

Developing a Systematic Process for Mobile Surveying and Analysis of WLAN security

UNIVERSITY OF TURKU

Department of Future Technologies

Master of Science in Technology Thesis

Networked Systems Security

June 2020

Saku Lindroos

Supervisors:

D.Sc. (Tech) Seppo Virtanen

D.Sc. (Tech) Antti Hakkala

The originality of this thesis has been checked in accordance with the University of Turku quality assurance system using TurnitinOriginalityCheck service.

UNIVERSITY OF TURKU
Department of Future Technologies

SAKU LINDROOS: Developing a Systematic Process for Mobile Surveying and Analysis of WLAN security

Master of Science in Technology Thesis, 109 p.
Networked Systems Security
June 2020

Wireless Local Area Network (WLAN), familiarly known as Wi-Fi, is one of the most used wireless networking technologies. WLANs have rapidly grown in popularity since the release of the original IEEE 802.11 WLAN standard in 1997. We are using our beloved wireless internet connection for everything and are connecting more and more devices into our wireless networks in every form imaginable. As the number of wireless network devices keeps increasing, so does the importance of wireless network security.

During its now over twenty-year lifecycle, a multitude of various security measures and protocols have been introduced into WLAN connections to keep our wireless communication secure. The most notable security measures presented in the 802.11 standard have been the encryption protocols Wired Equivalent Privacy (WEP) and Wi-Fi Protected Access (WPA). Both encryption protocols have had their share of flaws and vulnerabilities, some of them so severe that the use of WEP and the first generation of the WPA protocol have been deemed irredeemably broken and unfit to be used for WLAN encryption. Even though the aforementioned encryption protocols have been long since deemed fatally broken and insecure, research shows that both can still be found in use today.

The purpose of this Master's Thesis is to develop a process for surveying wireless local area networks and to survey the current state of WLAN security in Finland. The goal has been to develop a WLAN surveying process that would at the same time be efficient, scalable, and easily replicable. The purpose of the survey is to determine to what extent are the deprecated encryption protocols used in Finland. Furthermore, we want to find out in what state is WLAN security currently in Finland by observing the use of other WLAN security practices. The survey process presented in this work is based on a WLAN scanning method called Wardriving. Despite its intimidating name, wardriving is simply a form of passive wireless network scanning. Passive wireless network scanning is used for collecting information about the surrounding wireless networks by listening to the messages broadcasted by wireless network devices.

To collect our research data, we conducted wardriving surveys on three separate occasions between the spring of 2019 and early spring of 2020, in a typical medium-sized Finnish city. Our survey results show that 2.2% out of the located networks used insecure encryption protocols and 9.2% of the located networks did not use any encryption protocol. While the percentage of insecure networks is moderately low, we observed during our study that private consumers are reluctant to change the factory-set default settings of their wireless network devices, possibly exposing them to other security threats.

Keywords: wireless networks, encryption, security, wardriving, wireless standard, IEEE 802.11

Table of contents

1.	Introduction	1
1.1.	WLAN security	2
1.2.	Research questions	5
1.3.	Study methodology and scope	6
1.4.	Thesis structure.....	7
2.	Background	8
2.1.	A brief history of wireless networking.....	8
2.2.	ALOHAnet	8
2.3.	Pure and slotted ALOHA	9
2.4.	The Ethernet and collision detection	10
2.4.1.	Carrier-Sense Multiple Access with Collision Detection	11
2.4.2.	Carrier Sense Multiple Access with Collision Avoidance.....	12
3.	IEEE 802.11 Standard	14
3.1.	Standardisation organisations.....	15
3.1.1.	The Institute of Electrical and Electronics Engineers.....	16
3.1.2.	The Wi-Fi Alliance	17
3.2.	IEEE 802.11 1997 Legacy standard	18
3.3.	802.11 a and b amendments	19
3.3.1.	802.11a.....	19
3.3.2.	802.11b.....	20
3.4.	802.11g.....	21
3.5.	802.11n.....	22
3.6.	802.11ac.....	25
3.7.	802.11ax	28
3.8.	The new Wi-Fi Alliance 802.11 amendment naming system	31
4.	802.11 WLAN Security	33
4.1.	The basic principles of cryptography	34
4.1.1.	Symmetric shared-key cryptography	34
4.1.2.	Asymmetric public-key cryptography	35
4.1.3.	Stream and Block ciphers	36

4.2.	802.11 security	37
4.2.1.	Legacy 802.11 security	39
4.2.2.	Wired Equivalent Privacy WEP.....	40
4.3.	802.11i security amendment, WPA-TKIP and WPA2	44
4.3.1.	WPA-TKIP	45
4.3.2.	WPA2 CCMP/AES	49
4.3.3.	WPA-TKIP vulnerabilities.....	52
4.3.4.	WPA password cracking and WPA2 vulnerabilities	53
4.3.5.	802.11 Denial of Service Attacks	58
4.4.	WPA3	59
4.4.1.	WPA3 SAE handshake	60
4.4.2.	Opportunistic Wireless Encryption OWE.....	62
4.4.3.	WPA3 vulnerabilities.....	64
5.	Research methodology	67
5.1.	Wardriving.....	67
5.2.	Operating system, software, and hardware for wardriving	70
5.2.1.	Wardriving software	71
5.2.2.	Wardriving hardware	73
5.3.	Data sampling and analysis	75
5.4.	The legality of wardriving and the GDPR.....	78
5.4.1.	Wardriving and the GDPR.....	79
5.5.	Ethics of wardriving	82
5.5.1.	Utilitarianism and Virtue ethics	83
5.5.2.	Wardriving and Utilitarianism	84
5.5.3.	Wardriving and Virtue ethics.....	85
6.	Research findings	87
6.1.	The three surveyed locations	88
6.1.1.	The industrial district	89
6.1.2.	The city centre.....	90
6.1.3.	The suburb	91
6.1.4.	Use of encryption protocols in the three locations.....	93
6.2.	The bigger picture of WLAN security practices	95
6.2.1.	Encryption protocol use	95

6.2.2. SSID security practices	98
6.2.3. Popular device manufacturers	102
6.2.4. Popular wireless channels	103
7. Conclusions	106
7.1. Potential future research	109
References	110
Appendix A.	122
Abbreviations	122

List of figures

Figure 1 WEP encryption protocol [9]	41
Figure 2 Recovered WEP encryption key	44
Figure 3 Second EAPOL message	48
Figure 4 The four-way handshake process [9]	49
Figure 5 AES in counter mode [9]	50
Figure 6 AES CBC-MAC process [9]	51
Figure 7 Successful WPA dictionary attack	55
Figure 8 SAE handshake [94].....	61
Figure 9 OWE process [98]	63
Figure 10 Kismet user interface.....	72
Figure 11 GPS receiver and WLAN adapter used in this work.....	74
Figure 12 Kismet log files	76
Figure 13 Wardriving survey process.....	77
Figure 14 The industrial district survey route (Screenshot from Google Maps).....	89
Figure 15 The city centre survey route (Screenshot from Google Maps)	90
Figure 16 The suburb survey route (Screenshot from Google Maps).....	92
Figure 17 Encryption distribution between locations in percentages.....	93
Figure 18 Insecure and secure network distribution between locations, in percentages	94
Figure 19 Insecure and secure network distribution when guest networks included in secure networks	95
Figure 20 Encryption distribution.....	96
Figure 21 Encryption distribution in percentages.....	96
Figure 22 Ratio of WPA2-PSK and WPA-PSK mixed mode networks	97
Figure 23 Ratio of insecure and secure networks.....	97
Figure 24 Distribution of visible and cloaked networks in combined results	99
Figure 25 Distribution of visible and cloaked networks in the surveyed areas.....	99
Figure 26 Encryption use in networks with cloaked and visible SSID	100
Figure 27 Percentage of networks with altered and default SSIDs	100
Figure 28 Percentage of altered and default network SSIDs in the surveyed areas	101
Figure 29 Device manufacturers with most default SSIDs	101
Figure 30 Popular manufacturers in percentages	102
Figure 31 Distribution of networks operating on the 2.4 GHz and 5 GHz bands	104
Figure 32 Channel popularity on the 2.4 GHz band.....	104

Figure 33 Channel popularity on the 5 GHz band.....	105
---	-----

List of tables

Table 1 The new Wi-Fi Alliance 802.11 amendment naming system	32
Table 2 XOR process.....	37
Table 3 Encryption distribution between locations	93
Table 4 Insecure and secure network distribution between locations	94
Table 5 Popular manufacturers.....	102

1. Introduction

During the past twenty years we have become accustomed to wirelessly connecting to the internet through a *Wireless Local Area Network* (WLAN) connection, more familiarly known by its marketing name *Wi-Fi*. In twenty years' time, having a WLAN connection has become a commodity to us, something that we expect to be there for us anywhere we go. We might feel lost or get upset if our favourite local coffee shop or the hotel we are visiting on our vacation does not provide us with a wireless internet connection to connect our laptop, tablet, and smartphone to. Many of us might still have a strong memory of the first time we used a WLAN connection, and why wouldn't we? The cumbersome wires previously needed to connect to the internet were replaced with nothing but air. With WLAN, we could browse the internet anywhere we go without having to plug our computer into the other end of and cable and free ourselves to work from the comfort of our own sofa or go to our local coffee shop and connect to a public Wi-Fi hotspot and start answering e-mails.

In this relatively short twenty year period, ever since Apple became the first company to have built-in Wi-Fi support in laptop computers in 1999 [1], we have started connecting almost everything we can imagine to our wireless networks. We can easily state that today Wi-Fi is everywhere and in everything. In 2009 the *Wi-Fi Alliance* (WFA), the organisation responsible for creation and marketing the Wi-Fi brand, announced that the billionth WLAN chipset was sold [2]. In 2012 the annual amount of shipped WLAN devices hit 1.75 billion [2]. The amount of shipped WLAN devices has been estimated to rise over 2.20 billion in the year 2019 and up to 4 billion by the year 2024 [3].

In their 2018 report [4] one of the world's largest network device manufacturers, Cisco, estimated that there would be nearly 549 million public Wi-Fi hotspots worldwide in the year 2022. This would mean a four-time increase from the estimated 124 million public Wi-Fi hotspots in 2017. According to a study conducted by the analytics company Strategy Analytics [5], WLAN capable smart home IoT devices will increase the amount of in-home WLAN devices up to 16 million by the year 2030 from the current 2019 estimate of 4 million in-home devices.

The first steps toward the modern wireless data networking communication were taken in the Hawaiian islands in the late 1960s and early 1970s [1]. A group of faculty members

at the University of Hawaii's Department of Electrical Engineering began a project to wirelessly link computers between the Hawaiian Islands. This project would eventually become the world's first wireless packet data network, the *ALOHAnet* [6]. The influence and legacy of this early wireless network can still be seen in our wireless local area networks, for example in the form of the *Carrier Sense Multiple Access With Collision Avoidance* (CSMA/CA) medium access control scheme [1].

The next big steps for the development of WLAN technology were taken in the mid-1980s when the United States *Federal Communications Commission* (FCC) deregulated the use of the radio spectrum bands needed for our wireless data connections [1]. After the deregulation of the wireless bands the first manufacturers started coming out with their first wireless networking products in the late 1980s and early 1990s [7]. These initial wireless networking systems were marketed mostly toward larger businesses and universities, but because of their complexity, price, and low data rates they did not achieve much success at the time [7].

The early manufacturers often used proprietary technologies and protocols in their systems making the products incompatible between different manufacturers [7]. This sparked the need for industry-wide standardisation of WLAN technology. This need for standardisation would lead the wireless network industry coming into talks with the *Institute of Electrical and Electronics Engineers* (IEEE) standardisation organisation about the possible standardisation of the WLAN technology [1].

After meetings among the different already existing networking standard working groups in the IEEE it was decided that a new group would be needed for the task of developing and governing the standardisation of WLAN technology. The new IEEE 802.11 working group would start its work in the autumn of 1990 [1] and the first official standard for WLAN would be approved in June of 1997 [8]. Since its establishment the 802.11 working group has been responsible for the development of the WLAN standard and is at the time of writing this work finalising the newest 802.11ax amendment for the standard.

1.1. WLAN security

As the amount of WLAN devices has kept on increasing since the early days of the 802.11 standard in the late 1990s and early 2000s the security issues of the devices have become more prevalent. For the common consumer securing a home or a small office WLAN

network has traditionally meant the process of setting up a password to their *Wireless Access point* (AP) that is then shared to authorise and authenticate users on the network. However, there is much more to the issue of security in WLAN networking than mere passwords.

The problems of WLAN security lie in its core idea of being wireless. In wireless networking the main issue is that the information is being propagated through the air around us in radio waves. This means that anyone with the equipment for using a WLAN connection also has the means for eavesdropping on the wireless medium and capturing the transmitted information, much like anyone with a traditional FM radio can tune in to any radio station. To battle against the threats that WLAN connections face the 802.11 standard working group has constructed various security measures and solutions during the standard's lifetime. At the same time some device manufacturers and other entities have developed and implemented their own sometimes proprietary security solutions for WLAN networks.

Due to the security threats WLAN networking faces because of its wireless nature we must secure our communication by scrambling parts of the communicated information so that it cannot be interpreted by the possible eavesdroppers. This scrambling is done by the means of cryptographic encryption algorithms and protocols. In discussions about WLAN security it is not uncommon to see the two terms *encryption algorithm* and *encryption protocol* to be used interchangeably. For the future of this work it is beneficial to make a brief distinction between the two terms.

Mathematical algorithms are procedures designed to solve mathematical problems step by step. A common example of a simple step by step mathematical algorithm is the long division procedure. Cryptographic encryption algorithms are mathematical procedures that have been designed to scramble communicated information into a form that cannot be interpreted without knowing the secret encryption key. These encryption algorithms are utilised as parts of more complete security systems often referred to as security or encryption protocols. Encryption protocols are processes designed to address security issues in a particular application such as wireless networking by applying mathematical encryption algorithms and defining how the algorithms should be used [9]. In the case of WLAN networks, encryption protocols have been designed to offer user authorisation

and authentication as well as confidentiality and integrity for the communicated information.

In this Master's Thesis we will be concentrating on the cryptographic encryption protocols that have been defined in the 802.11 standard. The first cryptographic encryption protocol defined in the original 1997 ratified 802.11 standard was the *Wired Equivalent Privacy* (WEP) encryption protocol [8]. As the number of WLAN products started to rise and the devices became more affordable for the average consumers the interest in research of WLAN security also rose. Through research it was soon found out that WEP was fatally flawed and had to be replaced.

As an interim solution to fix the vulnerabilities and flaws found in WEP it was to be replaced by the *Wi-Fi Protected Access* (WPA) protocol. The first generation of WPA was based on a draft version of the at the time unfinished 802.11i amendment [2]. This first generation of WPA is better known as TKIP or *Temporal Key Integrity Protocol*, WPA-TKIP for short. The TKIP encryption protocol is based on the same cryptographic algorithm as the vulnerable WEP and was originally meant to serve as an extra layer of security over the broken WEP encryption [9]. The reason for using the same encryption algorithm as WEP is based on the idea that WPA-TKIP was to be only a short-term solution and a bandage over the flawed WEP. The use of the same algorithm also meant that consumers would not have to buy new hardware and could simply update the device software [9].

In 2004, the 802.11i amendment was officially ratified and the complete version of WPA, dubbed WPA2, was introduced to significantly improve the security from that offered by WEP and WPA-TKIP [2]. WPA2 provided a stronger cryptographic encryption algorithm compared to WEP and WPA-TKIP and fixed several other insufficiencies found in the WEP protocol [9]. The downside was that by changing the encryption algorithm consumers would have to invest in new hardware instead of just simply updating the device software. As years passed on from the release of the 802.11i amendment, more research was devoted to the security of the WPA protocols and eventually both WPA-TKIP and WPA2 were found to be vulnerable to different types of attacks. At the time of writing this work a newer version of the WPA protocol, named consequently as WPA3, has been released and has again been designed to replace the older protocols. WPA3 has however already

had its hardships as researchers found vulnerabilities in its implementation before it had even been officially released to the market.

Since WEP and WPA-TKIP protocols have long since been broken and WPA2 has been deemed to be vulnerable to several types of attacks, some manufacturers have already started implementing the new WPA3 protocol into their devices. This transition period between protocols provides us with a great opportunity to study the current state of WLAN security and to follow the market acceptance of the new encryption protocol as more WPA3 capable devices come available for consumers. Furthermore, it is interesting to see if there indeed still are devices in use that utilize the broken WEP and WPA-TKIP protocols.

The matter of broken and outdated security protocols has become more pressing due to the growing number of smart and IoT devices sold. We are always adding ever more increasing amounts of devices into our WLAN networks in varied forms of smart and IoT devices that may have poorly implemented or otherwise insufficient security. Outdated or in other ways faulty security implementations in any of the wireless devices connected to a WLAN network could potentially open an attacker a way into the network providing them with the possibility to compromise the security of the entire network and its users.

1.2. Research questions

The purpose of this Master's Thesis is to develop a process for surveying wireless local area networks and to survey the current state of WLAN security in Finland. The goal has been to develop a WLAN surveying process that would at the same time be efficient, scalable, and easily replicable. The purpose of the survey is to determine to what extent are the obsolete and deprecated encryption protocols used in Finland. Furthermore, we want to find out in what state is WLAN security currently in Finland by observing the use of other WLAN security practices.

Based on the presented study objectives, the following research questions have been defined:

1. What is the current state of WLAN security in Finland?
 - a. What encryption protocols are in use today?
 - b. Are there large numbers of unencrypted networks in use?

- c. How frequent is the use of other wireless network security practises?
 - d. Can we find any networks or devices supporting the newest 802.11 amendment and encryption protocol?
2. What is the most effective way to survey wireless networks?
- a. What kind of hardware and software is needed to effectively survey wireless networks?
 - b. How can we develop the surveying process so that it can be easily replicated, and scaled to larger environments?
 - c. What are the possible legal, regulatory, and ethical constraints for surveying wireless networks?

1.3. Study methodology and scope

To answer the set research questions we sought out to develop an effective process of surveying WLAN networks and conducted a survey of WLAN networks in a typical middle-sized Finnish city. The survey was conducted by utilizing a passive wireless network scanning technique called Wardriving. Despite its intimidating name, wardriving is simply a form of passive wireless network scanning. Passive wireless network scanning is used for collecting information about the surrounding wireless networks by listening to the messages broadcasted by wireless network devices to make their existence known to other surrounding devices.

The survey data has been collected on three separate occasions between the spring of 2019 and early spring of 2020. On each survey session, three different areas each representing a different part of the city was surveyed. By surveying three different parts of the city we could collect more diverse results. Each location was surveyed three times on each survey session to ensure that we discover as many networks as possible within each area. The collected data has then been processed and assembled into databases for further analysis.

From the collected results we sought to find information about the use of different encryption protocols and to observe the use of other WLAN security practices. At the end of the study, we should have a better understanding of the current state of WLAN security in Finland and have identified possible problems and deficiencies in current WLAN security practices in Finland. Moreover, we should have sufficient knowledge on how to

successfully conduct WLAN surveys by the means of wardriving in a manner that our research can be easily replicated in different locations.

1.4. Thesis structure

The rest of the thesis is structured as follows:

- In the following second chapter we give a brief history of wireless networking and provide the reader with the needed background information and basic concepts of wireless networking.
- The third chapter further expands the concepts of wireless networking and walks the reader through the different development phases of the 802.11 standard from its humble beginnings up to the current day.
- Chapter four is devoted to the security of 802.11 networks and the encryption protocols promoted in the 802.11 standard. In this chapter we familiarise the reader with the different encryption protocols used in the 802.11 networks alongside the flaws and vulnerabilities found in the protocols.
- Chapter five presents the research methodologies used during this work. We familiarise the reader with the concept of wardriving and present the software and tools needed for conducting WLAN surveys by the means of wardriving. In addition, we discuss the legality and ethics of wardriving.
- Chapter six presents the results of our survey study on the security of WLAN networks.
- Chapter seven concludes the thesis providing a discussion about the results of our survey, the possible avenues for future studies, and improvements.

2. Background

2.1. A brief history of wireless networking

The year 2020 marks the 35th anniversary of the crucial decision that eventually led to the development of our beloved WLAN wireless internet connection, familiarly known today as Wi-Fi. For us to be able to understand the current state of our wireless internet connection, it is important to discuss its history and origins. In this section, we are taking a brief look into the history of the wireless communication system that we today call Wi-Fi and what developments eventually led to its emergence.

The key event that would eventually lead to the innovation of many of the wireless communication technologies we have in use today can be led back to May 9th 1985 [1]. At that date, the US Federal Communications Commission (FCC) Report and Order on docket number: 81-413 “*Amendment of the rules to authorize spread spectrum and other wideband emissions in the Public Safety and Industrial, Scientific, Medical Bands*” was adopted and would later be released on the 24th of the same month [10].

In this document, the FCC allowed the unlicensed and individual use of Spread Spectrum radio communication systems on the *Industrial, Scientific, and Medical (ISM)* bands. The three ISM bands specified in the document were the 902-928 MHz, 2400-2483 MHz, and 5725-5875 MHz bands [10] which are still in use today. The ruling meant that companies and individuals did not have to apply for licences from the FCC to operate on these bands. These ISM bands were originally deemed as “garbage” bands in professional lingo since they were already in use by appliances such as microwave ovens and garage door openers and were thusly contested from the start [1]. In the document FCC defined some basic restrictions and rules for the use the ISM bands. For example, the devices operating on the bands were to be limited to a level of 1 watt maximum peak transmission power and that the systems must accept interference from other devices [7].

2.2. ALOHAnet

Even though the FCC ruling might have been the final push that enabled the development of WLAN technology and its eventual rise as the commercial success it is today, the foundation for the modern wireless data networks was laid 17 years prior in the Hawaiian Islands. In the fall of 1968, a faculty group in the University of Hawaii’s Department of

Electrical Engineering began the planning for an experimental radio-linked computer network. The original goal of the project was to determine the situations where radio communication is preferable over wired communication in computer networking [6].

From this project spawned the world's first wireless packet data network, known as ALOHAnet or the ALOHA system. The first ALOHA terminals went into operation in June 1971. By the year 1973, ALOHAnet was expanded to connect seven computer terminals on four different Hawaiian Islands and would eventually evolve into the first system to utilize a satellite connection for packet-switched networking. The satellite link first connected ALOHAnet to the wired ARPA network used on the US mainland. The satellite connection would eventually become the Pacific Network or "PacNet" for short connecting the University of Hawaii, NASA Ames Research Centre in California, the University of Alaska, Tohoku University in Sendai Japan, the University of Electro-communications in Tokyo Japan, and finally the University of Sydney in Australia. [6]

The network was assigned two 100 kHz bandwidth channels on the *Ultra High Frequency* (UHF) band. A random-access channel at 407.350 MHz was used for communication between the user terminals and the central computer [6]. The second channel at 413.475 MHz was used as the central computers broadcast channel. The theoretical maximum speed of the transmission on these channels was 9600 bits per second [6]. In today's perspective, this would mean 0.001 Megabits or 0.00001 Gigabits per second. The network was originally built as a star topology, meaning that the user terminals did not communicate directly with each other [6]. Instead of communicating directly with each other, the users sent their data on the random-access channel to the central computer where it would then be processed and re-broadcasted on the broadcast channel by the central computer to all of the clients connected to the network [6].

2.3. Pure and slotted ALOHA

Because the network was set up in this manner where every user had to send their data packets on the same contested fixed wireless channel, the network speed was highly affected by simultaneous transmissions and packets colliding with each other and never reaching the central computer. Instead of trying to prevent collisions from happening altogether, the original ALOHA protocol, or the "pure ALOHA" protocol as it is referred to today, was designed to only alleviate the possible effects that packet collisions had on the network [11].

In pure ALOHA, when a user successfully sends a packet to the central computer, an *acknowledgement message* (ACK) is sent by the central computer to the initial sender. If the sender does not receive an ACK message within a certain time period, the sending system calculates a random back-off time to wait until resending the packet to avoid further collisions [11]. The number of retransmissions was limited to three consecutive attempts after which the user would have to manually reinitiate the data transmission. Limiting the retransmission interval down to three consecutive transmissions increased the chances of other clients having successful transmissions, avoiding an infinite loop of re-sent packets flooding the wireless channel. Using the “pure ALOHA” protocol on the network meant that as the number of users grew, so did the chance of packet collisions. This led to a situation where the network was working only at around 18% of its full capacity of 9600 bits per second [11].

After noticing the shortcomings of the pure ALOHA protocol it was refined by Lawrence G. Roberts in 1972 [12]. Roberts noted that the capacity (or throughput) of the network could be improved by dividing the time users could send the data into discrete slots which gave the scheme its name “Slotted ALOHA”. In the slotted ALOHA protocol, a centralized system clock would be used to indicate timeslots in which users could send their packets in [12]. Even though packet collisions are still guaranteed in the slotted ALOHA protocol since two or more clients could send packets at the same time, Roberts was able to double the ALOHA networks throughput to around 37% of full capacity, up from the 18% achieved with the pure ALOHA protocol [12].

2.4. The Ethernet and collision detection

The ALOHA network project and especially the work done on the pure and slotted ALOHA protocols had a great influence on the development of our modern wired and wireless local area networks. In 1972 a man named Robert Metcalfe took an interest in the ALOHA network’s architecture while working on his doctoral thesis for Harvard University [6]. After working on the ALOHA project for some months, Metcalfe was hired by the Xerox company to work on their endeavour to create a network for their newly released Alto workstations [6]. The Xerox Alto workstations themselves were pioneering in the field of personal computing. Already in 1972, they had a graphical user interface, multitasking capabilities and a mouse coupled with a graphical user interface. These would not become standards in personal computing until much later. Metcalfe's work at

Xerox would in 1973 lead to the first version of what we today know as the Ethernet (at the time dubbed the Alto Aloha Network) [1].

2.4.1. Carrier-Sense Multiple Access with Collision Detection

The Xerox network was meant to connect the Alto workstations to each other as well as to file servers and printers in office spaces [1]. For the Alto network, Metcalfe took the principles of ALOHAnet and refined them to work on the wired medium. From this, Metcalfe defined a new and improved medium access control method dubbed “*Carrier-Sense Multiple Access with Collision Detection*” (CSMA/CD) that would eventually serve as the basis of what we today consider the Ethernet [1]. Metcalfe noted that on a wired medium it is possible to “listen” if the medium is free for the user to send their data packets. By listening to the medium to determine whether it is free it was possible to avoid collisions instead of just coping with them as done in the ALOHA network. This innovation would after many phases evolve into the “IEEE 802.3 Carrier Sense Multiple Access with Collision Detection (CSMA/CD) Access Method and Physical Layer Specifications” Ethernet standard to be published in 1985 [13].

The concept of CSMA/CD is fairly simple: the client first listens if the shared medium is idle (Carrier sense), and if there is a signal on the channel it means that some other client is sending data over it and other clients must wait until the transmission is over. Once the medium is silent, every client has an equal opportunity to start transmitting their data (Multiple access). If multiple clients transmit their data at the same time there will be a collision. When a collision occurs the clients sense the collision on the medium which in turn tells the clients to stop transmitting (Collision detection). After a collision is detected by the sending clients, each client calculates a random back-off time, after which they can try retransmitting their data. The same type of back-off method was already used in the ALOHA system although the back-off algorithm was modified by Metcalfe for CSMA/CD. [13]

The CSMA/CD method is no longer necessary on our wired *Local Area Network* (LAN) connections because of the newer Full-duplex mode which enables clients to operate in both directions simultaneously and because hubs have been replaced by switches, alleviating the need to control access on the wired medium [13]. However, a derivation of the CSMA/CD scheme known as the *Carrier Sense Multiple Access with Collision Avoidance* (CSMA/CA) is still in use today in our WLAN connections.

2.4.2. Carrier Sense Multiple Access with Collision Avoidance

CSMA/CA tries to solve the issues of packets colliding when two or more clients transmit data at the same time on the wireless medium. On a wired medium, we could listen to the signal on the wire but with a wireless connection, we cannot detect traffic on the medium in the same manner and certainty [1]. Collisions on wireless networks can happen, for example, when two clients are out of each other's range and cannot, therefore, sense each other's attempts to send data. This issue is known as the *hidden node problem* [1]. As a result of these issues, we are still trying to avoid and reduce the likelihood of collisions instead of detecting them on our WLAN connections.

CSMA/CA works mostly on the same basic principles as the CSMA/CD described in previous section. In CSMA/CA a client first monitors the medium for traffic. If the medium is occupied, a client waits for a random back-off time and check again if the medium is free. If the medium now is free, the client is ready to send its data and wait for an acknowledgement message from the receiving wireless device, most often a wireless *Access Point* (AP). It is important to recall that a client can only hear the traffic from other wireless clients on the network if they are within its range of operation. Because of this hidden node problem, the CSMA/CA protocol has been optimized with an optional extension that adds a two-way handshake between the sending and receiving device. [14]

When the sending client detects that the medium is idle, it sends a *Request to Send* (RTS) frame to the wireless access point. If the medium is free, the wireless access point sends a *Clear to Send* (CTS) frame to the sending client which will start transmitting data after receiving the frame. The CTS frame will be broadcasted to other wireless clients on the network. Based on the information in the broadcasted CTS frame, the other clients calculate a timeframe or a *Network Access Vector* (NAV) during which not to transmit any information. This system helps to solve the issue with two hidden nodes transferring data at the same time, but with an added cost of longer waiting times on the network because of the additional traffic the RTS and CTS packets create. [14]

Although ALOHAnet had a relatively short lifespan of only 5 years between 1971 - 1976 and did not yet utilize the spread spectrum technologies we are using today in wireless communication (spread spectrum technologies were considered to be used in ALOHAnet but were not feasible at the time because of the high costs of the needed technology at the time [6].) its influence on modern wireless data communication networks and the birth of

the Ethernet is undeniable. We still have remnants of the original ALOHA project in our modern cellular, wired, and wireless networking standards. The work on ALOHAnet and its collision detection protocols eventually led to the development of CMA/CD and CSMA/CA medium access control schemes, from which the latter is still in use today in WLAN networks. For us to better understand the current networking systems and protocols, it is important to recognise the influence of these pioneering systems and how they aided in the eventual development of our current networks.

3. IEEE 802.11 Standard

In the previous chapter we briefly discussed the origins of wireless networking and some of the pioneering systems that influenced the development of the wireless communication technologies we have today. The purpose of this section is to trace back some of the steps it took for WLAN standardization to become reality and to go through the development process and the most significant amendments of the IEEE 802.11 standard.

For something to become a *de facto* technology, there must be industry-wide standardization to ensure interoperability between devices by different manufacturers. After the FCC 1985 ruling allowed unlicensed use of the ISM bands, there was no huge surge of wireless networking devices on the market [1], [7]. The early WLAN systems of the late 1980s were overpriced, power-consuming and too large for individual consumer markets and were therefore mostly marketed and sold to larger businesses and universities [7]. Moreover, the lack of industry-wide standardization meant that the early WLAN products produced by different manufacturers were mostly proprietary leading to a situation where different products would be incompatible between manufacturers [7].

The first steps toward the WLAN standardization were taken in 1988 when the NCR corporation (at the time known as *National Cash Register*) sought to develop a wireless *network interface card* (NIC) to enable wireless LAN communication for their point of sale terminals. NCR had already made efforts to wirelessly connect their retail cash register and point-of-sale terminal systems wirelessly to back-end mainframe computers ever since the FCC's 1985 ruling and had been testing the viability of WLAN communication technology. NCR had decided that the wireless NIC should operate on the 902-928 MHz band to provide maximum range and for the lower costs of technology on the lower frequency compared with the higher 2.4 and 5 GHz frequencies. [1]

For WLAN technology to be on par with its wired counterparts, the issues with medium access control that had riddled wireless networking systems since the days of ALOHAnet had to be resolved first. In their wireless NIC design, NCR decided to leverage the already existing protocols relying on the IEEE 802 family standards and on the *Open System Interconnection* (OSI) network model, which had become a common networking industry practice since the IEEE 802.3 Ethernet standard was published in 1985 [1].

After considering the existing options for the medium access control protocol, NCR took interest in the IEEE 802.4 working group, which at the time was responsible for the development of the token-passing bus access method for LANs [1]. NCR's Bruce Touch and Victor "Vic" Hayes (the latter would eventually become known as "the father of Wi-Fi") took part in the meetings of the IEEE 802.4 task group, but it was soon noted by Hayes that the token bus protocol would not be sufficient for the means of wireless LANs [1].

The token bus protocol relies on a "token" frame which would be passed between clients on the network. Only the client holding the token would be allowed to transmit and, in this manner, controlling the access to the medium. If there would be an error and the token was lost, a token recovery algorithm would be initiated. This kind of access control method would work fine on a more reliable physical wired medium where lost tokens would be a rare event. However, it was soon noted that the token bus would not be feasible on the less reliable wireless medium that can suffer from possible interference from other devices and where collisions are more common. [1]

After meetings with the IEEE 802.3 and 802.4 working groups, it was decided to establish a new working group for the wireless LAN standard with Hayes as the chairman of the group, a place he would hold over ten years and earn him the title "father of Wi-Fi". The first official meeting of the IEEE 802.11 working group was held in September of 1990 [1]. The standard would have to define action on the two lowest layers of the OSI network model, the *Physical (PHY)* and the *Data Link Layer (DLL)* or to be more precise on the *Medium Access Control (MAC)* sublayer of the data link layer. In addition, it was further decided that separate task groups would be established for both layers [1]. For the sake of simplicity, we will be using the MAC layer to describe the second layer of the OSI model for the rest of this work.

3.1. Standardisation organisations

Before venturing further into the 802.11 standard it would be beneficial to familiarise ourselves with the different standardisation organisations that are discussed during this work as well as their roles and operations. The two main organisations responsible for the standardisation and certification of WLAN technology that are discussed in this work are the Institute of Electrical and Electronics Engineers (IEEE) and the Wi-Fi Alliance (WFA). There are of course many other organisations involved in the standardisation and

regulation processes of different aspects considering the WLAN technology. Organisations such as the *International Organization for Standardisation (ISO)*, the *European Telecommunications Standards Institute (ETSI)*, the *Internet Engineering Task Force (IETF)* and the already mentioned *Federal Communications Commission (FCC)* all have a role in the standardisation of WLAN technology [2]. As our focus will be on the actions and relationship between the *Institute of Electrical and Electronics Engineers IEEE* and the *Wi-Fi Alliance*, it is only appropriate to discuss them briefly.

3.1.1. The Institute of Electrical and Electronics Engineers

The Institute of Electrical and Electronics Engineers IEEE was founded in 1963 when two organisations, the *American Institute of Electrical Engineers (AIEE)* and the *Institute of Radio Engineers (IRE)*, merged together [15]. IEEE is a professional association for electronic and electrical engineers with over 422,000 members in over 160 countries [15]. IEEE's mission statement is to “foster technological innovation and excellence for the benefit of humanity” [16]. For us, this mission statement means creating, developing, and overseeing standards we use in various communication technologies.

For this work, the most notable standards governed by the IEEE are the 802.3 Ethernet and 802.11 WLAN standard families. The standards IEEE provides are written documents that describe how the technical processes and equipment governed over by the standard should function [2]. The system, unfortunately, leaves space for different interpretations when the standards are being developed and drafts of them are released [2]. This can lead to a situation where some early products based on a draft of a standard are not compatible with other products based on the same draft or eventually with the finalised standard, as was the case with the early 802.11 products [7].

Each standard has its own “working group” in charge of its development. The working group's number is assigned as the groups are formed. For example, the 802.11 working group is the 11th working group in the IEEE 802 project family [2]. When a need for a revision of the standard or some other issue arises, the working group assigns a task group to resolve the issue. The task groups are assigned a letter and that letter has traditionally been added to the end of the standard amendment as can be seen for example in 802.11a and 802.11b [2]. As time has progressed and all the available letters have been used, multiple letters have been assigned to the task groups as seen for example in 802.11ac and 802.11ax. This system has caused some confusion in consumers and since 2018 the

Wi-Fi Alliance has promoted a new system for naming the amendments based on simple numbers [17]. According to this new system, 802.11b devices would be dubbed as “Wi-Fi 1”, 802.11a devices as “Wi-Fi 2”, and so forth.

3.1.2. The Wi-Fi Alliance

At the wake of the popularisation of WLAN devices in 1999 major device manufacturers decided to come together and form a non-profit association to battle the evident interoperability issues between the different device manufacturers [2]. Starting as the *Wireless Ethernet Compatibility Association* (WECA) and later in 2002 changing its name to the Wi-Fi Alliance, the association consists of over 550 member companies and has certified over 50 000 products [2], [18]. The Wi-Fi Alliance is responsible for creating and marketing the Wi-Fi brand in addition to promoting new 802.11 WLAN solutions to consumers when they become available [2]. In addition to the significant marketing and promotional responsibilities, the most important task of the Wi-Fi Alliance is to ensure the interoperability of different 802.11 devices through its certification programs. This certification ensures that consumers can be sure that their new WLAN device will be interoperable with any other certified wireless device.

Interoperability is achieved by providing certification programs and testing for 802.11 products. At the time of writing this work, the Wi-Fi Alliance has released its certification program for the newest 802.11ax or as according to the already mentioned newly promoted naming system “Wi-Fi 6” products [18]. The certification programs do not only consider the interoperability of the 802.11 radios used in different products. The certification also covers the security, *Quality of Service* (QoS), coverage, and multimedia capabilities of the products [2]. For a manufacturing company to be able to use the Wi-Fi Alliance’s “Wi-Fi certified” logo on its product packaging or marketing, it must meet some requirements. Firstly, the company must be a member of the Wi-Fi Alliance. Secondly, the product must pass the certification program and tests conducted in a Wi-Fi Alliance authorised test laboratory [19].

Although Wi-Fi Alliance consists of a large number of device manufacturers and has certified vast quantities of products, it should also be stated here that the Wi-Fi Alliance certification is not in any case obligatory for any device manufacturer [2]. It is also important to understand that the IEEE and Wi-Fi Alliance, although working side by side, are two very different organisations and have different tasks. The IEEE provides the

802.11 standard describing the technological functions of WLAN devices and is responsible for the development of the standard. In contrast, the Wi-Fi Alliance is more of an advocacy group for the Wi-Fi brand responsible for the Wi-Fi certification programs and marketing the brand [2]

3.2. IEEE 802.11 1997 Legacy standard

After its first meeting in September of 1990, it would take until June 1997 for the IEEE 802.11 working group to approve the original 802.11 standard, now dubbed the “Legacy standard” [8]. Delays in the process were caused by arguments between different manufacturers competing over whose designs and proposals would be considered for the standard [1]. The main subjects of the arguments considered the medium access control protocols and which spread spectrum modulation technique should be used on the physical layer, either *Frequency Hopping Spread Spectrum* (FHSS) or *Direct Sequence Spread Spectrum* (DSSS) [1].

Because of the arguments over the standard’s PHY layer, the original standard defines three different solutions: Frequency Hopping Sequence Spread and Direct Sequence Spread Spectrum techniques on the 2.4 GHz band and the more obscure Infrared PHY at 316-353 THz [20]. The infrared PHY had no actual implementations, but it remains part of the standard [20]. Data rates defined in the standard are 1 Mbps with 2 Mbps as optional for FHSS and infrared. For DSSS both speeds were defined as mandatory which in practice meant that it would operate on 2 Mbps at close range and 1 Mbps at greater distances [1].

The standard describes two supported network topologies, ad-hoc and centralized mode. An ad-hoc WLAN network is a peer-to-peer type network where the network clients connect directly to each other without the need for external networking infrastructure. In centralized mode the clients connect to a wireless access point, sometimes referred to as a *Base station* (BS), which is connected to a higher speed backbone network connection. [1]

The legacy standard could be considered as a beta standard for the 802.11 family. At the time of its release in 1997, the speeds it provided were insufficient compared to its wired Ethernet counterpart which had already evolved to deliver 100Mbps transfer rates in the

mid-1990s and would reach Gigabit rates by the end of the 1990s [13]. Because the standard provided three different PHY layer options, interoperability between manufacturers was still an issue that needed to be addressed in the future [1].

3.3. 802.11 a and b amendments

At the time of the legacy standard's approval, the insufficiencies in the transfer rates and interoperability were fully known within the working group [1]. These realisations led to the establishment of two task groups that would tackle these issues. Task group *a* was established formally in September 1997 with the mission of developing the standard to support higher data rates on the 5 GHz band [1]. Task group *b*, formed in December 1997, was tasked with improving the data rates on the 2.4 GHz band. Both amendments would eventually be approved in September 1999. The IEEE 802.11a would officially be released on the 30th of December in 1999 [21] and IEEE 802.11b following on the 20th of January 2000 [22]. Both amendments kept the base MAC layer of the legacy standard intact whilst bringing improvements to the PHY layer.

3.3.1. 802.11a

The most influential change the 802.11a made to the standard besides increasing the theoretical maximum data rate up to 54 Mbps is the new modulation scheme, *Orthogonal Frequency-Division Multiplexing* (OFDM). OFDM is a multicarrier modulation technique that divides one wider frequency channel into smaller subcarriers or “tones” each used to transmit data [23]. 802.11a originally defined 12 non-overlapping 20 MHz channels which are then divided into 52 OFDM subcarriers with the separation of 0.3125 MHz. From these 52 subcarriers, 48 are used to carry data and 4 are used for carrying pilot data which is used for error correction [23].

The data rates on 802.11a can be reduced to 48, 36, 24, 18, 12, 9 and finally 6 Mbps based on the signal condition. Different modulation schemes are used depending on the signal condition. On a higher condition signal, *Quadrature Amplitude Modulation* (QAM) is used and on a lower signal condition, either *Quadrature Phase Shift Keying* (QPSK) or *Binary Phase Shift Keying* (BPSK) is used. [24]

802.11a had its advantages in the considerably increased data rate compared with the legacy standard and with having less interference because of the utilisation of the less contested higher 5 GHz frequency band. One of the biggest downsides of the 802.11a

amendment also lies in the 5 GHz band. Because of the higher frequency it has a shorter range compared to the 2.4 GHz band since higher frequency radio waves have less penetration through solid objects. Moreover, because of the change in modulation and frequency, the amendment was not backwards compatible with legacy standard equipment. [24]

3.3.2. 802.11b

The 802.11b amendment extends the legacy standard on the 2.4 GHz band increasing the maximum data rate up to 11 Mbps. To achieve the higher data rates 802.11b improves the legacy standard's Direct Sequence Spread Spectrum (DSSS) modulation technique. This improved technique dubbed *High Rate Direct Sequence Spread Spectrum* (HR-DSSS) utilizes the *Complementary Code Keying* (CCK) modulation scheme on the higher 5.5 and 11 Mbps rates. On the lower 1 and 2 Mbps rates *Differential Binary Phase Shift Keying* (DBPSK) and *Differential Quadrature Phase Shift Keying* (DQPSK) modulation schemes are used. [24]

802.11b defines 14 channels each with 22 MHz bandwidth. Channels from 1 to 11 are allowed in the United States and channels from 1 to 13 are allowed Europe. Because the channels' middle frequencies are only 5 MHz apart, they will overlap with adjacent channels and cause interference. In US channels 1, 6 and 11 are non-overlapping whereas in Europe channels 1, 5, 9 and 13 are non-overlapping. This overlapping should be considered when creating new WLAN networks with multiple access points. [24]

Although the 802.11a amendment had its advantages over 802.11b with its higher data rates and 12 non-overlapping channels on the less crowded 5 GHz band. The 802.11b amendment became the one that would eventually launch WLAN products to mass market success. Because 802.11b did not make any drastic changes on the legacy standard it provides backwards compatibility and a chance for manufacturers to use much of their already existing designs and technology [7].

The first commercial 802.11b devices got to the market already in 1999 at the time of the standard's release. In July of 1999, Apple became the first manufacturer to have built-in support for WLAN communication when it released its first iBook laptops and the AirPort product line [1]. The first 802.11a devices got to the consumer market much later in 2001 [7], [9] at which point the 802.11b products had largely taken over the market. 802.11a

had to battle against regulatory issues in Europe considering the use of the 5 GHz band, which in part had its effect on the popularity of 802.11a [25].

3.4. 802.11g

Already in March of 2000 IEEE decided to establish a new task group to bridge the gap in data rates between the newly released 802.11a and 802.11b [7]. Approved on the 12th of June 2003 the IEEE 802.11g amendment combines the best efforts made on the PHY layers of its two predecessors [26]. 802.11g combines the theoretical maximum 54 Mbps data rate of 802.11a by applying OFDM modulation and the longer range of 802.11b using the 2.4 GHz band. The different PHY layer options presented in 802.11g are in some cases referred to as *Extended Rate Physical* (ERP) as it is referred in the official IEEE 802.11g documentation [26], [27].

802.11g defines four different modulation techniques to be used on the PHY layer, two mandatory and two optional. ERP-DSSS-CCK and ERP-OFDM are defined as mandatory and DSSS-OFDM and ERP-PBCC (*Packet Binary Convolutional Coding*) are optional. Despite the ERP prefix attached to them, the modulation techniques are essentially the same as they have been described in the preceding amendments with only some minor necessary changes. [28]

DSSS-CCK modulation is used on the lower data rates from 1 to 11Mbps to provide backwards compatibility with legacy and 802.11b devices. OFDM modulation is used on the new 802.11g devices to provide higher theoretical data rates from 6 Mbps all the way to 54 Mbps. The two optional modulation techniques provided some improvements, but their implementation was deemed voluntary and neither was widely implemented by manufacturers. [28]

Because the newer 802.11g devices had to be backwards compatible with the older 802.11b devices, some compromises were made on the amendment to achieve this. The new 802.11g devices using OFDM and older 802.11b devices using DSSS-CCK modulation can coexist, but the two cannot hear each other [25]. In other words, when an 802.11b device operates with 802.11g access point, it cannot detect any possible OFDM communication and therefore cannot determine if the wireless channel is occupied causing collisions/interference [25]. This presents a new type of the “hidden node problem” described earlier in section 2.4.2.

For the 802.11g and 802.11b devices to interoperate, a protection mechanism had to be implemented to prevent the possible interference between the two. The concept of the mechanism is fairly simple. When an 802.11b device joins a network operated by an 802.11g access point, the RTS and CTS frames (described in section 2.4.2) are sent by using the slower DSSS-CCK modulation used by legacy and 802.11b devices. In this manner, the 802.11b devices can recognize if the medium is occupied by OFDM transmissions. [25]

The disadvantages of the 802.11g amendment stem from the need for interoperability with legacy devices and the use of the 2.4 GHz band. The possible improvements achieved by implementing OFDM modulation on the 2.4 GHz band can be cancelled by the presence of legacy and 802.11b devices on the 802.11g network. Because of the great success of 802.11b devices, it is very probable that an 802.11g network has 802.11b devices operating in it. The extra strain the protection procedure causes on the network combined with the inherent issues included in the crowded 2.4 GHz band can slow the network down significantly. [28], [29]

3.5. 802.11n

The work for the next new amendment to the 802.11 standard family had once again started soon after the finalization of its predecessor. The work on 802.11n lasted from the year 2002 until its release in October of 2009 [30], [31]. While the 802.11n was in development between the years 2003 and 2009, IEEE released several small amendments and specifications improving various aspects of the 802.11 standard. Some of the improvements included the 802.11j disclosing regulatory issues considering the use of the 5 GHz band in Japan (2004), 802.11e enhancing the *Quality Of Service* (QoS) on the MAC layer (2005), and 802.11k for better radio resource management improving the way traffic is distributed in a WLAN (2008) [1], [32].

The increase in the number of WLAN devices and the emergence of higher data rate multimedia services such as online video streaming services, *Voice over IP* (VoIP) services, and online gaming created a need for drastically increased throughput in WLAN. Up to this point the increases achieved in the real-world throughput had been mild, only increasing from 2 Mbps to around 25 Mbps from the legacy standard to 802.11g. In a real-world situation users could now expect throughput between 80 Mbps and 150 Mbps [33], [34]. In an optimal setting users could possibly achieve a throughput of over 200

Mbps [35], the maximum theoretical data rate of 802.11n being 600 Mbps [2]. For this drastic increase in data rates, the 802.11n amendment is sometimes also referred to as *High Throughput* (HT).

As the 802.11n standard presents a plethora of different new technologies and improvements and going through the higher details of all of them would be out of the scope of this work we are only going to briefly discuss the basic principles and features of the most important presented new technologies. A more detailed explanation of the technologies can be found for example from [2], [35], and [36].

The defining elements of 802.11n are its improvements on the PHY layer by utilising multiple transmitting and receiving antennas or as the technology is usually referred to as *Multiple-Input Multiple-Output* (MIMO). Other improvements on the PHY layer include the utilisation of 40 MHz channels by using channel bonding. In channel bonding, two adjacent 20 MHz channels are bonded together doubling the frequency available for transmissions. When we bond two 20 MHz channels together, we must choose which 20 MHz channel we are using as a primary channel for carrier sensing to check that no other device is transmitting at the same time on the same channels. This is important when there are several access points in the same area so that we do not have channels overlapping and causing interference. [35]

802.11n also makes use of dual-band technology. This means that 802.11n can operate either on the 2.4 or 5 GHz spectrum. The possibility of using both 2.4 and 5 GHz spectrums also means that 802.11n is backwards compatible with 802.11a/b/g devices. Improvements were also made on the MAC layer by introducing *frame aggregation* [2].

The introduction of MIMO technology is the bread and butter of the 802.11n amendment. The use of MIMO technology requires the utilization of multiple antennas hence the name multiple-input, multiple-output. One of the great benefits of MIMO is that it takes advantage of the fact that signals tend to reflect from different surfaces or can be blocked by a natural object causing two or more versions of the same signal to arrive at the receiver at different times and with different amplitudes [37]. This signal propagation phenomenon is called *multipath*.

In traditional 802.11 networks, multipath had a negative effect since copies of the same signal could get to the receiver at different times or natural object could fade the strength

of the signal. With the utilization of MIMO and multiple antennas, we can take advantage of this issue. In a MIMO setup, the receiving device with multiple antennas can use each of the signals arriving at different times and process them separately and combine them as one. It should be stated here that even though we have multiple antennas in use at the same time, it is not possible for an access point to serve multiple clients at once and we are still confined to serving one client at a time. [2], [35]

In a MIMO setup where we have multiple antennas in use, we can send multiple independent data streams with each data stream containing unique data. This technique is known as *Spatial Multiplexing* (SM) or *Spatial Diversity Multiplexing* (SDM) [2]. The benefit of spatial multiplexing, of course, is that sending two unique data streams will give us a drastic increase in throughput. In theory, if a MIMO access point sends two unique data streams to a MIMO capable client, we have doubled our throughput. With the same idea, we can say that in a setup with three sending and receiving antennas we have tripled our throughput [37]. We could imagine a situation where an access point uses three spatial streams with a throughput of 85 Mbps for each stream: the combined throughput would equate to 255 Mbps. 802.11n defines the possibility to use up to four antennas in a MIMO setup.

The improvements on the MAC layer in the 802.11n are directed towards reducing the overhead and congestion caused by all the different frames being sent between clients, for example, the acknowledgements sent for each transmitted frame. To reduce the number of these frames *frame aggregation* was introduced. The basic idea behind frame aggregation is to combine two or more frames into a single transmission. Two methods for frame aggregation are introduced in the amendment, *Aggregate MAC Service Data Units* (A-MSDU) and *Aggregate MAC Protocol Data Unit* (A-MPDU). [2]

One natural limitation to frame aggregation is that all the aggregated frames must be addressed to the same client or access point meaning for example that an access point cannot aggregate frames destined to two different clients. Another limitation for this technique is that the aggregated frames must be sent at the same time which in some cases can cause delays on the network. With frame aggregation, it is possible to decrease the probability of packets colliding and reduce the client back-off times while waiting for acknowledgements between transmissions. [37]

802.11n was a revolutionary step for the 802.11 standard. Until this point, the new amendments had been mere updates and small improvements with only slight increases in the real-world throughput. With the utilization of multiple antennas, MIMO, spatial streams, channel bonding of two 20 MHz into a 40 MHz channel and with the improvements on the MAC layer with frame aggregation the real-world throughput got increased to over 200 Mbps from mere 25 Mbps. This increase meant that WLAN could start to compete with the speeds of traditional wired LAN.

3.6. 802.11ac

In response to the need for faster data rates IEEE sought to develop improvements on the 802.11 standard and in December 2013 802.11ac was ratified. The 802.11ac amendment promises to take WLAN technology throughput from Megabits per second to Gigabits per second and for this reason, 802.11ac is referred to as *Very High Throughput* (VHT), inheriting the High Throughput part from its precursor. 802.11ac does not bring new technologies to the standard in the same magnitude as 802.11n did and is more of an update to 802.11n. 802.11ac takes the new advancements of its predecessor and improves on them to achieve the next level in data rates and robustness. [2]

One very distinctive feature of 802.11ac when compared with its predecessor is that it operates only on the less congested 5 GHz band leaving the 2.4GHz band behind. The reasons for this might lie in the introduction of 40 MHz channels in 802.11n and the fact that the 2.4 GHz band is not wide enough to host multiple non-overlapping 40MHz channels. As discussed in section 3.3.2, we can only have 3 to 4 non-overlapping 20 MHz channels on the 2.4 GHz band meaning that having multiple 40 MHz channels without interference would be impossible. On the 5 GHz band we have up to 25 non-overlapping 20 MHz channels available depending on the regional restrictions [38], [39]. Moreover, it might be that IEEE is trying to nudge consumers to leave the more congested 2.4 GHz band and move on to the wider 5 GHz band.

There are other regulatory restrictions concerning the use of the 5 GHz band. For example, there are restrictions on which channels are permitted to be used indoor and which are to be used only outdoors. In practice, these regulations mean that in the EU area there are five 80 MHz and two 160 MHz non-overlapping channels we can use [40]. More details on the regulatory issues can be found in [39] and [40].

In the wider 5 GHz band there is much more frequency space available and therefore 802.11ac makes use of the channel widths of 40, 80 and even 160 MHz. It should be noted that the support for 160 MHz channels is deemed optional in the amendment [2]. Just as in 802.11n that forms 40 MHz channels by combining two 20 MHz channels together, 802.11ac uses the same channel bonding technique as 802.11n and combines two 40 MHz channels into one 80 MHz channel and two 80 MHz channels into a 160 MHz channel [41].

The use of 40, 80 and 160 MHz channels proposes issues on the channel selecting process in the spaces with multiple access points. We may have more non-overlapping channels and frequency space on the 5 GHz band than on the 2.4 GHz band, but if we want to utilize the higher throughput provided by 80 MHz channels we must make sure that none of the access points in the area are operating on the same channels and causing interference between each other.

As in 802.11n, we are choosing which one of the bonded channels is the primary channel to be used for carrier sensing and packet detection to see if any other access points are transmitting at the same time on the same channel [41]. If there would be overlapping between access points operating 80 MHz wide channels, the access points would downgrade to 40 MHz channels to avoid overlapping and therefore lose the benefits of the 80 MHz channel. This technology is called *dynamic bandwidth operation* [2]. This feature brings its own complexity to the channel selection process in areas with multiple access points [2].

802.11ac brings with it changes to the MIMO technology introduced in 802.11n. As we noted in section 3.2, even though we have multiple antennas in our use we could not serve multiple clients at once. 802.11ac changes this by presenting *multiuser MIMO* or MU-MIMO. For the sake of clarity, the MIMO technique presented in 802.11n can be seen referred to as *Single-User MIMO* or SU-MIMO. With MU-MIMO, an access point can communicate with up to four clients simultaneously [2]. MU-MIMO also increases the number of spatial streams from 802.11n four up to eight [2]. We could compare this transition to replacing an Ethernet hub with a switch. MU-MIMO enables us to serve multiple clients simultaneously with less delay and higher data rates.

There are of course limitations to this technology, the first one being that most of the client devices that we use (e.g. smartphones) do not support even the basic MIMO technology due to costs in technology and issues with device battery life. Another limitation to MU-MIMO is that it can be used in downlink transmissions only from an access point to multiple clients due to the advanced signal processing required for MU-MIMO. [42]

802.11ac also makes use of *Beamforming*. Different forms of beamforming were introduced already in 802.11n amendment but it was neither widely adopted by chipset manufacturers nor does the WI-FI Alliance test it for 802.11n certification [2], [42]. The antennas in access points and clients usually radiate their signals omnidirectionally and the signals travel horizontally away from the antennas [42]. With beamforming, we can focus and direct the signal toward a client device. In beamforming, the multiple antennas on the access point or client transmit the same information through different antennas. The transmissions are timed so that they arrive at the receiver at the same time and in phase [2]. Beamforming should increase signal strength and, in this way, make it possible to use better modulation schemes thus increasing throughput.

The method of beamforming used in 802.11ac is referred to as *Explicit Beamforming*. Explicit beamforming uses an interactive calibration process between the devices to identify how to perform the transmission with multiple antennas, this process is known as *channel sounding*. In an MU-MIMO setup, beamforming can be used for guiding the signal toward multiple individual clients simultaneously, not only toward one client at a time. Going in further details of beamforming would be out of the scope of this work. For more details on beamforming, see [2], [38], [39], and [42].

Another improvement in the 802.11ac amendment is the use of 256-QAM modulation scheme which further improves throughput. On the MAC layer 802.11ac makes use of the same frame aggregation technique presented in 802.11n. The only change compared with 802.11n is that all packets are transmitted in the Aggregate MAC Protocol Data Unit (A-MPDU) format, even if only one frame is being transmitted. In addition, some extensions to the Ready To Send / Clear To Send (RTS/CTS) mechanism are added. [2]

802.11ac takes the technologies presented in 802.11n and improves on them to bring the WLAN technology to the Gigabit transmission rate era. Theoretically with 802.11ac a maximum data rate of 6.9 Gbps could be reached on a 160 MHz channel with eight spatial streams [43]. This is a very drastic increase compared with 802.11n's maximum data rate

of 600 Mbps. Of course, these numbers are only theoretical and do not translate to real-life situations with all the variable interference factors that affect WLAN connections. Taking in to account all the factors in optimal real-world situations on the wider channels, with only a couple of clients and multiple spatial streams it could be possible to achieve the over 1 Gbps data rates [43].

One of the disadvantages of 802.11ac is that it is only backwards compatible with 802.11a/n devices due to the use of the 5 GHz band. This issue can be tackled by purchasing an 802.11ac wireless access point with dual-band capabilities. In general, this means that the access point has both 802.11ac and 802.11n radios with the other radio working on the 5 GHz band serving 802.11ac capable devices and the other working on the 2.4 GHz band serving 802.11 n/g/b devices.

3.7. 802.11ax

At the time of writing this work, a new amendment to the 802.11 standard is in its final steps. The Wi-Fi Alliance has launched its certification program for the new standard and the first devices have already been certified for the new standard [44]. A version 3.0 of the draft was released in July 2018 and the new devices are based on the published draft versions of the amendment. The final amendment is set to be officially ratified after mid-2020 [45]. The distinctive difference of 802.11ax compared with its precursors is that instead of seeking ways to maximize data rates for a few users, 802.11ax tries to improve the user experience in more crowded environments varying from large offices, mass events to apartment buildings.

This change in direction is driven by the rise in client devices ranging from smartphones to the Internet of Thing (IoT) devices. To be able to serve the ever-increasing number of devices with good average throughput some changes to the standard must be made. Due to this change of pace from improved throughput to improved efficiency, the 802.11ax amendment has been titled *High efficiency* (HE), sometimes referred to as high-efficiency WLAN (HEW) [46].

One of the most important changes toward the more efficient use of the wireless medium in WLAN is the changes that have been done to the Orthogonal Frequency Division Multiplexing OFDM modulation scheme which has stayed quite the same since its introduction in 802.11a. As already discussed in section 3.3.1 on 802.11a, OFDM takes a wireless

channel and divides it into closely spread frequencies known as subcarriers or “tones”. The improved technology has been dubbed as *Orthogonal Frequency Division Multiple Access* (OFDMA). Similarly to OFDM, OFDMA takes a wireless channel and divides it into subcarriers, but this time we are also splicing the subcarriers into multiple groups known as *resource units* (RU) [47].

In practice, this means that instead of giving the channel to one device at a time, we can now serve multiple devices in parallel by allocating a slice of the channel according to the needs of the receiving device instead of giving the whole channel only to one device at a given time. OFDMA divides a 20 MHz channel into 256 subcarriers which can then be divided into blocks containing either 26, 52, 106 or 242 subcarriers [48]. On higher channel widths, we can divide the channel up to 996 subcarriers [47]. These divided blocks are the *resource units* we can allocate to devices. A resource unit containing 26 subcarriers equates roughly to a 2 MHz slice of a whole 20 MHz channel.

We can divide a 20 MHz channel into 9 resource units each containing 26 subcarriers. In the same manner, a 40 MHz channel can be divided into 18 units, 80 MHz channel into 37 units and finally a 160 MHz channel into 74 units, each unit containing 26 subcarriers. In theory, this would mean that depending on the channel width we are operating on we could serve up to 74 clients simultaneously. This is because we are not occupying the whole channel for one device while transmitting, just a small slice of it. [49]

OFDMA can be used in both downlink and uplink transmissions. For this reason, it can also be seen dubbed as *Multi-User OFDMA* [50]. The uplink OFDMA transmission functions similarly to the downlink transmission, but the uplink transmission calls for more coordination. To coordinate the uplink transmissions the access point sends a control frame or a *trigger frame* (TF) to the clients to inform them which subcarriers they can use for their transmission [47]. In other words, multiple client devices can now transmit simultaneously on different resource units allocated to them. The access point then receives the transmitted frames and demodulates them in parallel [51]. OFDMA has been implemented in cellular data networks before, but now 802.11ax brings the technology into WLANs.

Another improvement 802.11ax makes to pre-existing technologies considers the MU-MIMO technology presented in 802.11ac. MU-MIMO in 802.11ac tried on its part to relieve the issues created by the increasing amount of WLAN devices by allowing multiple simultaneous transmissions in downlink traffic from access points to clients. 802.11ax brings improvements to MU-MIMO technology by allowing it to be used in both downlink and uplink traffic from clients to access points [51]. 802.11ax also increases the number of supported spatial streams from four up to eight, increasing the number of individual clients an access point can serve simultaneously [51]. The uplink MU-MIMO is not going to be featured in the first wave of 802.11ax consumer devices because of its complex nature. We will probably see it implemented sometime in the future after the official release of the amendment. A more detailed presentation of the uplink MU-MIMO can be found in [47].

The main purpose of 802.11ax is to improve efficiency in WLAN networks in dense situations where many access points must be deployed at close range to serve many client devices. A situation where several access points have been deployed in close proximity, they can cause interference with each other reducing the wireless network efficiency and throughput. 802.11ax tries to tackle this issue with the introduction of *Basic Service Set* (BSS) Colouring [52]. Basic Service Set is used to define a set of wireless network devices that communicate together in 802.11 a network. As an example of a basic service set, we can think about a basic wireless network in a household, which usually consists of an access point and client devices [2].

In a dense situation, nearby access points must operate on the same channels and must take turns for transmissions. The colour of a BSS is based on a numerical value from zero 0 to 63. The identifier is added as a 6-bit value on the PHY-header of 802.11 frames. From this numerical value, the 802.11ax access point can identify where the frames are originating from. Every client associating with an access point takes on the same colour as the access point. If an access point detects a transmission on the same channel with the same colour it backs off from transmitting. If on the other hand, an access point detects a transmission on the same channel but with a different colour and weak signal, it can use the channel for transmission since the transmission is originating from a different BSS with a weak signal. [52]

The numbers indicating the colour of a BSS are assigned randomly so two neighbouring access points could be assigned the same number and collisions on the wireless medium might happen. To mitigate this issue, in the case of a collision an access point starts a procedure for changing its BSS colour and then advertises its new colour in beacon frames for clients nearby. This issue can be seen referred to as *colour collision*. [52]

802.11ax also brings with it many smaller changes. 802.11ax adds a slight increase in data rates by introducing a higher QAM modulation scheme moving from 256-QAM presented in 802.11ac to 1024-QAM. 1024-QAM modulation combined with the increase in spatial streams and more efficient use of the wireless medium the theoretical data rate up to 9.6 Gbps from the 6.9 Gbps presented in 802.11ac. [47]

One distinctive change in 802.11ax is the reintroduction of the 2.4 GHz band which was left out from 802.11ac. The incentive for bringing back the 2.4 GHz band lies in the fact that most of the small IoT devices are using the cheaper 2.4 GHz radios. 802.11ax also tries to better accommodate IoT devices by introducing a mechanism to improve battery life in client devices. The *Target wake up time* (TWT) lets devices to negotiate when to power on for sending and receiving data. This will greatly improve battery life in battery powered IoT devices. The device can stay in a power-saving state and only power on for a short period of time it needs to send or receive data. [51]

The new amendment has not yet been finalized and officially released at the time of writing this work. Despite this, some manufacturers have already released some preliminary devices such as smartphones and access points supporting the 802.11ax amendment [44]. It is interesting to see how long it will take until we see these new devices more widely adopted after the amendment has been officially released and more devices come available at the consumer market.

3.8. The new Wi-Fi Alliance 802.11 amendment naming system

In conjunction with the news of the release of the 802.11ax amendment, the Wi-Fi Alliance announced that they would start using a new naming system for the 802.11 standard amendments (Table 1) [17]. As we have discussed the various amendments of the 802.11 standard it is quite understandable that for an average consumer following the current naming system could pose some challenges. It can be difficult to know which amendment is the newest one or which versions are interoperable based on the current more technical-

sounding naming system. Starting from the release of 802.11ax the Wi-Fi Alliance proposes a new naming system based on numbers instead of letters [17]. 802.11ax will be known as Wi-Fi 6, 802.11ac as Wi-Fi 5, 802.11n as Wi-Fi 4 all the way to 802.11b which will be known as Wi-Fi 1. Even though amendment 802.11a was officially released before 802.11b, the reason for 802.11b becoming Wi-Fi 1 could lie in the fact that 802.11b devices got to the market before 802.11a and was, therefore, more widely adopted.

Assigned Number	Amendment	Approved
Wi-Fi 1	802.11b	1999
Wi-Fi 2	802.11a	1999
Wi-Fi 3	802.11g	2003
Wi-Fi 4	802.11n	2009
Wi-Fi 5	802.11ac	2013
Wi-Fi 6	802.11ax	2018

Table 1 The new Wi-Fi Alliance 802.11 amendment naming system

The new system provides a corresponding number for each of the major amendments of the 802.11 standard. This system should make it easier for the average consumers to understand which version of the standard their devices operate with. Every new iteration of the standard provides some improvement on its precursor and it will be beneficial for consumers to distinguish the newer devices from the older ones more easily. Wi-Fi Alliance, of course, wants to this become an industry-wide scheme and advocates for device manufacturer and operating system developers to implement the new system into their user interfaces to visually aid the user for knowing which version of the standard is used [17]. A similar kind of system is already used for the mobile 2G, 3G, 4G, and the upcoming 5G networks.

4. 802.11 WLAN Security

Until this point we have discussed the history of WLAN networking and have gone through the 802.11 standard iterations from its humble beginnings up to the present day. In this chapter, we are taking the discussion on to the main theme of this work, the security of 802.11 WLAN networks. To better understand the founding principles behind wireless communication security, we first provide the reader with the basic knowledge and terms considering the science of *cryptography*. After covering the basic aspects of cryptography, we discuss the issues in WLAN security by presenting the security mechanisms defined in the 802.11 standard and discussing the vulnerabilities found in them. For the scope of this work, we are taking a more practical approach to the subject and try to avoid a thorough mathematical analysis of the presented cryptographic encryption algorithms and security mechanisms. A more in-depth theoretical and mathematical discussion of the subject can be found from [53],[54] and [55].

Security in wireless networking has its own unique challenges compared to its wired counterpart. In wired networking, we are confining the signals and communication into network switches, routers and to the wires that connect them. The nature of wireless communication is quite different since the wireless radio signals are not confined and propagate around us in the air. Furthermore, since wireless access points most often act as gateways into the larger wired network infrastructure, we should be adding extra emphasis on the importance of security in our wireless communication. A wireless network in many cases is more convenient than its wired counterpart, but we are trading out a portion of security for that convenience.

The defining security issue in any wireless communication lies in the fact that the communicated data is propagating all around us in radio waves, meaning that it can be intercepted by anyone listening in the range of our communication. To make the communication secure it must be encrypted so that the possible eavesdropper cannot interpret what is being communicated between the message sender and the receiver. In our case, this means using cryptographic encryption algorithms to ensure the confidentiality and integrity of our wireless communications. There are multiple different cryptographic algorithms that are used in networking for several different operations. The two most notable encryption algorithms that will be discussed in this work are the *Rivest Cipher 4 (RC4)* and the *Advanced Encryption Standard (AES)*.

4.1. The basic principles of cryptography

As said in the previous section, to secure the wireless traffic propagating in the air around us we must take on some protective measures, mostly in forms of different mathematical cryptographic encryption processes. The founding idea behind the science of cryptography is to scramble information in such a way that it cannot be easily interpreted without knowing the secret *encryption key* which in modern cryptographic encryption algorithms come in the form of long and difficult to calculate numbers [9].

The basic function of any general cryptographic process is fairly simple. We have a piece of information known as the *plaintext* which we want to encrypt from everyone else. We take the plaintext and use a cryptographic algorithm, referred to usually as a *cipher*, of our choosing and turn the plaintext into its encrypted format also known as *ciphertext*. The ciphertext can be *decrypted* back into its plaintext form only by those who know the used cipher or have knowledge of the secret encryption key. [56]

This system does pose a problem for us in networking since we first must somehow communicate the secret encryption key between the message sender and recipient that have possibly never met and are long distances apart from each other prior to establishing the secure communication channel. These kinds of cryptographical systems where both parties must know the used cipher or the secret key to decrypt the encrypted message are categorised as *symmetric-key* cryptography or more familiarly as *shared-key* cryptography [56]. To tackle the problems symmetric key cryptography poses in networking, more suiting cryptographic systems known as *asymmetric* or *public-key* cryptography have been developed [9].

4.1.1. Symmetric shared-key cryptography

The earliest primitive symmetric-key cryptography systems have been in use since ancient times and are the oldest standing form of encryption[9]. One of the most known examples of these primitive cryptographic systems is the *Caesar cipher*. With the Caesar cipher, the message is encrypted by moving each letter of the plaintext a set number of times down in the alphabet [56]. The characterising feature of symmetric-key cryptography is the fact that both the encryption and decryption are done by using a single shared secret key, explaining the names of symmetric and shared-key cryptography.

These kinds of symmetric systems might have worked best when both of the parties could rely on each other not to lose or share the secret keys, for example in crude military-type communications. The use of symmetric cryptography becomes a problem when the communication is done on a network between long distances amongst unknown people or stand-alone devices. Because we are using the same secret key to encrypt and decrypt the sent messages, managing the encryption key becomes a problem. We cannot send the key through the same insecure medium as the encrypted message because an eavesdropper could be listening to our communication and use the intercepted secret encryption key to decrypt our message.

Still, it should be stated here that even though symmetric-key algorithms have their downsides, they are still widely implemented in many different cryptographic security systems [56]. They are used, for example, in the encryption algorithms designed for 802.11 networks, which will be disclosed in more detail later in this chapter. The reason for the widespread use of symmetric algorithms is that they are by nature simpler and faster to compute than asymmetric algorithms. By using symmetric algorithms we can save costs in the device manufacturing process and most importantly increase the network speed and bandwidth which are always the priority in networking [56].

4.1.2. Asymmetric public-key cryptography

The basis of the modern cryptographic system we use on our networking today started to take form in the early 1970s when the need for new types of security methods started to arise alongside our first modern computer networks [9]. To tackle the inherent problems in shared-key cryptographic systems the first concepts of asymmetric algorithms were developed in the *British government communication headquarters* (GCHQ) already in 1969 [53]. The GCHQ's involvement in creating the first was kept secret until the 1990s and until then the credit for coming up with the concept of asymmetric cryptography was given to Whitfield Diffie and Martin E. Hellman for their 1976 paper [53], [57]. The premise of asymmetric algorithms is to mitigate the issue of sharing one secret key by using two mathematically linked keys for the encryption and decryption process.

The first key, known as the *private key*, is to be kept as a secret and the second corresponding *public key* can be shared with anyone. The public key can only be used for encrypting messages and only the private key can be used for the decryption process. By

never having to communicate the private key we can safely send messages over an unsecured channel without having to fear someone eavesdropping on our communication since only the owner of the private key can decrypt the sent messages. One of the most widely known and still universally adopted public-key algorithms is the *RSA* algorithm, named after its inventors Ron Rivest, Adi Shamir and Len Adelman who released their work in 1978 [53], [58].

The brilliance of the RSA algorithm is that we can use it in two different ways. We can use it for the basic asymmetric function of using the public key to send encrypted messages to the owner of its corresponding private key. In addition to the basic asymmetric functions, the RSA allows the owner of the private key to also encrypt messages that can then be decrypted with its corresponding public key. By using the algorithm this way we can identify that the message truly was sent by the owner of the private key that matches its public key, allowing authentication over a network [9].

Even though it may seem that asymmetric systems are more secure and should have made the symmetric systems obsolete, this is not the case. As already mentioned in the previous section, symmetric encryption algorithms are still widely used because of their efficiency over the more computationally heavy asymmetric systems [54]. Both symmetric and asymmetric systems have their positive and negative qualities, but there are no factors that would make one superior over the other. After all, the security of any cryptographic system depends on the length of the used key and the used algorithm [54].

4.1.3. Stream and Block ciphers

The symmetric-key algorithms discussed in this work come in the forms of *stream* and *block* ciphers. A stream cipher takes the plaintext input data stream and encrypts it bit by bit. This feature makes it ideal for situations where there are limited amounts of computing power available and time is of the essence [59]. By their nature, stream ciphers fit in the world of wireless communication where the information is transmitted in a stream of radio waves rather than in fixed-size chunks [59]. When using a stream cipher the plaintext message is typically processed through a substitution scheme to create the encrypted ciphertext.

In our case of 802.11 security, the plaintext inputs are often processed by using a Boolean *Exclusive-OR* (XOR) operation presented in Table 2. The XOR operation combines the plaintext information with a pseudorandom bit-stream known as the *keystream* producing the encrypted ciphertext. The XOR process is fairly simple as there are only two possible values, 1 and 0. If the two input values are the same the XOR operation will produce the value 0 and if they are not the same the produced value will be 1. Because the XOR operation is an inverse of itself it means that a shared key can be used to encrypt and decrypt the produced ciphertext making it vulnerable if not implemented correctly. This also means that the same key should never be used twice when encrypting messages [53]. If an attacker can get hold of two messages that use the same encryption key, the messages can be easily decrypted.

Plaintext	1	0	1	0	1	1	0	0
Keystream	1	1	0	0	1	0	0	1
XOR output	0	1	1	0	0	1	0	1

Table 2 XOR process

Contrary to the stream cipher, block cipher encrypts the plaintext messages in fixed-length blocks and generates a block of ciphertext of the same length [53]. For example, a block cipher can be set to use a *block size* of 128 bits, in which case the input plaintext messages will be divided into blocks of 128 bits and the output ciphertext will be the same length of 128 bits. Because the sent message will most often be longer than the fixed block size of the cipher and the bits won't divide evenly into blocks, a padding of redundant bits must be added at the end of the last block to meet the 128-bit block size. Depending on the block ciphers mode of operation, they are usually designed to use a simpler cryptographic function repeatedly on a block [56]. Each of the repeated cycles are simply referred to as *rounds*. Even though the number of rounds can increase the level of security, each iteration will have an effect on the ciphers performance making block ciphers slower to process than a stream cipher [53].

4.2. 802.11 security

The security mechanisms that are defined in the 802.11 standard have evolved a lot since the standard was originally released in 1997. The original legacy standard defined two authentication options, the *Open System Authentication* (OSA) and *Shared Key Authentication* (SKA) [56]. As the cryptographic encryption protocol, the standard presents *Wired Equivalent Privacy* (WEP), which was soon found to be vulnerable due to poor

implementation and errors in its design. Because of these found vulnerabilities new and improved security mechanisms had to be implemented into the standard.

As an intermittent solution to replace the vulnerable WEP protocol the Wi-Fi Alliance took as its task to bring out a solution that could be implemented into the existing hardware with a simple firmware update [56]. The solution was taken from the at the time still unfinished 802.11i amendment and the *Wi-Fi Protected Access (WPA) with Temporal Key Integrity Protocol (TKIP)* was released. In July of 2004, the 802.11i amendment was approved defining the new enhanced security measures meant to replace the vulnerable legacy security options [60]. The 802.11i presents the new security mechanisms under the names *Robust Security Network Associations (RSNA)* and *Robust Security Network (RSN)* [60]. In the officially released 802.11i amendment the legacy security mechanisms have been dubbed as the *pre-RSNA* security mechanisms, but in this work we will be referring to them as legacy security mechanisms.

The 802.11i included the already released WPA-TKIP protocol as well as an improved version of the WPA protocol now named WPA2. WPA2 provides a stronger encryption algorithm as well as improved authentication mechanisms compared to WEP and WPA-TKIP. In addition, the 802.11i includes a stronger enterprise-level authentication framework 802.1x that uses the *Extensible Authentication Protocol (EAP)* authentication protocol to validate the network users [56]. 802.1x was not originally targeted towards wireless networks. It defines a system called *Port-Based Access Control* that was originally designed for 802.3 wired Ethernet networks [2]. The enterprise-level authentication architecture involves an external authentication server and other elements which are out of the scope of this work and will not be therefore discussed in more detail in this work. For a more detailed explanation of the 802.1x architecture, we refer to [9] and [56].

At the time of writing this work, the Wi-Fi Alliance has released a new version of the WPA consequently named as WPA3 [61]. WPA3 is yet again designed to fix some of the vulnerabilities and flaws found in its predecessor during the past 16 years. In this section, we are taking a closer look into the functions and vulnerabilities found in the 802.11 security mechanisms. There are of course other security solutions that have been implemented to enhance the WLAN security such as *Virtual Private Network (VPN)* solutions, SSID cloaking, MAC address filtering or the vulnerable *Wi-Fi Protected Setup (WPS)* developed by the Wi-Fi Alliance. However, these solutions are not defined in the 802.11

standard and for the scope of this work, we will be concentrating our discussion to the aforementioned security mechanisms defined in the 802.11 standard.

4.2.1. Legacy 802.11 security

To ensure secure communication in the WLAN network, the original 1997 802.11 legacy standard's security measures were defined to provide a basic authentication method as well as confidentiality and integrity for the communicated information [9]. As we focus on WLAN networks, we can think of authentication as the methods we use to ensure that only the authorised clients can connect to our wireless access point and join into our network. Confidentiality and integrity could be thought of as the cryptographic algorithms we use to conceal and add integrity values on to the communicated information.

The first defined authentication method in the legacy 802.11, the open system authentication. As the name implies, it is open. OSA does not validate the connecting client's identity and allows any willing client to join the network [9]. As actual confirmation of the connecting client's identity does not occur in OSA, it is sometimes referred to as a *null authentication* algorithm [56]. When the authentication method on an AP is set to OSA, the client wishing to join the network sends an authentication request to the AP that then replies with an authentication response authenticating the client to join the network. After exchanging the authentication frames the AP and client exchange association frames connecting the client into the network. Although WEP is not by default used in the OSA process it can be used to encrypt the communication between the client and AP after the client has successfully been authenticated [56].

Shared Key Authentication (SKA) is probably the most known way of authentication for any WLAN user. The system relies on a known shared key, usually in the form of a password to authenticate the wireless network's users. The key must be shared to users by some out of band mechanism outside the WLAN since the legacy 802.11 standard does not provide a system for distributing the keys [9]. For the SKA to work, a static WEP key must be configured to both the wireless AP and the client.

The SKA uses a four-way frame exchange in the authentication process. The client first sends an authentication request to the AP, and the AP responds by sending a cleartext challenge as an authentication response. The client then uses its WEP key to encrypt the challenge text and sends the encrypted challenge back to the AP. If the AP can decrypt

the challenge by using the static WEP key it will send a final authentication frame for the client to confirm the successful authentication. If the decryption is unsuccessful, the AP will respond negatively and will not allow the client to authenticate. After successful authentication, the static WEP key used for authentication will be used to encrypt the 802.11 data frames communicated between the client and AP. [56]

At first sight, it might seem that the shared key authentication method would be more secure than the open system authentication, but this is not necessarily the case. Even though the shared WEP key is used in the user authentication process there is always the risk that the key gets compromised. If the key is compromised, all the encrypted information on the network can be decrypted with the compromised key. On OSA mode, the AP simply discards those packets that it cannot decrypt [23]. Still, it must be emphasised here that neither of the legacy authentication coupled with WEP encryption should not be used anymore in any situation. Even though still included in the latest 2016 revision of the 802.11 standard, they are deemed obsolete and deprecated [62]. The reason for this is the numerous flaws found in the WEP encryption algorithm. The inner workings and flaws in WEP are discussed in more detail in section 4.2.2.

Before venturing further into the inner workings of the encryption algorithms it should be briefly explained at which point the encryption happens and which part of the 802.11 data frame known as *MAC Protocol Data Unit (MPDU)* is encrypted by the algorithms. The 802.11 standard defines actions that occur on the two lowest layers of the OSI-network model, namely the physical (PHY) and medium access control (MAC) layers. The encryption happens on the second-lowest MAC layer and the information that is being encrypted is the data from the upper (3-7) layers of the OSI-model [2]. To be more precise, the encrypted part of an MPDU that contains the encrypted upper-level information is known as the *Mac Service Data Unit (MSDU)* and is encapsulated as part of the MPDU after the encryption process. A good rule of thumb here is to remember that an MSDU is communicated down the OSI-protocol stack layers and an MPDU is the 802.11 frame that is being transferred between the communicating radios [56].

4.2.2. Wired Equivalent Privacy WEP

Included in the 1997 legacy standard, the Wired Equivalent Privacy WEP encryption protocol was designed to provide basic data encryption and authentication comparable to a wired LAN network [1], [8], [9] and [23]. The encryption protocol is based on the stream

cipher algorithm Rivest Cipher 4 (RC4). Sometimes in literature, the name can also be seen written as *Ron's code* or as *Ron's cipher* which might cause some confusion [36], [53]. The algorithm was developed by Ron Rivest, one of the inventors of the RSA algorithm mentioned in section 4.1.2, Rivest developed the algorithm for the RSA security company in 1987. The algorithm was a trade secret of the RSA company until it was reverse engineered and leaked to the public in 1994 [9], [36], [54] and [63]. As a consequence of the algorithm being leaked and it being a trade secret of the RSA company, it can sometimes be seen referred to as *Alleged RC4* or ARC4 or ARCFOUR adding to the confusion [9], [56].

The WEP encryption protocol (Figure 1) relies on a pre-established and shared set of keys that usually are in the form of plaintext passwords. The keys come in two lengths, either 64 bits or 128 bits long. Although both 64 and 128-bit encryption keys were defined in the legacy standard, only 64-bit encryption was originally available because of a restriction set by the United States government. After the restrictions were lifted manufacturers could produce and export devices with 128-bit encryption implemented. The keys are a combination of two separate parts, a 40 or 104 bit long pre-set key and a 24-bit *Initialization Vector (IV)*. The keys can be set in either hexadecimal (0-9 and A-F) or ASCII characters. Restricting the key length to only 40 or 104 bits leads to very short and predictable passwords. A 40-bit long key can consist of either 10 hexadecimal or 5 ASCII characters. The 104-bit keys can be either 26 hexadecimal or 13 ASCII characters long. [56]

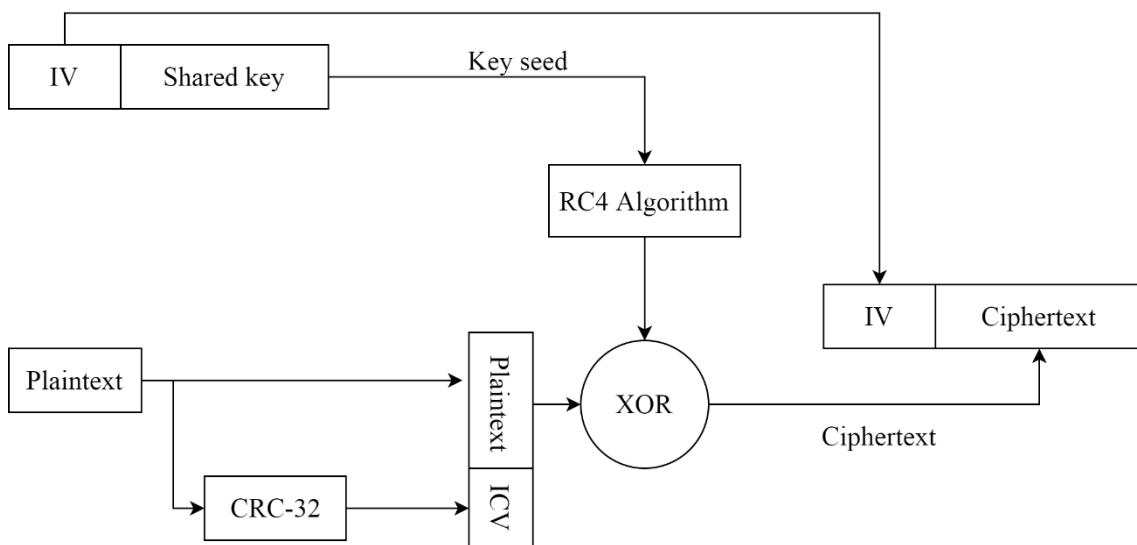


Figure 1 WEP encryption protocol [9]

The WEP encryption process can be broken into the following steps. (1) The first step is to calculate an *Integrity Check Value* (ICV). This is done by running a 32-bit *Cyclic Redundancy Check* (CRC) operation on the plaintext data and the resulting ICV checksum value will then be appended to the end of the plaintext data to provide message integrity. In practice, this means that if the receiving device notices during the decryption process that the ICV has been altered, the packet is not valid and will be discarded. (2) Next, a 24-bit IV is generated and combined with the shared encryption key. This combination known as the *key seed* is then processed by the RC4 algorithm that generates a *keystream*. (3) The keystream is then combined with the plaintext data from step 1 by using the XOR process. (4) The resulting encrypted ciphertext is then prefixed with the generated IV and is ready to be sent forward for transmission. [9]

The goal of the WEP encryption process is to provide security against unauthorized access to the wireless network, provide integrity checks and encryption on each communicated MSDU [9]. The 802.11 standard specifies that WEP is designed for protecting authorized users of the protected WLAN network from casual eavesdroppers [8]. At the time of the legacy standard's release, WEP might have achieved its goal of protecting users from casual eavesdroppers because of the low numbers and higher costs of WLAN devices. The situation changed rapidly when WLAN devices became more affordable and widely available for the average consumers. As the number of wireless devices rose, so did the research dedicated to the security of WEP encryption [1].

The first studies describing the possible vulnerabilities in WEP were made public in the early 2000s by Walker [64] and by Borisov, Goldberg and Wagner [65]. The ground under WEP encryption truly started to cave in after Fluhrer, Martin and Shamir released their work in 2001 [66]. In their work they describe several different vulnerabilities in WEP protocol and implementation of the RC4 algorithm. Two of the presented attacks named "Related-Key Attack Based on the Invariance Weakness" and the "Related-Key Attack Based on Known IV Weakness" would later become known as the FMS attacks [67].

Although Fluhrer et al. [66]. did not demonstrate their attacks in a real-life scenario, soon after Stubblefield, Ioannidis, and Rubin [68] improved on the previous studies and implemented the attacks into real life. They were able to recover the secret key by using their own simulations and off-the-shelf WLAN equipment and software. For their attack to be

successful, they needed to collect 5 to 6 million frames when using the attacks described by Fluhrer et al. and around 1 million frames when using their improved attack [68]. The final nail in the coffin of WEP was struck in 2007 when Tews, Weinmann and Pyshkin [69] presented their attack that only needed between 40,000 and 85,000 captured frames to retrieve a 128-bit WEP key. This meant that the encryption could now be broken in minutes using only a basic laptop computer and an off-the-shelf WLAN interface.

The core issue with WEP does not necessarily lie in the RC4 algorithm itself but more in the way it is used [9]. The core problems are to be found in the WEP protocol itself and could be pinpointed to the already discussed issues with the wireless medium and into how WEP uses the initialization vectors. The IVs have two major built-in issues. The first issue is the fact that they are used as part of the RC4 key seed with the shared static key and are then sent in cleartext as a part of the MPDU frame revealing part of the key. The second issue is the short length of the IVs. The IVs are only 24 bits long meaning that there can only be 16,777,216 different combinations [56]. A busy wireless access point operating only at 11 Mbps would use up all the IVs in about five hours leading into a situation where the key is re-used [9]. The situation is even worse since the 802.11 standard does not define how the IVs should be created and used, leaving it to the manufacturer to decide and further increasing the chance of the key being re-used [9], [68].

The presented issues combined with the short 5 or 13 character manually configured static WEP keys makes the protocol fatally vulnerable to attacks. The weakness in WEP boils down to the fact that it uses a new IV as part of the per-packet encryption key and the number of possible IVs is rather small. This leads into a situation where the key will be re-used and they are coupled with short and predictable static keys, making it possible for an attacker to recover the plaintext static key with ease (Figure 2) [9]. It is this weakness in the IVs that most of the attacks described in the before-mentioned studies leverage upon. The only way to battle against this issue would be to manually change the static key very frequently and even that will not have any effect anymore [9]. For a more detailed description of the attacks on WEP, we refer the reader to the articles presented in this section.

Although it might seem that discussing the long-since broken legacy security mechanisms is a waste of time that is not necessarily the case. According to the global statistics by Wigle.net, even though the amount of WEP encrypted devices has been steadily declining

from its peak (45%) in 2010, still at the time of writing this work, 5.44% of all the 625 million reported devices still use WEP encryption [70]. It is also interesting to notice that according to the same statistics WPA-TKIP that was to replace and ease issues in WEP, never reached similar popularity (11.81% at its highest) and is still less used with a market share of 5.16%. This could be explained by WPA-TKIP's short lifespan that was originally intended to be only five years to give consumers time to migrate to WPA2 devices [1].

Aircrack-ng 1.5.2

[00:00:00] Tested 753 keys (got 64102 IVs)

KB	depth	byte	(vote)																	
0	0/	9	62(85504)	6D(74752)	47(74240)	A1(73472)	27(72704)	41(72704)	3D(72448)	BD(72448)	64(72192)									
1	0/	1	06(93696)	67(78592)	51(76288)	DD(75008)	16(73984)	62(73216)	8F(73216)	60(72960)	F9(72704)									
2	0/	1	64(92928)	75(77568)	AF(77568)	07(73216)	90(72192)	BF(72192)	A9(71680)	63(71424)	83(70912)									
3	3/	3	C7(75264)	65(73472)	B6(72704)	DB(72704)	23(72192)	FA(72192)	17(71936)	63(71936)	A8(71936)									
4	55/	4	EC(67072)	18(66816)	51(66816)	6A(66816)	7A(66816)	91(66816)	95(66816)	CE(66816)	20(66560)									

KEY FOUND! [62:61:64:70:61:73:73:77:6F:72:64:31:32] (ASCII: badpassword12)
Decrypted correctly: 100%

Figure 2 Recovered WEP encryption key

4.3. 802.11i security amendment, WPA-TKIP and WPA2

To replace the faulty security mechanism presented in the legacy standard, the IEEE established a new task group to start the work on enhancing the now broken security. The 802.11i task group was originally part of the 802.11e task group that sought to improve on the QoS as well as security of the standard. Task group *e* was split in half and task group *i* was officially formed in April of 2001 [1]. As a result of the task group's work, the 802.11i amendment was approved in June of 2004 [60]. As already outlined in section 4.2, the 802.11i amendment presents two new encryption methods: WPA-TKIP and WPA2 in both personal and enterprise-grade versions.

The new enhanced security features were branded under the names Robust Security Network (RSN) and Robust Security Network Association (RSNA). RSN is a term that defines the whole wireless network that only allows the use of the new and improved association and authentication measures defined by the RSNA [56]. To put things more simply, a network that only allows the use of WPA-TKIP and WPA2 enhanced authentication and encryption can be called an RSN. The goal of 802.11i was to provide improved

encryption for the 802.11 frames, provide enhanced privacy and integrity as well as new and improved authentication methods [2].

As neither of the mentioned protocols is not as profoundly flawed and vulnerable as WEP, presenting both in greater detail would be an extensive task and out of the scope of this work. For this reason, we are not going to have as in-depth of a discussion about them as with WEP encryption. Instead, we are concentrating on the most notable improvements and changes, as well as on the found vulnerabilities in the WPA protocol. We will also be excluding the enterprise-level authentication from our discussion and focus on *WPA-Personal* authentication that considers small office and home environments. For a more detailed description of the WPA enterprise-level protocols, we refer the reader to [9] and [56].

4.3.1. WPA-TKIP

Before the official approval of the 802.11i amendment, the urgency to alleviate the distress caused by the vulnerability of WEP caused the Wi-Fi Alliance to take initiative. They took a piece of the unfinished amendment and released it to the public. The released new security protocol was called Wi-Fi Protected Access (WPA). As the improved encryption protocol, WPA uses the Temporary Key Integrity Protocol (TKIP) defined in the 802.11i amendment. The Wi-Fi Alliance started its WPA certification program in April of 2003, over a year before the official release of the 802.11i amendment [56], [71]. Like WEP, TKIP is based on the RC4 encryption algorithm [56]. The use of RC4 can be explained by the simple fact that WPA-TKIP was designed as an intermediate solution to fix the issues in WEP without the need for manufacturers to design or consumers to buy new hardware. Instead of consumers needing to buy new WLAN equipment the changes could be implemented on the existing hardware with a simple firmware update.

The improvements made in the TKIP protocol are designed to target the profound weaknesses in the WEP protocol. The most significant improvements being (1) doubling the size of the initialization vector to 48 bits. In TKIP the IV values also referred to as *TKIP Sequence Counter* (TSC) since they are also used to sequence the sent MPDUs [56]. (2) Changing to a stronger data integrity check algorithm known as *Michael*. Michael calculates a value known as *Message Integrity Check* (MIC) by only using simple and fast shift and addition operation [9]. Because of its simplicity, the TKIP MIC process is known to have vulnerabilities and some countermeasures had to be added to mitigate the issues. If

the calculated MIC value fails to match two times in a short period of time during decryption, the connection between the client and access point will be terminated and new temporal keys must be calculated [9]. (3) Finally, the most important changes were done to the key management and creation process. The most significant change is moving from static encryption keys to dynamically created temporal encryption keys. Any two communicating devices create a dynamic encryption key that is then used in the encryption process and is unique for every connection. The dynamic encryption keys are created during the four-way handshake authentication process [56].

At its heart, WPA-TKIP encryption protocol still has the same core elements as WEP but new elements have been introduced into the process to address flaws in the WEP protocol and to add complexity into the per-packet key formation process. The used keys are still essentially 128 bits long to be compatible with WEP, but the key has gone through a two-phase key mixing process to add complexity along with the extended 48-bit IV values[9], [56]. The result of the key mixing process is then fed to the RC4 algorithm and goes through the same XOR process in the same manner as in WEP protocol to encrypt the message. The difference is that the second phase of the key mixing process is done for every packet creating stronger per-packet encryption than in WEP [9].

WPA-TKIP in personal mode still uses pre-shared secret keys for authenticating a wireless client with an access point but does not use the shared key during the data encryption process in the same manner as WEP. In WPA-TKIP there are two types of keys, the *Pair-Wise Master Key* (PMK) which we can think of as the shared password and the *Pair-wise Transient Key* (PTK) which is derived from the PMK [9]. The PTKs are only temporal and are session related, meaning that they are calculated newly every time a client associates with an access point. To calculate the PTKs we need a few elements, the PMK, the client MAC-address, the access point MAC-address and nonce values from the client and access points [56]. The nonce values can be thought of as random numbers that are calculated to be only used once. When the PTK has been calculated the per-packet keys used for the encryption are derived from the PTK.

Before venturing further into the key creation process, it should be noted that there is a difference between the Shared Key Authentication defined in the legacy standard and the *Pre-Shared Key* (PSK) or as it sometimes referred WPA-Personal authentication scheme

defined in the 802.11i amendment. Although in both systems the same secret key or password must be shared among the network users, the way the keys are used is completely different. To put the difference in more simple terms, the difference comes from the fact that in the shared key method the shared key is used as part of the data encryption process with the initialization vector, whereas in the PSK system the shared key is used to dynamically derive temporal encryption keys that are then used in the data encryption process [56].

The number of different keys that are used and created during the WPA PSK authentication and encryption processes can understandably cause some confusion. To ease some of the confusion we should emphasize that technically the PMK is not the same as the set pre-shared password used to authenticate to a wireless network. In technical terms, the PMK is a combination many elements, the shared password between 8 and 63 ASCII characters and the network *Service Set Identifier* (SSID) that is then run through a hash function to produce a 256-bit PMK [56]. Adding into the confusion sometimes the terms PSK and PMK are being used interchangeably because in the PSK system they are practically the same thing. To avoid confusion, we are henceforth using PMK when addressing the computed pre-shared key, PSK when speaking about the authentication system, and password when talking about the manually set and shared plaintext password.

As one of the biggest weaknesses in WEP was the decision to use static encryption keys, WPA sought out to fix the issue by transitioning to dynamically created encryption keys [9]. The elements needed to derive the dynamic PTKs discussed previously are obtained through a process called the four-way handshake (Figure 4). Before the handshake process between a client and an access point can be started both must know the PMK. As the name tells us the four-way handshake consist of four messages sent between the access point and client. The process is handled by exchanging four *Extended Authentication Protocol over LAN* (EAPOL) key-frames defined in the 802.1x amendment [56].

1. The first message is sent from the access point to the client containing an *Anonce* value. After receiving the message, the client has all the elements to derive the PTK. Recall that the PTK is derived from the PMK, client and access point MAC-addresses and the nonce values exchanged between the client and access point.
2. The second message (Figure 3) contains the client's nonce value titled the *Snonce* and the MIC value to ensure message integrity. After receiving the message, the

access point has all the elements needed to derive the PTK. Now both parties have the PTK to use for data encryption. It is at this point that the authentication process will fail if the shared password set on the client and the access point do not match and the PTKs will not be established properly.

3. The third message contains yet again another key calculated by the access point. The *Group Temporal Key* (GTK) which is used for encrypting broadcast traffic that needs to be sent to all client on the network. The GTK is encrypted using the newly created PTK and sent to the client along with a MIC value.
4. After receiving the GTK, the client sends the fourth and final message to the access point confirming that the PTKs have been set and the traffic henceforth can be encrypted.

```

Key Descriptor Type: EAPOL RSN Key (2)
[Message number: 2]
Key Information: 0x010a
Key Length: 0
Replay Counter: 5
WPA Key Nonce: 2a845ed4f22f9b725112d30cb9082cf7812a4dbdb17da615...
Key IV: 00000000000000000000000000000000
WPA Key RSC: 0000000000000000
WPA Key ID: 0000000000000000
WPA Key MIC: 103f6a0fd984195ada5e4bb9632f4388
WPA Key Data Length: 24
WPA Key Data: 30160100000fac040100000fac040100000fac023c000000

```

Figure 3 Second EAPOL message

The TKIP process sought to fix some of the holes poked into the WEP protocol while remaining backwards compatible with the already existing legacy hardware. The improvements did achieve most of its goals on plugging the holes found in WEP and its planned five-year lifespan without major breaks or vulnerabilities. This is mostly due to the introduction of dynamic encryption keys, improved per-packet encryption combined with the doubled IV size and the improved four-way handshake authentication method. Still, TKIP has its share of vulnerabilities although not as severe as the ones in WEP. Most of the vulnerabilities in TKIP utilize the Michael MIC protocols weaknesses and the four-way handshake process. For these found vulnerabilities the current 2016 802.11 standard states that *“The use of TKIP is deprecated. The TKIP algorithm is unsuitable for the purposes of this standard”* [62] and should not, therefore, be used anymore. We will be discussing the vulnerabilities in both WPA-TKIP and WPA2 further in this chapter.

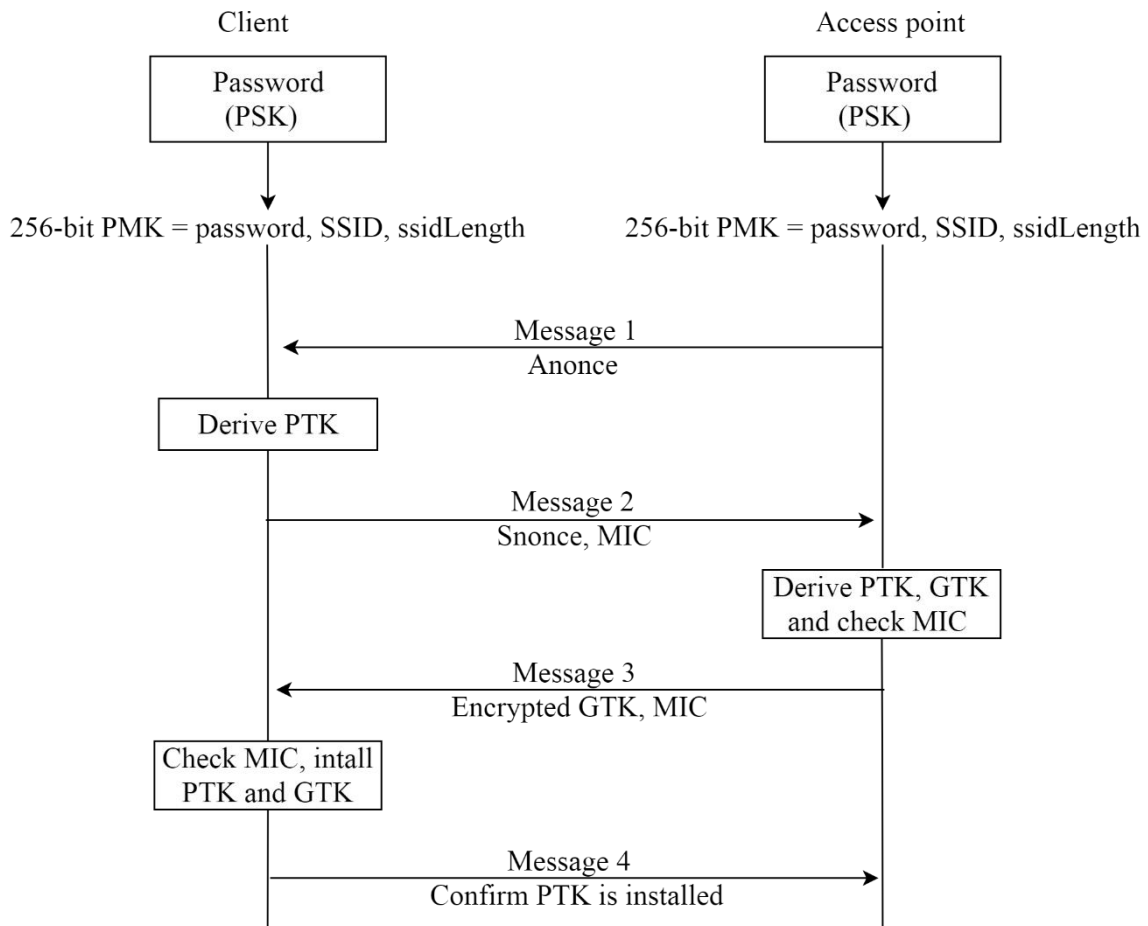


Figure 4 The four-way handshake process [9]

4.3.2. WPA2 CCMP/AES

WPA-TKIP was only a portion of the 802.11i amendment brought in early by the Wi-Fi Alliance to ease the issues with WEP encryption. After the 802.11i amendment was fully approved, the second and much-improved encryption protocol designed to replace both WEP and TKIP was released under the name Wi-Fi Protected Access 2 or WPA2. As both WPA and WPA2 solutions are part of the 802.11i amendment they are in many ways the same [9]. WPA2 uses the same authentication and key establishment processes as WPA-TKIP, the difference being that WPA2 uses the same encryption key for encryption and message integrity protection [9]. The most significant changes from the scope of our work are, of course, the change of encryption algorithm from RC4 to the Advanced Encryption Standard (AES) and the change of message integrity algorithm away from Michael. As most of the changes from WEP to WPA are already discussed in section 4.3.1, we will be concentrating on briefly explaining the changes in the encryption and message integrity methods.

The AES encryption standard was established by the United States *National Institute of Standard and Technology* (NIST) in 2001 after its public search for a replacement to its predecessor *Data Encryption Standard* (DES) [56]. The AES standard is based on the *Rijndael cipher* developed by Belgian Cryptographers Vincent Rijmen and Joan Daemen who submitted their work to NIST in 1999 during the open search process. Therefore, sometimes AES and Rijndael can be seen used synonymously as will be done in this work for the sake of simplicity. The Rijndael algorithm is a block cipher that encrypts data in 128-bit blocks and uses either 128, 192 or 256-bit keys. The algorithm runs either 10, 12 or 14 of computational rounds depending on the key length [56]. For a more detailed mathematical description of the Rijndael algorithm, we refer the reader to [53], [54], [55] and [72].

As the improved and stronger encryption protocol, the 802.11i amendment presents *Counter Mode with Cipher-Block Chaining Message Authentication Code Protocol* or as it is sometimes shortened *Counter Mode with CBC-MAC Protocol* (CCMP) which is based upon the AES block cipher algorithm (Figure 5). The two words are sometimes used in tandem as CCMP/AES but often separate, which can cause some confusion.

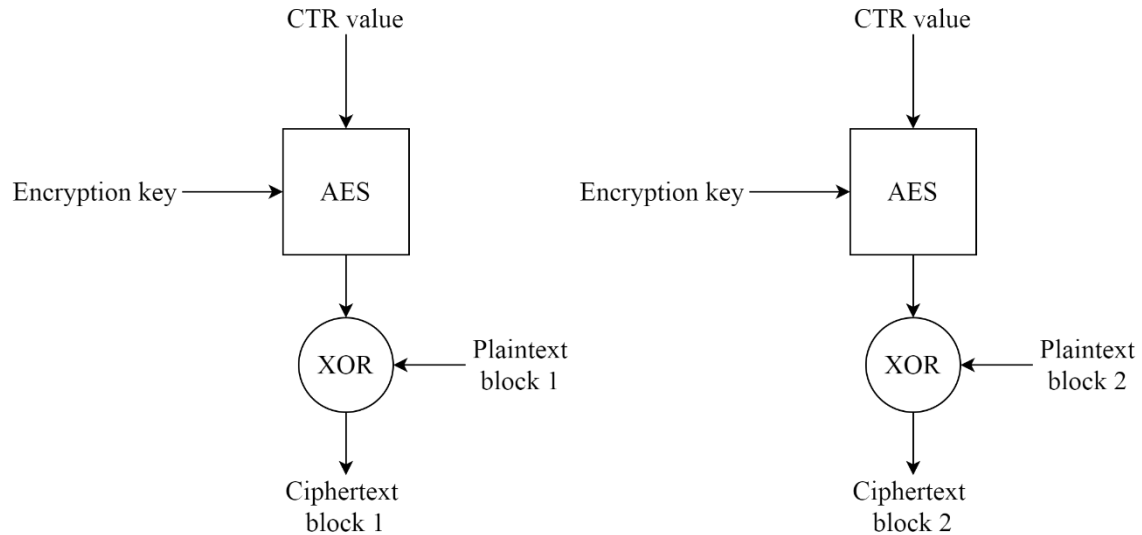


Figure 5 AES in counter mode [9]

The name Counter Mode with Cipher Block Chaining Message Authentication Code Protocol contains a couple of different operations. Counter mode is used to provide data confidentiality and is sometimes represented as CTR. Because we are using AES in a counter mode it makes it possible for us to use it as a stream cipher although at heart it is still a block cipher [9]. The initial CTR counter value is derived from a nonce value that then

changes for each consecutive message block. The encryption key and the counter value are then fed to the AES algorithm to produce a keystream that is then XORed with a 128-bit block of the original message [9]. The system is secure as long as the counter value is never repeated with the encryption key. In WPA2 this is achieved by using dynamically created PTKs for every new session as in TKIP [56].

Cipher-Block Chaining Message Authentication Code CBC-MAC is used for authentication and integrity and can be thought as kind of an extension to the counter mode process. The CBC-MAC process (Figure 6) is initially the same as the CTR process but it XORs a plaintext block with the previous blocks resulting ciphertext before encrypting it. This process means that any change to a ciphertext block changes the decrypted output of the last block thus changing the remainder MAC value. As block ciphers can have different modes of operation, we can think of CCMP as a combination of two different modes of operation which allow us to use AES as a stream cipher. [9]

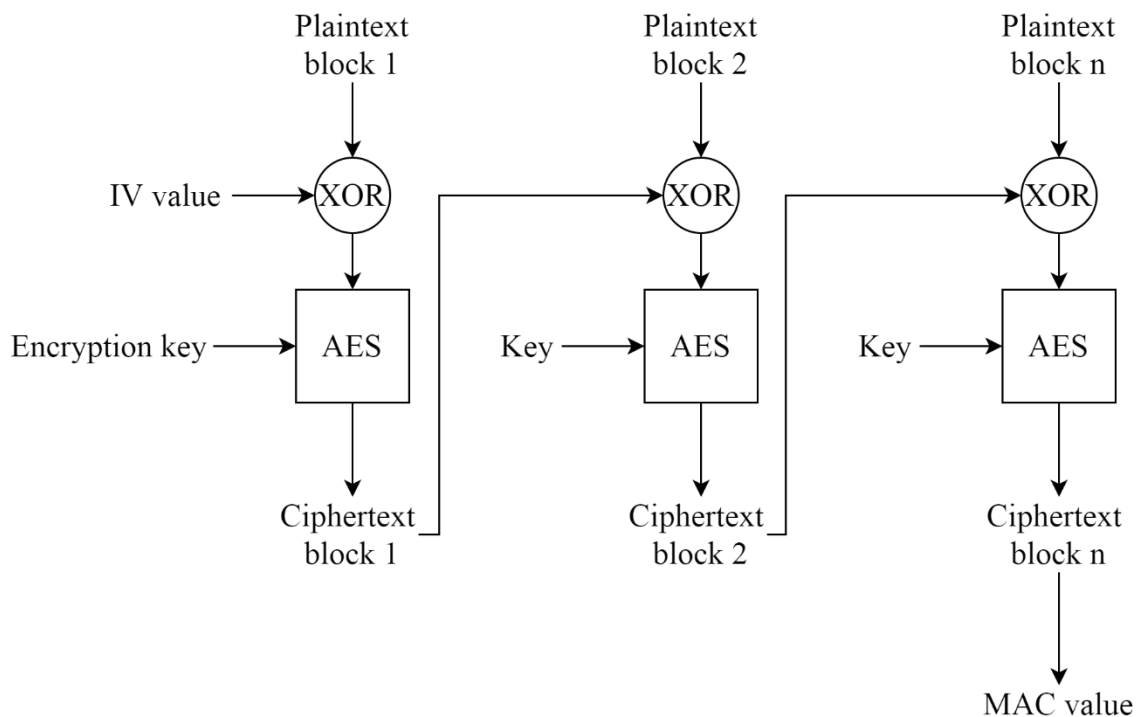


Figure 6 AES CBC-MAC process [9]

The CCMP process allows us to use one encryption key for the actual data encryption and decryption processes as well as to provide authentication and integrity for the communicated information. As previously implied, the CCMP encryption process includes several elements.

(1) To start the encryption process we need to have derived the 128-bit PTKs during the four-way handshake process. (2) CCMP uses a 48-bit *Packet Number* (PN) for sequencing the 802.11 frames. The PN identifies each transmitted frame and is incremented with each consecutive transmission. (3) The third needed element is the already mentioned nonce value. The 104-bit value is derived from the PN, the sending devices MAC address and data used for varied QoS purposes. The nonce value together with the PN could be thought as the IV value for the CCMP process and should not be confused with the nonce values that are created during the four-way handshake process. (4) The last needed element along with the plaintext data is the *Additional Authentication Data* (AAD) which consist of portions of the MPDU header information such as MAC addresses, sequence control value and QoS values. ADD is used for data integrity at the receiving end as well as seeding material for the CCMP encryption. After all the needed values have been derived, they are fed to the CCMP module which uses the AES Rijndael cipher to encrypt the plaintext data in 128-bit blocks, along with the message authentication code value. [9], [56]

When comparing WPA2 to its predecessors it seems to be more robust and not as vulnerable to attacks. Unlike WEP and TKIP, it has not been deprecated from the 802.11 standard and is at the time the most commonly used encryption protocol of the three [70]. Disregarding its stronger encryption algorithms WPA2 still has its flaws and it has never been a silver-bullet solution for all the vulnerabilities in WLAN networks, despite being more robust than WEP or TKIP. The WPA2 protocol has now been in use for twenty years and has been through a very comprehensive amount of study during that time and is at the time of writing this work being superseded by a newer generation encryption, WPA3. In the following section we present vulnerabilities found in both WPA-TKIP and WPA2. After discussing the vulnerabilities of the original WPA protocols we will be further discussing the upcoming WPA3 protocol in section 4.4.

4.3.3. WPA-TKIP vulnerabilities

As already mentioned in the previous sections, WPA-TKIP and WPA2 protocols have their share of vulnerabilities, although maybe not as severe as the ones found in WEP. Most of the vulnerabilities found in WPA-TKIP are rather impractical and mostly consider the MIC protocol Michael. The first practical attack against TKIP MIC was presented by Beck and Tews in 2009 [73] and a year later Tews improved on the attack [74]. The

attack Beck and Tews describe makes it possible for an attacker to recover plaintext from an encrypted frame without knowing the encryption key, recover the MIC value and inject forged frames into the network.

After the release of the Beck and Tews attack, it has been refined on multiple occasions. Ohigashi and Morii propose a Man-In-the-Middle type version of the attack shortening the attack time and stripping away some of the requirements of the original attack [75]. Ohigashi and Morii too would eventually provide a new and improved version of their attack and released their work in 2012 [76]. More recent improvements on the attacks against TKIP MIC have been released by Vanhoef and Piessens in 2013 [77] and in 2019 by Schepes, Ranganathan and Vanhoef [78]. Even though the Beck and Tews attack has been refined many times the attack is still somewhat arbitrary and not very practical to implement because of the many variables that must align for the attack to be successful, although the latest attacks presented by Schepes, Ranganathan and Vanhoef [78] take away some of those variables. To put things into perspective, the presented attacks against WPA-TKIP do not allow the malicious actor to retrieve the networks pre-shared key and are not in any way as practical to implement and execute as the attacks against the WEP protocol.

4.3.4. WPA password cracking and WPA2 vulnerabilities

The previously discussed vulnerabilities found in WPA-TKIP Michael MIC process are not applicable for WPA2 since WPA2 uses the CBC-MAC to provide encryption as well as integrity for the communicated frames [78]. Since the CCMP/AES encryption algorithm is much more robust and secure when compared to RC4 used in TKIP and WEP. Many of the most significant vulnerabilities considering the WPA2 protocol are to be found in the four-way handshake process and weak pre-set passwords. The more common attacks used against WPA2 and WPA-TKIP respectively are based on either brute-forcing the pre-shared password or trying to calculate it based on the information we can derive from the four-way handshake process. The weaknesses and possible attacks against the WPA four-way handshake were made public by Robert Moskowitz already in 2003 before the 802.11i amendment official released in 2004 [79]. Simply brute-forcing the pre-shared password is not by any means very practical. In a brute-forcing attack, we would be exhausting every single possible variation of the 8 to 63-character password and the

process would take endless amounts of time and computational power to complete even if the password is weak.

A much more efficient way of cracking a WPA2 password is to use a pre-made large dictionary of passwords. The dictionary attack (Figure 7) process is fairly simple and we know based on our discussion in section 4.3.1 that the password is used to generate the 256-bit PMK by combining the network SSID and by running the combination through a hash function. The PMK is then used to derive the PTK used for the data encryption which is then used for calculating a MIC value for the last two frames exchanged during the handshake process.

For the attack to be successful, an adversary needs to capture the EAPOL handshake packets exchanged between a client and an access point [80]. This can either be done by simply waiting for someone to connect to the network or by forcing already connected clients to disconnect and reconnect to the network (see section 4.3.5). If the adversary can capture the handshake process between a client and access point, he can compare the PTK and MIC values to the ones calculated based on the passwords in our dictionary [80]. If the adversary can find matching PTK and MIC values, he can confirm that a certain word in our list is the plaintext password used to derive the PMK [80].

The problem with dictionary attacks is that they are not very efficient. Efficiently calculating the key values and comparing them with the captured values requires plenty of processing power. To speed up the password cracking process, it is useful to calculate the needed key values beforehand. Since we know that the wireless network's PMK is derived from the password and network SSID, we can create a list of the possible PMK values. These lists are known as "Rainbow Tables" [80].

Calculating the values before starting the cracking process will save us time and resources making the process more efficient. Many of these dictionaries and rainbow tables can also be found freely online. Many of these lists are made up from the results of large password leaks and as the rainbow table creation process has become crowdsourced in security and hacker communities, they have become quite extensive and effective. One example of a freely available dictionary and rainbow table is the one released by "The Church of Wi-Fi" group available at [81]. The dictionary and rainbow table is comprised by combining over 1 million common passwords and a thousand of the most common network SSIDs [81], [82].

Aircrack-ng 1.5.2

[00:00:23] 55096/7120712 keys tested (2036.92 k/s)

Time left: 57 minutes, 50 seconds

0.77%

KEY FOUND! [password1234]

```
Master Key      : 3D B6 83 80 08 DC 1A FB 32 FE AB DB 80 66 D3 63
                  07 34 F7 4B 03 08 23 F7 F6 AC 6D B3 8B 2B 7B 93

Transient Key   : A3 37 AA ED 87 32 DB E7 EB EF E9 FF C6 2A 66 48
                  F6 34 02 9F E4 7D 83 43 2E 1D 5C 87 F5 6C 8A A5
                  2A E7 E0 19 43 9D DE 01 DD 31 EA 32 A9 9A FE 8E
                  CB F4 8F 20 D0 B7 3D D3 9A 2E 94 18 D4 7C B1 93

EAPOL HMAC      : 70 CE 0F 03 55 D6 7A 06 B6 AD EB 58 4B F6 95 5A
```

Figure 7 Successful WPA dictionary attack

Even when using premade password dictionary or rainbow table the cracking processes are still computationally heavy and require a lot of processing power to be time efficient. To access the needed computational power, password cracking software have started supporting multicore *Central Processing Units* (CPU) as well as utilising off-the-shelf *Graphical Processing Units* (GPU). By utilising CPUs with multiple cores and the processing power of modern GPUs it is possible to go through tens or even hundreds of thousands of passwords per second even with an average computer. [80]

The solutions for password cracking come in many different forms varying from commercial and open-source software to web sites and cloud computing-based solutions. One of the most known commercial software is the Russian software company ElcomSoft's Wireless Security Auditor EWSA [83], which makes it possible to utilize both the CPU and GPU in the cracking process. Some of the most notable open source software used in password cracking are Pyrit, coWPAtty, Hashcat and Aircrack [84]–[87].

The basic WPA password cracking process was improved in 2018 when one of the creators of Hashcat posted his findings on the Hashcat forum [88]. What he found out was that it is possible to derive the PMK by capturing only one EAPOL frame sent by the access point and attacking a PMKID hash value found in the EAPOL frame. The attack eliminated the need for capturing a complete four-way handshake and could be done in a network without any connected users. The limitation of the attack was that the PMKID value is something added by the device manufacturers and is not implemented by every

manufacturer making the attack unpractical to carry out successfully in some real-life scenarios.

To get the maximum amount of computing power available, hackers have started using powerful rental cloud computers for their password cracking endeavours. One of the first cases where such rental cloud computers were used for WPA password cracking was reported in 2011 by security expert Thomas Roth who presented his findings during the Black Hat security conference [89]. Roth utilized Amazon's rentable cloud-based computers that contained multiple multi-core CPUs and multiple GPUs. In his experiment, Roth used the already mentioned cracking software Pyrit along with a 39 million word dictionary. Roth was able to try a little under 400,000 passwords per second when distributing the workload to eight rented computers at the same time, leaving the workload of a single computer to between 45,000 and 50,000 passwords per second [89]. The numbers Roth presents in his work must have only gone up because of the increased computing power over the past nine years. Unfortunately, we were not able to find more recent numbers to present for this work.

Even though cracking WPA passwords might seem easy and convenient with the right tools and high amounts of computing power, it is still a game of guessing the right password. The simplest and best way to mitigate the presented password attacks is to use strong passwords. It is even stated in the 802.11 standard that "*Keys derived from the pass-phrase provide relatively low levels of security, especially with keys generated from short passwords, since they are subject to dictionary attack. Use of the key hash is recommended only where it is impractical to make use of a stronger form of user authentication. A key generated from a passphrase of less than about 20 characters is unlikely to deter attacks.*"[62]. The lengthier and complex the password is, the less likely it is for an adversary to be able to guess or to calculate it even with high amounts of computing power.

The more recent and novel attack against the WPA2 handshake process dubbed as KRACK (*Key Reinstallation Attack*) was found in 2017 by Vanhoef and Piessens [90], improving on their work a year later [91]. Related to the work of Vanhoef and Piessens, the latest vulnerability on WPA dubbed as Kr00k was found and presented by researchers of the Slovakian antivirus company ESET in February 2020. The KRACK vulnerability found by Vanhoef and Piessens leverages a vulnerability in the implementation of the

four-way handshake allowing a malicious actor to manipulate and replay the EAPOL handshake messages making it possible to force the victim wireless device into reinstalling an already used encryption key [90]. As we recall from section 4.3.1 when a wireless client joins a network the four-way handshake process negotiates a new PTK key for the session. The client installs the key after it receives the third message from the access point and once it is installed it will be used for encrypting data during that session.

The catch of the attack lies in the fact that it is very plausible to lose frames on the wireless medium. If the third frame of the four-way handshake is lost, an access point will retransmit it until it receives an answer from the client. This means that it is possible to send the third frame multiple times to a wireless client and force it to reinstall the already used PTK key and reset the packet number nonce and replay counter values [90]. This makes it possible to decrypt and forge traffic between a client and access point without knowing the encryption key. The attack was at the time most devastating for devices using Linux based operating systems, which instead of reinstalling the already used key, installed a new key comprised only of zeros [90]. Even though the attack is distressing, even after improving on their work the attack is not very practical, it does not affect Windows or IOS operating systems, and it has quite many other variables to meet to be successful.

The Kr00k vulnerability could be considered more of a bug in how the vulnerable WLAN chips operate than an actual vulnerability in the WPA protocol. The researchers found out that after a client disassociates itself from a network the PTK between the client and access point is set to all zeros [92]. This is normal since, as we know the PTKs are renegotiated for every connection, but there is a fault in this system. After a client disassociates and the PTK has been set to all zeros, there may still be some unencrypted frames left in the device's memory, waiting to be encrypted and transmitted.

These leftover frames, possibly containing sensitive information, are encrypted with the zero PTK before transmission. A malicious actor could exploit this bug by forcefully disassociating clients repeatedly and collect frames with weak encryption. The vulnerability only affects devices that use WLAN chips manufactured by Broadcom and Cypress. According to the research, many well-known device manufacturers such as Google, Samsung, Apple, Asus and Huawei use chips by the mentioned manufacturers [92]. The good thing here is that, because we are talking about something that is more of a bug in the chip's operation the issue can be solved with a software update.

4.3.5. 802.11 Denial of Service Attacks

Some of the more practical attacks presented against the WPA protocol are Denial of Service (DoS) attacks. Many of these attacks are applicable against WPA-TKIP, WPA2 and WEP, respectively [56]. DoS attacks can be easily executed, for example by sending disassociation or deauthentication frames to the wireless network. When sent to the network, deauthentication frames terminate the connection between the access point and connected clients, forcing clients to renegotiate the authentication with the access point [56]. After a client's connection has been terminated, the negotiation between the client and access point can be distracted indefinitely, keeping the client from connecting back to the network. The use of deauthentication frames in DOS attacks is very practical because they are sent unencrypted making it possible to easily forge them [56]. Another way to execute a similar DoS attack is to simply exhaust the access point's processing power by sending bursts of probe request frames or false authentication requests [56].

Although sending forged frames to the network is a convenient way of causing DoS attacks, it should be noted that newer 802.11ac and 802.11ax devices are capable of encrypting and authenticating deauthentication frames. These fixes were included in the 802.11 standard after 2009 when the 802.11w amendment was approved, alleviating some of the DoS issues [2], [56]. There are, of course, ways to cause interference in wireless networks by jamming the radio frequencies WLAN networks operate in and execute a DoS attack that way. As these radio frequency jamming attacks can be either intentional or unintentional and are to be blamed on the nature of wireless communication rather than a fault in the 802.11 standard, they are excluded from this conversation.

Although the attacks and vulnerabilities discussed in the previous sections might seem intimidating, their effects can be easily mitigated and possibly prevented altogether. The simplest ways are to keep the software in your wireless devices updated and to always use the strongest encryption possible. Today this usually means using WPA2 along with a strong over 20-character password. If your device does not support WPA2 and there is no way of upgrading to a newer device, then WPA-TKIP should be used instead of WEP.

Another simple improvement is to change the networks default SSID to something random, as many of the pre-made rainbow tables use a combination of the most common passwords and factory default SSIDs. When using online services it is always good to check that the connection is HTTPS encrypted or to invest in a VPN software for an extra

layer of encryption. Using a VPN and making sure that used online services are HTTPS encrypted is even more important when using an unencrypted WLAN network or a public network where a single password is shared among many users. This is most often the case when using a network provided by hotels or cafes. In a scenario where everyone shares a single password, it should be remembered that anyone who knows the shared password and has captured the network user's four-way handshakes can record the wireless traffic and decrypt it with the known shared key coupled with the captured handshakes.

4.4. WPA3

Alongside with the news about the coming release of the latest 802.11ax amendment, it was announced by the Wi-Fi Alliance that the next generation of WPA encryption dubbed WPA3 would also be released [61]. As WPA3 is the next generation of WPA, it is in many ways similar to the preceding its protocols. For the average user, the changes will not be in any way visible or affect the user experience. There will still be both personal and enterprise modes, CCMP/AES is still used for data encryption and client authentication is still done by sharing one single password.

As the most profound vulnerabilities in WPA2 were found in the four-way handshake process, the fundamental changes in WPA3 have been aimed at improving the handshake process. In addition, WPA3 patches the KRACK vulnerability and adds an extra layer of security for open networks. WPA3 also aims to fix the issues caused by DoS attacks by forcing the use of encryption in the various management frames. Similarly, to the previous sections, we will be excluding the enterprise mode of WPA3 from our discussions and focus on the enhancements made to the personal mode of WPA in this section.

To prevent the various dictionary password attacks against the WPA four-way handshake discussed in section 4.3.4, WPA3 introduces *Simultaneous Authentication of Equals* (SAE) handshake presented by Dan Harkins in 2008 [93]. The name Simultaneous Authentication of Equals refers to the fact that either involved party can initiate the SAE handshake, although in our case of classic WLAN communication the client will be the initiating party. The different variations of the SAE handshake are based on the *Dragonfly* key exchange protocol and for this reason, SAE is sometimes seen referred to as the *Dragonfly Handshake* [94]. The SAE handshake has been a part of the 802.11 standard since 2011 as part of the 802.11s amendment [95]. The 802.11s amendment defines the MAC layer and security functions for WLAN mesh networks.

The SAE handshake process could be thought as an addition on top of the WPA four-way handshake process, replacing the system where the PMK is derived from the pre-shared password. In WPA3, the SAE handshake is used to process the shared password to derive a PMK with high cryptographic entropy. The PMK will be then used during the four-way handshake process to derive the PTK used for data encryption. The higher cryptographic entropy is achieved by utilising a form of public-key cryptography known as *Elliptic Curve Cryptography* (ECC) during the SAE handshake process. [94]

When using elliptic curves, the operations are performed on x and y axis of a graph, where $x < p$ and $y < p$ with p being a prime number and the equation $y^2 = x^3 + ax + b \pmod p$ must hold. The strong security in this system thrives from a mathematical problem known as the *Discrete Logarithm Problem* (DLP), where it is extremely hard to find the value of x in the equation $Y = G^x \pmod q$ even if given the values of Y , G and q [96]. The larger the given values the harder it gets to compute the value of x . Beneath the mathematical equations, the important thing to take from the DLP would be the following; the result of the equation is easy to calculate, but reverting from the result to the values used in the equation is the computationally hard task. Explaining the full mathematical processes considering ECC in more detail would be a lengthy task and out of the scope of this work so we refer the reader to [53]–[55] for the detailed mathematic description of elliptic curves and ECC.

4.4.1. WPA3 SAE handshake

Before initiating the SAE handshake (Figure 8) the shared password is converted into a value known as the *password element* by using a hash-to-curve algorithm by both the client and the access point. The algorithm hashes the password together with a counter value along with the MAC addresses of both the client and access point [94]. This hash is then used as the x coordinate on the elliptic curve. It then finds the solution for y over the equation $y^2 = x^3 + ax + b \pmod p$ where p is a prime and the values a , b , and p depend on the used elliptic curve [94]. If a solution exists, the solution then becomes the password element value. If there is no solution the counter value is incremented and a new attempt to find the value of y over the new x value is made. To increase the security of the function the process is executed several times even if the solution for y is found. The extra rounds are based on a randomly generated password instead of the real one. According to

Vanhoef and Ronen [94], some older implementations executed the process only 4 times and newer versions execute the process 40 times.

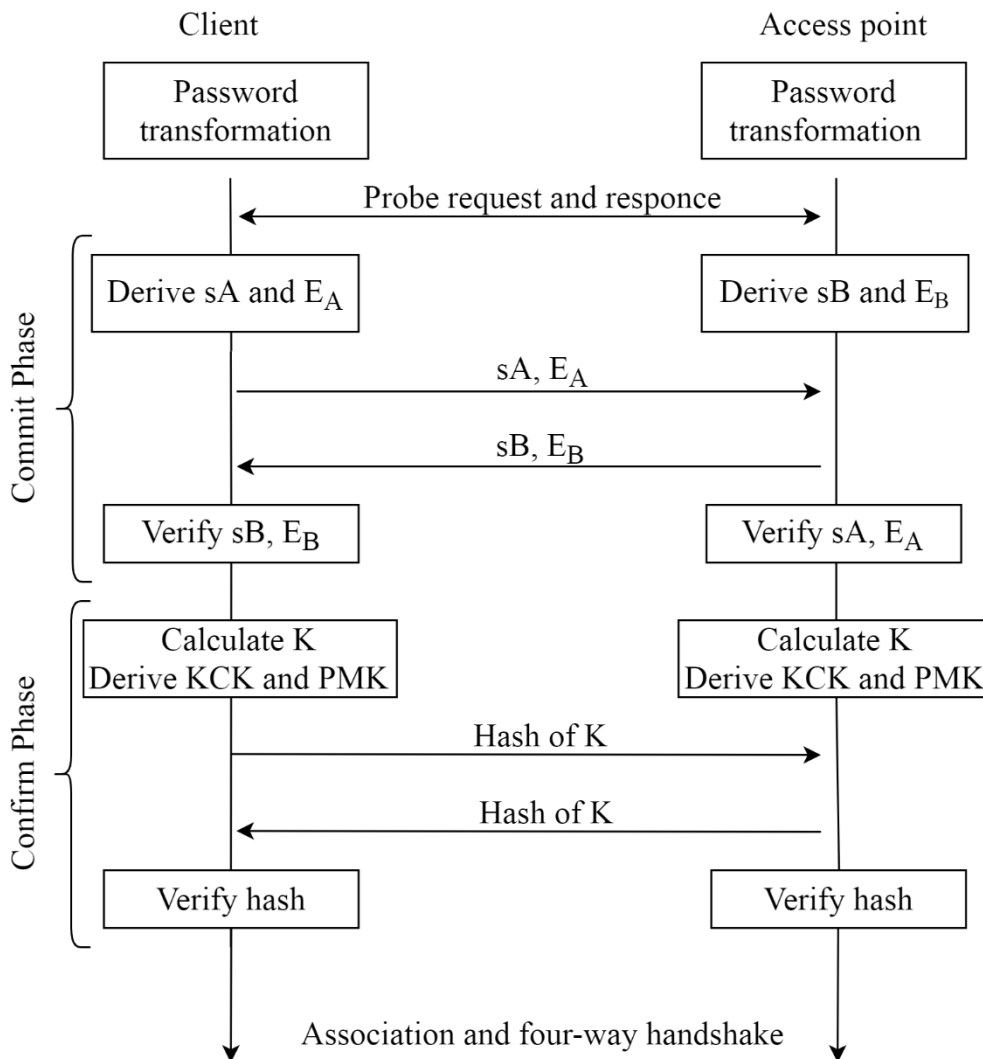


Figure 8 SAE handshake [94]

The SAE handshake itself has two phases; the *commit phase* followed by the *confirm phase*. The idea of the commitment phase is to have the communicating peers to exchange a single guess of the pre-shared password. The following confirmation phase is used to prove that the guessed password is correct [97]. In the commit phase both the client and the access point pick two random values r_a, r_A and r_b, r_B and then calculate the *scalar* value $s_A = (r_a + r_A) \bmod q$ where q is a large prime number. Next, we take the password element derived from the password as previously explained and raise it minus to the power of r_A turning it into *group element* $E_A = s_A^{r_A}$. The client and access point will then exchange the calculated scalar values s_A and s_B and group element values. After receiving the values both parties verify the values that should match if both used the same pre-

shared password. If the validation of the values fails, the handshake is aborted [94]. The point here is that even if an eavesdropper is capable of capturing the scalar value and group element, computing the two random values ra and rA is unfeasible even while having the knowledge of sA and the group element because of the already mentioned DLP [94].

In the confirm phase both client and access point calculate a shared secret value K using their own scalar value, the received shared scalar value and the password element $K = rA * (rB * E_B)$. From the computed shared secret value K , two 256-bit keys are derived, the *Key Confirmation Key* (KCK) and the PMK [96]. Both client and access point then use the KCK as part of a hash function calculated over the secret value K [94]. This hashed value is then exchanged between the peers and verified. If the verification is successful, the PMK derived before will then be used during the four-way handshake process to derive the PTK [94]. For a more detailed mathematical description of the ECC process in SAE and the SAE handshake, we refer the reader to [62] and [93]–[95].

The idea behind the SAE handshake is to provide the same pre-shared password-based authentication as before in WPA but with much-increased cryptographic protection by never communicating parts of the password or the PMK [97]. The SAE handshake process makes it practically unfeasible for a malicious actor to derive the password or the PMK from a captured SAE handshake. Even if a malicious actor was able to record the handshake and compute the used password in a few days' time, it cannot be used for decrypting traffic recorded in the past or future. This is because the SAE handshake is renewed for every connection, making it resistant to offline dictionary attacks, providing forward secrecy for the password [94]. Moreover, because the commit - confirm process forces a client to guess and confirm the password for every connection attempt, the number of guesses can be limited making active dictionary attacks unfeasible [94].

4.4.2. Opportunistic Wireless Encryption OWE

To alleviate the inherent encryption issues with networks using the open system authentication (section 4.2.1), WPA3 introduces *Opportunistic Wireless Encryption* (OWE). OWE has been defined in the Internet Engineering Task Force (IETF) RFC8110 specification [98]. OWE provides encryption in open networks by generating an individual encryption key for every connected client. The OWE process is based on the Diffie-Hellman key exchange (Section 4.1.2) and the same DLP problem as SAE. OWE and SAE have

the same goal of generating individual PMK to be used in the four-way handshake process to compute the encryption key.

The benefit of the OWE process (Figure 9) is that even in an open network every user has an individual encryption key, deterring eavesdroppers from simply recording and reading the traffic of an open unencrypted network. This also means that in a situation where a large group of users share the same password, an eavesdropper cannot decrypt the wireless traffic by using the known shared password and captured four-way handshakes. However, it should be stated that OWE does not provide authentication for the identities of the communicating devices. This leaves the system open for active attack where a malicious actor impersonates a legitimate access point and tricks a client to connect to it instead of the legitimate access point.

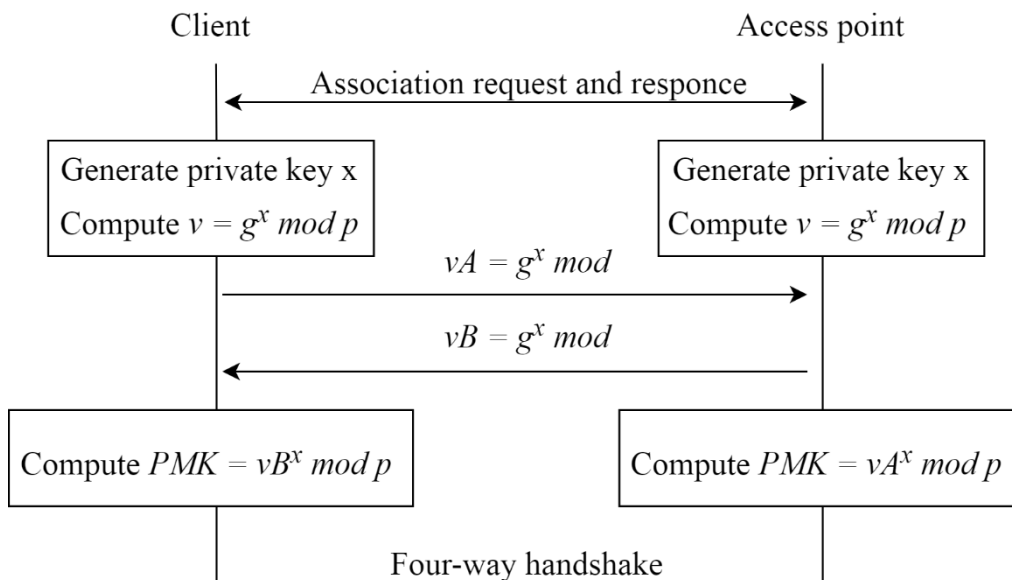


Figure 9 OWE process [98]

The OWE process is very similar to the SAE handshake. The following is a simplified description of the OWE process. For a more in-depth description of OWE, we refer to [98]. Before the OWE process can begin in an open network, both the client and access point must agree on a group size $p-1$ where p is a prime number. Both must also agree on a *group element* g based on the chosen number group. These values are public and are transmitted in 802.11 association request and response frames. Next, both parties choose private key value x from the agreed number group to keep as a secret and calculate a value $v = g^x \text{ mod } p$ to exchange between the two parties. After the exchange, both can calculate a *key* $= v^x \text{ mod } p$. Because of the DLP, an eavesdropper cannot calculate the key value v

without the knowledge of x even if given the g and p values. After both parties have calculated a key it will then be used as the PMK in the four-way handshake process to derive the encryption key PTK. [98]

4.4.3. WPA3 vulnerabilities

Even with the mentioned improvements made to the WPA process in WPA3, it too has been found to be vulnerable and susceptible to attacks. Vanhoef and Ronen first released their findings in April of 2019 [94] and improved on their work in early 2020 [99]. In their work, Vanhoef and Ronen present multiple vulnerabilities that can be used against both the enterprise and personal modes of WPA3. The found vulnerabilities mostly consider weaknesses in the hash-to-curve algorithm used during the SAE handshake process to convert the plaintext password into the password element. The presented vulnerabilities make it possible for a malicious adversary to conduct dictionary attacks to recover the pre-shared password as well as to perform DoS attacks against WPA3 capable access points. For the sake of consistency, we will be disclosing the vulnerabilities affecting the personal mode of WPA3. For a description of all the found vulnerabilities we refer the reader to the original articles [94] and [99].

To provide backwards compatibility and accommodate older devices only capable of using WPA2 encryption WPA3 access points can be set to use a transition mode which makes it possible to use WPA3 and WPA2 at the same time with the same pre-set password [99]. What Vanhoef and Ronen [99] noticed was that they were able to make a WPA3 capable client device connect to a rogue WPA2 access point by imitating a legitimate WPA3 capable access point and by broadcasting a signal stronger than the WPA3 access point.

By launching a rogue WPA2 access point, an attacker can forge 802.11 beacon frames and trick a WP3 capable client device into thinking that the legitimate WPA3 network uses WPA2 encryption. If the rogue access point is close enough to the client device and broadcasts a stronger signal than the legitimate WPA3 access point, the client device will try to establish a connection with the rogue access point downgrading its encryption to WPA2. The client and the rogue access point will exchange the first two frames of the four-way handshake, but because of security measures set in the handshake, the connection will be aborted on the third message.

After receiving the second message of the handshake from the client, the adversary has the information needed to perform a dictionary attack against the legitimate target networks password as described in section 4.3.4. The downside of this “downgrade attack” is that the adversary must be relatively close to the client device to make sure that the client tries to connect to the rogue access point instead of the legitimate WP3 access point making the attack somewhat unpractical. [99]

Vanhoef and Ronen [99] also found out that it is possible to conduct a dictionary attack against the SAE handshake. The attack is based on measuring the time it takes for an access point to compute the hash-to-curve algorithm (section 4.4.1) used for converting the plaintext password into a group element. Because the plaintext password and device MAC addresses are used as elements in the algorithm, the time it takes to compute the algorithm will differ based on passwords length and device addresses. This will result in a situation where every password has a unique “signature” time. By forging a large number of commit frames with different MAC addresses and by measuring the time it takes for the access point to respond, it is possible to gather information about the used passwords signatures.

Based on the collected password signatures, an attacker could conduct an offline dictionary attack and possibly recover the password by running a dictionary of passwords through the hash-to-curve algorithm and comparing the results against the collected passwords signatures. The research presents several vulnerabilities that could be used to launch dictionary attacks against the hash-to-curve algorithm but for the scope of this work, the discussed variations are the most relevant ones. For a more detailed explanation of the presented attacks, we refer the reader to the Vanhoef and Ronen research paper [99].

As a final vulnerability, Vanhoef and Ronen present a DoS attack also leveraging on the hash-to-curve algorithm. As we can recall from section 4.4.1 the algorithm processes the password element 40 times during the procedure making the process computationally heavy for a wireless access points CPU. A malicious actor can send multiple commit frames with spoofed MAC addresses to the access point, forcing it to start multiple hash-to-curve processes exhausting the access points CPU. This can prevent other clients from connecting to the network and possibly drain the battery in battery-powered access points.

The designers of WPA3 knew that the commit frames could potentially be abused to congest the access point and set up a defence mechanism. The defence mechanism consists of a short security frame sent by the access point which the client must reflect back before the commit frames are processed by the access point [99]. The defence mechanism should prevent an adversary from sending forged commit frames from spoofed device MAC addresses. Despite the defence mechanism, Vanhoef and Ronen were able to use forged MAC addresses to send commit frames and reflect the security frames to the access point with relative ease [99].

Even though the vulnerabilities presented by Vanhoef and Ronen might seem severe, the situation is not as dire as it first might seem. Vanhoef and Ronen have informed the Wi-Fi Alliance and device manufacturers about the vulnerabilities and fortunately, most of the issues can be fixed with software updates [99]. In addition, there are only a few WPA3 capable devices available in the consumer market meaning many of the presented vulnerabilities can still be patched for the coming consumer devices.

In the earlier version of their article [94], Vanhoef and Ronen do present critique toward the Wi-Fi Alliance on how WPA3 was fashioned without public review, leaving it vulnerable to attacks. They claim that the current situation could have been averted if the wider community of security experts and researcher would have been involved in the WPA3 implementation process. It is probable that by taking security experts into the process the flaws they presented could have been found earlier and even before WPA3's public release and early products reaching the consumer market. Hopefully, for the next generation of WPA, the Wi-Fi Alliance will consult researchers and experts before its imminent release.

5. Research methodology

In this work, we are undertaking quantitative research methods for collecting our research data about the state of WLAN security and encryption. Quantitative research methods are based on the use of numbers to describe what exists. The benefit of a quantitative research methodology is that because it is based on numbers, we can arrange our findings in easily comprehensible statistics, tables, charts, and figures. The use of quantitative research also gives us the benefit of easily storing our numerical data for possibly continuing on our study in the future. [100]

The main goal of our study is to conduct a survey of the current state of WLAN security in Finland. The objective is to find out to what extent are the broken and deprecated wireless network encryption protocols used today and what the current state of wireless network security exactly is in Finland. The survey data has been collected on three separate occasions between the spring of 2019 and early spring of 2020 in a typical medium-sized Finnish city. On each survey, we collected data from three separate locations within the city. The surveyed locations were chosen for their representation of different sides of the city, namely the industrial district, the more densely populated city centre, and the less densely populated suburb. Each location was surveyed three times on every survey session to collect as much data as possible with the greatest possible accuracy. The collected data has been then assembled into databases and converted into numerical values for statistical analysis and arranged into a graphical form.

For our data collection method we chose a passive WLAN scanning process labelled with the intimidating name *Wardriving*. In the following sections, we will give the reader a more detailed description of the wardriving process as well as descriptions of the software and hardware needed for conducting WLAN surveys via wardriving. The wardriving survey process developed for this work is presented in Figure 13 at the end of section 5.3. In addition to the technicalities included in the wardriving process, we will be disclosing some of the legal and ethical issues considering wardriving.

5.1. Wardriving

Despite the ominous-sounding name, wardriving is not hacking, criminal or in any other way harmful toward the “target” wireless network devices or to the owners of the said devices. The name wardriving stems from the term “*Wardialing*” inspired by the 1983

movie *Wargames* in which the main character is seen using a computer to dial consecutive phone numbers to locate computers [101]. In the early days of our modern computer-networking, computers communicated by using modems connected to the landline phone network. This meant that by simply calling every consecutive phone number within a given area, it was possible to locate modems and more importantly the computers connected to the modems [101]. Early hackers would exploit this and wrote programs that would dial consecutive phone numbers and make a record of every modem that answered to the call.

Wardriving does very much the same thing as wardialing but updated to WLAN networks and with greater efficiency [101]. To put things in simple terms, wardriving is the act of moving around a certain area and scanning wireless network devices and mapping the located devices for statistical purposes [102]. Today wardriving can be extended to different wireless communication technologies such as Bluetooth and ZigBee used in IoT and smart devices. The name of the activity can vary depending on the mode of transportation from warwalking, warbiking to warflying with drones [101]. Wardriving first became popularized in the early 2000s during the wake of the commercial success of WLAN devices. The first known wardriving survey was conducted in 2001 when a security researcher by the name of Peter Shipley first announced the results of his 18-month survey during the information security conference DefCon [102]. Shipley's presentation at the conference has been made available on the DefCon Conference Youtube channel [103].

Although many of the software used in wardriving as well as the gained information can be used for malicious purposes, it should be emphasised that the act of wardriving in itself is not hacking and is a legitimate tool used by information security researcher and professionals. The purpose is not to simply move around and seek for outdated or open access points to gain easy entrance into poorly configured networks. The main idea of wardriving is to collect information for statistics to raise awareness about the security of WLAN networks. Shipley already points this out in his 2001 presentation [103]. His idea was that because WLAN technology was at the time fairly new and becoming increasingly popular, most people probably would not know how to secure their new wireless network. For this reason he thought that it would be important to somehow raise awareness about the situation. To back up his claims he started to survey wireless access points and found out that most of them truly were unsecured.

The situation today is in many ways the same as it was in 2001 for two main reasons. Firstly, the number of WLAN devices is rising fast because new IoT and smart devices are becoming more common and consumers might not have enough knowledge about how to configure their devices securely or that the devices themselves might be inherently insecure. This was shown well by Kumar et al. [104] as they used data acquired from the information security company Avast to study the security of IoT devices and noted that over 50% of homes in Western Europe and over 66% in North America had one or more IoT device.

When looking at the security of the devices worldwide, Kumar et al. [104] found out that large numbers of IoT devices supported old and somewhat obsolete protocols such as FTP (7.8%), Telnet (7.1%) and that nearly half supported the insecure HTTP protocol. They also found out 14.6% of common consumer wireless network routers supported either FTP or Telnet. Another distressing finding was that out of all the mentioned devices, 17.4% exhibited weak factory-set default FTP credentials and 2.1% had weak Telnet credentials [104]. These numbers might not seem large but to put things into perspective, we are talking about millions of devices and more are produced and sold every day.

Secondly, as discussed in the previous chapter the outdated WLAN encryption protocols have been either broken or are in other ways vulnerable to attacks and are at the moment in the middle of transitioning to a new generation. As a consequence of these two reasons, we find ourselves once again in a situation where people are acquiring more new wireless network devices without being familiar with their security aspects. This is once again creating an increased need for raising awareness about the security of WLAN networks.

Fortunately, there is an active community of wardrivers today who share their findings online for statistical purposes. One of the most prominent websites providing WLAN statistics acquired by wardriving is the already mentioned (Section 4.2.2) Wigle.net that has been active online since 2001 [105]. In addition to the research dedicated to the WLAN security protocols and measures discussed in chapter 4, some studies have been dedicated to mapping WLAN networks and the used security protocols by means of wardriving.

In our brief research, we were able to find 17 different published studies similar to ours. Out of the 17 found studies six had been conducted in New Zealand and three in Malaysia

making them the most represented countries. The most prominent of the New Zealand studies have been done by Lin, Sathu and Joyce [106] in 2004, continued by Sathu and Sarrafzadeh [107] in 2015, Nisbet in 2012 [108] and 2013 [109]. The latest of the New Zealand studies was conducted by Kyaw, Agrawal and Cusack [110] in 2016. Only three out of the found 17 studies had been conducted in Europe, the latest one conducted by Valchanov, Edikyan, Aleksieva [111] in Varna, Bulgaria in 2019.

5.2. Operating system, software, and hardware for wardriving

For this work, we decided to use a Linux based operating system and software. The decision was made after testing different solutions with different operating systems. It was soon noted that the Linux based systems were simply best suited for our wardriving endeavours. This is because Linux based operating systems allow using WLAN interfaces in monitor mode allowing passive scanning. In addition, many of the software designed for wireless network scanning and wardriving are designed for Linux operating systems for the mentioned reason.

Secondly, we wanted to better accommodate the set requirement of presenting an easily repeatable process using built-in and out of the box solutions. The operating system used in this work is *Kali Linux* developed and maintained by the Offensive Security group [112]. Kali Linux is based on the Debian Linux distribution and is aimed toward security professionals to be used for penetration testing and security auditing [80]. Kali Linux has all the tools needed for wardriving pre-installed, is well documented and has great built-in support for many off-the-shelf wireless network adapters [112]. A very similar Linux operating system dubbed Parrot Os was also considered to be used in this work, but Kali Linux was chosen for its popularity and its better documentation and support.

It was also chosen to run Kali Linux as a virtual machine on a Windows 10 laptop to discard the need for installing a whole new operating system on to a device making the process as user friendly as possible. Using a virtual machine also has many other benefits over physically installing a completely new operating system. For instance, virtual machines further increase the support for different device drivers and have the possibility of taking snapshots of the virtual machine, making experimenting with the operating system easier for new users. Offensive Security also offers ready to deploy virtual machine images which can be downloaded and deployed as is without the need for having any prior

knowledge on how to create virtual machines or how to install an operating system. Offensive Security offers virtual machine images for VMware Workstation, Oracle Virtual-Box and Microsoft Hyper-V which are all free hypervisor software for personal use [113]. In this work, we are using VMware Workstation 15 Pro version 15.5.1 build-15018445.

The downside of using a virtual machine is that it runs on top of the host device and therefore uses the host devices resources when deployed, which increases the amount of processing power and other resources needed from the host. Fortunately running a virtual Linux distribution does not take very much resources to run smoothly and most modern laptops have more than enough processing power to operate a virtual machine. If you are already using a Linux operating system such as Ubuntu or Debian on your physical device, the Linux based software discussed later in this work can be installed on the operating system, therefore, eliminating the need to use a virtual machine.

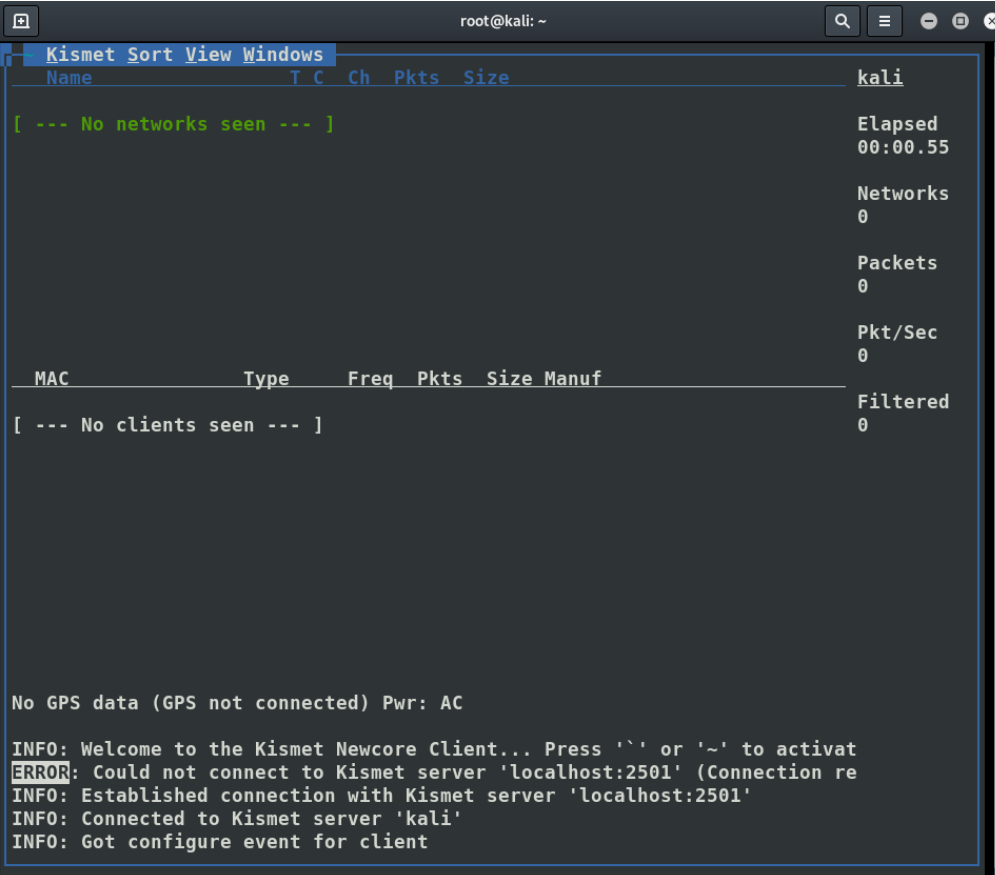
As one of the aims of this thesis was to develop an effective and easily repeatable process for conducting WLAN surveys via wardriving, we have chosen as many out of the box and off-the-shelf solutions as possible. One of the easiest ways for anyone to try wardriving is to use a simple smartphone application. Smartphones with their built-in WLAN and *Global Positioning System* (GPS) capabilities makes them a good starting point for anyone wanting to test wardriving without the need for purchasing new equipment. The people behind Wigle.net have produced their own easy to use wardriving application for Android smartphones which can be downloaded free of charge from the Google Play store [114]. The downside of using a smartphone for the wardriving process is that the mobile WLAN interfaces designed for smartphones are not as powerful as those designed for computers. This will impact on the results as the less powerful mobile WLAN interfaces have more limited range compared to an external wireless interface designed for computers.

5.2.1. Wardriving software

A plethora of different software has been produced over the years for different operating systems each having with their different purpose and features. One of the first freely available WLAN mapping software published for the Windows operating system is the *Netstumbler* [101]. Two of the more recent software that have been developed for Windows are the updated version of Netstumbler dubbed as *Vistumbler* [82] and *Acrylic Wi-Fi* by Tarlogic Research [115]. The biggest difference between the three is that Acrylic is

a commercial software with home and enterprise versions available, whereas Netstumbler and Vistumbler are completely free of charge and are open-source projects. The downside of using Windows-based software is that they are often active scanners meaning that they have to interact with the target devices to collect information [80]. In addition, as Netstumbler is one of the oldest scanners available for Windows it has poor support for the newest versions of Windows and it is updated infrequently compared to Acrylic and Vistumber [82]. Different versions of Netstumbler have also been released for the Apple OS X operating system under the names MacStumbler and iStumbler.

After testing different options for the software to be used in our wardriving survey process, it was decided that we would use *Kismet*. Kismet is an easy to use, open-source 802.11 wireless network detector, sniffer and it can also be used as a wireless *Intrusion Detection System* (IDS) [80]. Kismet was chosen for its versatility, GPS support, simple *User Interface* (UI) (Figure 10) and for its ability to use multiple WLAN adapters at the same time. Kismet also has a large active user base, is well documented and maintained. In addition, Kismet comes pre-installed with Kali Linux so new users not yet familiar with Linux do not have to install it separately.



```
root@kali: ~
Kismet Sort View Windows
Name          T C Ch Pkts Size
[ --- No networks seen --- ]
Elapsed
00:00.55
Networks
0
Packets
0
Pkt/Sec
0
MAC          Type   Freq Pkts  Size Manuf
[ --- No clients seen --- ]
Filtered
0

No GPS data (GPS not connected) Pwr: AC
INFO: Welcome to the Kismet Newcore Client... Press '' or '~' to activate
ERROR: Could not connect to Kismet server 'localhost:2501' (Connection re
INFO: Established connection with Kismet server 'localhost:2501'
INFO: Connected to Kismet server 'kali'
INFO: Got configure event for client
```

Figure 10 Kismet user interface

Kismet has also been in development for a long time and many of its functions have been automated making Kismet a very easy to use software for new users. Kismet automatically creates log files in many different file formats making the data sampling process (section 5.3) quite simple. There are two versions of Kismet: the newer version uses a web browser-based UI and whereas the older uses a much simpler terminal UI. In this work we are using the older version for its simplicity and because at the time of writing this work the newer version did not support GPS and was not stable enough for our use. The version of Kismet used in this work is 2016-07 R1.

Behind its easy to use and automated features, Kismet is much more than just a simple network scanning software. It is a complete framework for capturing and analysing 802.11 networks [82]. The framework is comprised of two separate components, the Kismet client seen by the user and the Kismet server which performs most of the work in the background [82]. When a user first starts Kismet, both the client and server components are launched and the user will be interacting with the server through the client user interface.

Another aspect of wardriving is to visualise the found WLAN devices on a map. To be able to pinpoint the location of the device on to a map we need to use GPS alongside Kismet. For this purpose we used *GPSD*. *GPSD* is open-source software that allows us to connect a GPS device through a USB connection to our system and makes the GPS data available for us [82]. After the GPS device has been successfully connected to *GPSD*, the GPS data can then be queried by Kismet and tagged with the found WLAN devices [82]. *GPSD* also comes pre-installed in Kali Linux so again the user does not need to install it by hand.

5.2.2. Wardriving hardware

The process of choosing the right software for wardriving can be an exhausting task for those who are not yet familiar with wardriving. The good thing here is that choosing and acquiring the needed hardware is a much simpler task. Today, wardriving can be done with very minimal and simple everyday off-the-shelf hardware. In its most simple form, the only things needed are an Android smartphone and a free application. For more accurate results, it is recommended to have a laptop computer, a GPS receiver, and an external WLAN adapter (Figure 11). As already mentioned in the previous section, we are using a laptop computer with a Windows 10 operating system and Kali Linux as a virtualised

operating system on top of Windows 10. In addition to the laptop, we used an external USB WLAN adapter and a USB GPS receiver.



Figure 11 GPS receiver and WLAN adapter used in this work

The wireless adapter model used in this work is TP-Link AC600 Archer T2UH with MediaTek MT7610U chipset. The adapter was chosen for its high-gain external antenna and its dual-band capabilities. These features provided us with better range and the ability to include devices operating in both 2.4 GHz and 5. GHz bands. When choosing which WLAN adapter to use we recommend to first consult manufacturer websites to make sure that they provide device drivers for Linux operating systems. Many websites, which can be found with a simple Google search, have collected lists of WLAN adapters supported by different Linux operating systems.

As the GPS receiver, we used GlobalSat G-Start IV BU-353S4. Because we acknowledged that not everyone has a USB GPS receiver at hand, we also tried using an Android smartphone as the GPS receiver. Using a smartphone as the GPS receiver involves some extra work such as installing an application to the smartphone for sharing the GPS info

with Kali Linux, enabling the developer options on the smartphone, and installing *Android Debug Bridge* (ADB) tools to Kali Linux. As we tested both the GlobalSat GPS receiver and smartphone solution we concluded that we had more accurate results with the smartphone setup and chose to use it for our endeavour. Because of the extra work caused by setting up the smartphone connection, we would recommend the much simpler USB receiver for the less experienced user. To share the GPS data from our smartphone to Kali Linux, we used the *Share GPS* application which can be downloaded for free from the Google Play store. There is a possibility in the application to connect a smartphone wirelessly to Kali Linux via Bluetooth. For this work, we chose to use a USB connection since Bluetooth and WLAN operate on the same 2.4 GHz band and using them closely together might have caused interference during our survey process. It should also be stated that, if you are not interested in mapping the device locations, the use of GPS is naturally not required.

5.3. Data sampling and analysis

The data sampling and analysis processes used in this work are simple and straight forward. We chose the following applications and procedures. Firstly, to accommodate the requirement of presenting an easily repeatable process for wardriving. Secondly, we needed to efficiently store and organise our survey data into a form that allows us to easily find only the needed information and present our findings. In addition, we wanted to be able to see our findings visualised on a map. To meet our requirements for storing, finding, and presenting our research data, we decided that storing the information into databases would be the best option. This allows us to easily take only the information we need from our datasets, as well as wipe out the data we do not need in our research or want to store for the future.

After searching for and testing different solutions we soon noticed that Kali Linux comes pre-installed with an application called *GISKismet*. GISKismet allows us to easily transfer data into an *SQLite* database enabling us to pull wanted information from the database with simple SQL queries as well as to visualise our findings on a map [82]. By default, Kismet creates five different log files (Figure 12) *.alert*, *.gpsxml*, *.nettxt*, *.netxml* and *.pcadump*. What we are most interested in is the *.netxml* file. GISKismet uses the *.netxml* file as input and parses information from it into an *SQLite* database. GISKismet is extra useful for us since we ran each route three times on every survey meaning we had multiple

netxml files. When importing data into the database GISKismet automatically leaves out duplicate networks from the database conserving us valuable time.

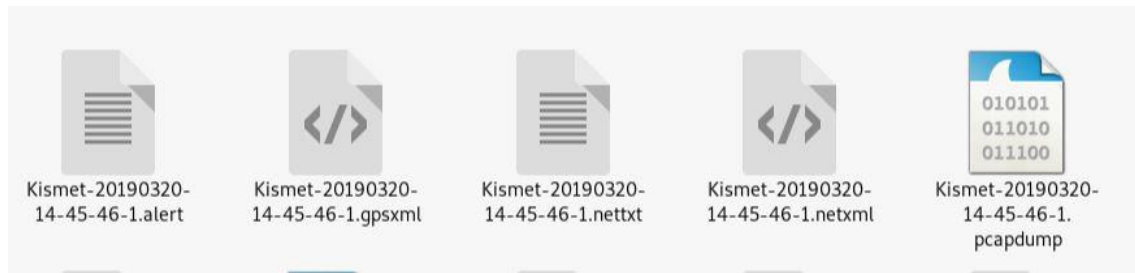


Figure 12 Kismet log files

The SQLite database itself contains two tables, one listing all the found WLAN networks and the other listing all the client devices connected to the said networks. The table of found networks has the following information: network SSID, MAC addresses, network device manufacturer, used wireless channel, is the SSID cloaked, used encryption, first and last time network is seen, variable GPS information, and other technical information about the network devices. For our study, we used the SSID, MAC address, manufacturer, channel, and encryption information. In addition, the GPS information was used for the map visualisation. For extracting the wanted information from the database we used the *sqlite3* application, which also comes pre-installed with Kali Linux. There are several different graphical and easy to use software which can be used for browsing and pulling information from an SQLite database, for example, the *DB Browser for SQLite* which can be downloaded free of charge for Windows, Linux and Apple macOS [116].

After we selected the data we want from the database we parsed it into *Comma-Separated Value* (CSV) files. The CSV files can be then opened in spreadsheet software such as Microsoft Excel and Google Sheet for further inspection and analysis. For visualising the found networks on a map we used GISKismet for pulling and parsing the needed GPS and network information from the databases into *Keyhole Markup Language* (KML) files which can then be opened for example in Google Maps or Google Earth.

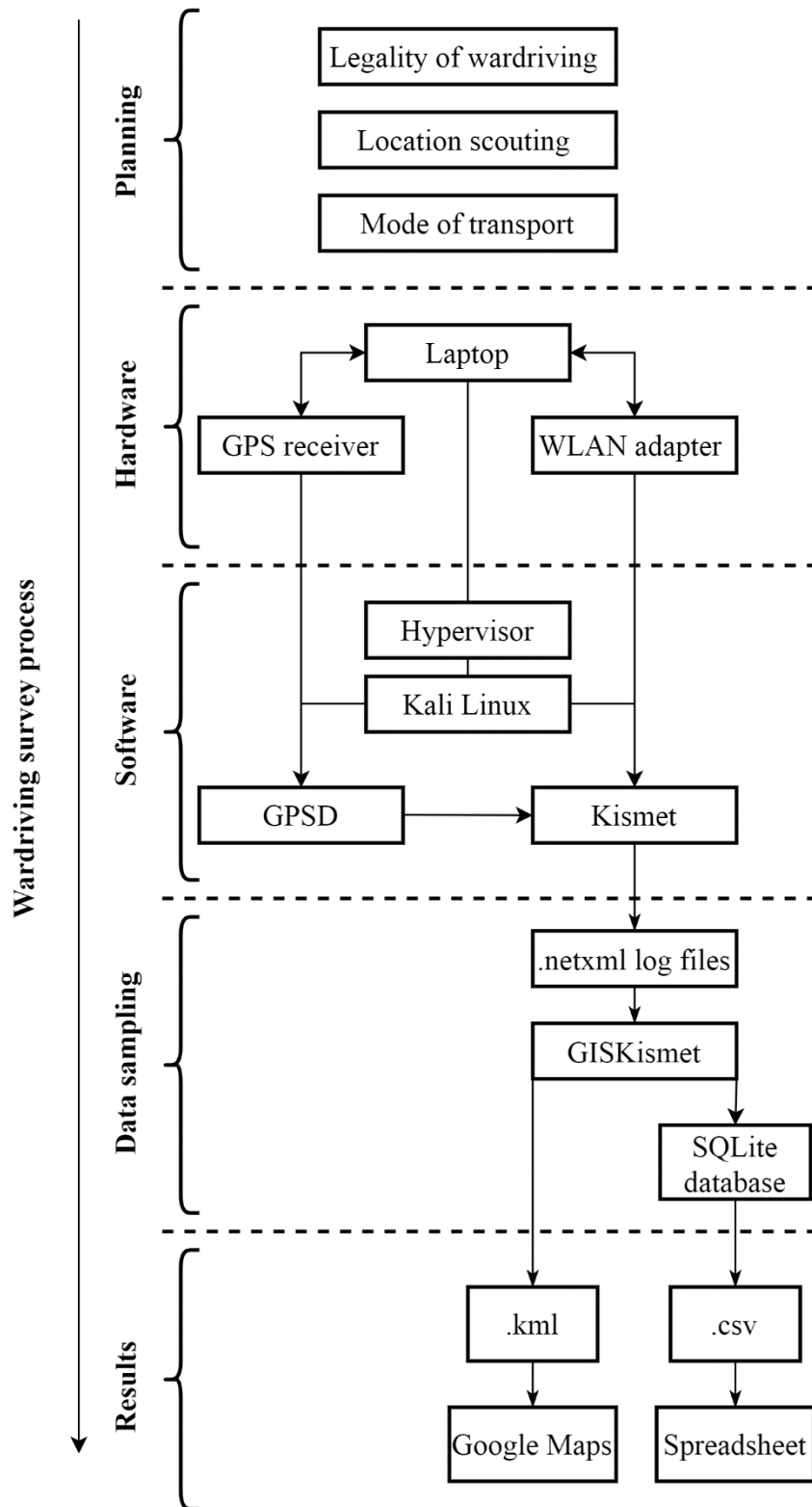


Figure 13 Wardriving survey process

5.4. The legality of wardriving and the GDPR

The Finnish law is quite clear when it comes to the legality of wardriving. According to the Finnish criminal law the use of an open unencrypted WLAN network is completely legal. This means that one can use any open WLAN network for web surfing or online gaming [117]. As connecting into an unknown password-protected WLAN network would mean breaching the security of the network in one way or the other, these kinds of actions are naturally deemed as criminal in the eyes of the Finnish criminal law.

Furthermore, the use of an open WLAN network becomes criminal if the connection is used for connecting and snooping into other devices and services in the network without the owner's permission [118]. In practice, this would mean that you cannot, for example, login into the wireless access point's admin panel and make changes to the network settings or browse files on someone else's computer connected to the network. As we are using passive network scanning methods, we are not making any connection attempts or will in any other way interact with the scanned networks, our wardriving endeavours are fully legitimate in the face of the Finnish criminal law.

Passive network scanning is merely just listening to the beacon and probe request frames broadcasted by the wireless devices. WLAN capable devices such as laptops and smartphones are doing this by default to find and associate with wireless access points. 802.11 beacon frames are one subtype of 802.11 management frames [80]. During this work we have already mentioned other management frame types such as deauthentication, probe request and response frames. Beacon frames are broadcasted by wireless network devices at regular intervals. They are used by devices to announce their presence to other devices and for starting the association process between devices [80]. Beacon frames are sent unencrypted and carry information about the wireless network such as the SSID, used encryption, used channel, MAC address and vendor information. By setting our wireless interface into monitor mode we can hop from wireless channel to channel capturing as many broadcast frames as possible from as many wireless devices as possible [80].

5.4.1. Wardriving and the GDPR

As Finland is a member of the *European Union* (EU), we must consider the possible regulations set by the EU. The EU *General Data Protection Regulation* (GDPR) came into effect on May 25th of 2018 and is a set of rules and regulations that have an effect on how different entities can collect and use one's personal information and data [119]. The aim is to give people more rights and power over their personal data and force transparency on how that personal data is being collected and used for example by governments or large online businesses such as Google or Facebook. One good example of users having more power over their data is the "right to be forgotten" which according to the GDPR gives the users power to have their personal data to be deleted from online services [119].

As the GDPR documentation has very profound and lengthy descriptions about the regulations considering the use of personal data and fully describing the GDPR documentation would be out of the scope this work, we are not going to have an in-depth discussion of GDPR in this work. Instead, we will be concentrating on those portions of the regulation that affect our research. We refer those more interested in the subject to familiarize themselves with the GDPR documentation [119].

In the case of wardriving where we are collecting and storing information about WLAN networks and network devices, we should be considering how the GDPR defines personal data and how it regulates collecting and storing the possibly personal data. The GDPR documentation defines personal data as; "*any information relating to an identified or identifiable natural person ('data subject'); an identifiable natural person is one who can be identified, directly or indirectly, in particular by reference to an identifier such as a name, an identification number, location data, an online identifier or to one or more factors specific to the physical, physiological, genetic, mental, economic, cultural or social identity of that natural person*" [119].

Because we are collecting location data and information such as network SSIDs and device MAC addresses which could possibly be interpreted as being online identifiers we must further investigate how they are defined by the GDPR documentation. The GDPR documentation states the following about online identifiers; "*Natural persons may be associated with online identifiers provided by their devices, applications, tools and protocols, such as internet protocol addresses, cookie identifiers or other identifiers such as*

radio frequency identification tags. This may leave traces which, in particular when combined with unique identifiers and other information received by the servers, may be used to create profiles of the natural persons and identify them” [119].

Based on the this statement we could say that device MAC addresses can be thought of as personal data since they are unique to a single device and could, in theory, combined with the device SSID and location data be used to directly or indirectly identify the owner of the said device. Although on the other hand in the case of WLAN access points, there are usually several people using a single access point making it impractical to pinpoint and identify the true owner of the device. The exception here being smart and IoT devices or laptop computers, but even then, identifying the device owner based solely on its MAC address and SSID is not practical in most cases.

This is because even though the MAC addresses are unique to a device, no one can ever truly tell the owner of the device since they are not connected to a person’s name, nor are there any databases which connect any other personal information to a specific device MAC address. Even by combining the device SSID, MAC address and location information the probability that the data could be used to identify a single person is very low. The collected location data is only a rough estimate based on the scanned device’s signal strength and location of the scanning device. Moreover, as we know many factors affect the strength of the signal and as we are moving by car, making accurate enough estimations about the location of the device based on the location data is very much implausible.

Still, even though accurately identifying a person based on our research data is implausible and as we are collecting and processing some data that could be deemed as personal, we have to make sure that we are complying with the GDPR regulations on processing personal data and the lawfulness of processing personal data. Because we are conducting scientific research, GDPR gives us some specific guidelines and freedoms on collecting and storing personal data. For example, GDPR article 5 states that personal data shall be:

“collected for specified, explicit and legitimate purposes and not further processed in a manner that is incompatible with those purposes; further processing for archiving purposes in the public interest, scientific or historical research purposes or statistical purposes shall, in accordance with Article 89(1), not be considered to be incompatible with the initial purposes (‘purpose limitation’);” [119]

“kept in a form which permits identification of data subjects for no longer than is necessary for the purposes for which the personal data are processed; personal data may be stored for longer periods insofar as the personal data will be processed solely for archiving purposes in the public interest, scientific or historical research purposes or statistical purposes in accordance with Article 89(1) subject to implementation of the appropriate technical and organisational measures required by this Regulation in order to safeguard the rights and freedoms of the data subject (‘storage limitation’);”. [119]

Based on these statements we can collect and store our research data for scientific research purposes, as long as we store the data in a manner that prevents it from leaking outside of our research, and is stored securely and in a form that even in the case of the data being leaked no one can identify a person based on the data. To comply with the presented statements, we have removed the last three octets of the device MAC addresses from our final datasets as they are assigned uniquely to every device, leaving the first three octets that contain information about the device manufacturer. This way the MAC address cannot be used to identify a single device. We have also deleted all the unnecessary and unused Kismet log files and have stored only the data necessary for our present and future research. After we have completed our research, the used data will not be shared to third parties and will be stored accordingly on the university's information systems for possible future research use.

Article 6 of GDPR presents six conditions based on which processing personal data shall be lawful. Article 6 states the following:

Processing shall be lawful only if and to the extent that at least one of the following applies:

- a) the data subject has given consent to the processing of his or her personal data for one or more specific purposes;*
- b) processing is necessary for the performance of a contract to which the data subject is party or in order to take steps at the request of the data subject prior to entering into a contract;*
- c) processing is necessary for compliance with a legal obligation to which the controller is subject;*
- d) processing is necessary in order to protect the vital interests of the data subject or of another natural person;*
- e) processing is necessary for the performance of a task carried out in the public interest or in the exercise of official authority vested in the controller;*
- f) processing is necessary for the purposes of the legitimate interests pursued by the controller or by a third party, except where such interests are overridden by the*

interests or fundamental rights and freedoms of the data subject which require protection of personal data, in particular where the data subject is a child. [119]

In our case of legitimate scientific research conducted by a public university, we see that sections e and f give us the lawful basis of processing our research data. The “controller” as defined in the GDPR “*means the natural or legal person, public authority, agency or other body which, alone or jointly with others, determines the purposes and means of the processing of personal data;*” [119] will in our case be the university. Moreover, as we are conducting our research for the public interest and to promote WLAN security, we see that based on this, section e will give us the legitimacy to conduct our research in the eyes of the GDPR.

5.5. Ethics of wardriving

Now that we have made sure about the legality of wardriving as a research method, we should also think about the ethical issues wardriving might pose to us. In the world of information security, it is a well-known fact that in order to stay on top of their craft, information security professionals must master many of the same tools and skills as their malicious hacker counterparts. This line of information security work is known as “ethical hacking” [120]. Ethical hackers are most often researchers or professionals who are hired by businesses to run security audits and penetration testing on their information systems with the intent of finding and fixing possible vulnerabilities before malicious hackers can exploit them [120]. Ethical hackers are by definition hackers, but they are bound by ethical guidelines to using their skills for good. As wardriving is a tool that could potentially be used by hackers and ethical hackers alike, it is important to ask ourselves if it can truly be ethical to conduct research by the means of wardriving [120].

Although extensive literature exists on the topics of computer ethics, ethics of technology, and ethics of information security, very little if nothing has been written about the ethics of wardriving. Fully and comprehensively investigating the ethicality of wardriving is a new and much greater subject that would need more in-depth discussion to examine in the full extent that the topic deserves than we can offer in the confines of this work. Nevertheless, we can still offer a brief discussion about the ethics of wardriving. In this section, we try to reason and find out if it is ethically and morally right to use wardriving as a research method in the manner that we have used it in this work.

Before venturing further into the ethics of wardriving we must first start by shortly defining what we mean by ethics. When searching for a definition of ethics one usually finds a statement along the lines of *ethics is the philosophical study of morality* [121]. Through the study of ethics and morality, we can get a better understanding of what constitutes as either morally right or wrong. There are several different ethical theories, all of which try to teach and explain to us how to act ethically and morally right. For us to be able to reflect on if the act of wardriving is in fact ethically right, we must familiarise ourselves with some ethical theories. For the purposes of this work, we chose the ethical theories of *Utilitarianism* and *Virtue ethics*. These two theories were chosen because they are quite different in nature, but at the same time pose very fitting questions in the scope of this work.

5.5.1. Utilitarianism and Virtue ethics

Two of the most important advocates of utilitarianisms were the British philosophers, Jeremy Bentham, and John Stuart Mill. Utilitarianism is one of the *consequentialist* ethical theories, it tries to sort out morally right actions by looking at the outcomes and consequences of one's actions and how much happiness or "utility" an act produces when compared with other actions that could have been taken in the same situation [121]. Utilitarianism branches out to two different schools, *act* and *rule* utilitarianism [122].

Act utilitarians believe that when we are deciding on how to act, the morally right course of action is the one that would produce the greatest good for the greatest number [122]. Rule utilitarians consequently believe in the importance of justified and pre-defined moral rules. An act utilitarian would say that an action is right if it is based on a justified moral rule that yields more utility than any other possible moral rule [122]. The key difference between the two branches is that act utilitarians base their moral thinking to the individual actions and its consequences. Whereas rule utilitarians apply their moral thinking on moral rules and then evaluate if the action complies with the moral rule [122].

Inspired by the Greek philosopher Aristotle, virtue ethics emphasises one's character and virtue, instead of action and their consequences when judging if an act is morally right. Virtue ethics tries to answer much broader questions than the consequentialist ethical theories. Instead of focusing on finding universal ethical principles that apply in moral decision making, virtue ethics tries to answer questions such as "how ones should live?" and "what is a good life?". The very basic idea of Aristotelian virtue ethics is that, if a person

has a good character and acts according to his good character traits, he will be virtuous and therefore act morally right. [123]

Because Aristotle believed that these good traits come rationally from us but need to be practised and nurtured for them to become natural for us. A virtuous person, therefore, is one that acts kindly toward others because it is part of his nature and character, not because it would maximise utility or fill one's duty. Moreover, virtuous acts should be considered as those that are in the "golden mean". Virtuous acts are those that are between excess and insufficiency, for example, the virtue of courage lies somewhere between recklessness and fearfulness [124]. If one can live a life of practising and nurturing his virtuous character traits he will flourish, make the right ethical choices, and be a good moral exemplar to others [123].

5.5.2. Wardriving and Utilitarianism

For the scope of this work and to keep this section brief we will be considering wardriving only from the act utilitarian perspective instead of both branches of utilitarianism. As already discussed in the previous section, act utilitarians base their thinking on the consequences of one's actions. Therefore, we must first contemplate the possible consequences our wardriving endeavours might produce. Furthermore, we can recall that wardriving and passive wireless network scanning is not in any way harmful to the target network devices, to the network the devices reside in, nor to the device owners. In fact, the device owner and network users do not even know that their wireless network or network devices are being surveyed.

The fact that the device owners do not know about their devices being scanned does raise questions about the device owners' consent on being scanned. Moreover, the device owners could see wardriving as an invasion of their privacy since we are collecting location as well as other information that could be seen as personal and could at least, in theory, be used to identify the device owner. The issue of consent is quite problematic for us as we cannot possibly ask every device owners consent and we cannot know beforehand how many devices are on our route let alone whose devices will be on our range while wardriving. Under these circumstances asking for consent would be an impossible task and would, in addition, defer us from conducting our research and cause much more distress than utility.

Now that we know some of the possible negative consequences of our wardriving endeavours could cause, we should think about the possible utility our research might produce and if they do outweigh the negatives. As researchers our aim is at producing valuable information for a greater community and raise awareness about the current state of WLAN security. By conducting our research according to laws and regulation and by publicly releasing our results, we are serving a greater good through bringing up the possible issues and raising awareness in hopes that our findings will aid on improving the state of WLAN security. As we will not be releasing any personal information that could endanger the device owner's privacy and our actions are not visible to the network users or device owners, we could say that our research will produce very little harm. By counting on the possible utility our research might produce to the community, we could conclude that wardriving in the context of research will bring up more good than harm to the greater community and would thus be ethically and morally right according to act utilitarianism.

If we would be acting out of the context of research, solely for the sake of wardriving and going our way posting the gathered information to an online service such as Wigle.net, our actions would possibly be causing more harm than utility for the network device owners. We would be deliberately releasing private information to the public and handing out information about possibly vulnerable networks that could then be used by a malicious actor. In this case, we would be acting immorally in the eyes of utilitarianism.

5.5.3. Wardriving and Virtue ethics

Analysing wardriving from the perspective of virtue ethics is not as straightforward as from the perspective of act utilitarianism because of all the different aspects virtue ethics takes into account when judging moral actions. For the sake of simplicity and the scope of this work, we will meditate the morality of our actions through two aspects of virtue ethics we see as the most profound in the eyes of this work. Firstly, as we can recall, according to virtue ethics, morally right actions are those that a virtuous person acting out of good character would do in a given situation [123]. In other words, a morally right person does the right thing at the right time for the right reasons. Secondly, morally right acts are those in the golden mean of virtue between excess and deficiency [123].

In our research, we are wardriving for the sake of seeking information about the state of WLAN security. We practise our learned technical skills in the hopes that our published research results will bring attention to the issues of WLAN security. Moreover, we are acting at a time where the number of WLAN devices is increasing rapidly, and a new encryption protocol is being implemented to replace the old and vulnerable protocols. We are acting in accordance with reason and within the limits of laws and regulations when collecting the needed information for our research. We are not going out of our way to expose possibly vulnerable wireless networks or to violate anyone's privacy by releasing detailed personal information to the public. As we are using our acquired skills as researchers out of good character, for the right reason, at the right time and within reason, it seems that wardriving in our case would be morally right in the context of virtue ethics.

6. Research findings

In this chapter, we present the results of our wardriving survey. More importantly, we are trying to find answers to our main research questions about the current state of WLAN security in Finland. We are most interested in the encryption protocols used and if there still indeed are large numbers of networks configured with the obsolete and deprecated encryption protocols. In addition, we try to make more general observations about WLAN security by observing the prevalence of other security practices such as cloaking or changing the network default SSID and channel use. The use of a default network SSID could, for example, indicate that the network owners are using possibly insecure default factory settings in their devices.

We will first present the findings from each of the three different locations for comparison to see how they differ from each other and if there are some defining trends or features between the areas. After comparing the results from the three individual areas we will be observing the survey results as a whole and try to compose a picture about the current state of WLAN security and take note of the possible deficiencies or flaws in current WLAN security practices.

Before venturing further into the results, some matters should be first clarified about the presented results.

- Firstly, when talking about open unencrypted networks we must remember that most often they have been left open intentionally. Businesses, organisations, and schools, for example, usually have dedicated open networks for customers, guests and students which are often named as guest and visitor networks. As the surveyed areas naturally contain businesses, schools, government, and administrative buildings which have dedicated open unencrypted guest networks, we try to make distinctions between the intentionally and unintentionally open unencrypted networks based on the network SSID and location information and only take into account the possibly unintentionally open networks when discussing truly insecure networks.
- Secondly, we acknowledge the fact that one access point might have multiple wireless network interfaces each with their own MAC address. Kismet will, therefore, list them as individual networks and network devices. This will, of course,

have an effect on the amount of found networks, but as we are most interested in the used encryption protocols and accurately sorting out the networks that originate from the same device would be a rigorous task, we have left the final amount of networks as is.

- Thirdly, it should be clarified that when we are talking about default network SSIDs we mean factory set easily identifiable SSIDs. Very often the default network SSID is simply the device manufacturer, device model or service providers name coupled with a series of numbers and letters, for example, TP-LINK_1X01, HUAWEI-B315-XXX, TW-EAV510 or 4G-Gateway-101A. In addition, it is not uncommon to see for example the MAC address of the device being used as the default network SSID.
- Lastly, we want to acknowledge the WPA-TKIP and WPA2-AES mixed mode. In mixed mode, the wireless network is configured to accommodate both TKIP and AES capable devices making the network possibly vulnerable to attacks against TKIP if such devices are connected to the network. Despite this possibility, we have decided to group the networks operating in mixed-mode as part of WPA2-AES networks instead of WPA-TKIP networks to keep the presented comparisons and results more simple and clear for the reader.

6.1. The three surveyed locations

As already disclosed during this work, we conducted our survey in a middle-sized Finnish city on three separate occasions and in three different locations each representing a different part of the city, the industrial district, the city centre, and the suburb. The idea behind surveying three different parts of the city is to have more diverse data than we would have gotten by only concentrating on one specific area in the city. Each location was surveyed three times in each survey session to ensure that we discover as many networks as possible in each area. Before the three actual survey runs, we performed a test survey where we scouted the chosen locations and tested different tools, hardware, and software. After our test survey, we chose to change one of the locations due to poor road and traffic design in the area. Furthermore, we decided to upgrade our WLAN adapter to one with a higher gain external antenna with dual-band capabilities to accommodate both the 2.4 GHz and 5 GHz bands for better results. Results from the test survey run have not been incorporated into the final survey results.

6.1.1. The industrial district

The industrial district has very little housing and mostly consists of a large variety of different businesses varying from car dealerships to metal workshops and small technology start-ups to CrossFit gyms. Our chosen route (Figure 14) ran 2.9km long, and along the route we managed to locate 338 networks during our surveys translating into one detected access point on every 8.5 meters. The 338 located networks constitute 19.3% of the combined number of networks found during our research.

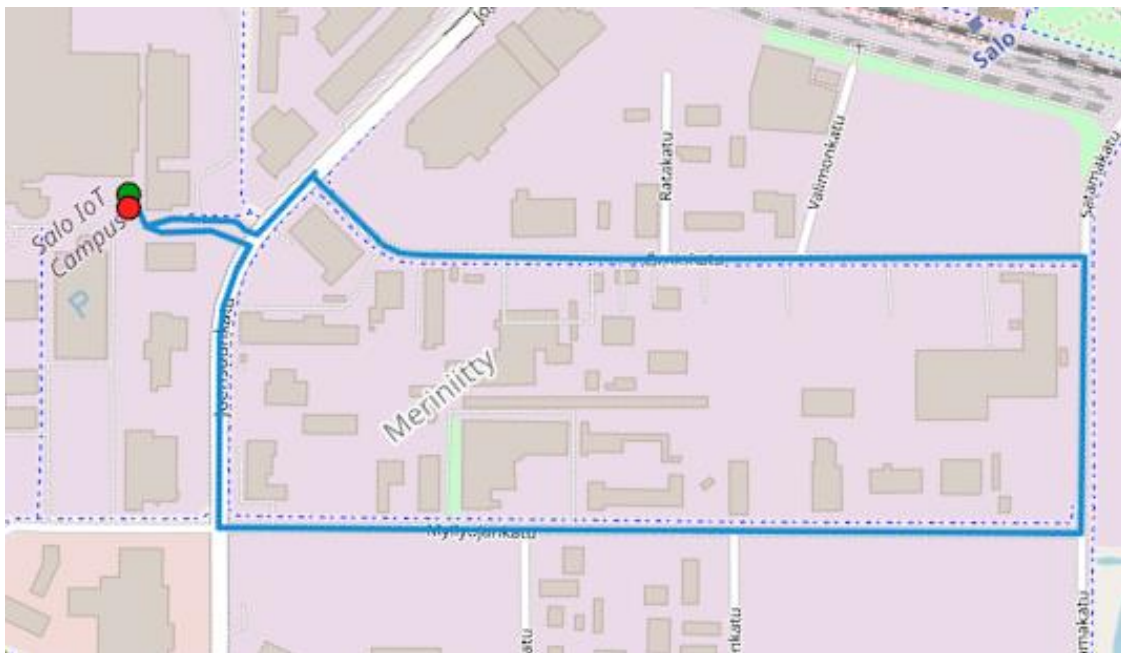


Figure 14 The industrial district survey route (Screenshot from Google Maps)

From the located 338 networks we derived the following results:

- 45 or 13.3% of the networks were unencrypted.
- 6 or 1.8% of the networks used WEP encryption
- 15 or 4.4% of the networks used WPA-TKIP encryption
- 272 or 80.5% of the networks used WPA2
 - 226 or 66.9% used WPA2-PSK encryption
 - 46 or 13.6% used WPA-Enterprise encryption

Further inspecting the unencrypted WEP and TKIP networks we can make a few observations about the networks and network devices. The most worrying findings are found among the five unencrypted networks and the six WEP networks. The unencrypted networks seem to originate from an Android smartphone, an IoT device, a printer, and two wireless routers. The IoT and printer devices are very worrisome since we are in an area

consisting largely of different size businesses and these devices could potentially provide entry for hackers into the larger company network. Four out of the six WEP networks have their SSIDs cloaked which would indicate that there is some attempt to make them more secure but are none the less still very much insecure and outdated. Looking into the WPA-TKIP networks it seems that 14 out of the 15 them originate from handheld PDA devices and only one from an actual wireless access point, making the situation somewhat better than with the unencrypted and WEP networks.

6.1.2. The city centre

For the second location in our survey we chose the city centre for its denser population and variety of housing, businesses, and administrative buildings. Along our 3.7 km route (Figure 15) we have the city hall, police station, school, the market square, apartment buildings and a variety of businesses from restaurants and nightclubs to flower shops. During our surveys in the city centre we located 1195 networks constituting 68.2% out of the total amount of networks found during our research and roughly translating into one detected access point on every 3.1 meters.

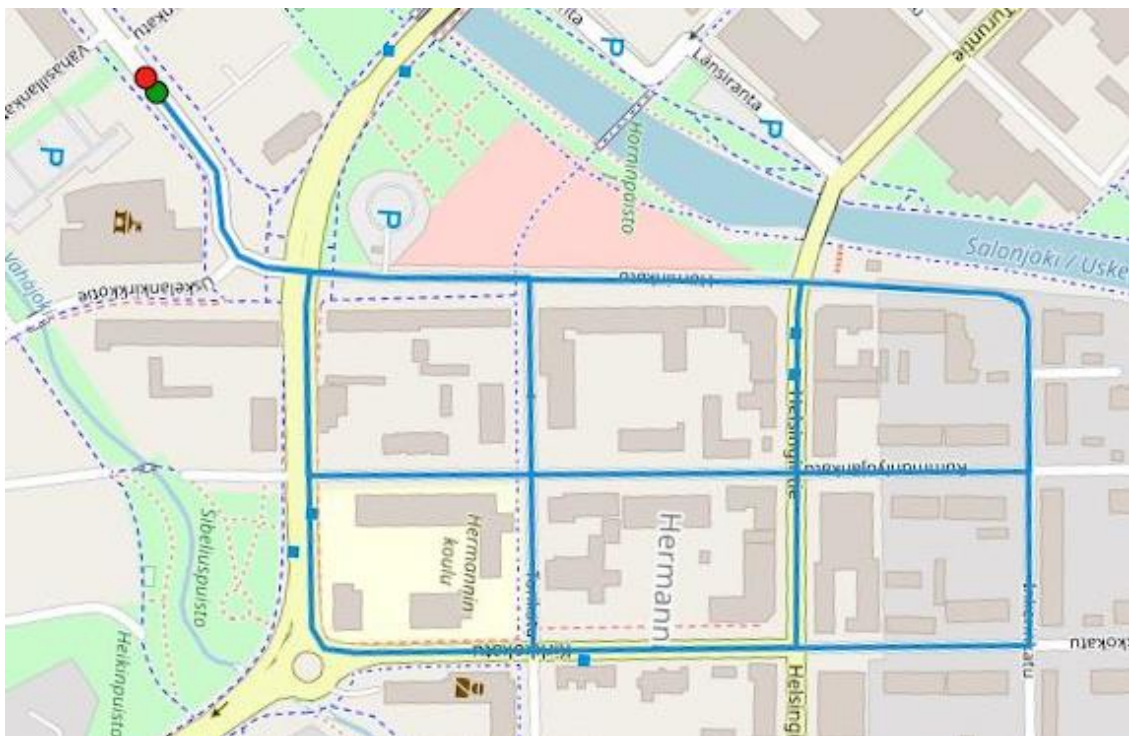


Figure 15 The city centre survey route (Screenshot from Google Maps)

From the located 1195 networks we derived the following results:

- 108 or 9.0% of the networks were unencrypted.
- 7 or 0.59% of the networks used WEP encryption
- 8 or 0.67% of the networks used WPA-TKIP encryption
- 1072 or 89.7% of the networks used WPA2
 - 995 or 83.3% used WPA2-PSK encryption
 - 77 or 6.4% used WPA-Enterprise encryption

After a deeper look into the unencrypted and insecure WEP and TKIP networks, we can make some more detailed observations. While looking into the unencrypted networks we see that half of them originate from Google Chromecast devices which generate a temporary WLAN network when they have not been connected to any other network. Among the other unencrypted networks, we again find printers, Android smartphones, mass storage devices, IoT devices, and a few wireless access points with default settings. As some of the more peculiar findings in the area we could mention an unencrypted geomatic surveying device, an IoT vacuum cleaner and a digital information screen. Again, the most concerning findings being the unencrypted wireless routers, printers, IoT devices, and mass storage devices.

The situation with WEP devices seems to be very much the same as in the industrial district. Only two of the WEP devices broadcast their network SSIDs indicating that there have been some efforts made to make them more secure. Based on the information about the found WPA-TKIP networks it would seem that the devices are mostly by the same manufacturer and have their default configurations on. This leads us to believe that the networks most likely originate from outdated devices that have been handed out by a local internet service provider at some point in time. The concerning matter about the TKIP networks is that based on the SSIDs, two of them belong to local businesses making them vulnerable to attacks.

6.1.3. The suburb

As the third location and route (Figure 16) we chose a less densely populated area representing a typical Finnish suburb with mostly single-family housing and apartment buildings outside of the immediate city centre. By surveying this area, we are aiming to get a picture of the current WLAN security practices in typical Finnish homes.



Figure 16 The suburb survey route (Screenshot from Google Maps)

Our chosen route ran 2.20km long and we were able to locate a total of 219 networks roughly giving a new access point on every 10 meters. These 219 networks constitute 12.5% of the total amount of located networks.

From the found 219 networks we derived the following results:

- 9 or 4.1% of the networks were unencrypted.
- 0 of the networks used WEP encryption
- 2 or 0.9% of the networks used WPA-TKIP encryption
- 208 or 95% of the networks used WPA2
 - 208 used WPA2-PSK encryption
 - 0 used WPA-Enterprise encryption

Based on our findings the situation with WLAN security seems to be in good form in the surveyed suburb area. There are no WEP encrypted devices and there are very few unencrypted devices. Most of the few unencrypted networks seem to again originate from Google Chromecast devices and dedicated guest networks based on their SSIDs. The only worrying unencrypted device is yet again a printer with WLAN capabilities. Moreover, we could locate only two WPA-TKIP networks originating from what seem like typical wireless routers.

Even though things seem to be in order in terms of encryption, some observations can still be made about the security practices in this area. A large majority of the found networks used default SSIDs and the usual pre-set channels 1, 6 and 11 which could indicate that the networks are using the pre-set configurations. Also noticeable in this area is the

common use of IoT security cameras in addition to the fair amount of Google devices, including the only Google Home Mesh network found during our survey.

6.1.4. Use of encryption protocols in the three locations

Here we will present comparisons of the use of different encryption protocols between the three surveyed locations. Table 3 shows the distribution of encryption protocol between the locations in precise numbers and Figure 17 present the encryption distribution in percentages.

Location	Unencrypted	WEP	WPA-TKIP	WPA2-PSK	WPA-Enterprise	Total
Industrial district	45	6	15	226	46	338
The city centre	108	7	8	995	77	1195
The suburb	9	0	2	208	0	219
Total	162	13	25	1429	123	1752

Table 3 Encryption distribution between locations

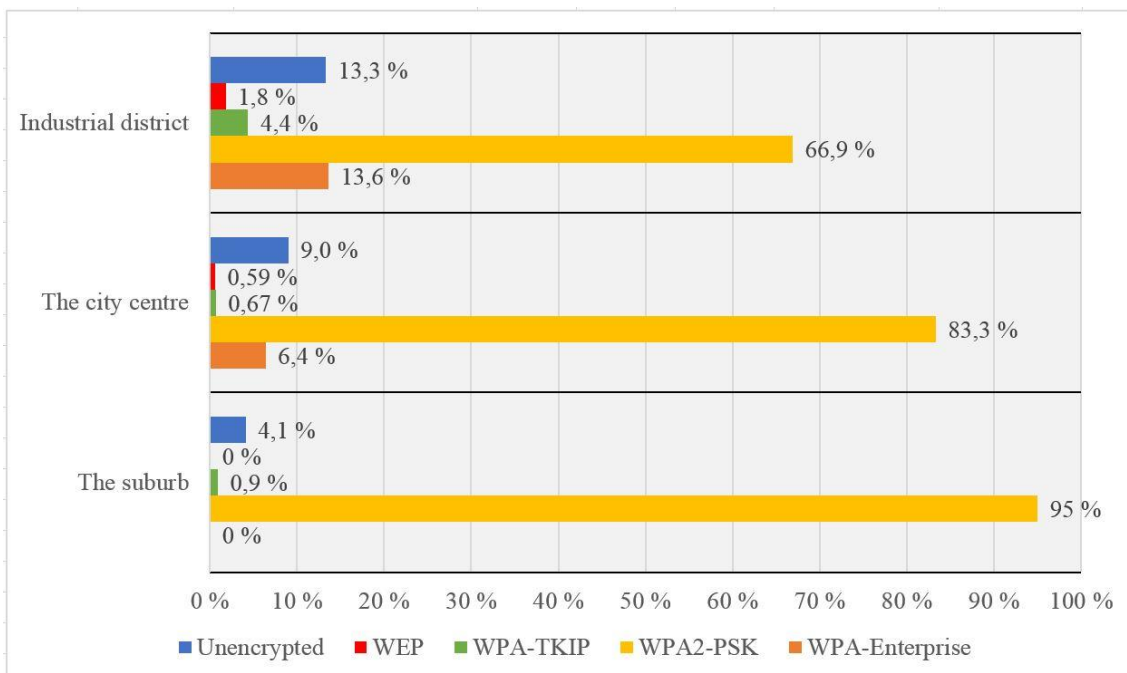


Figure 17 Encryption distribution between locations in percentages

When comparing the use of WPA2-PSK and WPA-Enterprise encryption with the use of WEP and WPA-TKIP encryption the situation looks optimistic. In each location, most networks have been secured with the stronger WPA2 encryption protocols and the use of the broken and deprecated encryption protocols is very low. The number of WPA-TKIP encrypted networks in the industrial district seems higher compared to the city centre and suburb due to the already mentioned 14 TKIP encrypted PDA devices.

The number of unencrypted networks, on the other hand, seems more alarming when compared with the use of WEP and WPA-TKIP encryption. In each location, we found more unencrypted networks than WEP and WPA-TKIP networks combined. However, we must remember that the numbers in Table 3 also includes the networks intentionally left open and unencrypted provided by different businesses and organisations, especially in the industrial district and city centre areas.

When strictly comparing the ratio between insecure and secure networks in the three different areas we get the following results. In table 4 we have categorised the unencrypted, WEP and WPA-TKIP networks as insecure and the WPA2-PSK and Enterprise networks as secure. In Figure 18 we present the same results in percentages.

Location	Unencrypted, WEP + TKIP	WPA2-PSK + WPA Enterprise	Total
Industrial district	66	272	338
The city centre	123	1072	1195
The suburb	11	208	219
Total	200	1552	1752

Table 4 Insecure and secure network distribution between locations

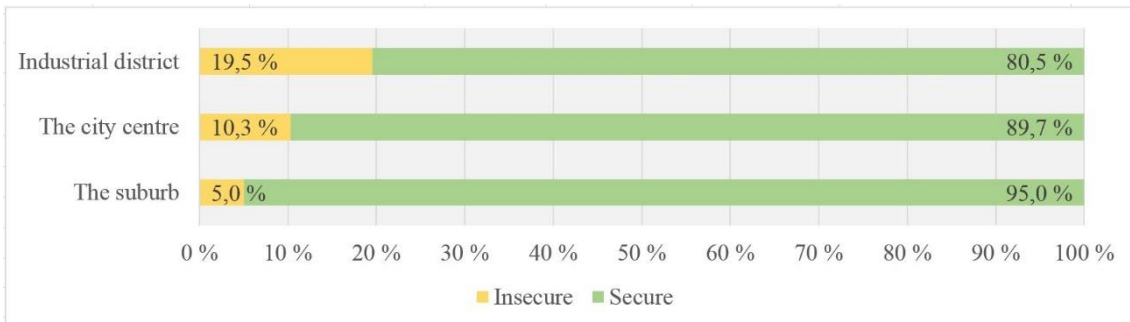


Figure 18 Insecure and secure network distribution between locations, in percentages

It seems that the industrial district has the most issues in WLAN security with almost a fifth of the found networks categorised as insecure. This finding is alarming since the area mostly comprises out of different sized businesses. However, we must again remember that the area has many businesses and organisations, many of which provide unencrypted guest and customer networks, which raises the number of insecure networks.

For this reason, in Figure 19 we present the results presented in Table 4 and Figure 18 with the unencrypted guest networks combined with the secure WPA2-PSK and Enterprise networks. As we can see, when we include the networks left intentionally open with secure networks the ratio between insecure and secure networks evens out to between 4.3% and 7.7% in the different locations. However, it should still be stated that making

this distinction between networks that have been intentionally and unintentionally left open and unencrypted does not mean that using an unencrypted wireless network is by no means safe even when it is provided by a trusted party and should always be used with caution.

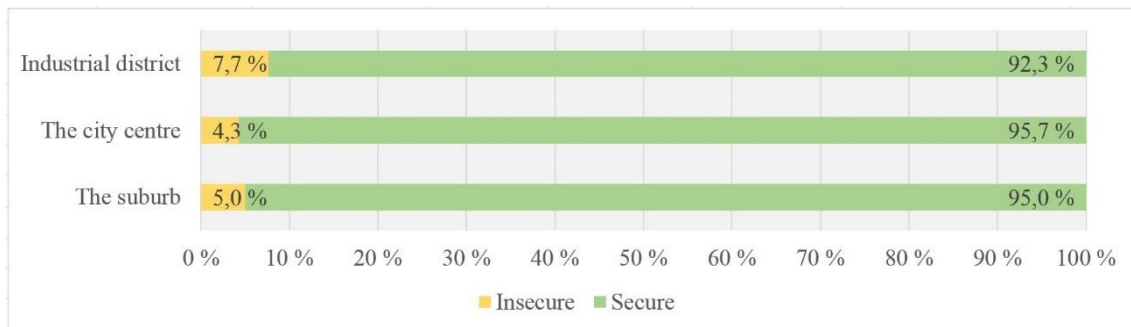


Figure 19 Insecure and secure network distribution when guest networks included in secure networks

6.2. The bigger picture of WLAN security practices

In this section, we present our findings based on the entire combined dataset collected from the three presented areas during our research to have a more complete picture about the state of WLAN security and security practices in the surveyed area. In addition to disclosing our finding considering the use of encryption protocols, we will be presenting our findings on the use of SSID cloaking and default SSIDs, the use of wireless channels, and the most popular device manufacturers

6.2.1. Encryption protocol use

When combined, our results show that during our research, we have located a total of 1752 networks. The total amount of insecure unencrypted, WEP and WPA-TKIP networks add up to a total of 200 networks representing 11.4% out of the 1752 networks leaving a total of 1552 or 88.6% networks categorised as secure WPA2 networks. Further dividing the located networks, we could derive the following results:

- 162 or 9.2% of the found networks were unencrypted.
- 13 or 0.7% of the found networks used WEP encryption
- 25 or 1.4% of the found networks used WPA-TKIP encryption
- 1552 or 88.6% of the found networks used WPA2
 - 1429 or 81.6% used personal PSK encryption
 - 123 or 7.0% used WPA-Enterprise encryption

Figure 20 presents the above information as a chart where the blue column represents the number of networks and the yellow column represents the corresponding percentage. Figure 21 presents the encryption distribution in percentages. Figure 22 presents the distribution between WPA networks using strictly WPA2-PSK encryption and networks using WPA-PSK mixed mode encryption accommodating both WPA-TKIP and WPA-AES devices.

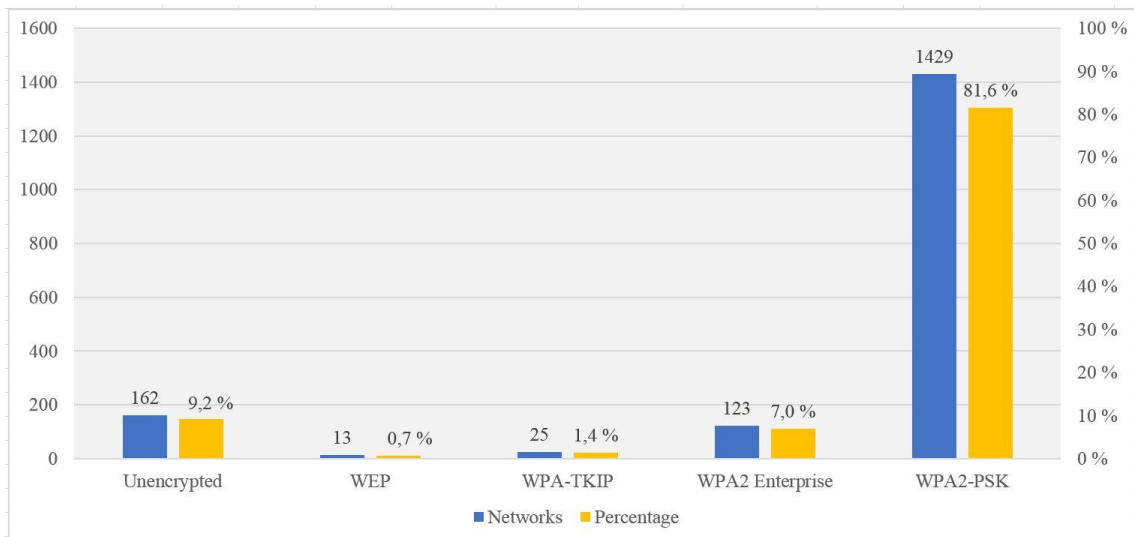


Figure 20 Encryption distribution

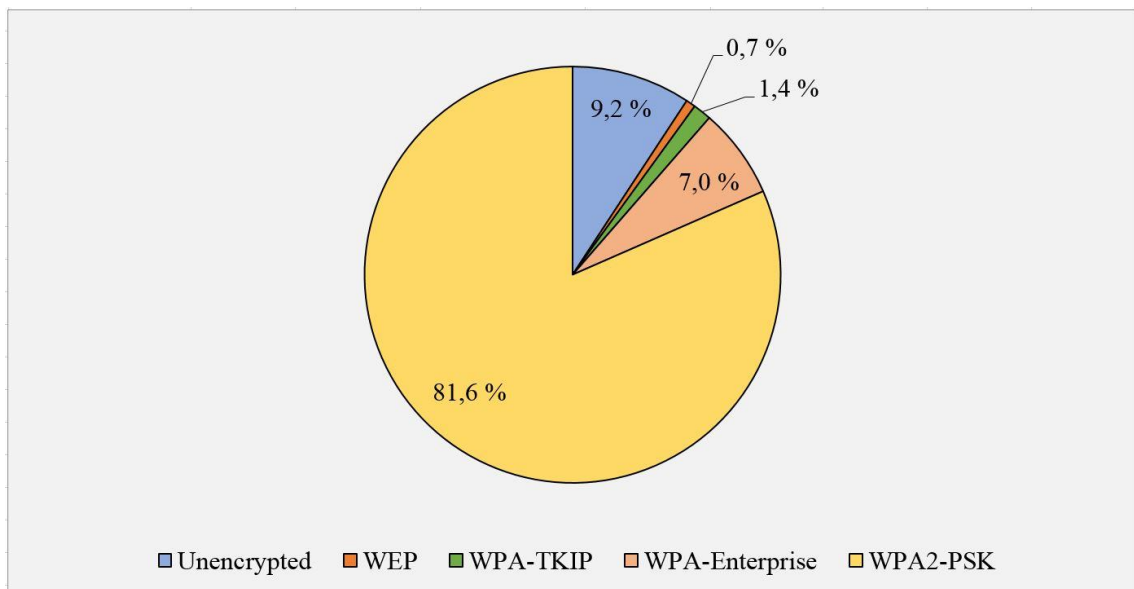


Figure 21 Encryption distribution in percentages

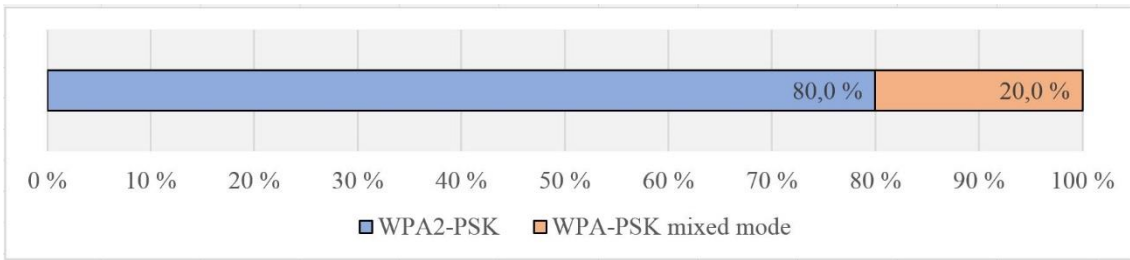


Figure 22 Ratio of WPA2-PSK and WPA-PSK mixed mode networks

When looking at the numbers of WEP and WPA-TKIP encrypted networks the situation seems quite optimistic when compared against the number of stronger WPA2-PSK and WPA-Enterprise encrypted networks. Still, even while the WEP and TKIP encrypted networks represent only a small fraction of the found networks, the number of unencrypted networks is concerning. Even when filtering out the intentionally unencrypted guest and customer networks the number of unencrypted networks (53 or 3%) is still higher than the number of WEP and WPA-TKIP networks combined.

Looking into Figure 23 where we have combined the insecure networks into one column and secure networks into another, we can see the ratio between insecure and secure networks better. From the figure we can see that the number of unencrypted networks (9.2%) is over four times greater than the combined amount of WEP and TKIP networks (2.2%) and is even taking over the number of WPA-Enterprise networks (7.0%). Furthermore, when looking at the distribution between the use of WPA2-PSK and WPA mixed mode networks presented in Figure 22, it seems that the use of the more insecure mixed mode option is four times lower than the more secure WPA2 only mode. Based on these realisations, it would seem that the security issues in WLAN secure are not necessarily to be found from the use of old and deprecated encryption protocols and have much more to do with the use of unencrypted networks.

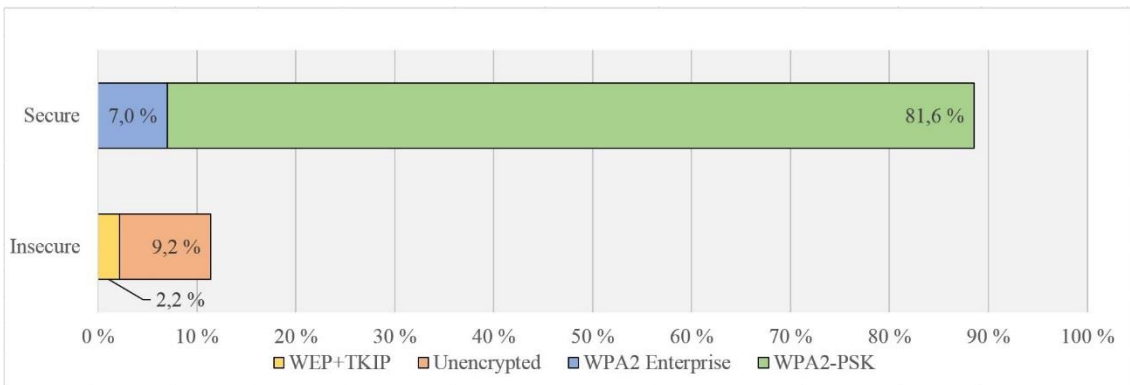


Figure 23 Ratio of insecure and secure networks

6.2.2. SSID security practices

Often wireless access points have an option for stopping the network SSID from being broadcast as well as an option for changing the network SSID. Although these two methods do not add to the initial security of the wireless network and cause extra work when setting up the network, they do have some perks to them. When you buy a new wireless access point it is pre-configured to use certain settings including a default SSID, channel, and passwords which can in the worst case be very weak. A known weak default password together with a pre-configured network SSID that reveals the device manufacturer and model can give an attacker all the needed information to launch an attack against the wireless network.

SSID cloaking can be easily circumvented with wireless network scanners which often can tell the wireless device manufacturer information based on the device MAC address and reveal the cloaked SSID by monitoring connecting client devices. Nevertheless, both cloaking and renaming the SSID are still recommended actions for any home or small office wireless network. By taking these simple measures wireless network device owners can keep the casual eavesdroppers at bay and add extra layers of privacy and security to their wireless network by making it less visible.

During our research we found a good example of how much information a poorly chosen and visible network SSID can reveal. Today many newer cars have a built-in WLAN access point. From our research data we could pinpoint a few of these access points. While processing the found networks and network SSIDs we noticed that one SSID contained a car manufacturer's name and possibly a car's licence plate number. As car owner information is public in Finland and is freely available online, we were able to find very detailed information about the car, as well as the car owners' personal information, by making a simple search with the licence plate number. This example shows well the benefits that changing the network SSID into a form that does not contain information about the wireless device or device owner and cloaking the network SSID can have.

Observing the use of SSID cloaking in the three surveyed areas, we got the following results:

- From the 338 networks located in the industrial district, 31 or 9.2% had cloaked SSIDs leaving 307 or 90.8% of the networks with visible SSIDs

- From the 1195 networks located in the city centre, 147 or 12.3% had cloaked SSIDs leaving 1048 or 87.7% of the networks with visible SSIDs
- From the 219 networks located in the suburb, 12 or 5.5% had cloaked SSIDs leaving 207 or 94.5% of the network with visible SSIDs

By observing the combined results we could see that from the total amount of 1752 located networks 190 or 10.8% had cloaked SSIDs leaving 1562 or 89.2% of the networks with visible SSIDs. Figure 24 presents the distribution in combined results and Figure 25 presents the distribution in the three surveyed areas.

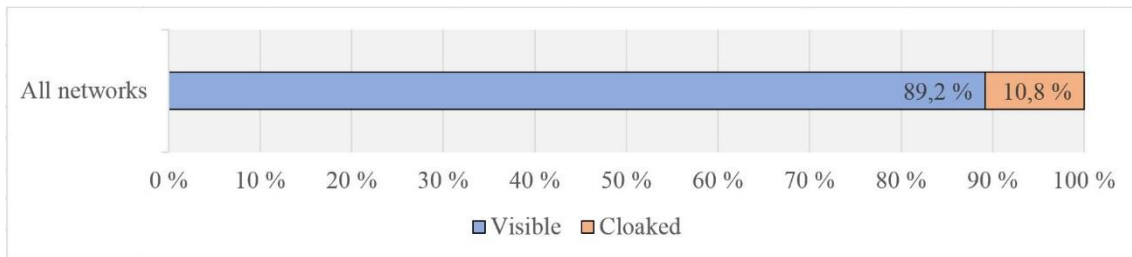


Figure 24 Distribution of visible and cloaked networks in combined results

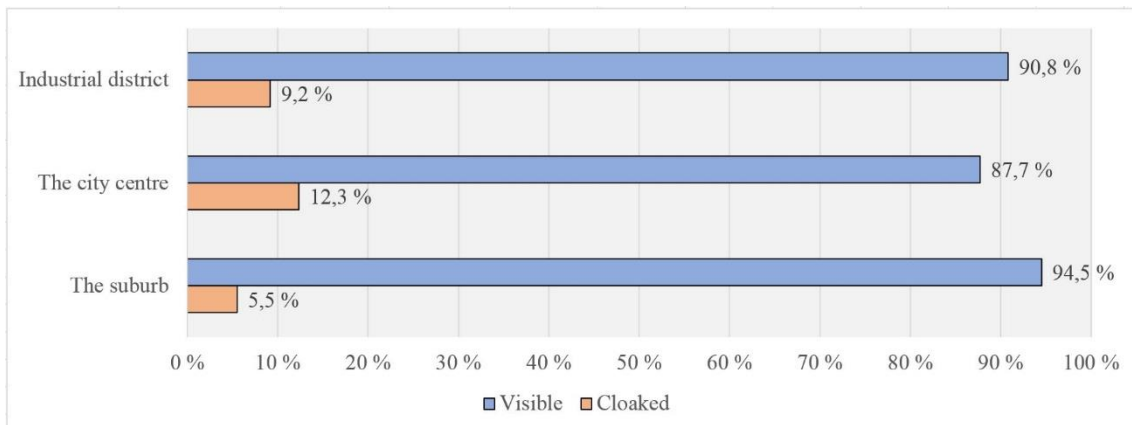


Figure 25 Distribution of visible and cloaked networks in the surveyed areas

Based on the presented results it seems that SSID cloaking is an infrequent habit in the surveyed area. Noticeable is that the use of insecure encryption protocols is more frequent in networks with cloaked SSIDs when comparing the use of encryption protocols between cloaked and visible networks, as shown in Figure 26. We found that out of the 190 cloaked networks 35 or 18.4% used insecure encryption protocols, whereas out of the 1562 networks with visible SSIDs 165 or 10.6% used insecure encryption protocols. This finding would indicate that at least some of the network owners know that they are using insecure encryption protocols and are making efforts to add security in their network by cloaking the SSID.

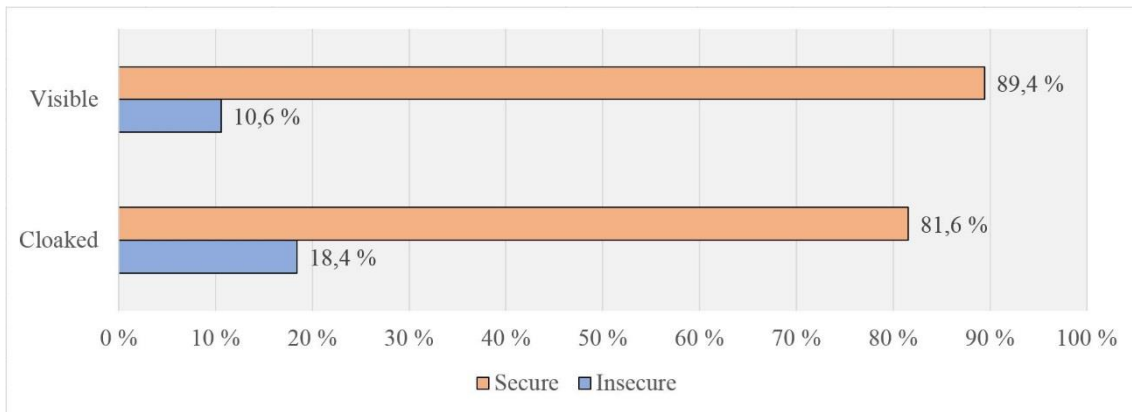


Figure 26 Encryption use in networks with cloaked and visible SSID

Just as SSID cloaking seems to be a very uncommon practice in the surveyed areas, the same can be said about altering the networks default SSID. According to our findings, from the total amount of 1752 located networks 905 or 51.7% used a pre-set default SSID. This leaves under half 847 or 48.3% of the networks with either altered or cloaked SSIDs, as seen in Figure 27.

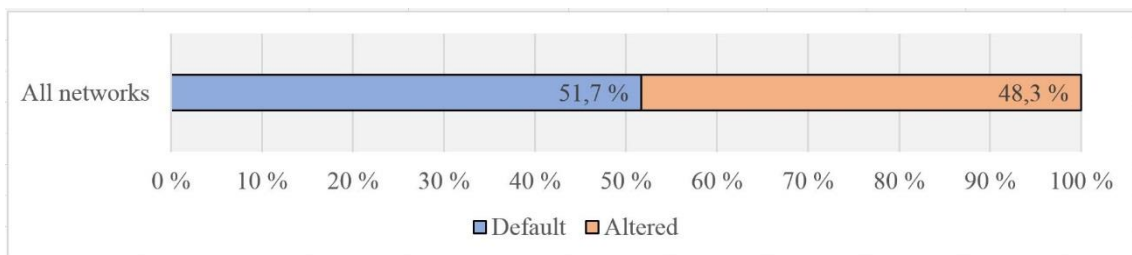


Figure 27 Percentage of networks with altered and default SSIDs

Looking into our data from the three surveyed areas we can conclude the following results:

- From the 338 networks located in the industrial district, 62 or 18.3% had default SSIDs, leaving 276 or 81.7% of the networks with altered SSIDs
- From the 1195 networks located in the city centre, 666 or 55.7% had default SSIDs, leaving 529 or 44.3% of the networks with altered SSIDs
- From the 219 networks located in the suburb, 177 or 80.8% had default SSIDs, leaving 42 or 19.2% of the network with altered SSIDs

It seems that in the city centre and suburb areas where there is much more housing the use of default SSIDs is more common than in the industrial district, as seen in Figure 28. The lower number of default SSIDs in the industrial district could be explained by the lack of housing and by the larger number of businesses and organisations that reside in

the area. It is also a common practice and only sensible for businesses to accordingly name their wireless networks to distinguish them from the neighbouring businesses wireless networks. Based on the presented information, it would seem that a large portion of private consumers do not alter their wireless network SSIDs by either cloaking or changing the pre-set SSID, despite it being a fairly simple task and for its benefits for security and privacy.

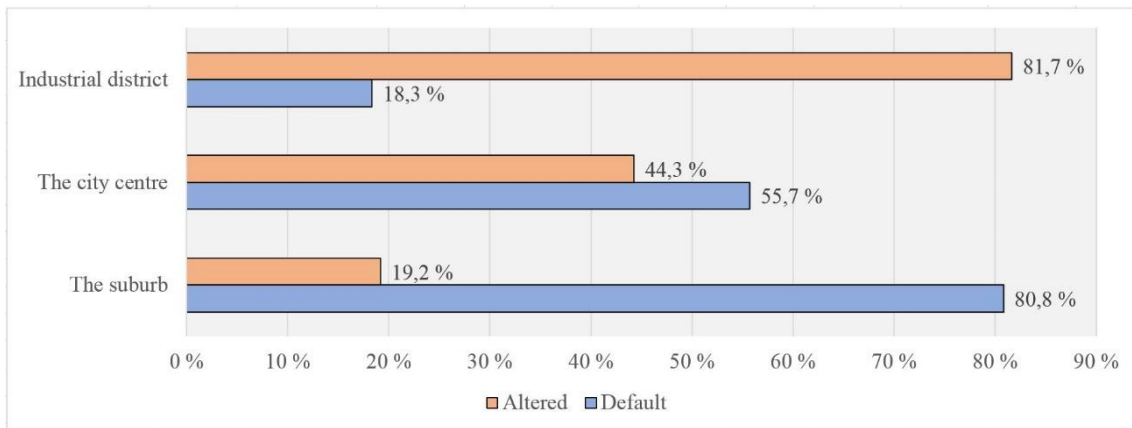


Figure 28 Percentage of altered and default network SSIDs in the surveyed areas

Furthermore, it was interesting to note that the majority out of the 905 networks with default SSIDs originate from devices manufactured by only three different companies. As seen in Figure 29 devices manufactured by Huawei, Telewell and TP-Link account for over 558 or 61.7% of the networks with default SSIDs, leaving the rest (347 or 38.3%) of the networks to other and unknown device manufacturers. This would leave us to assume that internet service providers in the area offer devices by the top three device manufacturers to their customers alongside with new internet connection plans.

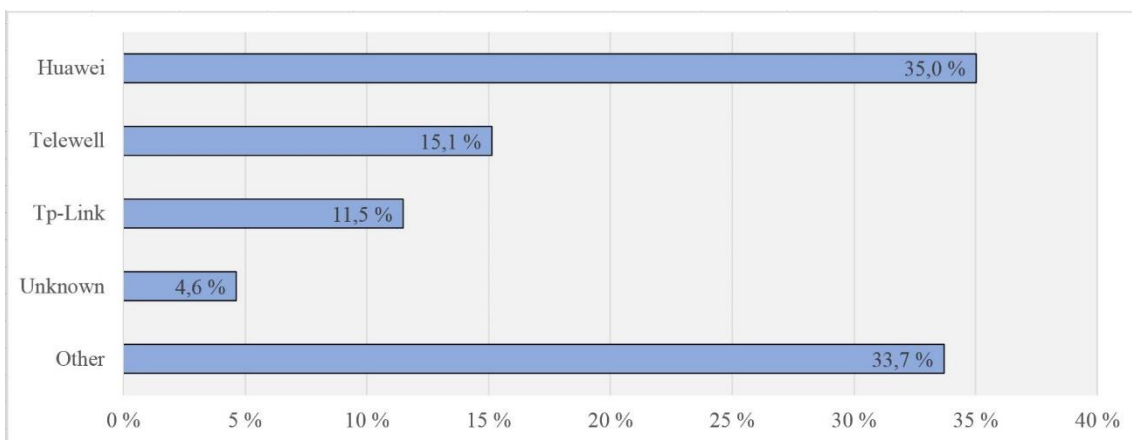


Figure 29 Device manufacturers with most default SSIDs

6.2.3. Popular device manufacturers

During our research, we found devices manufactured by 62 different manufacturers. From these 62 different manufacturers, the top six of the most popular manufacturers account for 70% of the found networks. Table 5 presents the most popular manufacturers in precise numbers and Figure 30 correspondingly presents the popular manufacturers in percentages.

Manufacturer	Industrial district	The city centre	The suburb	Total
Huawei	26	271	72	369
Ruckus Wireless	63	162	0	225
Cisco	96	112	2	210
Tp-Link	18	131	47	196
Telewell	4	127	15	146
Unknown	17	91	11	119
Hewlett-Packard	28	51	1	80
Other	86	250	71	407
Total	338	1195	219	1752

Table 5 Popular manufacturers

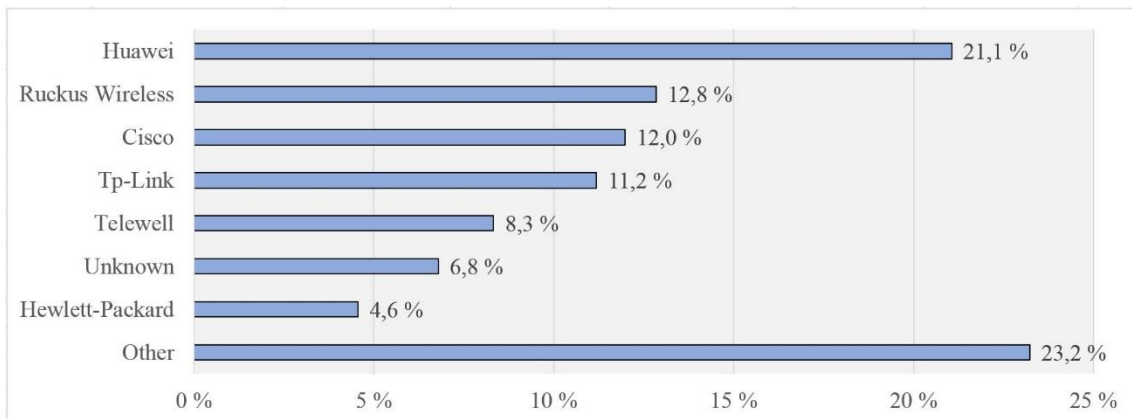


Figure 30 Popular manufacturers in percentages

When observing encryption protocol use among the most popular manufacturers, it was surprising to notice that only three of them had issues with the use of insecure encryption protocols. The most insecure of the most popular manufacturers was Cisco with 16 insecure networks, followed by Telewell with 4 and Huawei with 3 insecure networks.

Out of all the different manufacturers we encountered during our research, most issues with the use of insecure encryption protocols were found in devices manufactured by Google, Buffalo, Hon Hai Precision and Inteno.

- Out of the located 34 Google devices, 24 used insecure encryption. Some of the insecure networks originating from Google devices can be explained by the temporary unencrypted networks originating from the Google Chromecast devices.
- Out of the located 16 Buffalo devices, 8 used insecure encryption
- Out of the located 7 Hon Hai Precision devices, 4 used insecure encryption
- Out of the located 7 Inteno devices, 4 used insecure encryption

6.2.4. Popular wireless channels

Lastly in this section, we will present our findings considering the use of wireless channels in the surveyed area. On one hand channel selection has very little to do with WLAN security and rather has more of an effect on the performance of the wireless network, as discussed earlier in chapter three. Nevertheless, it is very common for the device manufacturer to pre-set their devices on one of the three non-overlapping channels 1, 6 or 11 on the 2.4 GHz band making them the most crowded channels. By observing the popularity of the pre-set wireless channels, we can get information about the network owners' proclivity to change the factory-set default settings.

In addition, by looking into the channel a wireless device resides on, one can get additional information about the device. For example, we noticed that two of the located devices used channel 14 which is not allowed in Europe and is instead only allowed in Japan. Based on this information we could locate the device manufacturer and eventually the model of the said device based on the network SSID. Furthermore, it is interesting to see the difference in popularity between the presumably more crowded 2.4 GHz band and less occupied 5 GHz band.

For reasons unknown to us Kismet has logged 48 out of the located 1752 networks to be operating on channel 0, leaving 1704 networks for us to analyse in this section. We suspect that this might occur when the logged network is far away, and the signal strength is too weak. Weak signal strength might lead to packet losses and to a situation where parts of the packet are lost in transmission, leading Kismet to log the channel as zero due to the lack of information. From the 1704 networks left for us to analyse, 1267 or 74.4% reside on the 2.4 GHz band. This leaves the rest 437 or 25.6% of the networks on the 5 GHz band, making the 2.4 GHz band nearly three times more popular than the 5 GHz band as seen in Figure 31.

As expected, the most popular channels on the 2.4 GHz band are the three non-overlapping channels 1, 6 and 11. Together they account for 72% of the 1267 networks. On the 5 GHz band, three of the most popular channels 36, 44 and 52 make up for 60.2% of the 437 networks, channel 36 being the most popular with a 35.5% share of all the networks. These findings are presented in Figures 32 and 33 in further detail.

In conclusion, it could be said that based on our findings our presuppositions about WLAN channel use are correct. The 2.4 GHz band is indeed much more crowded than the 5 GHz band and the pre-set channels 1, 6 and 11 are the most popular ones on the 2.4 GHz band. Moreover, based on the great popularity of only a few channels on both the 2.4 GHz and 5 GHz bands network owners are reluctant on changing the pre-set wireless channels. This coupled with the reluctance for altering the network SSIDs discussed in the previous section indicates that very few wireless network device owners are willing to change the factory-set default settings on their wireless network devices.

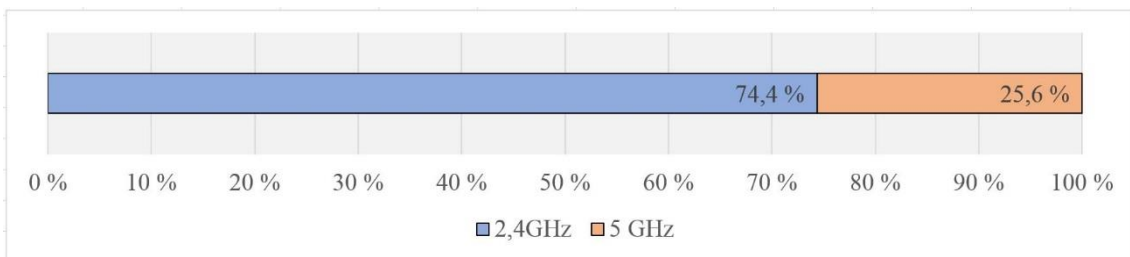


Figure 31 Distribution of networks operating on the 2.4 GHz and 5 GHz bands

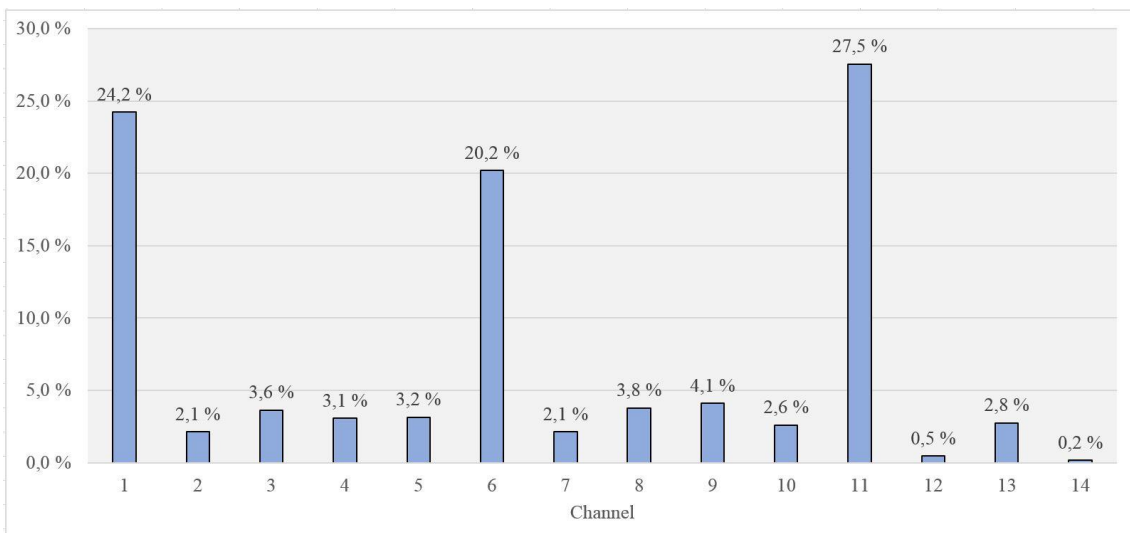


Figure 32 Channel popularity on the 2.4 GHz band

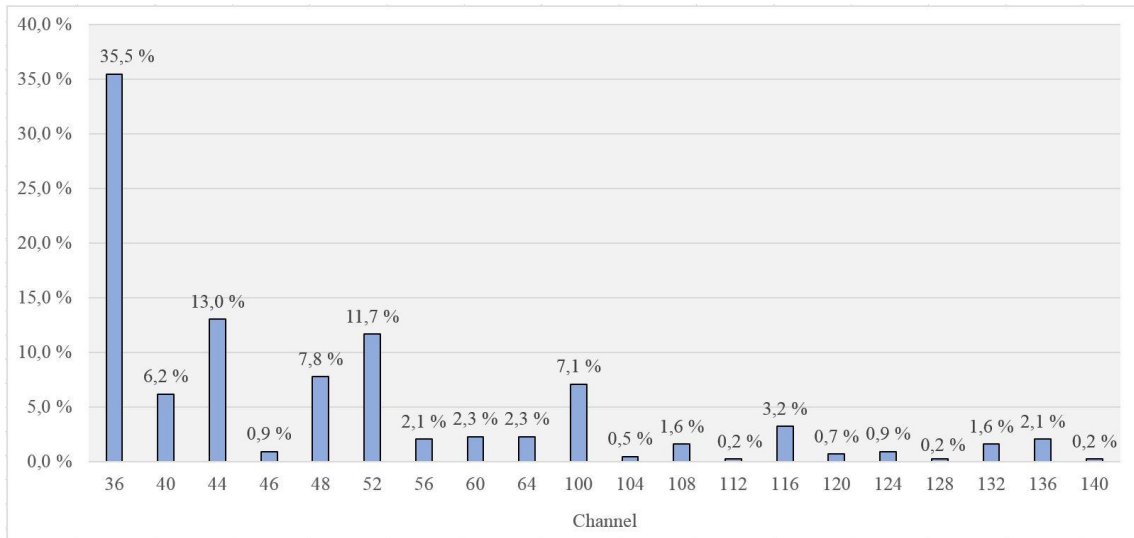


Figure 33 Channel popularity on the 5 GHz band

7. Conclusions

Despite WLAN networks having been part of our lives for over 20 years, discussions about the security of WLAN networks have recently become increasingly contemporary and predominant. The reasons for this increased attention toward WLAN security can be explained by current events. Firstly, the number of WLAN devices is on a steep rise because of the surge in popularity of various WLAN capable IoT and smart devices. Secondly, as discussed in chapter 4 of this work, incurable flaws have been found even in the most current widely available WLAN encryption protocol. Lastly, at the time of writing this work, we are on the verge of transitioning into the next generations of the 802.11 WLAN standard and the WPA encryption protocol presented in sections 3.7 and 4.4.

In this Master's Thesis, we sought to develop a process for surveying wireless local area networks and to survey the current state of WLAN security in Finland. The goal has been to develop a WLAN surveying process that would at the same time be efficient, scalable, and easily replicable. We wanted the survey process to be effective, but also practical in a manner that could be easily replicated and adapted in other environments. The purpose of the survey was to determine to what extent are obsolete and deprecated encryption protocols currently used in Finland. Furthermore, we wanted to find out in what state is WLAN security currently in Finland by observing the use of other WLAN security practices.

To fulfil the study objectives set for this Master's Thesis, the following research questions were set in the introduction section 1.2:

1. What is the current state of WLAN security in Finland?
 - a. What encryption protocols are in use today?
 - b. Are there large numbers of unencrypted networks in use?
 - c. How frequent is the use of other wireless network security practises?
 - d. Can we find any networks or devices supporting the newest 802.11 amendment and encryption protocol?
2. What is the most effective way to survey wireless networks?
 - a. What kind of hardware and software is needed to effectively survey wireless networks?

- b. How can we develop the surveying process so that it can be easily replicated and scaled to larger environments?
- c. What are the possible legal, regulatory, and ethical constraints for surveying wireless networks?

To be able to evaluate our success in answering the set research questions and set thesis objectives in the chronological order of this work, we must start our evaluation from question two *What is the most effective way to survey wireless networks?*. Based on the discussions presented in chapter 5 and the successful WLAN surveying results presented in chapter 6 it is safe to say that sufficient answers to the presented research question have been found. The wireless network surveys conducted during this work have been done by the means of wardriving. Wardriving is a passive wireless network scanning method used for locating wireless networks and network devices in a certain area. A more in-depth discussion about the wardriving process has been presented in section 5.1.

To accommodate the requirements presented in questions 2a and 2b only freely available software and off-the-shelf hardware were used during the WLAN surveying and following data sampling processes. Detailed descriptions of the used software and hardware can be found in sections 5.2.1, 5.2.2 and 5.3. The answer to questions about the legitimacy and ethicality of wardriving presented in question 2c have been presented in sections 5.4 and 5.5. In section 5.4, we found wardriving as a research method to be legitimate in the eyes of the Finnish criminal law and the EU General Data Protection Regulation. In section 5.5 the ethicality of wardriving was examined through the ethical theories of utilitarianism and virtue ethics. In both cases, wardriving was found to be ethical in the context of conducting legitimate research.

To answer the main research question *What is the current state of WLAN security in Finland?* a survey of WLAN networks was conducted in a middle-sized Finnish city. Based on the findings presented in chapter 6 it would seem that WLAN security in Finland is in a relatively good state when considering encryption. During our survey, we located a total of 1752 networks and out of those networks only 13 (0.7%) used the broken WEP encryption and 25 (1.4%) used WPA-TKIP encryption. When combined, only 38 (2.2%) of the located networks were found to be using the broken and deprecated encryption protocols. From the located networks 162 (9.2%) had no encryption enabled. A clear majority of 1552 (88.6%) used the strongest widely available WPA2 encryption protocols. Based

on our findings it would seem that the issues of WLAN security do not necessarily lie in the use of old and deprecated encryption protocols, as it would seem that the number unencrypted networks is four times over that of WEP and WPA-TKIP networks.

A comparison with the latest similar study conducted in Europe would also seem to reaffirm our presented conclusion. It would seem that the use of deprecated encryption protocols is lesser in Finland whereas the number of unencrypted networks is higher. In their 2019 study conducted in Varna Bulgaria, Valchanov, Edikyan and Aleksieva [111] analysed 11534 wireless networks and found that 1% of the located wireless networks used WEP encryption, 6% used WPA-TKIP encryption and 7% were unencrypted. The number of WPA2 encrypted networks seems to be in line with our findings with 86% portion of the located networks.

To answer the question 1c we looked into the wireless network device owner's proclivity for altering their wireless network SSID and wireless channel. The survey results show that out of the total amount of 1752 located networks 190 (10.8%) had cloaked SSIDs leaving 1562 (89.2%) of the networks with visible SSIDs. Furthermore, 905 (51.7%) of the located networks used a pre-set default SSID and 847 (48.3%) had either an altered or a cloaked SSID. Our results about the use of wireless channels were very much in line with our expectations. On the 2.4 GHz band the three non-overlapping channels 1, 6 and 11 together account for 72% of the analysed networks. On the 5 GHz band three of the most popular channels 36, 44 and 52 make up for 60.2% of the analysed networks.

From the presented results we can conclude that wireless network device owners are not inclined to alter their devices factory-set default settings. When further looking into the networks with cloaked SSIDs, we found out that out of the 190 networks that had a cloaked SSID nearly a fifth (35 or 18.4 %) used insecure encryption protocols. Whereas out of the 1562 networks with visible SSIDs just a tenth (165 or 10.6%) used insecure encryption protocols. This finding could indicate that those network device owners who are using insecure encryption protocols are more inclined to add security to their network by altering the wireless device default settings.

Based on the presented results and discussion it could be concluded that this thesis has successfully fulfilled its set objective of surveying the current state of WLAN security in Finland. We have successfully surveyed the use of insecure encryption protocols and the use of other wireless network security practices. In addition, it has been shown that the

constructed Wardriving survey process can be easily repeated with freely available software and off-the-shelf hardware. However, we were not able to find answers to question 1d concerning the newest 802.11ax amendment and WPA3 protocol. This might be because of the low number of 802.11ax and WPA3 capable devices available in the consumer market at the moment or because Kismet cannot yet recognise the newest WLAN devices.

7.1. Potential future research

The presented study leaves many possibilities for future research. The next logical step would be to further improve on the used surveying software, hardware, practices and to upscale the research to larger environments. By enhancing the surveying process to be more efficient and scaling up to larger areas, it would be possible to have more accurate and diverse results. It would also be beneficial to use a separate WLAN adapter for the 2.4 and 5 GHz bands and use more powerful antennas on the adapters. Changing the software from Kismet to for example Acrylic Wi-Fi could provide more precise information about the surveyed wireless networks, such as the used 802.11 standard. At the time of writing this work, Kismet will only list the surveyed wireless networks as either 802.11b or undefined.

On the other hand, because Kismet can be installed on any Linux based operating system it would be interesting to produce a more portable device for wardriving by using a small single-board computer such as the Raspberry Pi. These smaller-scale devices could also be left to survey different location for an indefinite time to follow the changes in WLAN network activity in the chosen area. A smaller device could also be attached to other means of transport such as drones or could be easily kept in a car and used whenever needed.

There is also potential for follow up studies in the already surveyed areas. It would be interesting to follow up on the developments in encryption protocol use and at what pace the newer protocols are being adopted. Changing the location would open up possibilities to upscale the surveyed area and to conduct comparisons between two cities. The discussion about the ethics of wardriving also presents a potential for future studies as very little has been written on the topic and only a brief discussion on the topic was possible in the confines of this work. It would seem that the opportunities for future research on the topic of wireless network security are limited only by one's imagination.

References

- [1] W. Lemstra, V. Hayes, and J. Groenewegen, *The Innovation Journey of Wi-Fi: The Road to Global Success*, 1st ed. Cambridge, United Kingdom: Cambridge University Press, 2011.
- [2] D. A. Westcott and D. D. Coleman, *Certified Wireless Network Administrator: Official Study Guide*, 4th ed. Indianapolis, USA: John Wiley & Sons Inc, 2014.
- [3] Abiresearch, “Wi-Fi Celebrates 20 Years with More Than 20 Billion Anticipated Device Shipments over the Next Six Years,” *Abiresearch*, 2019. [Online]. Available: <https://www.abiresearch.com/press/wi-fi-celebrates-20-years-more-20-billion-anticipated-device-shipments-over-next-six-years/>. [Accessed: 08-Nov-2019].
- [4] Cisco Systems Inc, “Cisco Visual Networking Index: Forecast and Trends, 2017–2022 White Paper,” *Cisco Visual Networking Index*. Cisco Systems Inc, 2019.
- [5] Businesswire, “Smart Home Will Drive Third Wave in Wireless Home Evolution Strategy Analytics,” 2019. [Online]. Available: <https://www.businesswire.com/news/home/20190807005530/en/Smart-Home-Drive-Wave-Wireless-Home-Evolution>. [Accessed: 07-Nov-2019].
- [6] N. Abramson, “Development of the ALOHANET,” *IEEE Transactions on Information Theory*, vol. 31, no. 2, pp. 119–123, 1985.
- [7] K. J. Negus and A. Petrick, “History of wireless local area networks (WLANs) in the unlicensed bands,” *INFO*, vol. 11, no. 5, pp. 36–56, 2009.
- [8] IEEE, “IEEE Standard for Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications,” *IEEE Std 802.11-1997*. pp. 1–445, 1997.
- [9] P. Chandra, D. Bensky, T. Bradley, C. Hurley, S. Rackley, J. Rittinghouse, J. Ransome, T. Stapko, G. Stefanek, F. Thorton, C. Lanthem, and J. Wilson, *Wireless Security: Know It All*, 1st ed. Burlington, USA: Elsevier Science & Technology Books, 2008.
- [10] J. McNulty and M. Marcus, “Amendment of the rules to authorize spread spectrum and other wideband emissions in the Public Safety and Industrial, Scientific,

- Medical Bands.” Federal Communications Commission FCC, 1985.
- [11] F. F. Kuo, “Computer Networks - the ALOHA System,” *Journal of Research of the National Bureau of Standards*, vol. 86, no. 6, p. 591, 1981.
- [12] L. G. Roberts, “ALOHA packet system with and without slots and capture,” *ACM SIGCOMM Computer Communication Review*, vol. 5, no. 2, pp. 28–42, Apr. 1975.
- [13] C. E. Spurgeon and J. Zimmerman, *Ethernet: The definitive guide*, 2nd ed. Berlin, Germany: O’Reilly Media Inc, 2014.
- [14] Ionos Digital Guide, “CSMA/CA What is CSMA with Collision Avoidance?,” 2019. [Online]. Available: <https://www.ionos.com/digitalguide/server/know-how/csmaca-carrier-sense-multiple-access-with-collision-avoidance/>. [Accessed: 28-Aug-2019].
- [15] IEEE, “History of IEEE.” [Online]. Available: <https://www.ieee.org/about/ieee-history.html#the-societies-converge-and-merge>. [Accessed: 11-Nov-2019].
- [16] IEEE, “IEEE Mission & Vision,” 2019. [Online]. Available: https://www.ieee.org/about/vision-mission.html?WT.mc_id=lp_ab_mav. [Accessed: 11-Nov-2019].
- [17] Wi-Fi Alliance, “Wi-Fi Alliance introduces Wi-Fi 6,” 2018. [Online]. Available: <https://www.wi-fi.org/news-events/newsroom/wi-fi-alliance-introduces-wi-fi-6>. [Accessed: 07-Jan-2020].
- [18] Wi-Fi Alliance, “Wi-Fi Certified 6 delivers new Wi-Fi era,” 2019. [Online]. Available: <https://www.wi-fi.org/news-events/newsroom/wi-fi-certified-6-delivers-new-wi-fi-era>. [Accessed: 08-Jan-2020].
- [19] J. Lee, “What is WI-FI CERTIFIED and the Wi-Fi Alliance?,” *CommScope Blog*, 2018. [Online]. Available: <https://www.commscope.com/blog/2018/what-is-wi-fi-certified-and-the-wi-fi-alliance/>. [Accessed: 08-Jan-2020].
- [20] G. Hiertz, D. Denteneer, L. Stibor, Y. Zang, X. Costa, and B. Walke, “The IEEE 802.11 universe,” *IEEE Communications Magazine*, vol. 48, no. 1, pp. 62–70, Jan. 2010.

- [21] IEEE, “802.11a-1999 - IEEE Standard for Telecommunications and Information Exchange Between Systems - LAN/MAN Specific Requirements - Part 11: Wireless Medium Access Control (MAC) and physical layer (PHY) specifications: High Speed Physical Layer in the 5 GHz ba,” *IEEE Std 802.11a-1999*. pp. 1–102, 1999.
- [22] IEEE, “802.11b-1999 - IEEE Standard for Information Technology - Telecommunications and information exchange between systems - Local and Metropolitan networks - Specific requirements - Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) sp,” *IEEE Std 802.11b-1999*. pp. 1–96, 2000.
- [23] J. Ross, *The book of Wireless*, 2nd ed. San Francisco, USA: No Starch Press Inc, 2008.
- [24] P. Nicopolitidis, M. S. Obaidat, G. I. Papadimitriou, and A. S. Pomportsis, *Wireless Networks*. Chichester, United Kingdom: John Wiley & Sons Ltd, 2002.
- [25] J. Raynolds, *Going Wi-Fi: A Practical Guide to Planning and Building an 802.11 Network*, 1st ed. Boca Raton, USA: Routledge CRC Press LLC, 2003.
- [26] IEEE, “802.11g-2003 - IEEE Standard for Information technology - Local and metropolitan area networks - Specific requirements - Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications: Further Higher Data Rate Extension in the 2,” *IEEE Std 802.11g-2003*. pp. 1–104, 2003.
- [27] D. Vassis, G. Kormentzas, A. Rouskas, and I. Maglogiannis, “The IEEE 802.11g standard for high data rate WLANs,” *IEEE Network*, vol. 19, no. 3, pp. 21–26, May 2005.
- [28] M. Gast, *802.11 Wireless Networks: The Definitive Guide*, 2nd ed. Berlin, Germany: O’Reilly Media Inc, 2005.
- [29] P. Roshan and L. Jonathan, *802.11 Wireless LAN fundamentals*, 1st ed. Indianapolis, USA: Cisco Press, 2003.
- [30] Y. Xiao, “IEEE 802.11n: Enhancements for higher throughput in wireless LANs,” *IEEE Wireless Communications*, vol. 12, no. 6, pp. 82–91, 2005.

- [31] E. Perahia and R. Stacey, *Next Generation Wireless LANs*, 2nd ed. Cambridge, United Kingdom: Cambridge University Press, 2013.
- [32] S. Banerji and R. S. Chowdhury, "On IEEE 802.11: Wireless Lan Technology," *International Journal of Mobile Network Communications & Telematics*, vol. 3, no. 4, pp. 45–64, 2013.
- [33] B. O'Brien, "Review: Speedy next-gen Wi-Fi equipment that works now," 2007. [Online]. Available: <https://www.computerworld.com/article/2543814/review--speedy-next-gen-wi-fi-equipment-that-works-now.html>. [Accessed: 03-Oct-2019].
- [34] D. Haskin, "FAQ: 802.11n wireless networking," 2007. [Online]. Available: <https://www.computerworld.com/article/2544290/faq--802-11n-wireless-networking.html>. [Accessed: 03-Oct-2019].
- [35] T. Paul and T. Ogunfrunmiri, "Wireless LAN Comes of Age: Understanding the IEEE 802.11n Amendment," *IEEE Circuits and Systems Magazine*, vol. 8, no. 1, pp. 28–54, 2008.
- [36] A. Holt and C. Huang, *802.11 Wireless Networks*, 1st ed. London, United Kingdom: Springer London, 2010.
- [37] Cisco Systems Inc, "802 .11n : The Standard Revealed," *Cisco Public White Paper*. Cisco Systems Inc, 2009.
- [38] L. Verma, M. Fakharzadeh, and S. Choi, "Wifi on steroids: 802.11AC and 802.11AD," *IEEE Wireless Communications*, vol. 20, no. 6, pp. 30–35, 2013.
- [39] Cisco Systems Inc, "802 . 11ac : The Fifth Generation of Wi-Fi," *Cisco Public White Paper*. Cisco Systems Inc, 2014.
- [40] A. Nagy, "802.11ac Channel Planning," 2013. [Online]. Available: <http://www.revolutionwifi.net/revolutionwifi/2013/03/80211ac-channel-planning.html>. [Accessed: 24-Oct-2019].
- [41] O. Bejarano, E. Knightly, and M. Park, "IEEE 802.11ac: from channelization to multi-user MIMO," *IEEE Communications Magazine*, vol. 51, no. 10, pp. 84–90, 2013.

- [42] M. Gast, *802.11ac: A Survival Guide*, 1st ed. Sebastopol, USA: O'Reilly Media Inc, 2013.
- [43] J. Lendino, "What is 802.11ac Wi-Fi, and how much faster than 802.11n is it?," 2016. [Online]. Available: <https://www.extremetech.com/computing/160837-what-is-802-11ac-and-how-much-faster-than-802-11n-is-it>. [Accessed: 16-Oct-2019].
- [44] C. Jeffrey, "Wi-Fi Alliance launches 802.11ax certification program - TechSpot," 2019. [Online]. Available: <https://www.techspot.com/news/81917-wi-fi-alliance-launches-80211ax-certification-program.html>. [Accessed: 17-Oct-2019].
- [45] A. Irei, "802.11ax release date: Here's what has to happen first," 2019. [Online]. Available: <https://searchnetworking.techtarget.com/infographic/80211ax-release-date-Heres-what-has-to-happen-first>. [Accessed: 17-Oct-2019].
- [46] D.-J. Deng, K.-C. Chen, and R.-S. Cheng, "IEEE 802.11ax: Next generation wireless local area networks," in *10th International Conference on Heterogeneous Networking for Quality, Reliability, Security and Robustness*, Rhodes, Greece, 2014, pp. 77–82.
- [47] E. Khorov, A. Kiryanov, A. Lyakhov, and G. Bianchi, "A tutorial on IEEE 802.11ax high efficiency WLANs," *IEEE Communications Surveys and Tutorials*, vol. 21, no. 1, pp. 197–216, 2019.
- [48] D. Coleman, P. Correl, and A. Gates, "802.11ax Tech Brief," *Aerohive Networks White Paper*. 2018.
- [49] H. Yang, D.-J. Deng, and K.-C. Chen, "Performance Analysis of IEEE 802.11ax UL OFDMA-Based Random Access Mechanism," in *GLOBECOM 2017 - 2017 IEEE Global Communications Conference*, Singapore, 2017, pp. 1–6.
- [50] Huawei Technologies, "Wi-Fi 6 (802.11ax) Technology White Paper," *Huawei Technologies White Paper*. Huawei Technologies Co. Ltd, 2018.
- [51] Aruba Networks, "802.11ax," *Aruba Networks White Paper*. Aruba networks, 2018.
- [52] D. Coleman, "How Does BSS Coloring Work in 802.11ax? - Aerohive Blog

- Aerohive Blog,” 2018. [Online]. Available: <https://blog.aerohive.com/how-does-bss-coloring-work-in-802-11ax/>. [Accessed: 24-Oct-2019].
- [53] N. P. Smart, *Cryptography Made Simple*, 1st ed. Cham, Switzerland: Springer International Publishing, 2016.
- [54] W. Stallings, *Cryptography and Network Security Principles and Practice*, 6th ed. San Francisco, USA: Pearson Education, Inc, 2014.
- [55] S. Y. Yan, *Cybercryptography: Applicable Cryptography for Cyberspace Security*, 1st ed. Cham, Switzerland: Springer International Publishing, 2019.
- [56] D. D. Coleman, D. A. Westcott, B. Harkins, and S. Jackman, *CWSP - Certified Wireless Security Professional - Official Study Guide*, 1st ed. Indianapolis, USA: John Wiley & Sons Inc, 2010.
- [57] W. Diffie and M. Hellman, “New directions in cryptography,” *IEEE Transactions on Information Theory*, vol. 22, no. 6, pp. 644–654, Nov. 1976.
- [58] R. L. Rivest, A. Shamir, and L. Adleman, “A Method for Obtaining Digital Signatures and Public-Key Components.,” *Communications of the ACM*, vol. 21, no. 2, pp. 120–126, 1978.
- [59] T. J. Shimeall and J. M. Spring, *Introduction to Information Security - A Strategic-Based Approach*, 1st ed. Waltham, USA: Syngress Publishing Inc, 2014.
- [60] IEEE, “802.11i-2004 - IEEE Standard for information technology- Telecommunications and information exchange between systems-Local and metropolitan area networks-Specific requirements-Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) speci,” *IEEE Std 802.11i-2004*. pp. 1–190, 2004.
- [61] Wi-Fi Alliance, “Wi-Fi Alliance introduces Wi-Fi CERTIFIED WPA3 security,” 2018. [Online]. Available: <https://www.wi-fi.org/news-events/newsroom/wi-fi-alliance-introduces-wi-fi-certified-wpa3-security>. [Accessed: 07-Mar-2020].
- [62] IEEE, “IEEE Standard for Information technology - Telecommunications and information exchange between systems Local and metropolitan area networks— Specific requirements - Part 11: Wireless LAN Medium Access Control (MAC)

- and Physical Layer (PHY) Specifications,” *IEEE Std 802.11-2016 (Revision of IEEE Std 802.11-2012)*. pp. 1–3534, 2016.
- [63] L. Stošić and M. Bogdanovic, “RC4 stream cipher and possible attacks on WEP,” *International Journal of Advanced Computer Science and Applications*, vol. 3, no. 3, 2012.
- [64] J. R. Walker, “Wireless LANs Unsafe at any key size; An analysis of the WEP encapsulation,” *IEEE 802.11-00/362*, pp. 1–9, 2000.
- [65] N. Borisov, I. Goldberg, and D. Wagner, “Intercepting mobile communications: The Insecurity of 802.11,” in *Proceedings of the 7th annual international conference on Mobile computing and networking - MobiCom '01*, New York, USA, 2001, vol. 25, no. 9, pp. 180–189.
- [66] S. Fluhrer, I. Mantin, and A. Shamir, “Weaknesses in the Key Scheduling Algorithm of RC4,” in *Lecture Notes in Computer Science*, vol. 2259, 2001, pp. 1–24.
- [67] H. F. Tipton and M. Krause, *Information Security Management Handbook, Sixth Edition, Volume 2*, 6th ed. Boca Raton, USA: Auerbach Publications, 2008.
- [68] A. Stubblefield, J. Ioannidis, and A. Rubin, “Using the Fluhrer, Mantin, and Shamir Attack to Break WEP,” in *Proceedings of the 2002 Network and Distributed Systems Security Symposium*, San Diego, USA, 2002, pp. 1–11.
- [69] E. Tews, R.-P. Weinmann, and A. Pyshkin, “Breaking 104 Bit WEP in less than 60 seconds,” in *IACR Cryptology ePrint Archive*, 2007, pp. 188–202.
- [70] Wigle.net, “Statistics,” 2020. [Online]. Available: <https://wagle.net/enc-large.html>. [Accessed: 18-Feb-2020].
- [71] Wi-Fi Alliance, “Wi-Fi Protected Access Security Sees Strong Adoption,” 2004. [Online]. Available: <https://www.wi-fi.org/news-events/newsroom/wi-fi-protected-access-security-sees-strong-adoption>. [Accessed: 22-Feb-2020].
- [72] R. Prodanovic and D. Simic, “A Survey of Wireless Security,” *Journal of Computing and Information Technology*, vol. 15, no. 3, p. 237, 2007.

- [73] M. Beck and E. Tews, “Practical attacks against WEP and WPA,” in *Proceedings of the second ACM conference on Wireless network security - WiSec '09*, Zurich, Switzerland, 2009, p. 79.
- [74] M. Beck, “Enhanced TKIP Michael attacks,” 2010. [Online]. Available: <https://arxiv.org/abs/1410.6295>.
- [75] T. Ohigashi and M. Morii, “A practical message falsification attack on WPA,” in *Proceedings of Joint Workshop on Information Security, JWIS (2009)*, Kaohsiung, Taiwan, 2009.
- [76] Y. Todo, Y. Ozawa, T. Ohigashi, and M. Morii, “Falsification Attacks against WPA-TKIP in a Realistic Environment,” *IEICE Transactions on Information and Systems*, vol. E95-D, no. 2, pp. 588–595, 2012.
- [77] M. Vanhoef and F. Piessens, “Practical verification of WPA-TKIP vulnerabilities,” in *Proceedings of the 8th ACM SIGSAC symposium on Information, computer and communications security - ASIA CCS '13*, Hangzhou, China, 2013, pp. 427–436.
- [78] D. Schepers, A. Ranganathan, and M. Vanhoef, “Practical Side-Channel Attacks against WPA-TKIP,” in *Proceedings of the 2019 ACM Asia Conference on Computer and Communications Security*, Auckland, New Zealand, 2019, pp. 415–426.
- [79] R. Moskowitz, “Weakness in Passphrase Choice in WPA Interface,” 2003. [Online]. Available: https://wifinetnews.com/archives/2003/11/weakness_in_passphrase_choice_in_wpa_interface.html. [Accessed: 09-Mar-2020].
- [80] B. Sak and R. Jilumudi Raghu, *Mastering Kali Linux Wireless Pentesting*. Birmingham, United Kingdom: Packt Publishing, 2016.
- [81] The Church of Wi-Fi, “The Church of Wifi WPA-PSK Lookup Tables.” [Online]. Available: <https://www.renderlab.net/projects/WPA-tables/>. [Accessed: 04-Mar-2020].
- [82] J. Cache, J. Wright, and V. Liu, *Hacking Exposed Wireless, Second Edition: Wireless Security Secrets and Solutions*, 2nd ed. New York, USA: McGraw-Hill, 2010.

- [83] Electronics Notes Co.Ltd, “Elcomsoft Wireless Security Auditor.” [Online]. Available: <https://www.elcomsoft.com/ewsa.html>. [Accessed: 04-Mar-2020].
- [84] J. Mora and L. Lueg, “Pyrit: The famous WPA precomputed cracker.” [Online]. Available: <https://github.com/JPaulMora/Pyrit>. [Accessed: 04-Mar-2020].
- [85] J. Wright, “coWPAtty.” [Online]. Available: <https://www.willhackforsushi.com/>. [Accessed: 04-Mar-2020].
- [86] J. Steube, “hashcat.” [Online]. Available: <https://hashcat.net/hashcat/>. [Accessed: 04-Mar-2020].
- [87] T. D’Otreppe and C. Devine, “aircrack-ng.” [Online]. Available: <https://www.aircrack-ng.org/doku.php?id=aircrack-ng>. [Accessed: 04-Mar-2020].
- [88] J. Steube, “New attack on WPA/WPA2 using PMKID,” *Hashcat forum*, 2018. [Online]. Available: <https://hashcat.net/forum/thread-7717.html>. [Accessed: 05-Mar-2020].
- [89] T. Roth, “Breaking encryptions using GPU accelerated cloud instances,” in *Black Hat Technical Security Conference*, Las Vegas, USA, 2011.
- [90] M. Vanhoef and F. Piessens, “Key Reinstallation Attacks,” in *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, Dallas, USA, 2017, pp. 1313–1328.
- [91] M. Vanhoef and F. Piessens, “Release the Kraken,” in *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*, Toronto, Canada, 2018, pp. 299–314.
- [92] M. Čermák, Š. Svorenčík, R. Lipovský, and O. Kubovič, “KR00K - Serious vulnerability deep inside your wi-fi encryption,” *ESET White paper*. 2020.
- [93] D. Harkins, “Simultaneous Authentication of Equals: A Secure, Password-Based Key Exchange for Mesh Networks,” in *2008 Second International Conference on Sensor Technologies and Applications (sensorcomm 2008)*, Cap Esterel, France, 2008, pp. 839–844.
- [94] M. Vanhoef and E. Ronen, “Dragonblood: A Security Analysis of WPA3’s SAE

- Handshake,” *Papers.Mathyvanhoef.Com*, 2018.
- [95] IEEE Std 802.11s -2011, *Part 11 : Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications Amendment 10 : Mesh Networking* IEEE Computer Society. 2011.
- [96] B. Buchanan, “How A Dragonfly Aims To Fix Delicate Wi-Fi’s Wings,” 2018. [Online]. Available: <https://medium.com/asecuritysite-when-bob-met-alice/how-a-dragonfly-aims-to-fix-delicate-wi-fis-wings-f26d82798010>. [Accessed: 10-Mar-2020].
- [97] D. D. Coleman, D. A. Westcott, and B. Harkins, *CWSP Certified Wireless Security Professional Study Guide CWSP-205*, 2nd ed. Indianapolis, USA: John Wiley & Sons Inc, 2016.
- [98] D. Harkins and W. Kumari, “Opportunistic Wireless Encryption,” *Internet Engineering Task Force (IETF) RFC8110 specification*, 2017. [Online]. Available: <https://rfc-editor.org/rfc/rfc8110.txt>.
- [99] M. Vanhoef and E. Ronen, “Dragonblood: Analyzing the Dragonfly Handshake of WPA3 and EAP-pwd,” in *IEEE Symposium on Security and Privacy*, San Francisco, USA, 2020.
- [100] P. S. Gray, J. B. Williamson, D. A. Karp, and J. R. Dalphin, *The Research Imagination - An introduction to qualitative and quantitative methods*, 1st ed. Cambridge, United Kingdom: Cambridge University Press, 2007.
- [101] C. Hurley, R. Rogers, F. Thornton, D. Connelly, and B. Baker, *WarDriving and Wireless Penetration Testing*, 1st ed. Rockland, USA: Syngress Publishing Inc, 2007.
- [102] B. Haines, M. J. Schearer, and F. Thornton, *Kismet Hacking*, 1st ed. Burlington, USA: Syngress Publishing Inc., 2008.
- [103] P. Shipley, “DEF CON 9 - 802.11b War Driving and Lan Jacking,” 2001. [Online]. Available: <https://www.youtube.com/watch?v=bWH-3OZJ0vo>. [Accessed: 17-Mar-2020].
- [104] D. Kumar, K. Shen, B. Case, D. Garg, G. Alperovich, D. Kuznetsov, R. Gupta, and

- Z. Durumeric, “All Things Considered: An Analysis of IoT Devices on Home Networks,” in *Proceedings of the 28th USENIX Conference on Security Symposium*, Santa Clara, USA, 2019, pp. 1169–1185.
- [105] Wigle.net, “Wigle.net FAQ.” [Online]. Available: <https://wigle.net/faq>. [Accessed: 17-Mar-2020].
- [106] C. Lin, H. Sathu, and D. Joyce, “Network security of wireless LANs in Auckland’s central business district,” *WSEAS Transactions on Communications*, vol. 3, no. 2, pp. 511–516, 2004.
- [107] A. Sarrafzadeh and H. Sathu, “Wireless LAN security status changes in Auckland CBD: A case study,” in *2015 IEEE International Conference on Computational Intelligence and Computing Research (ICIC)*, Madurai, India, 2015, pp. 1–6.
- [108] A. Nisbet, “A tale of four cities: Wireless security & growth in New Zealand,” in *2012 International Conference on Computing, Networking and Communications (ICNC)*, Maui, Hawaii, USA, 2012, pp. 1167–1171.
- [109] A. Nisbet, “A 2013 study of wireless network security in New Zealand: Are we there yet?,” in *Proceedings of the 11th Australian Information Security Management Conference, ISM 2013*, Perth, Australia, 2014, pp. 75–82.
- [110] A. K. Kyaw, P. Agrawal, and B. Cusack, “Wi-Pi: a study of WLAN security in Auckland City,” in *Proceedings of the Australasian Computer Science Week Multiconference on - ACSW '16*, Canberra, Australia, 2016, pp. 1–9.
- [111] H. Valchanov, J. Edikyan, and V. Aleksieva, “A Study of Wi-Fi Security in City Environment,” in *IOP Conference Series: Materials Science and Engineering*, Plovdiv, Bulgaria, 2019, vol. 618, p. 012031.
- [112] Offensive Security, “What is Kali Linux? Kali Linux Documentation.” [Online]. Available: <https://www.kali.org/docs/introduction/what-is-kali-linux/>. [Accessed: 28-Mar-2020].
- [113] Offensive Security, “Kali Linux Custom Image Downloads - Offensive Security.” [Online]. Available: <https://www.offensive-security.com/kali-linux-vm-vmware-virtualbox-image-download/>. [Accessed: 28-Mar-2020].

- [114] Wigle.net, “WiGLE.net Tools and Downloads.” [Online]. Available: <https://wagle.net/tools>. [Accessed: 28-Mar-2020].
- [115] Tarlogic Research, “Acrylic Wi-Fi Home.” [Online]. Available: <https://www.acrylicwifi.com/en/wlan-wifi-wireless-network-software-tools/wlan-scanner-acrylic-wifi-free/>. [Accessed: 28-Mar-2020].
- [116] DB Browser for SQLite, “DB Browser for SQLite.” [Online]. Available: <https://sqlitebrowser.org/>. [Accessed: 01-Apr-2020].
- [117] Finlex, “Laki rikoslain 28 luvun 7 §:n muuttamisesta 190/2011.” [Online]. Available: <https://www.finlex.fi/fi/laki/alkup/2011/20110190>. [Accessed: 29-Mar-2020].
- [118] Finlex, “Rikoslaki 19.12.1889/39.” [Online]. Available: <https://www.finlex.fi/fi/laki/ajantasa/1889/18890039001#L28>. [Accessed: 29-Mar-2020].
- [119] European Parliament and the Council of the European Union, “Regulation (EU) 2016/679 of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC (General Data Protection Regulation),” *Official Journal of the European Communities*, vol. 59, no. 119/1, pp. 1–88, 2016.
- [120] M. Walker, *Certified Ethical Hacker All-in-One Exam Guide*, 4th ed. New York, USA: McGraw-Hill Education, 2019.
- [121] F. Feldman, *Introductory Ethics*, 1st ed. Englewood Cliffs, USA: Prentice-Hall Inc, 1978.
- [122] S. Nathanson, “Utilitarianism, Act and Rule,” *Internet Encyclopedia of Philosophy*. [Online]. Available: <https://www.iep.utm.edu/util-a-r/#H1>. [Accessed: 30-Mar-2020].
- [123] N. Athanassoulis, “Virtue Ethics,” *Internet Encyclopedia of Philosophy*. [Online]. Available: <https://www.iep.utm.edu/virtue/#H3>. [Accessed: 30-Mar-2020].
- [124] M. J. Quinn, *Book of ethics for the information age*, 6th ed. Harrisonburg, USA: Pearson Education Inc, 2014.

Appendix A.

Abbreviations

AAD	<i>Additional Authentication Data</i>
ACK	<i>Acknowledgement Message</i>
ADB	<i>Android Debug Bridge</i>
AES	<i>Advanced Encryption Standard</i>
AIEE	<i>American Institute of Electrical Engineers</i>
A-MPDU	<i>Aggregate MAC Protocol Data Unit</i>
A-MSDU	<i>Aggregate MAC Service Data Units</i>
AP	<i>Wireless Access Point</i>
AP	<i>Access Point</i>
BPSK	<i>Binary Phase Shift Keying</i>
BS	<i>Base Station</i>
BSS	<i>Basic Service Set</i>
CCK	<i>Complementary Code Keying</i>
CCMP	<i>Counter Mode with CBC-MAC Protocol</i>
CPU	<i>Central Processing Unit</i>
CRC	<i>Cyclic Redundancy Check</i>
CSMA/CA	<i>Carrier Sense Multiple Access with Collision Avoidance</i>
CSMA/CD	<i>Carrier-Sense Multiple Access with Collision Detection</i>
CSV	<i>Comma-Separated Value</i>
CTS	<i>Clear to Send</i>
DBPSK	<i>Differential Binary Phase Shift Keying</i>
DES	<i>Data Encryption Standard</i>
DLL	<i>Data Link Layer</i>
DLP	<i>Discrete Logarithm Problem</i>
DQPSK	<i>Differential Quadrature Phase Shift Keying</i>
DSSS	<i>Direct Sequence Spread Spectrum</i>
EAP	<i>Extensible Authentication Protocol</i>
EAPOL	<i>Extended Authentication Protocol over LAN</i>
ECC	<i>Elliptic Curve Cryptography</i>
ERP	<i>Extended Rate Physical</i>
ETSI	<i>European Telecommunications Standards Institute</i>
EU	<i>European Union</i>
FCC	<i>Federal Communications Commission</i>
FHSS	<i>Hopping Spread Spectrum</i>
Gbps	<i>Gigabits per second</i>
GCHQ	<i>British Government Communication Headquarters</i>
GDPR	<i>General Data Protection Regulation</i>
GHz	<i>Gigahertz</i>
GPS	<i>Global Positioning System</i>
GPU	<i>Graphical Processing Units</i>

GTK	<i>Group Temporal Key</i>
HR-DSSS	<i>High Rate Direct Sequence Spread Spectrum</i>
HT	<i>High Throughput</i>
ICV	<i>Integrity Check Value</i>
IDS	<i>Intrusion Detection System</i>
IEEE	<i>Institute of Electrical and Electronics Engineers</i>
IETF	<i>Internet Engineering Task Force</i>
IoT	<i>Internet of Things</i>
IRE	<i>Institute of Radio Engineers</i>
ISM	<i>Industrial, Scientific, and Medical</i>
ISO	<i>International Organization for Standardisation</i>
IV	<i>Initialization Vector</i>
KCK	<i>Key Confirmation Key</i>
KML	<i>Keyhole Markup Language</i>
KRACK	<i>Key Reinstallation Attacks</i>
LAN	<i>Local Area Network</i>
MAC	<i>Media Access Control</i>
Mbps	<i>Megabits per second</i>
MHz	<i>Megahertz</i>
MIC	<i>Message Integrity Check</i>
MIMO	<i>Multiple-Input Multiple-Output</i>
MPDU	<i>MAC Protocol Data Unit</i>
MSDU	<i>Mac Service Data Unit</i>
MU-MIMO	<i>Multiuser MIMO</i>
NAV	<i>Network Access Vector</i>
NCR	<i>National Cash Register</i>
NIC	<i>Network interface card</i>
NIST	<i>National Institute of Standard and Technology</i>
OFDM	<i>Orthogonal Frequency-Division Multiplexing</i>
OFDMA	<i>Orthogonal Frequency Division Multiple Access</i>
OSA	<i>Open System Authentication</i>
OSI	<i>Open System Interconnection</i>
PHY	<i>Physical Layer</i>
PMK	<i>Pair-Wise Master Key</i>
PN	<i>Packet Number</i>
PSK	<i>Pre-Shared Key</i>
PTK	<i>Pair-wise Transient Key</i>
QAM	<i>Quadrature Amplitude Modulation</i>
QOS	<i>Quality of Service</i>
QPSK	<i>Quadrature Phase Shift Keying</i>
RC4	<i>Rivest Cipher 4</i>
RSN	<i>Robust Security Network</i>
RSNA	<i>Robust Security Network Associations</i>
RTS	<i>Request to Send</i>

RU	<i>Resource Units</i>
SAE	<i>Simultaneous Authentication of Equals</i>
SDM	<i>Spatial Diversity Multiplexing</i>
SKA	<i>Shared Key Authentication</i>
SM	<i>Spatial Multiplexing</i>
SSID	<i>Service Set Identifier</i>
TF	<i>Trigger Frame</i>
TKIP	<i>Temporal Key Integrity Protocol</i>
TSC	<i>TKIP Sequence Counter</i>
TWT	<i>Target Wake Up Time</i>
UHF	<i>Ultra High Frequency</i>
UI	<i>User Interface</i>
WECA	<i>Wireless Ethernet Compatibility Association</i>
WEP	<i>Wired Equivalent Privacy</i>
WFA	<i>Wi-Fi Alliance</i>
VHT	<i>Very High Throughput</i>
WLAN	<i>Wireless Local Area Network</i>
VoIP	<i>Voice over IP</i>
WPA	<i>Wi-Fi Protected Access</i>
VPN	<i>Virtual Private Network</i>
WPS	<i>Wi-Fi Protected Setup</i>
XOR	<i>Exclusive-OR</i>