research, security cost is defined as the energy consumption which is additional to the packet transmission for security services. Most published researches have studied the cost of security based on packet-basis. Most of these studies has focused on studying the extra bytes added by security and the required time for Microcontroller Unit (MCU) to process complex security algorithms. This methodology is important and has been demonstrated by many researchers. Evaluation by this methodology has shown that security services such as encryption, replay protection and authentication add overhead to the packet being transmitted [6] [7] [8]. This overhead is expressed by the extra time required by radio to transmit the added bytes for authentication service and the time needed by the MCU to process complex encryption/decryption and authentication algorithms. However, this overhead is just represents the cost of security for transmitting one packet with the security header, and therefore is only part of the actual security overhead. What is missing is the impact of duty cycle protocols on the cost of security.

This research focuses on how duty cycle could affect the security cost at the data link layer. There are many different types of duty cycle protocols for an IoT embedded system, and these protocols ensure that radio is turned off as much as possible, while allowing different embedded devices to communicate. Security services affect radio energy consumption and therefore their cost is affected by the duty cycle. This has been neglected by previous researches when evaluating security cost. To study whether duty cycle protocols affect the security cost at data link layer, we must first understand the cost of security per packet, and prove that security does add cost in terms of energy. This will help us to understand the communication and computation overhead added by security. Then, it is necessary to study the work mechanism of the employed duty cycle protocol to investigate whether it affects security overhead.

II. Related Work

Many researchers have studied the cost of security for IoT embedded devices at the data link layer. In [7] the cost of encryption algorithms with different modes of operation has been studied. The evaluation uses MicaZ and TelosB hardware. The results show the cost of different types of symmetric encryption. Another research, conducted by [9], proposes an optimised implementation of Advanced Encryption Standard (AES) which uses a hardware accelerator with different modes of operation. The research compares it with software implementation in terms of energy consumption, and concludes that hardware implementation is more efficient. In [10], the authors have studied and compared AES, RC5 and RC6 encryption, and the results indicate that, among the three

978-1-5386-4366-2/18/$31.00 ©2018 IEEE

evaluated encryption algorithms, AES is the most expensive in terms of energy consumption. These researches are useful for explaining the security cost on a per packet-basis. However, what is missing is an evaluation of the impact of duty cycle on the security cost. The cost of security depends on the number of security services invocation. Hence, duty cycle protocols might affect the overall security cost. This, to the authors knowledge, is the only research that evaluates the relationship between duty cycle protocols and security cost for IoT embedded devices.

III. IEEE 802.15.4 Security Specification

In this section we present the security services covered in the evaluation. IEEE 802.15.4 security specification is used in this research to measure the security cost of IoT embedded devices. This is the most widely used security specification in IoT embedded devices at the data link layer. IEEE 802.15.4 defines the security requirements in the data link layer [11], and supports security techniques to protect the wireless communication from possible attack. These techniques assure the confidentiality, integrity, authenticity and replay protection on a per-frame basis [12]. IEEE 802.15.4 offers two operational modes: a beacon-enabled mode, and a non-beacon enabled mode [13]. In the former case, the network communication is managed by a coordinator [11]. The coordinator sends regular beacons to synchronise embedded devices and manage all communication. With beacon-disabled mode, every embedded device can access the channel through a CSMA/CA protocol which is the used mode in this evaluation. IEEE 802.15.4 security layer is controlled at the data link layer, and the security requirements are specified at the application layer [11]. If no security mechanism is chosen at the application layer, then communication will be unsecured.

IEEE 802.15.4 offers a security suite which supports eight different security levels, as shown in Table I. Each security level supports specific security requirements and has a different frame format. These security levels generally offer no security, encryption only (AES-CTR), authentication and integrity only (AES-CBC-MAC), or all three security services: encryption, integrity and authentication (AES-CCM). AES-CBC-MAC uses three different MIC lengths: 4, 8 and 16 bytes [14]. Also, CCM supports a high level of security with the option of 4, 8 and 16 bytes. MIC is used to guarantee that data has not been changed, and also guarantee that data has originated from a legitimate source. The length of MIC represents the strength of integrity and authentication. The name of each level consists of two to three parts. The first part indicates the cryptography scheme (AES if security parameters is enabled). The second part indicates the mode of operation used in the cryptography scheme. The last part, if applicable, indicates the Message Integrity Code (MIC), which can be of varying length. Advanced Encryption Standard (AES) cipher is used in the standard with a fixed block size of 128 bits and different key lengths of 128, 192 or 256 bits [13]. However, the employed key length is 128 bits. An unsecured frame consists of three fields: a MAC header with 7 to 23 bytes, data payloads with 0 to 115 bytes, and Frame Check Sequences (FCS) with 2 bytes [12]. A secured frame has one more field named Auxiliary Security Header (ASH), with 5 to 14 bytes. This is in addition to the Message Integrity Code (MIC) if authentication is enabled. One of the contents of ASH is the Frame Counter with 4 bytes for replay attack detection. Frame counter is set by the outgoing frame at the transmitter side. The frame counter field is checked at the receiving embedded devices, and is accepted if the value is higher than the previous received value.

TABLE I
Security Suites, Reproduced from [15]

SiuteID	Describtion	Security Services	MIC Length
0	No Security	Null	0
1	AES-CBC-MAC-64		4
2	AES-CBC-MAC-64	Authentication	8
3	AES-CBC-MAC-128		16
4	AES-CTR	Encryption only	0
5	AES-CCM-32	Authentication	4
6	AES-CCM-64	and	8
7	AES-CCM-128	Encryption	16

IV. Experimental Setup

This research uses the Contiki OS developed by a world-wide community of experts.Contiki is a highly portable open source OS and runs on many different wireless sensor platforms. It supports a simulator called Cooja [16], which allows developers to debug and simulate their applications on large-scale networks. Cooja is used in this research for simulation. Cooja provides a means of estimating the power consumption of the radio and MCU. The Powertrace tool [17], which is supported in Contiki, is used to provide detailed information about where power is being consumed (transmission, receiving, etc), and calculates the time each component spends in a particular mode. Cooja can emulate real hardware. Sky hardware is emulated in the simulator. The two components affected by security services are the MCU and radio. Hence, the following formula is used to measure the energy consumed by security:

$$E_{sec} = E_{sec-compu} + E_{sec-comm} \quad (1)$$

where E_{sec} the energy consumed by security, $E_{sec-compu}$ computes the energy consumed by MCU for security, and $E_{sec-comm}$ the energy required by the radio to send the extra bytes neccessary for security. The cost of transmitting a packet without security services will be used as a *baseline* to calculate security cost. A higher security level is associated with higher security services, which will generally consume more energy. Each security level will be evaluated in terms of energy, and the difference between the current and baseline level is the security cost for that particular security level. The current drawn by Tmote sky components in different modes is required

to calculate energy consumption. Tmote sky uses cc2420 as transceiver and MSP430 as a micro-controller. The current drawn by these components is shown in Table II.

TABLE II
TYPICAL CURRENT CONSUMPTION FOR TMOTE SKY

Component	Current drawn
MCU- Active state	2.4mA
Radio - Transmitting mode	17.4mA
Radio - Receiving mode	19.7mA

The parameters which used in this evaluation are shown in Table III

TABLE III
SIMULATION PARAMETERS

Parameter	Value
Platform	Tmote Sky
MAC protocol	CSMA
Radio Duty Cycle	ContikiMAC
Payload	24 byte
Transmission range	50 Meters
TX/RX success ratio	100%
Radio	CC2420
Microcontroller unit (MCU)	MSP430

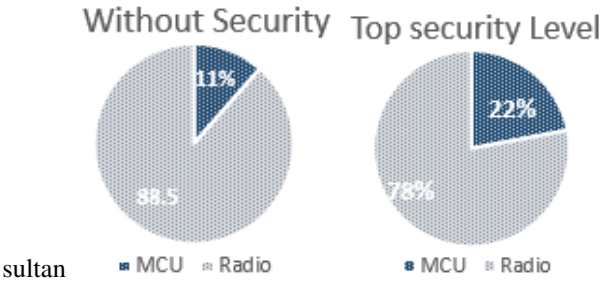
Energy consumption for each component can be measured by calculating the time each component spends in a certain mode (receiving, transmitting, idle...). The following formula is used to achieve this:

$$E = Energest\_Value * Voltage * Current \\ / RTIMER\_SECOND * runtime \quad (2)$$

Where, E is the amount of energy consumed by an embedded device's component in a particular mode, $Energest\_Value$ is the difference in ticks between two interval times, and $RTIMER\_SECOND$ is the number of ticks per second, which is equal to 32768 ticks/second in this simulation. The evaluation has been repeated for each security level. The results are shown in Table IV

V. SECURITY COST

Table IV shows the energy consumption of both MCU and radio when transmitting a packet with 24 bytes payload for all IEEE 802.15.4 security levels. As can be seen from the table IV and Figure 1 that radio is the main contributor to energy consumption. It constitutes 73.7% of the total energy consumption at level 0, and 88.5% at security level 7. This indicates the important of duty cycle protocol as it responsible for controlling radio. MCU consumes only 12% at level 0, and this increases as the code grows in complexity with higher security levels. In general the security cost at the first three levels fluctuates between 31.54%, at security level 1, and 60.46% at top security level based on the MIC lengths[4, 8, 16 bytes]. Increasing the MIC length used for authentication would keep the radio active for longer, allowing more bytes to be sent. This explains the great energy consumption when enabling authentication. The cost of security at the fourth level is almost 33% since only encryption is supported. The last three security levels support authentication, encryption, integrity, and replay protection attack. Hence, they are the most energy consuming levels among all security levels. The only difference between the last three levels is authentication length, which can be as described in the first three levels 4, 8, and 16 bytes. The results shows an overhead added by security services which could shorten the network lifetime significantly.



sultan

Fig. 1 Radio consumption vs MCU consumption for level 0 and 7

VI. DUTY CYCLE PROTOCOLS

The main objective of duty cycle protocols is to disable the radio as often as possible [18], while allowing low power devices to communicate with each other at minimum requirement. This technique of controlling the radio state to save energy is called duty cycling [19]. To achieve this, several protocols have been proposed in the literature which trade-off these requirements and extend the network lifetime. These protocols can be classified into synchronous, asynchronous and semi-synchronous duty cycles [20], in relation to the mechanism employed to control the radio module. In a synchronous scheme embedded devices are time synchronised [20], hence all embedded devices wake up and sleep at a set time. Contrary to this, asynchronous embedded devices do not need to work simultaneously, and they have no agreed wakeup/sleep schedule. A semi-synchronous duty cycle combines the two methods by grouping neighbour embedded devices into a synchronised cluster where different clusters communicate with each other asynchronously. This research will discuss duty cycle from the perspective of security cost.

A. Impact of Duty Cycle on Security Cost

Security cost is based on the number of security services invocation, which in this case is AES. The relation between security cost and duty cycle is that some duty cycle protocols increase the number of transmitted packet by re-transmission. This means greater AES invocation, which in-turn leads to greater energy consumption. To clarify, a typical duty cycle protocol, named ContikiMAC [18], is discussed as an example. This protocol uses an asynchronous mechanism, and supports two methods of transmission: unicast and broadcast.

TABLE IV
ENERGY CONSUMPTION OF TRANSMITTING ONE PACKET WITH A PAYLOAD OF 24 BYTE IN DIFFERENT SECURITY LEVELS

Security level	MCU energy consumption (μJ)	Radio energy consumption (μJ)	Total energy consumption (μJ)	Percentage of increased security overhead over non-secure packet (%)
0	9.53	73.28	82.81	-
1	24.01	84.91	108.926	31.54%
2	24.15	92.39	116.54	40.72%
3	24.32	103.546	127.87	54.4%
4	28.95	81.24	110.19	33%
5	28.5	83.8	112.3	35.6%
6	29.11	90.80	119.91	44.8%
7	29.33	103.55	132.88	60.46%

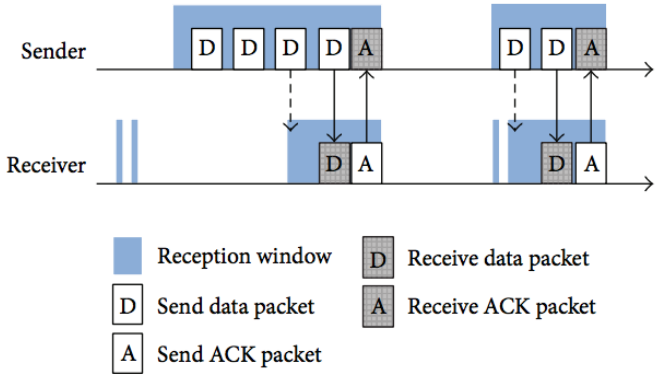


Fig. 2 Unicast transmission of Contiki-MAC.

1) Unicast Transmission: In a unicast transmission, as shown in Figure 2, the sender checks the channel before transmission, and if it is free, the whole packet is sent repeatedly until the receiver wakes up and returns an acknowledgement. It is clear that successful delivery of one packet might require the transmission of many packets from the sender side. With every packet transmission the employed security services are invoked. Hence, the energy consumption for security processes is increased by the ContikiMAC duty cycle. To evaluate that in the simulation, we will conduct an experiment containing two nodes. Each node transmits packets to the other node. The employed duty cycle protocol in this experiment is ContikiMAC. Figure 3, which obtained from the simulator, shows the transmission for both $node1$ and $node2$. As can be noticed that one successful packet delivery requires multiple transmissions. The figure shows that, $node1$ transmits 4 packets before the receiver $node2$ wakes up and sends an acknowledgement, whereas $node2$ needs to send 7 packets before receiving acknowledgement from $node1$. The number of AES invocations in node 1 is less than in node 2. This retransmission is repeated with each packet delivery, which clearly demonstrates the effects of duty cycle on security cost.

To evaluate the effect of duty cycle on security cost, the PowerTracker tool supplied with the Cooja simulator is used. PowerTracker can present a detailed information on the total time the radio spends in active, receiving and transmitting modes. Figure 4 depicts the percentage of duty cycle for both

Fig. 3 Unicast transmission of Contiki-MAC.

$node1$ (Sky1 in the figure) and $node2$ (Sky2 in the figure) in all three modes. The $node1$ radio module is active for 5.72% of the time, while the radio in $node2$ is active for 6.87%. In both nodes, most energy is consumed during transmission mode, which explains the retransmission and AES invocations effects on the actual packet energy consumption. Security cost fluctuates between 31% and 60% per packet based on the selected security level, and this means that with each transmission attempt caused by duty cycle, there is extra overhead for security. Hence, the the actual security cost evaluation should take duty cycle into consideration.

Mote	Radio on (%)	Radio TX (%)	Radio RX (%)
Sky 1	5.72%	1.39%	0.37%
Sky 2	6.87%	2.41%	0.37%
AVERAGE	6.29%	1.90%	0.37%

Fig. 4 Duty cycle evaluation for the two nodes.

To get more realistic evaluation for actual implementation, the number of delivered packets versus the retransmission tries for each node have been recorded for five minutes, as shown in Figure 5. Each node delivered 306 successful packets to the other node. However, $node1$ retransmitted 1224 packets to get the 306 packets delivered, while in $node2$ 2142 packets. This significant difference in retransmission is definitely affect the number of security features invocations, and consequently, affect the total security energy consumption.

2) Broadcast Transmission: in contrast, the broadcast transmission in ContikiMAC constantly sends the same packet repeatedly for a full interval wake-up time to ensure that all embedded devices have received the packet. There is no acknowledgement in the broadcast transmission, as illustrated in Figure 6. This mechanism consumes more energy compared to unicast transmission, as the security

features will be invoked repeatedly for a full interval wake-up time.

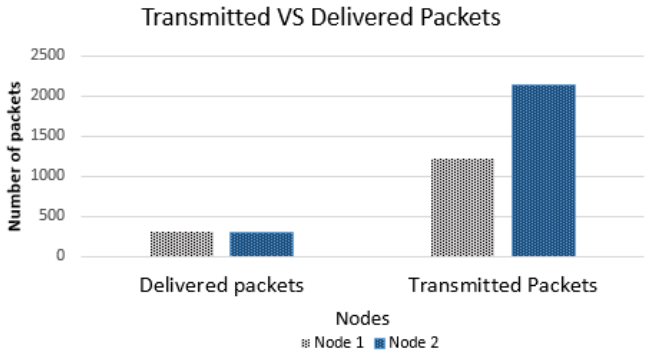


Fig. 5 Comparison between delivered and transmitted packets.

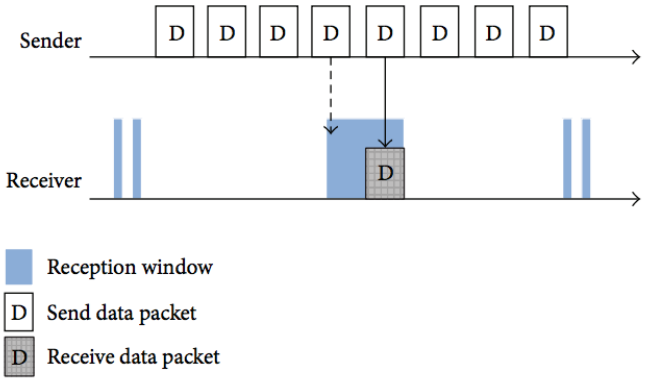


Fig. 6 Broadcast transmission of ContikiMAC.

The duty cycle protocols do not always affect the security cost; this depends on their work mechanism. The ContikiMAC protocol, as discussed previously, sends the whole packets repeatedly which requires multiple security invocations. Other protocols may use different mechanism allow for transmission to take place only when the receiver is active, so there is only one transmission for packet delivery. This is better in terms of security cost as there is only one security services invocation. For example, X-MAC [21] and LPL(Low Power Listening) [22]protocols uses short and long pre-amble to guarantee that the embedded receiver device has sufficient time to detect the pr-eamble and remain active before transmitting any packet from the sender. Hence, there is only one packet transmission, which means only one AES invocation.

VII. CONCLUSION

Security is crucial to IoT embedded devices. However, security comes at a cost to these constraint devices. The main issue here is that the exact security cost is still not known, as there are many influencing factors, such as the utilised security services and the used duty cycle protocol. This research has highlighted the impact of duty cycle protocols on security cost, an area which has been neglected by previous papers. First, it evaluates the security cost on a per packet-basis, and then prove that it adds significant overhead. The results show that security cost depends on the number of security features which are invoked. The research goes on to evaluate how duty cycle protocols could affect security cost. It clarifies the effect of increasing the number of AES invocations by some duty cycle protocols. Also, It has shown that not all duty cycle protocols affect security cost, and that this depends on the number of packet retransmissions for a single packet. Different duty cycle protocols lead to different results. Therefore, engineers should take duty cycle into consideration when evaluating security cost.

REFERENCES

[1] J. A. Stankovic, "Research Directions for the Internet of Things," *IEEE Internet of Things Journal*, vol. 1, no. 1, pp. 3–9, feb 2014. [Online]. Available: http://ieeexplore.ieee.org/document/6774858/

[2] H. Modares, R. Salleh, and A. Moravejosharieh, "Overview of security issues in wireless sensor networks," in *Computational Intelligence, Modelling and Simulation (CIMSiM), 2011 Third International Conference on*. IEEE, 2011, pp. 308–311.

[3] S. Sciancalepore, G. Piro, E. Vogli, G. Boggia, and L. A. Grieco, "On securing ieee 802.15. 4 networks through a standard compliant framework," in *Euro Med Telco Conference (EMTC), 2014*. IEEE, 2014, pp. 1–6.

[4] S. K. Singh, M. Singh, and D. Singh, "A survey of energy-efficient hierarchical cluster-based routing in wireless sensor networks," *International Journal of Advanced Networking and Application (IJANA)*, vol. 2, no. 02, pp. 570–580, 2010.

[5] S. Alharby, N. Harris, A. Weddell, and J. Reeve, "The security trade-offs in resource constrained nodes for iot application," *International Journal of Electrical, Computer, Energetic, Electronic and Communication Engineering*, vol. 12, no. 1, pp. 52 – 59, 2018. [Online]. Available: http://waset.org/Publications?p=133

[6] N. Dziengel, N. Schmittberger, J. Schiller, and M. Günes, "Secure communications for event-driven wireless sensor networks," in *Proc. of the 3rd Int. Symp. on Sensor Networks and Applications SNA*, 2011.

[7] J. Lee, K. Kapitanova, and S. H. Son, "The price of security in wireless sensor networks," *Computer Networks*, vol. 54, no. 17, pp. 2967–2978, 2010.

[8] M. Luk, G. Mezzour, A. Perrig, and V. Gligor, "Minisec: a secure sensor network communication architecture," in *Proceedings of the 6th international conference on Information processing in sensor networks*. ACM, 2007, pp. 479–488.

[9] C. Panait and D. Dragomir, "Measuring the performance and energy consumption of aes in wireless sensor networks," in *Computer Science and Information Systems (FedCSIS), 2015 Federated Conference on*. IEEE, 2015, pp. 1261–1266.

[10] A. Trad, A. A. Bahattab, and S. B. Othman, "Performance trade-offs of encryption algorithms for wireless sensor networks," in *Computer Applications and Information Systems (WCCAIS), 2014 World Congress on*. IEEE, 2014, pp. 1–6.

[11] S. Saleem, S. Ullah, and K. S. Kwak, "A study of ieee 802.15. 4 security framework for wireless body area networks," *Sensors*, vol. 11, no. 2, pp. 1383–1395, 2011.

[12] R. Daidone, G. Dini, and G. Anastasi, "On evaluating the performance impact of the ieee 802.15. 4 security sub-layer," *Computer Communications*, vol. 47, pp. 65–76, 2014.

[13] I. Standard and I. C. Society, "IEEE Standard for Local and metropolitan area networks Part 15 . 4 : Low-Rate Wireless Personal Area Networks (LR-WPANs) IEEE Computer Society S ponsored by the," *IEEE Std 802.15.4-2011*, vol. 2011, no. September, pp. 1–294, 2011. [Online]. Available: http://ieeexplore.ieee.org/servlet/opac?punumber=6012485

[14] S. Raza, S. Duquennoy, J. Höglund, U. Roedig, and T. Voigt, "Secure communication for the internet of thingsa comparison of link-layer security and ipsec for 6lowpan," *Security and Communication Networks*, vol. 7, no. 12, pp. 2654–2668, 2014.

[15] A. V. Taddeo, M. Mura, and A. Ferrante, "Qos and security in energy-harvesting wireless sensor networks," in *Security and Cryptography (SECRYPT), Proceedings of the 2010 International Conference on*. IEEE, 2010, pp. 1–10.

[16] F. Osterlind, A. Dunkels, J. Eriksson, N. Finne, and T. Voigt, "Cross-level sensor network simulation with cooja," in *Local computer networks, proceedings 2006 31st IEEE conference on*. IEEE, 2006, pp. 641–648.

[17] A. Dunkels, J. Eriksson, N. Finne, and N. Tsiftes, "Powertrace: Network-level power profiling for low-power wireless networks," 2011.

[18] A. Dunkels, "The ContikiMAC Radio Duty Cycling Protocol," SICS, Tech. Rep., 2011. [Online]. Available: http://soda.swedish-ict.se/5128/1/contikimac-report.pdf

[19] J. Saraswat and P. P. Bhattacharya, "Effect of duty cycle on energy consumption in wireless sensor networks," *International Journal of Computer Networks & Communications*, vol. 5, no. 1, p. 125, 2013.

[20] R. C. Carrano, D. Passos, L. C. Magalhaes, and C. V. Albuquerque, "Survey and taxonomy of duty cycling mechanisms in wireless sensor networks," *IEEE Communications Surveys & Tutorials*, vol. 16, no. 1, pp. 181–194, 2014.

[21] M. Buettner, G. V. Yee, E. Anderson, and R. Han, "X-mac: a short preamble mac protocol for duty-cycled wireless sensor networks," in *Proceedings of the 4th international conference on Embedded networked sensor systems*. ACM, 2006, pp. 307–320.

[22] J. L. Hill and D. E. Culler, "Mica: A wireless platform for deeply embedded networks," *IEEE micro*, vol. 22, no. 6, pp. 12–24, 2002.

Available online at www.sciencedirect.com

ScienceDirect

IFAC PapersOnLine 51-6 (2018) 72–77

IFAC Papers Online CONFERENCE PAPER ARCHIVE

The Cost of Link Layer Security in IoT Embedded Devices ⋆

S. Alharby, A. Weddell, J. Reeve, N. Harris *

* *School of Electronics and Computer Science, University of Southampton, Southampton, United Kingdom (e-mail: sa1c15, asw, jsr, nrh@ ecs.soton.ac.uk).*

Abstract:
Security of Internet of Things (IoT) devices is important for the acceptance of IoT applications. Several security mechanisms have been proposed, however, due to the limited resources in IoT devices, their overhead should be evaluated carefully. Many existing security solutions have been evaluated by simulation. However, with the increasing deployment of real nodes, a more accurate result based on real hardware is necessary. This research aims to evaluate the impact of applying security features of IEEE 802.15.4 on example hardware. The evaluation includes the impact on the Microcontroller Unit (MCU), radio, latency, packet throughput and transmission power. The results show that enabling software security services impact limited devices significantly. This impact could be translated into a decrease in throughput to almost 44%, an increase in latency and energy consumption to almost 197% and 86% respectively with maximum security level. The outcome of this paper is intended to benefit network designers and researchers by allowing them to model security overhead, allowing them to choose the security level, if enabled, which suits their application requirements.

Keywords: Hardware Security Evaluation, Security Evaluation, IoT security, IEEE 802.15.4 Security, WSN Link Layer Security.

1. INTRODUCTION

Smart homes, the smart grid, environmental monitoring, smart irrigation, and other similar applications have helped make the world more connected than ever before. The common vision associated with such distributed systems is linked to the paradigm of the Internet of Things (IoT). The IoT is not associated with a particular technology, and can be used in different applications. IEEE 802.15.4 standard defines only the two bottom layers of IoT stack protocols: physical and MAC layers. There are other standards responsible for upper layers like: Zigbee and 6LoWPAN. IEEE 802.15.4 provides an efficient power MAC protocol, and it also defines a specification for preserving the security at MAC layer. The connectivity in the IoT embedded devices could be directly to the internet or through a multi-hop communication to a gateway. The multi-hop communication through a gateway is called Wireless Sensor Network (WSN). WSN is a foundational technology for IoT Turkanović et al. (2014) Xu et al. (2014). Sensors are the main tools for reporting events in "things" such as cars, home appliances, and any 'thing' to which a sensor can be attached. WSNs have recently grabbed the attention of the academic and industrial communities Zhong et al. (2014) with a new type of wireless network that has many popular military and civilian applications. WSNs consist of hundreds or thousands of distributed sensors that monitor physical and environmental events. These events can include pressure, humidity, temperature, pollution, and many other parameters. The integration of different types of WSN applications and the internet would be a major technological evolution. The limited resources of most IoT embedded devices, particularly of energy, brings some challenges. Owing to its energy constraint, security is considered one of the top barriers for the proliferation of IoT devices, but is essential to many IoT applications such as smart-home and medical technologies. Thus, in order to provide a sufficient level of security while properly utilizing the available resources, it is important to have a good understanding of both the energy cost and the features of the available security mechanisms such as encryption and authentication. This can be achieved by conducting evaluation on real hardware.

Most existing security solutions for WSN have been evaluated by only simulation Dener (2014). However, with the spread of real nodes and the availability of online testbeds, more accurate models based on typical hardware is needed. Due to the challenges introduced by WSNs such as using a shared medium for radio communication, results obtained from simulative and theoritical evaluation can only be considerd approximative Horneber and Hergenrder (2014). Even perfect approximations need to be confirmed by real hardware measurement Horneber and Hergenrder (2014). This does not mean that simulations are not useful. Indeed, simulation and emulation are very helpful when testing a new concept and evaluating a network at large-scale. They are essential steps ahead of real experimentation to decrease issues related to solution design Tonneau et al. (2015). Implicit issues exist in some simulators which

⋆ I wish to present my special thanks to Majmaah University in Saudi Arabia for their care and funding.

Peer review under responsibility of International Federation of Automatic Control.
10.1016/j.ifacol.2018.07.132

are revealed when using real hardware. This research is conducted as an essential step for developing an adaptive security protocol for IoT embedded devices at the data link layer. The aim is to measure the overhead introduced by enabling different security levels of IEEE 802.15.4. The results of this work will serve the progress of the adaptive protocol to change dynamically between security levels based on the application requirements. Also, it confirms that adaptive security protocol at the data link layer could be possible solution to save energy owing to the difference in energy consumption between security levels.

The security impact will be evaluated in terms of the following points:

- Energy Consumption: Security services increases energy consumption of the two components (MCU and Transceiver) differently. Cryptography may affect only the MCU, while authentication may impact both transceiver and the MCU.
- Network Parameters: Encryption and the extra bytes for authentication may affect packet throughput and latency parameters.
- Packet Size: the relationship between the packet length and security energy consumption will be evaluated.
- Transmission Power: The evaluation will investigate whether transmission power affects security energy consumption.

2. RELATED WORK

Previous research has evaluated the efficiency of different cryptographic algorithms in the context of IoT embedded systems. For example, in Trad et al. (2014) the performance of the following three block ciphers is evaluated: AES, RC5 and RC6. This evaluation is conducted based on the Mica2 platform. The result reveals that RC5 is the most efficient block cipher in terms of energy and time. In Lee et al. (2010), different encryption algorithms and modes of operation have been evaluated on Mica2 and TelosB platform. The result of this research shows the suitability of different algorithms for use in WSNs. Another related paper in Panait and Dragomir (2015) has evaluated the implementation of AES on software and on ATmega128RFA1 hardware, and they conclude that hardware implementation is more efficient than software implementation. In Alharby et al. (2018a), security specification of IEEE 802.15.4 has been evaluated using Cooja simulator, and the results indicates a significant overhead caused by enabling security. Also, the relation between duty cycle and security cost is evaluated in Alharby et al. (2018b), and their result shows how some duty cycle protocols increase security cost by retransmitting a packet for one delivery. The study concludes that the cost of security based on the number of encryption calls and retransmission.

However, none of these studies has evaluated the 802.15.4 security levels based on real world experiment. In addition, these studies present the security cost of different security algorithms, while the transmission cost over unsecure transmission is not demonstrated. Authentication and the relation between security cost and factors which affect security cost directly such as transmission power and packet length have been neglected.

3. IEEE 802.15.4 SECURITY SPECIFICATION

IEEE802.15.4 standard is adopted in a range of applications, including military surveillance, environmental monitoring, and industrial automation. These applications require secure communication to protect the exchanged data. IEEE 802.15.4 security specification supports optionally eight different security levels, and applications can operate at the preferred security level. This standard defines only the physical and medium access control (MAC) sub-layer Standard and Society (2011), therefore different protocols can work on top of that for the network and application layer. It defines the security requirements in the MAC layer Saleem et al. (2011), and supports security techniques to protect the wireless communication from possible attack. These techniques assure the confidentiality, integrity, authenticity and replay protection on a per-frame basis Daidone et al. (2014). The IEEE 802.15.4 link-layer security supports hop-by-hop security, where every node in the network should be trusted Raza et al. (2014). IEEE 802.15.4 security layer is controlled at the MAC layer, and the security requirements are specified at the application layer Saleem et al. (2011). Low energy communication using IEEE 802.15.4 std leave at most 102 bytes for the payload and upper layers protocols usage, as shown in Figure 1.

IEEE 802.15.4 can function in a secure mode or non-secure mode. It offers a security suite which supports eight different security levels. Each security level supports specific security requirements and has a different frame format. These security levels generally offer no security, encryption only (AES-CTR), authentication and integrity only (AES-CBC-MAC), or all three security services encryption, integrity and authentication (AES-CCM). AES-CBC-MAC comes with three different message integrity code (MIC) length: 4, 8 and 16 bytes Raza et al. (2014). Also, CCM supports a high level of security and comes with three options of 4, 8 and 16 bytes. MIC is used to guarantee that data has not been changed and also guarantee that data comes from a legitimate source. The length of MIC represents the strength of integrity and authentication. Table 1 and Figure 1 show the difference between security levels of IEEE 802.15.4 security suites. The name of each level consists of two to three parts. The first part indicates the cryptography scheme (AES if security parameters enabled). The second part indicates the mode of operation which used in the cryptography scheme. The last part, if applicable, indicates the MIC, which could be of varying length.

An unsecured frame consists of three fields: a MAC header with 7 to 23 bytes, and data payloads with 0 to 115 bytes, and Frame Check Sequences (FCS) with 2 bytes Daidone et al. (2014). A secured frame has one more field named Auxiliary Security Header (ASH), with 5 to 14 bytes. This is in addition to the Message Integrity Code (MIC) if authentication is enabled. One of the contents of ASH is the Frame Counter with 4 bytes for replay attack detection. The frame counter is set by the outgoing frame at the transmitter side. The frame counter field is checked

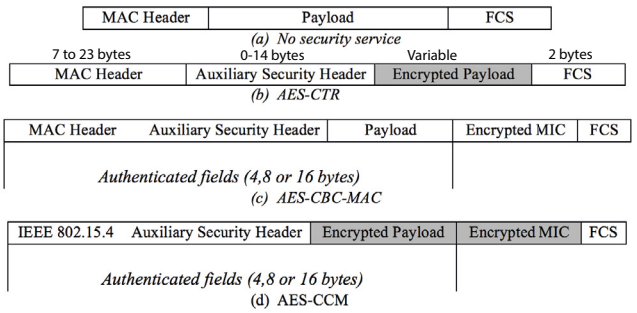


Fig. 1. Secure 802.15.4 frame format

at the receiving nodes, and is accepted if the value is higher than the previous received value.

Table 1. Security suites, reproduced from Tadeo et al. (2010)

Security Services			
SuiteID	Describtion	Services	MIC size (Byte)
0	No security	Null	0
1	AES-CBC-MAC-32	Authentication	4
2	AES-CBC-MAC-64		8
3	AES-CBC-MAC-128		16
4	AES-CTR	Encryption only	0
5	AES-CCM-32	Authentication and encryption	4
6	AES-CCM-64		8
7	AES-CCM-128		16

CTR mode encrypts only the payload, and ASH is sent in clear(as shown in Figure 1).

4. EXPERIMENTAL SETUP

Due to the limited resource in IoT embedded devices, it is important to evaluate energy consumption, latency and throughput. The architecture of this evaluation consists of one node acts as a sink, and other nodes communicating with the sink through IEEE 802.15.4 technology. All nodes are connected to a USB port and their activities are monitored and recorded. The mote hardware used in this experiment is CM5000 platform which is designed based on the original open-source TelosB / Tmote Sky platform, running Contiki as an operating system. CM5000 support Texas Instruments MSP430 micro-controller (48k Flash, 10k RAM) and CC2420 transceiver. Figure 2 shows the testbed of the experiment.

4.1 MAC Protocol Modifications

Contiki-MAC uses Clear Channel Assessment (CCA) during wake-up, which in its turn uses the RSSI to detect radio activities in a channel [44]. In case of a unicast transmission, the whole packet is transmitted repeatedly until an acknowledgement is received from the receiver, as shown in Figure 3. To get accurate measurement for the overhead caused by security, the energy consumption

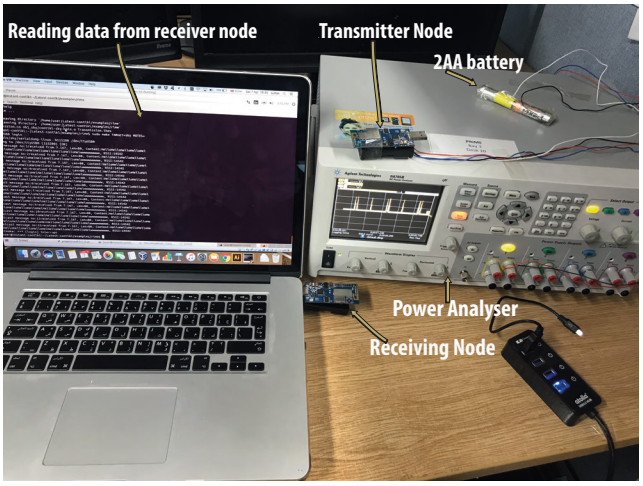


Fig. 2. Testbed of the experiment

caused by duty cycle should be avoided. To accomplish this, CCA before each transmission should be disabled. ContikiMAC perform number of CCA before each transmission to make sure the channel is free and avoid collision. Also, for measurement accuracy, the number of retransmission should be avoided. This can be accomplished by keeping the radio at the receiver side on, so the packet can be received from the first attempt. By this, the extra energy consumption caused by the duty cycle can be avoided. It should be noticed that this is the best case in terms of security cost, and it could be more based on the impact of retransmission.

5. RESULTS ANALYSIS

5.1 Energy Consumption

Energy is used to power the hardware of an IoT embedded devices. These devices are limited in terms of energy budget. Hence, energy usage is one of the most significant issues when dealing with these limited devices. Security services affect two components of an IoT devices: the MCU and transceiver. Cryptography in some cases may affect only the MCU, while authentication may impact both transceiver and the MCU. The transceiver usually consumes far more energy than the MCU, and is the greatest contributor to battery depletion. The aim of this experiment is to evaluate the energy consumption required by each security level of IEEE 802.15.4 for both the MCU and transceiver on real hardware. The energy consumption of security services can be evaluated using the following formula:

$$E_s = I_{Component-Mode} * T \quad (1)$$

Where E_s represents energy consumption of security overhead, $I_{Component-Mode}$ denotes the current draw in Amperes for a component (MCU or transceiver) in a specific mode (active, low power mode, etc.), and T is the time it takes for a component in specific mode in seconds. Figure 4 shows the energy consumed for transmitting one packet with 24 *bytes* payload, and it includes the energy needed for receiving an acknowledgement for the transmitted packet.

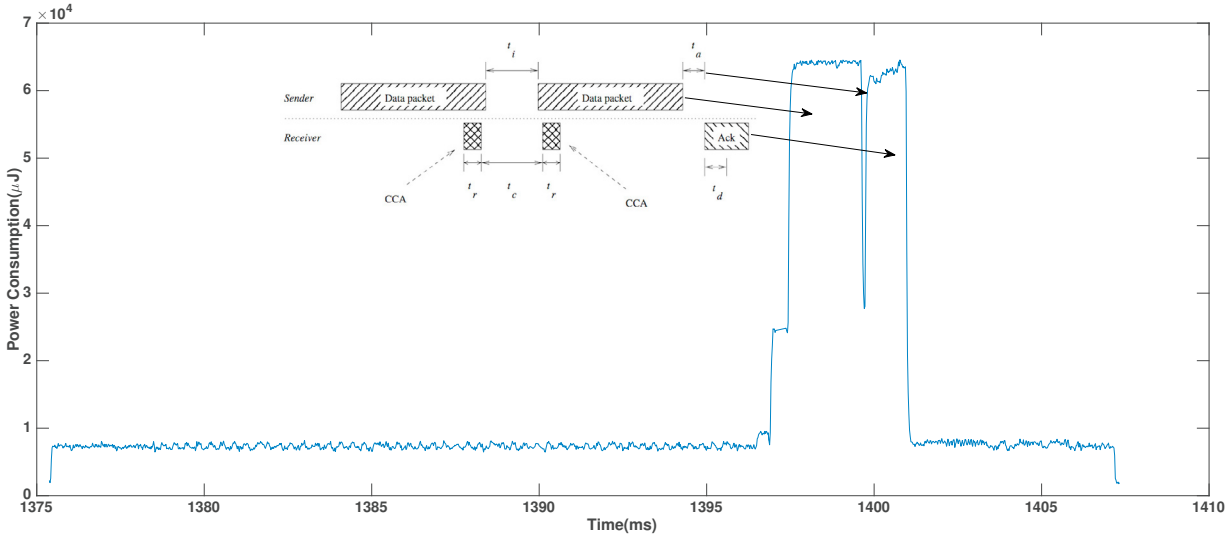


Fig. 3. ContikiMac Transmission Mechanism

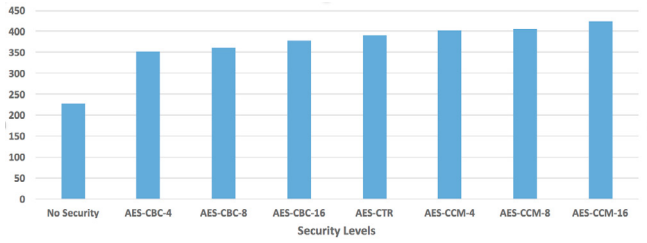


Fig. 4. Energy consumption of transmitting one packet including the acknowledgement

As can be seen from the figure 4, that energy consumption is varied for each security level. Security level 0 represents the energy consumption of transmitting one packet without security, and hence will be used as a *baseline* to calculate the extra energy added by other security levels. The energy consumption increases progressively from security level 0 to level 7. This increasing in energy consumption is due to enabling more complex security services with higher levels. Energy consumption increases from security level 1 to 3 and from 5 to 7 as a result of increasing authentication length which is represented by MIC. Each level employs a different MIC length, and it can be 4, 8,16 bytes based on the required strength of authentication. The extra bytes for authentication makes a transceiver stays in *active* mode longer, which is translated in more energy consumption. Security level 4 is added almost 73% over the baseline level, and this due to enabling encryption which keeps the MCU in *active* mode longer to compute AES encryption. Figure 4 shows a significant increase for energy consumption when enabling security. It can be observed that the minimum security level, level 1, adds almost 55% , and the maximum security level, level 7, adds almost 86% over non-secure transmission. This may affect the network lifetime significantly based on the utilized security services.

Packet Size Versus Security Cost Packet size is one of the important parameters which could affect security cost significantly. Figure 6 shows a comparison for security cost of all levels with 24 bytes and 80 bytes the maximum length can be used with top security level. As can be seen in Figure 6, the security cost increases with higher payload. This is due to the more time needed to encrypt and authenticate longer payload, which is translated to more energy consumption.

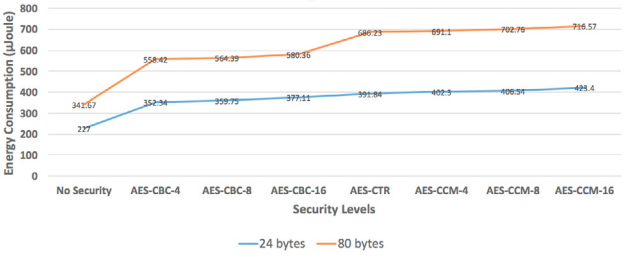


Fig. 6. Energy consumption for different payload size

5.2 Effects of Security on QoS

Security services must ensure the confidentiality, integrity, authentication, and freshness of the data being shared between sensor nodes. These requirements come at a cost to QoS parameters. Adding security to the communication of a WSN may degrade its QoS parameters, and the QoS requirements may affect the employed security mechanism. They are both needed for optimum service, especially when faced with limited resources. The following experiments evaluate the following QoS parameters: packet throughput, latency and transmission power.

5.3 Packet Throughput

In this experiment, the transmitter node sends 50 packets per second, and the number of successful received packets at the destination within 300 *seconds* is recorded. The transmission includes an acknowledgement for each packet. The experiment is repeated with each security level of

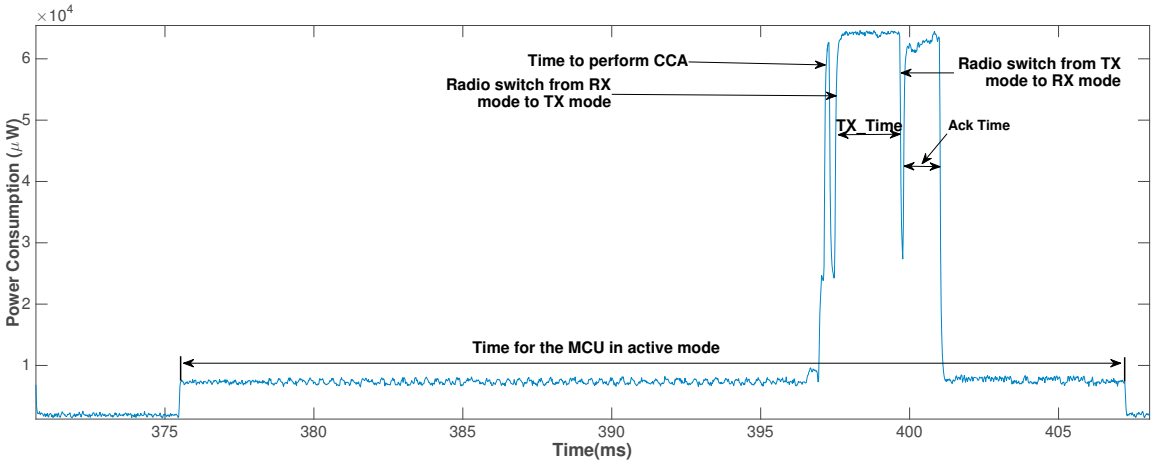


Fig. 5. Latency Timeline

IEEE 802.15.4 standard. The transmitter allows some time after each transmission to receive acknowledgement frame, as specified by the ContikiMAC protocol. The result, as shown in Figure 7, depicts a significant decrease in the number of received packet at the destination when security is enabled. The percentage decrease of packets fluctuates between 35% and 47% based on the employed security level. The result shows that different authentication length at the first three levels have almost the same impact with an almost 35% decrease on the number of received packets over non-secure level. Security level 4, which is encryption only, has a higher impact compared to the first three levels with a 47% decrease of received packets. This indicates that encryption does affect data throughput significantly. The last three security levels which combine encryption with authentication has a significant impact on data throughput as well, ranging from 42% and 46%.

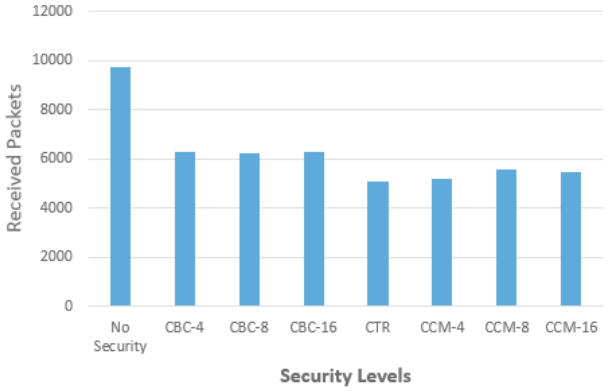


Fig. 7. Number of received packets over 300 seconds

5.4 Security Impact on Latency

In this experiment, latency means the time it takes from preparing a packet at the MCU until an acknowledgement is received from the receiver, as shown in Figure 5. This can be calculated as follows:

$$Latency = T_{CCA} + T_{mcu} + T_{tx} + T_{rs} + T_{wait} + T_{ack} \quad (2)$$

Where, T_{CCA}, is the time needed for a radio to perform a CCA before a transmission. T_{mcu} is the time needed for the MCU to prepare the packet and compute security services if enabled, T_{tx} is the time needed for transmitting the actual IEEE802.15.4 packet with extra bytes for security if enabled, T_{rs} is the time required for a radio to switch from receive to transmit mode or vice versa , T_{wait} is the time needed for a radio to wait for acknowledgement after each transmission, and T_{ack} is the time required by the MCU to process a received acknowledgement. This experiment will be repeated for all security levels. The aim of this experiment is to analyse the delay which caused by security services on both the MCU and radio. The result, as shown in Table 2, shows that latency has increased significantly when enabling security with a minimum of 129% for the first security level over non-secure transmission. Increasing authentication strength dose not incur any difference once authentication is enabled. Latency increases when encryption is enabled by almost 182% over non-secure packet. The last three security levels increase latency by almost 197%. It is obvious that enabling security increase latency significantly, and this can be noticed at all security levels.

Table 2. Packet Delivery Latency

Security Level	Payload	Delay (ms)
No security	24	13.1
AES-CBC-MAC-32	24	30
AES-CBC-MAC-64	24	31
AES-CBC-MAC-128	24	32
AES-CTR	24	37
AES-CCM-32	24	38.7
AES-CCM-64	24	39
AES-CCM-128	24	39

Impact of Transmission Power on Security Cost It can be noticed in Table 3 that the change in transmission power affects the security cost. Security cost is higher with minimum transmission power by almost 3% for all security levels except level 4. There is no extra bytes for

security at level 4, and hence shows no change in energy. MCU works independently, and hence is not affected by the transmission power change. MCU energy consumption becomes more significant, compared to radio, when decreasing transmission power.

Table 3. Percentage of Security Cost over Non-secure Packet Transmission with Minimum and Maximum Transmission Power

Sec_LVL\TX Power	With minimum transmission power	With maximum transmission power
No security	-	-
AES-CBC-MAC-32	57%	53%
AES-CBC-MAC-64	59%	56%
AES-CBC-MAC-128	66%	63%
AES-CTR	74%	74%
AES-CCM-32	77%	74%
AES-CCM-64	80%	76%
AES-CCM-128	87%	85%

6. CONCLUSION

Taking into consideration that security might be an enabling element in the proliferation of many IoT applications, this research perform an analysis for the cost of security techniques used in IEEE 802.15.4 at the data link layer on real hardware. This research uses a real hardware for the evaluation in order to avoid implicit issues which introduced by some simulators. The results show a considerable impact on energy consumption and network parameters when enabling security. The security overhead on energy fluctuates between 53% and 85% over non-secure transmission based on the utilized security services. Security does come at a cost, and should be enabled as needed. The results of this evaluation encourage the research to go further to develop an adaptive security protocol, which can adjust its security level dynamically at run time to save energy.

REFERENCES

Sultan Alharby, Nick Harris, Alex Weddell, and Jeff Reeve. The security trade-offs in resource constrained nodes for iot application. *International Journal of Electrical, Computer, Energetic, Electronic and Communication Engineering*, 12(1):52 – 59, 2018a. ISSN 1307-6892. URL `http://waset.org/Publications?p=133`.

Sultan Alharby, Nick Harris, Alex Weddell, and Jeff Reeve. Impact of duty cycle protocols on security cost of iot. In *International Conference on Information and Communication Systems*, Irbid, 2018b. IEEE.

Roberta Daidone, Gianluca Dini, and Giuseppe Anastasi. On evaluating the performance impact of the ieee 802.15. 4 security sub-layer. *Computer Communications*, 47:65–76, 2014.

Murat Dener. Security analysis in wireless sensor networks. *International Journal of Distributed Sensor Networks*, 10(10):303501, 2014.

J. Horneber and A. Hergenrder. A survey on testbeds and experimentation environments for wireless sensor networks. *IEEE Communications Surveys Tutorials*, 16 (4):1820–1838, Fourthquarter 2014. ISSN 1553-877X. doi: 10.1109/COMST.2014.2320051.

Jongdeog Lee, Krasimira Kapitanova, and Sang H Son. The price of security in wireless sensor networks. *Computer Networks*, 54(17):2967–2978, 2010.

Cristina Panait and Dan Dragomir. Measuring the performance and energy consumption of aes in wireless sensor networks. In *Computer Science and Information Systems (FedCSIS), 2015 Federated Conference on*, pages 1261–1266. IEEE, 2015.

Shahid Raza, Simon Duquennoy, Joel Höglund, Utz Roedig, and Thiemo Voigt. Secure communication for the internet of thingsa comparison of link-layer security and ipsec for 6lowpan. *Security and Communication Networks*, 7(12):2654–2668, 2014.

Shahnaz Saleem, Sana Ullah, and Kyung Sup Kwak. A study of ieee 802.15. 4 security framework for wireless body area networks. *Sensors*, 11(2):1383–1395, 2011.

IEEE Standard and IEEE Computer Society. IEEE Standard for Local and metropolitan area networks Part 15 . 4 : Low-Rate Wireless Personal Area Networks (LR-WPANs) IEEE Computer Society S ponsored by the. *IEEE Std 802.15.4-2011*, 2011(September):1–294, 2011. doi: 10.1109/IEEESTD.2011.6012487. URL `http://ieeexplore.ieee.org/servlet/opac?punumber=6012485`.

Antonio Vincenzo Taddeo, Marcello Mura, and Alberto Ferrante. Qos and security in energy-harvesting wireless sensor networks. In *Security and Cryptography (SECRYPT), Proceedings of the 2010 International Conference on*, pages 1–10. IEEE, 2010.

Anne-Sophie Tonneau, Nathalie Mitton, and Julien Vandaele. How to choose an experimentation platform for wireless sensor networks? a survey on static and mobile wireless sensor network experimentation facilities. *Ad Hoc Networks*, 30:115–127, 2015.

Abdelbasset Trad, Abdullah Ali Bahattab, and Soufiene Ben Othman. Performance trade-offs of encryption algorithms for wireless sensor networks. In *Computer Applications and Information Systems (WCCAIS), 2014 World Congress on*, pages 1–6. IEEE, 2014.

Muhamed Turkanović, Boštjan Brumen, and Marko Hölbl. A novel user authentication and key agreement scheme for heterogeneous ad hoc wireless sensor networks, based on the Internet of Things notion. *Ad Hoc Networks*, 20:96–112, sep 2014. ISSN 15708705. doi: 10.1016/j.adhoc.2014.03.009. URL `http://linkinghub.elsevier.com/retrieve/pii/S157087051400064X`.

Li Da Xu, Wu He, and Shancang Li. Internet of things in industries: A survey. *IEEE Transactions on Industrial Informatics*, 10(4):2233–2243, 2014. ISSN 15513203. doi: 10.1109/TII.2014.2300753.

Yuanchang Zhong, Lin Cheng, Liang Zhang, Yongduan Song, and Hamid Reza Karimi. Energy-efficient routing control algorithm in large-scale wsn for water environment monitoring with application to three gorges reservoir area. *The Scientific World Journal*, 2014, 2014.

World Academy of Science, Engineering and Technology
International Journal of Electronics and Communication Engineering
Vol:12, No:1, 2018

The Security Trade-Offs in Resource Constrained Nodes for IoT Application

Sultan Alharby, Nick Harris, Alex Weddell, Jeff Reeve

***Abstract*—**The concept of the Internet of Things (IoT) has received much attention over the last five years. It is predicted that the IoT will influence every aspect of our lifestyles in the near future. Wireless Sensor Networks are one of the key enablers of the operation of IoTs, allowing data to be collected from the surrounding environment. However, due to limited resources, nature of deployment and unattended operation, a WSN is vulnerable to various types of attack. Security is paramount for reliable and safe communication between IoT embedded devices, but it does, however, come at a cost to resources. Nodes are usually equipped with small batteries, which makes energy conservation crucial to IoT devices. Nevertheless, security cost in terms of energy consumption has not been studied sufficiently. Previous research has used a security specification of 802.15.4 for IoT applications, but the energy cost of each security level and the impact on quality of services (QoS) parameters remain unknown. This research focuses on the cost of security at the IoT media access control (MAC) layer. It begins by studying the energy consumption of IEEE 802.15.4 security levels, which is followed by an evaluation for the impact of security on data latency and throughput, and then presents the impact of transmission power on security overhead, and finally shows the effects of security on memory footprint. The results show that security overhead in terms of energy consumption with a payload of 24 bytes fluctuates between 31.5% at minimum level over non-secure packets and 60.4% at the top security level of 802.15.4 security specification. Also, it shows that security cost has less impact at longer packet lengths, and more with smaller packet size. In addition, the results depicts a significant impact on data latency and throughput. Overall, maximum authentication length decreases throughput by almost 53%, and encryption and authentication together by almost 62%.

***Keywords*—**Internet of Things, IEEE 802.15.4, security cost evaluation, wireless sensor network, energy consumption.

I. Introduction

THE concept of IoT has recently grabbed the attention of the academic and industrial communities [1]. The IoT is not associated with a particular technology, and can be used in different applications. However, the Wireless Sensor Network (WSN) is a foundational technology for IoT [2], [3]. Sensors are the main tools for reporting events in *things* such as cars, home appliances, and any object to which a sensor can be attached. However, IoT devices do suffer from major issues involving limited resources [4], particularly energy, and a vulnerability to various types of attack. Protecting the communication between IoT devices with limited resources is a complex task. Nodes are usually equipped with small batteries, which makes energy conservation crucial to WSNs. Every bit consumes energy [5], so conventional security mechanisms which introduce more overheads for both computation and communication are unsuitable for such limited devices [6]–[9]. It is clear that security and power consumption are opposite parameters. One of the main obstacles facing security solutions for IoT devices is energy consumption. Batteries are the main source of power in these devices, and the indicator of IoT device lifetime. Usually these devices are implemented in remote area or harsh environment which make changing a battery difficult. Thus, the energy limitation of these small devices necessitates a trade-off between security mechanism and energy consumption. Security has become essential to many IoT applications [10], [11], especially when dealing with sensitive data such as medical and military applications, but it does, however, come at a cost to resources. Nevertheless, security cost has not been studied sufficiently. Many research have used security specification of 802.15.4 for IoT MAC layer, but the cost of each security level is unknown. Basic security services include encryption to guarantee confidentiality, authentication to ensure packets are sent from a legitimate party, integrity to guarantee packets have not changed through transmission, and freshness of data to ensure that packet is recent and old packets are not being re-played. This paper investigates the overhead introduced by IEEE 802.15.4 security levels at MAC layer and their effect on the QoS parameters. To obtain accurate results, the effects of MAC and Radio Duty Cycle(RDC) protocols on the security cost has been excluded, since the purpose of this evaluation is only to get the extra overhead of security on sensor networks. However, the mechanism used ContikiMAC and the methods employed to avoid its effects are discussed, as it is the RDC protocol employed in this emulation. The obtained results assume a perfect communication environment, therefore packet delivery is 100% successful as long as the two nodes involved are within the same transmission coverage area. The results represent the minimum security overhead, and the actual overhead could be greater, depending on the mechanism employed for the Radio Duty Cycle. For example, re-transmitting packets increases security services' impact on performance. The overhead considered in this scenario is that introduced by the transmission mode of each security level. The evaluation focuses on the following performance parameters:

1) Per-packet energy Consumption E: The total energy needed for delivering one packet from source to destination at each security level. This includes the

S. Alharby is with the Department of Electronics and Computer Science, University of Southampton, UK (corresponding author, e-mail: sa1c15,@soton.ac.uk).

N. Harris, A. Weddell and J. Reeve are with the Department of Electronics and Computer Science, University of Southampton, UK (e-mail: nrh@ecs.soton.ac.uk, asw@ecs.soton.ac.uk, jsr@ecs.soton.ac.uk).

energy consumed by transmission mode E_{tx} and receiving mode E_{rx}, and the energy required by a relay nodes to forward a packet E_{fwd}. Hence, the total energy consumption of transmitting one packet E can be represented as follows:

$$E = E_{tx} + E_{rx} + n * E_{fwd} \quad (1)$$

2) Latency (L): This measures the time needed for a node to transmit a packet until it received by the destination.
3) Throughput (Thr): This is the number of packets received at the destination per unit time (one second in this research).

At the end of this paper, the most significant security levels will be identified based on their impact on sensor network performance, particularly in terms of energy consumption.

II. Related Work

Several studies have evaluated the cost of security at the IoT MAC layer, but the cost of each security level in IEEE 802.15.4 is unknown. For instance, [12] have analysed the energy consumption of AES, RC5 and RC6. They have evaluated the energy cost and memory requirements of these cipher algorithms.

Similar study [13] has evaluated the cost of AES, RC5 and RC6. This study also investigates the impact of key size on energy cost and concludes that RC5 is the most energy-efficient for limited resource devices. Also, [14] provides a method of optimising encryption hardware implementation. The study investigates the energy consumption and performance of AES in both software and hardware implementations. The results indicate that hardware is more efficient than software implementation. However, none of these studies discuss authentication cost, which is crucial to security services in WSNs. In addition, a network engineer cannot identify the cost of security over non-secure transmission, as these studies present only the cost of encryption.

Reference [15] have analysed the cost of using different encryption block ciphers such as AES and RC5 on two popular hardware platforms: MicaZ, and TelosB. The study evaluates the effects of different key sizes on energy cost, and also presents the energy cost of different MAC protocols. However, the study does not evaluate IEEE 802.15.4 and its implications on communication cost. Also, the cost of security over non-secure transmissions is undefined, as the study focuses on the comparison of cipher algorithms rather than security over non-secure transmission.

In contrast, the present study focuses on the security levels of IEEE 802.15.4. It identifies the impact of different security levels on energy consumption and QoS parameters such as latency and throughput, and illustrates how transmission power affects the security cost. In addition, the study clarifies the relationship between security cost and the packet data length. Furthermore, it covers aspects which have been neglected by previous studies, such as how security affects energy consumption indirectly by causing multiple transmissions. Finally, this study provides a methodology for evaluating the security overhead of the IoT MAC layer.

III. Simulation Setup and Parameters

There are many lightweight operating systems (OSs) which could be used in wireless sensor nodes. These operating systems provide similar services, but certain characteristics of these operating systems might affect the choice of the developers. Examples of these operating systems are Contiki, RIOT and TinyOS. However, Contiki operating system was selected in this experiment for its suitable features. The Cooja simulator, which comes with Contiki OS, is used to obtain the results in this paper. Also, Powertrace tool [16], which is supported in Contiki, is used to provide detailed information about where the energy is consumed (transmission, receiving, etc). It calculates the time each component takes in particular mode. This tool is claimed to be 94% accurate in measuring the energy consumed by a real device [16]. Table I shows the parameters which used in the simulator.

TABLE I
Simulation Parameters

Parameter	Value
Platform	Tmote Sky
MAC protocol	CSMA
Radio Duty Cycle	ContikiMAC
Payload	24 and 80 byte
Transmission range	50 Meters
TX/RX success ratio	100%
Radio	CC2420
Microcontroller unit (MCU)	MSP430

The simulation uses single hop communication to deliver packets from source to destination.

IV. IEEE 802.15.4 Security Specifications

This experiment uses a MAC layer security protocol which supports eight levels, as defined by the IEEE 802.15.4 security specifications (as shown in Table II). The minimum security level is 0, whereby no security mechanism is used, and the highest level is 7, which includes encryption,replay protection, integrity and authentication with AES-128.

TABLE II
Security Suites, Reproduced from [17]

Security Suites				
SuiteID	*Description*	*Services*	*Replay detection*	*MIC size (Byte)*
0	No Security	Null	-	0
1	AES-CBC-MAC-32		ON	4
2	AES-CBC-MAC-64	Authentication	ON	8
3	AES-CBC-MAC-128		ON	16
4	AES-CTR	Encryption only	ON	0
5	AES-CCM-32	Authentication	ON	4
6	AES-CCM-64	and	ON	8
7	AES-CCM-128	encryption	ON	16

The security services added at each security level are shown in Fig. 1. AES-CTR mode only provides encryption for the payload, hence it supports confidentiality. The length of the key used is 128 bits, as recommended by the IEEE 802.15.4 security specifications. This length will be fixed at all levels which support confidentiality in this experiment. Authentication can be achieved by appending a message authentication code in every packet. Message authentication

code is also named message integrity code (MIC). This research will use MIC to indicate to message authentication code, so we can differentiate between media access control (MAC) and message authentication code. Authentication can be of various lengths based on the required security strength [4, 8 or 16 byte].

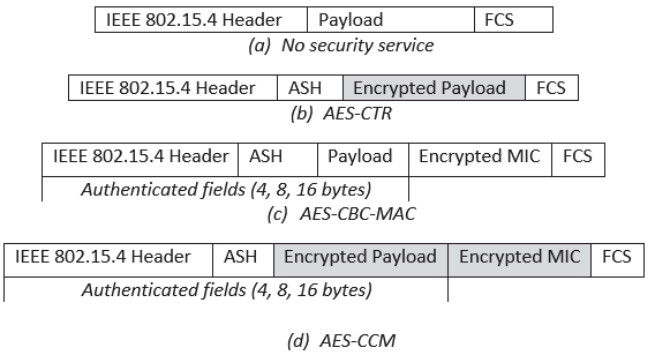


Fig. 1 Security services frame format

Auxiliary Security Header(ASH)(as shown in Table III) consists of three fields: security control, frame counter, and key identifier. ASH is added to the frame only when frame control bit field is set to one [18]. Security control specifies the security level employed for a frame, frame counter is used to provide replay protection against replay attack, and key identifier provides information about the key identifier mode.

TABLE III
AUXILIARY SECURITY HEADER

1 byte	4 byte	0 -9 byte
Security Control	Frame Counter	Key Identifier

V. ACCURACY OF THE SECURITY OVERHEAD RESULTS

There are many factors which affect the accuracy of the results obtained from the emulator, such as the padding mechanism and MAC protocol. The ContikiMAC protocol is used as a RDC protocol. Energy consumption is significantly affected by the employed RDC protocol. Under the ContikiMAC protocol, the sender checks the medium channel before transmitting, and if there is no radio activity, it sends a full data packet and continues to transmit until the receiver wakes up and acknowledges the message. This can affect the result of assessing the overhead of security, as the number of AES invocation varies. At the receiver side, a node checks the medium channel periodically for any activity [19]. Fig. 2 shows the work mechanism of ContikiMAC through unicast transmission. Node 2 represents the transmitter, and node 1 represents the receiver. ContikiMAC requires a minimum length packet size. This is to guarantee that the packet doese not fall down between two Clear Channel Assessment (CCA) [19]. This becomes more important in broadcast communication, as there is no acknowledgement returned to the sender. If the packet size is lower than the minimum size, then a padding mechanism is used to increase the packet size to the minimum. In order to avoid the impact of the padding mechanism on the experiment results, the packet size will always be larger than the minimum packet size.

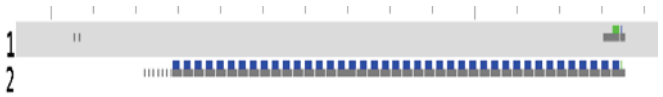


Fig. 2 ContikiMAC mechanism

It can be observed that the radio is turned on and off on regular basis to save power. This is determined by a parameter known as Channel Check Rate. There is an optimisation phase for ContikiMAC which reduces the number of re-transmissions by keeping a track of the receiver wake up period. This could help in making the sender transmit just before the receiver wakes up. Retransmission can significantly affect the energy consumption and assessment of security overhead. In order to avoid the impact of re-transmitting the packet and obtain an accurate result for transmitting one packet, the receiver node is kept on at all times (as shown in Fig. 3). Node number 1 is the transmitter and node number 2 is the receiver.

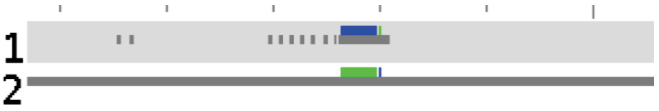


Fig. 3 The radio state for both sender and receiver

Fig. 3 depicts the CCA mechanism, at every transmission the radio checks the channel to make sure it is clear. To eliminate the impact of CCA on the obtained results for energy consumption, CCA is disabled before transmission (as it shown in Fig. 4).

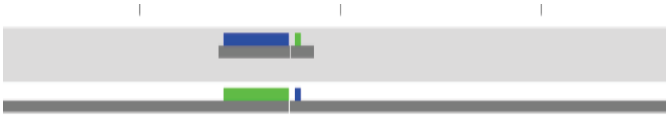


Fig. 4 CCA is disabled before transmission

VI. SIMULATION RESULTS

A. Energy Consumption Evaluation

In order to obtain the total security related energy consumption, all components which affects security cost should be investigated. There are two factors that contribute to the energy consumed by security processes: computation, and communication overhead. Security computation related energy consumption is caused by adding/removing security services such as cryptography. Computation processes makes the MCU run longer to compute complex algorithm. The communication cost is can be obtained by the energy consumed by the radio to transmit the extra byte for authentication. Hence, the total security energy consumption for single packet transmission can be represented as follows:

$$E_{sec-total} = \sum_{k=1}^{n} (E_{sec-compu} + E_{sec-comm}) \quad (2)$$

where, n indicates the number of nodes involved in the transmission, $E_{sec-total}$ the total security energy consumption, $E_{sec-compu}$ computes the energy required for computation overhead, which includes processing the actual transmission and cryptography algorithm, and $E_{sec-comm}$ the energy required for transmitting a packet, which includes transmitting the actual frame and the extra bytes needed for MIC authentication. In the following sections, energy cost is investigated for each security level of the IEEE 802.15.4 security standard. This will include both computation and communication energy cost. The obtained result is an energy cost for delivery of a single packet and expressed in μJoule units. Cost per packet delivery includes the generation of the packet by the MCU and transmission by the radio at the source. This will be acquired for each security level. Required security services are added/removed for each plaintext block according to the security level. The cost of transmission without security services will be taken as a *baseline* for comparison, since security overhead increases by selecting higher security level. In this evaluation, the powertrace tool is used to measure energy consumption. Powertrace records the time that a component (Radio or MCU) enters a specific mode, hence, the time that the MCU and Radio spend in each mode (active, low power mode, etc.) is recorded. The current drawn by the MCU and Radio in different modes should be known in order to estimate the energy consumption. Tmote sky uses CC2420 as a radio driver and MSP430 as a microcontroller. According to Sky mote datasheet [20], the current drawn by the radio and the micro-controller is shown in Table IV

TABLE IV
TYPICAL CURRENT CONSUMPTION FOR TMOTE SKY

Component	Current drawn
MCU- active state	2400μA
Radio - Transmitting mode	17.4mA
Radio - Receiving mode	19.7mA

The objectives of this experiment are as follows:

1) Measure the energy consumption in delivering a single packet at each security level for transmit mode.
2) Investigate the impact of frame length on the security cost.
3) Explore the most significant security level based on energy and also according to security services.
4) Investigate the impact of the power of transmission on performance in terms of energy consumption.

Scenario 1: Evaluation with a payload length of 24 byte in transmit mode The two components of sensor node which affected by security are the MCU and the radio. Hence, the energy consumption associated with these components will be studied. First, the energy consumption of transmitting single packet with 24 byte without security is measured. This will serve as a baseline for comparison with other levels which include different security services. The following formula is used to calculate the energy consumption of every node components:

$$E = Energest\_Value * Voltage * Current \\ / RTIMER\_SECOND * runtime \quad (3)$$

where, E is the energy consumption of a node's component at a specific mode, $Energest\_Value$ is the difference between two interval times, and $RTIMER\_SECOND$ is the number of ticks per second, which in the current simulation is 32768 ticks/second.

Table V shows the energy consumed by the MCU and radio transmitting a single packet with a 24 byte payload. As can be seen from Table V, the radio is the main contributor to energy consumption. MCU consumption at level 0 constitutes 11.5% of the total energy consumption, and it grows as the code increases in complexity with higher security services. However, at the top security level it constitutes only 22% of the total cost of energy. This extra consumption by the MCU at higher security levels is due to AES operation and the processing of extra bytes added by progressive levels of authentication. On the contrary, the radio is responsible for the majority of energy consumption during transmission (as shown in Fig. 5). It can be noticed that radio energy consumption at all levels fluctuates between 73.7% and 88.5% of overall packet consumption, which is a very high percentage. The Radio is responsible for transmitting packets, and it remains in use longer with a greater number of bits. This explains the high energy consumption when enabling authentication, as authentication adds more bytes to the packets.

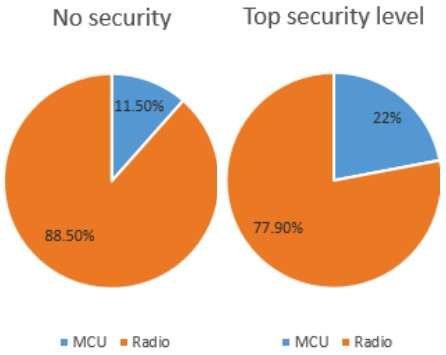


Fig. 5 Radio consumption vs MCU consumption for level 0 and 7

Also, it can be noticed that, total energy consumption increases gradually from security level 0 to level 3, and from 5 to level 7. This is due to the length of MIC, as every level employs a different MIC length. Security level 4 employs encryption only, therefore the radio consumes less energy comparing to authentication security levels. There is a slight difference in MCU energy consumption between security levels 1, 2 and 3. This also applies for security levels 4, 5, 6 and 7, which see only minor changes in MCU energy consumption. However, the increased energy consumption for the MCU at levels 5, 6 and 7 is almost 4 times of the energy consumed by level 0. According to Table V, the percentage

TABLE V
ENERGY CONSUMPTION OF TRANSMITTING ONE PACKET WITH A PAYLOAD OF 24 BYTE IN DIFFERENT SECURITY LEVELS

Security level	MCU energy consumption (μJ)	Radio energy consumption (μJ)	Total energy consumption (μJ)	Percentage of increased security overhead over non-secure packet (%)
0	9.53	73.28	82.81	-
1	24.01	84.91	108.926	31.54%
2	24.15	92.39	116.54	40.72%
3	24.32	103.546	127.87	54.4%
4	28.95	81.24	110.19	33%
5	28.5	83.8	112.3	35.6%
6	29.11	90.80	119.91	44.8%
7	29.33	103.55	132.88	60.46%

increase in security overhead over non-secure communication is high. It can be observed that the minimum security level, level 1, adds a 31.54% overhead, and the highest security level adds 60.46%. This significant overhead affects the network lifetime, and may shorten it significantly depending on the employed security level.

Scenario 2: Evaluation with a payload length of 80 byte in transmit mode The previous experiment was conducted a second time but with longer payload, to investigate the effects of frame length on security overhead. Table VI depicts the security overhead on a frame with an 80 byte payload. Obviously, the security overhead is less with a longer frame. The overall security overhead decreases at all security levels compared to the previous scenario which uses a 24 byte payload. This is because the security cost is the same in both scenarios at all security levels, but the security cost becomes more obvious when the overall energy consumption is small, and less obvious when the overall energy consumption is large. However, in both scenarios, the security cost is significant and makes difference in terms of the network lifetime. Sensor node hardware is limited in terms of payload size, therefore the extra byte added by authentication may lead to multiple packet transmissions if the message exceeds the maximum packet length. For instances, TinyOS uses 36 byte as a default packet length.

The most significant security levels are 0, 4, 6 and 7. The reason for selecting these levels is that the energy consumption at level 2 and 3, which provide authentication only, is similar to 5 and 6, which provide encryption, integrity and authentication. Hence, the latter are chosen since they provide more security with the same energy consumption. Level 4 has been chosen as it provides encryption only at an acceptable cost in case authentication is not required.

Scenario 3: Evaluating the effect of transmission power on security cost

It can be observed in Table VII and Fig. 7 that transmission power does affect security overhead in terms of energy consumption. This due to that MCU run independently and not affected by transmission power change. This makes MCU overhead more visible, in comparison to the radio cost, when the transmission power is reduced, and affects overall energy consumption. The overall security cost will be higher with low transmission power.

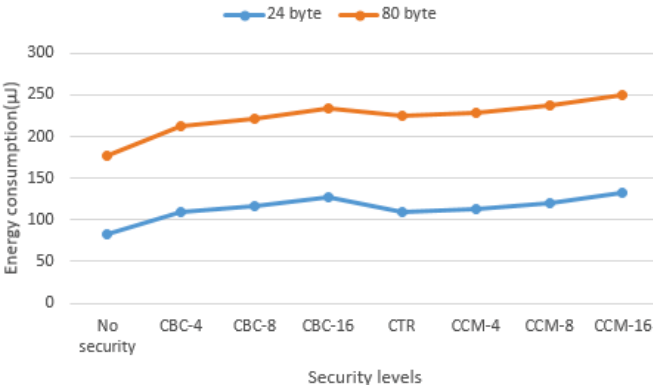


Fig. 6 Energy consumption for different payload size

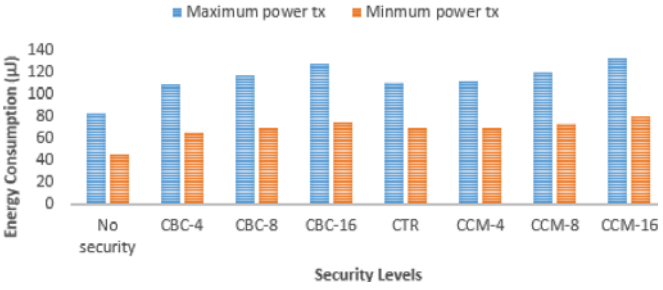


Fig. 7 Energy consumption of different security levels with minimum and maximum transmission power-24 bytes payload

B. Latency Evaluation

In this section, the trade-off between security level and latency is studied and evaluated. It is assumed that cryptography will increase the computation time when adding/removing security services. This also applies for communication overhead, as MIC adds an extra byte to the frame, consequently, a longer frame requires more time for transmission. There are many factors which affect the time required for delivering a single packet. Note that CCA is disabled here to prevent its impacting the results. Fig. 8 depicts the process for transmitting a frame with and without security services. It is demonstrated based on the functionality of ContikiMac. ContikiMac waits for an acknowledgement after each transmission to guarantee that a transmission has been received at the next hop.

Latency without security services can be calculated analytically as follows:

TABLE VI
ENERGY CONSUMPTION OF TRANSMITTING ONE PACKET WITH A PAYLOAD OF 80 BYTE IN DIFFERENT SECURITY LEVELS

Security level	MCU energy consumption (μJ)	Radio energy consumption (μJ)	Total energy consumption (μJ)	Percentage of increased security overhead over non-secure packet (%)
0	11.93	165.67	177.6	-
1	35.92	176.82	212.74	20%
2	36.2	184.79	220.99	24%
3	36.49	197.5	233.99	31.7%
4	51.26	173.63	224.89	26.62%
5	51.35	176.82	228.17	28.47%
6	51.65	184.8	236.45	33%
7	51.68	197.53	249.21	40.32%

TABLE VII
PERCENTAGE OF SECURITY COST OVER NON-SECURE PACKET TRANSMISSION WITH MINIMUM AND MAXIMUM TRANSMISSION POWER

Sec_LVL\TX Power	With maximum transmission power	With minimum transmission power
No Security	-	-
CBC-4	31.54%	44.47%
CBC-8	40.72%	52.77%
CBC-16	54.40%	65.05%
CTR	33%	51.40%
CCM-4	35.60%	53.32%
CCM-8	44.80%	62.05%
CCM-16	60.46%	76.10%

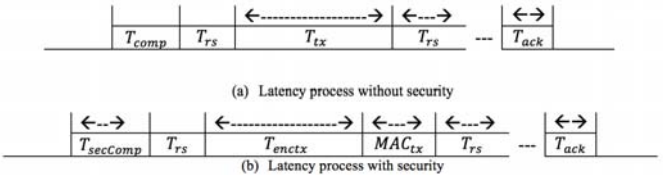


Fig. 8 Latency process

$$Latency = T_{comp} + T_{tx} + T_{rs} + T_{wait} + T_{ack} \quad (4)$$

where, T_{comp} is the time required to process a frame format by MCU, T_{tx} is the time required to transmit the actual frame, T_{rs} is the time required for a radio to switch from transmit mode to receive mode or from idle to transmission mode, T_{wait} the time needed to receive an acknowledgement from the destination, and T_{ack} the time required to process an acknowledgement frame. Fig. 8 (b), shows the required overhead when security services are added to the communication. It is assumed that Cryptography, Integrity and Authentication are enabled. This is demonstrated mathematically in the following formula:

$$Latency_{sec\_enabled} = T_{seccomp} + T_{enctx} + MIC_{tx} + T_{rs} + T_{wait} + T_{ack} \quad (5)$$

where, $T_{seccomp}$ is similar toT_{comp} but with one or more security services such as cryptography, T_{enctx} is the required time to transmit an encrypted actual frame. MIC_{tx} is the time it takes to transmit the extra bytes needed for authentication. The time needed for the extra bytes depends on the length of MIC, it can be 4, 8 or 16 byte.

Scenario 1: Latency evaluation with different payload lengths Network performance may be affected by security services in terms of latency. This might be due to the extra overhead incurred by processing and transmitting. Fig. 10 shows the simulation layout of this experiment. The latency is obtained by calculating the time it takes to transmit a packet from node 3 to node 1 passing through node 2. This includes the time needed to add data to buffer, add security services, transmitting time, receiving time and finally removing security services at the destination. The extra time added by security services for transmitting one packet with two different payload length has been measured. The first with 24 byte, and the second with the same setting but with an 80 byte payload length. In the simulation, to obtain an accurate result for latency added by security services at each level, the radio is kept on for all nodes to avoid latency caused by RDC protocol. The experiment is run without security services, at security level 0, then again for each progressive security level. The receiver in this experiment located in two hops distance, hence, the layer two acknowledgement cannot be received at the sender. Consequently, the extra time over level 0 is recorded as follows, where n is the number of received packets:

$$TotalLatency = \sum_{k=1}^{n} (rx_{time} - tx_{time}) \quad (6)$$

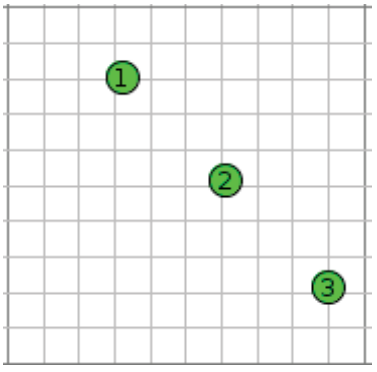


Fig. 9 Experiment layout for latency with three nodes

Fig. 10 shows the latency performance in ms for each security level of the IEEE 802.15.4 security standard. As can be seen, latency increases sharply when security services are enabled. For example, the latency is almost $42ms$ without

security, and with an 80 $byte$ payload, this rises dramatically by almost 328% when authentication is enabled (Level 1).

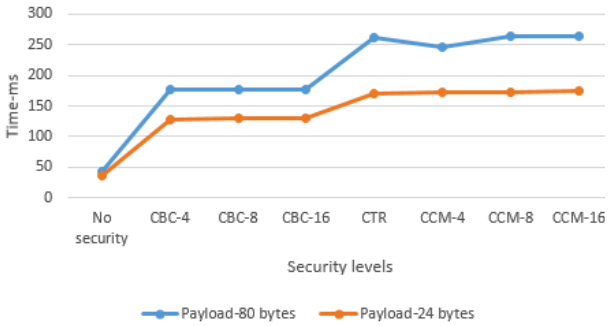


Fig. 10 Latency time per-packet with different payload length

The result shows that all authentication levels [*CBC-4, 8 and 16*] have similar latency performance once authentication is enabled. This may indicate that MIC length of [4, 8 and16] has similar effects on latency in the two payload experiments. CTR encryption increases the latency by almost 526% over level 0. This includes the latency of encryption at the source, decryption and encryption at the relay node, and decryption at the destination node. Overall, latency performance increases sharply with authentication by more than three times of the original packet cost, but it increases with encryption by almost 46% over authentication cost. This indicates that latency is affected more by processing overhead than with transmission overhead. In fact, the processing effects can be noticed clearly by observing the latency at different payload lengths. As can be depicted in Fig. 10, the greater the payload length, the greater the resulting latency, due to the need for more resources being required to encrypt or decrypt a packet.

C. Throughput Evaluation

The objective of this experiment is to assess whether security services have obvious impact on the throughput of packets. In this experiment, throughput refers to the number of packets received at the destination node over a certain time. Throughput has been calculated between two nodes with different security levels for 300 $seconds$. A payload of 24 $byte$ is used in all levels. To obtain an accurate result, the radio of the receiving node is kept 'on' to achieve the maximum throughput. Theoretically, security services are expected to affect the number of received packets, because the radio keeps on longer to transmit longer packet length with authentication, and the MCU computation takes longer to process security operations. As shown in Fig. 11, the percentage of received packets at $securitylevel1$ decreases by 53.5% when compared to a state of no security level throughput. This percentage is similar for levels 2 and 3, with only small variations. Levels 4, 5, 6 and 7 have a greater effect on throughput by reducing the percentage of received packets to almost 62%. At higher levels, cryptography and authentication are enabled, and this cause the drop in throughput. Overall, authentication decreases throughput by almost 53%, and encryption and authentication by almost 62%.

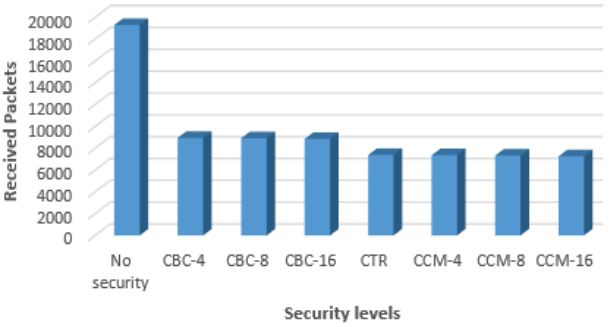


Fig. 11 Throughput of different security levels-300 seconds

TABLE VIII
MEMORY EVALUATION ON TTMOTE SKY HARDWARE

Scheme	text (B)	data (B)	bss (B)	RAM (KB)	ROM (KB)
Contiki OS	22415	142	5136	~5.2 KB	~22 KB
Contiki OS + Security	25449	182	5558	~5.6 KB	~25KB

VII. MEMORY FOOTPRINT EVALUATION

The memory required by the security modules might become an issue, especially with such constrained devices as WSN nodes. Evaluating memory usage is crucial, as it provides information on whether a security algorithm could run on constrained devices. This information can be obtained by determining the size of the compiled file with and without security modules. Memory size can be known by using the command ' $size$' followed by the compiled file name on Linux OS. Memory is divided into flash memory (ROM) and dynamic memory (RAM). Table VIII, Figs. 12 and 13 show the memory used by security processes on Contiki OS for different segments on tmote sky hardware. 'text' represents the read-only part of memory, 'data' represents the read-write part of memory, and 'bss' contains uninitialized data, global and static variables which are initialised to zero would be included in this part. RAM size can be determined by the sum of data+bss, and ROM size by text+data. Out of $48k$ in sky mote ROM size, Contiki OS with maximum security services consumes in total $\sim$ $25k$, with only $3k$ used for security, while RAM consumes $\sim$ $5.6k$ out of $10k$ available in sky mote. These results, as depicted in Figs. 12 and 13, show that the hardware could accommodate the security specification of IEEE 802.15.4, leaving 44% of RAM, and 48% of ROM free for application usage.

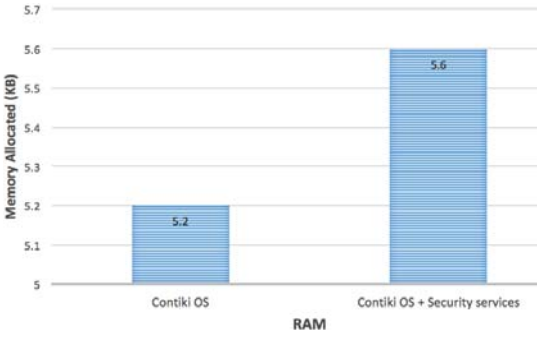


Fig. 12 RAM memory usage

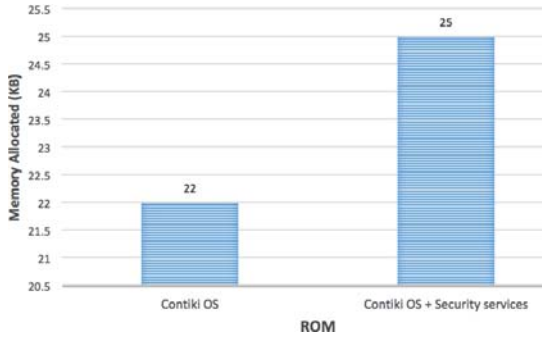


Fig. 13 ROM memory usage

VIII. Conclusion

Security has become essential in IoT devices, but it does, however, come at a cost to resources. Security costs are associated with two main components: the radio for transmission/receiving, and the MCU for processing security operations. The cost of security at several levels has been studied at the IoT MAC layer. The results show that security processes contribute a significant overhead, particularly, in terms of energy consumption, which is quite high. The energy consumed at high security levels may shorten network lifetime significantly. The results also reveal that high security levels increase latency by almost five times over that of the non-secure level when used with an 80 byte payload. It is also observed that latency increases with encryption at a higher rate than it does with authentication due to the greater number of security operations performed by the MCU. In addition, the results have shown that security cost is higher with low transmission power, as the cost of the MCU is not affected by transmission power. Experimental measurements show a significant impact on data throughput. It is reduced by 53% over non-secure packets when authentication is enabled, and 62% when both encryption and authentication are enabled. It is not easy to gather accurate data on security cost, as many factors such as packet length, power transmission, and the type of security service employed can affect the results. However, it is clear that security processes reduce the performance of IoT devices significantly, and energy consumption increases in line with ascending security levels. The results of this paper are aimed to benefit network designers and researchers in terms of security cost, and allow them to choose the level which suits their application requirements. In the future work, a calibration will be made, between emulation results and real-hardware results to check the credibility of the emulation. Also, A mechanism to trade-off security with QoS, and energy consumption will be proposed.

References

[1] Y. Zhong, L. Cheng, L. Zhang, Y. Song, and H. R. Karimi, "Energy-efficient routing control algorithm in large-scale wsn for water environment monitoring with application to three gorges reservoir area," *The Scientific World Journal*, vol. 2014, 2014.

[2] M. Turkanović, B. Brumen, and M. Hölbl, "A novel user authentication and key agreement scheme for heterogeneous ad hoc wireless sensor networks, based on the Internet of Things notion," *Ad Hoc Networks*, vol. 20, pp. 96–112, sep 2014. (Online). Available: http://linkinghub.elsevier.com/retrieve/pii/S157087051400064X

[3] L. D. Xu, W. He, and S. Li, "Internet of things in industries: A survey," *IEEE Transactions on Industrial Informatics*, vol. 10, no. 4, pp. 2233–2243, 2014.

[4] M. Elshrkawey, S. M. Elsherif, and M. E. Wahed, "An enhancement approach for reducing the energy consumption in wireless sensor networks," *Journal of King Saud University-Computer and Information Sciences*, 2017.

[5] Z. Jiang and Y. Pan, *From Problem to Solution: Wireless Sensor Networks Security*. Commack, NY, USA: Nova Science Publishers, Inc., 2009.

[6] S. Sicari, A. Rizzardi, L. A. Grieco, and A. Coen-Porisini, "Security, privacy and trust in internet of things: The road ahead," *Computer Networks*, vol. 76, pp. 146–164, 2015.

[7] M. K. Jain, "Wireless sensor networks: Security issues and challenges," *International Journal of Computer and Information Technology*, vol. 2, no. 1, pp. 62–67, 2011.

[8] D. K. G., M. K. Singh, and M. Jayanthi, Eds., *Network Security Attacks and Countermeasures*. IGI Global, 2016. (Online). Available: http://services.igi-global.com/resolvedoi/resolve.aspx?doi=10.4018/978-1-4666-8761-5

[9] S. B. Othman, A. A. Bahattab, A. Trad, and H. Youssef, "Confidentiality and integrity for data aggregation in wsn using homomorphic encryption," *Wireless Personal Communications*, vol. 80, no. 2, pp. 867–889, 2015.

[10] H. Modares, R. Salleh, and A. Moravejosharieh, "Overview of security issues in wireless sensor networks," in *Computational Intelligence, Modelling and Simulation (CIMSiM), 2011 Third International Conference on*. IEEE, 2011, pp. 308–311.

[11] S. Sciancalepore, G. Piro, E. Vogli, G. Boggia, and L. A. Grieco, "On securing ieee 802.15. 4 networks through a standard compliant framework," in *Euro Med Telco Conference (EMTC), 2014*. IEEE, 2014, pp. 1–6.

[12] S. B. Othman, A. Trad, and H. Youssef, "Performance evaluation of encryption algorithm for wireless sensor networks," in *Information Technology and e-Services (ICITeS), 2012 International Conference on*. IEEE, 2012, pp. 1–8.

[13] A. Trad, A. A. Bahattab, and S. B. Othman, "Performance trade-offs of encryption algorithms for wireless sensor networks," in *Computer Applications and Information Systems (WCCAIS), 2014 World Congress on*. IEEE, 2014, pp. 1–6.

[14] C. Panait and D. Dragomir, "Measuring the performance and energy consumption of aes in wireless sensor networks," in *Computer Science and Information Systems (FedCSIS), 2015 Federated Conference on*. IEEE, 2015, pp. 1261–1266.

[15] J. Lee, K. Kapitanova, and S. H. Son, "The price of security in wireless sensor networks," *Computer Networks*, vol. 54, no. 17, pp. 2967–2978, 2010.

[16] A. Dunkels, J. Eriksson, N. Finne, and N. Tsiftes, "Powertrace: Network-level power profiling for low-power wireless networks," 2011.

[17] A. V. Taddeo, M. Mura, and A. Ferrante, "Qos and security in energy-harvesting wireless sensor networks," in *Security and Cryptography (SECRYPT), Proceedings of the 2010 International Conference on*. IEEE, 2010, pp. 1–10.

[18] J. Misic and V. Misic, *Wireless personal area networks: Performance, interconnection, and security with IEEE 802.15. 4*. John Wiley & Sons, 2008, vol. 1.

[19] A. Dunkels, "The ContikiMAC Radio Duty Cycling Protocol," SICS, Tech. Rep., 2011. (Online). Available: http://soda.swedish-ict.se/5128/1/contikimac-report.pdf

[20] "Moteiv Corporation. SkyTmote Datasheet," 2006, (Online Document) Available: http://www.eecs.harvard.edu/ konrad/projects/shimmer/references/tmote-sky-datasheet.pdf.