Traffic Analysis of IEEE 802.11 on Physical Layer by using Software Defined Radio

Jiri Pokorny[1,3], Radek Fujdiak[1], Martin Kovanda[2], Martin Strajt[1], Jiri Hosek[1]
[1] *Department of Telecommunications, Brno University of Technology, Brno, Czech Republic*
[2] *GreyCortex, Brno, Czech Republic*
[3] *Unit of Electrical Engineering, Tampere University, Tampere, Finland*
Contact author's e-mail: fujdiak@vutbr.cz

***Abstract*—The security of wireless networks is a fairly discussed topic. There are many threads capable of attacking any personal, public, or company network. One way of protection is to analyze traffic on the desired wireless network and look for irregularities in the traffic. There are various approaches on how to capture and analyze wireless traffic. In this paper, we present three enablers on capturing wireless traffic, including Off-the-shelf wireless cards and software defined radios. The results provide the capabilities of used devices and possibilities on how to analyze the traffic. We also present our developed wireless traffic visualizer written in python that can be utilized to discover possible attacks on wireless networks visually.**

***Index Terms*—IEEE 802.11, Wi-Fi, Security, SDR**

I. Introduction

The wireless technology standard IEEE 802.11, also known as ”Wi-Fi”, spreads to many different areas, including home, enterprises, industry, or even the military [1]. However, the security point of view is still a very underestimated topic in Wi-Fi technologies. Once the attacker compromises the access point (AP), there is nothing that would possibly impede eavesdropping and other unwanted actions [2]. Moreover, the broadcast nature of Wi-Fi also allows very easy eavesdropping of the legitimate communication between users [3]. Therefore, the use of cryptography and security protocols is a must in the wireless environment [4], [5]. Unfortunately, the security protocols such as WEP (Wired Equivalent Privacy), WPA (Wi-Fi Protected Access) or WPA2 (Wi-Fi Protected Access 2), which are supported by current Wi-Fi equipment, are coming with several vulnerabilities and are known to be compromised [6]–[8]. Therefore, a new security protocol was developed to answer the cybersecurity vulnerabilities in Wi-Fi technologies, known as the WPA3 [9]. However, WPA3 was compromised even before it hit the market with the first devices implementing it and nowadays already exists again many different vulnerabilities described even for this new protocol such as the Dragonblood technique [10], bad-token vulnerability [11], deprivation attacks [12], and others [13], [14]. Furthermore, connected with Wi-Fi technologies, is the prolonged vulnerability fixation time, i.e., Bittau et al. [15] discovers that 76 % of APs in London still use an older version of WEP more than six years after it was found to be insecure.

The description of the IEEE 802.11 standard covers the first two layers of the OSI model (Data Link Layer - MAC and Physical Layer - PHY) [16]. Most of the vulnerabilities are coming from the higher layers. However, the physical layer security is important as well. There are many different papers focused on physical security of wireless technologies such as [17]–[24], including different surveys and tutorials such as [25]–[28]. The most relevant papers are [29]–[31], where several security measures and techniques were introduced for IEEE 802.11, including self-interference mitigation, classifier selection and location identification. Moreover, Zhu et al. [31] show the power of the software defined radio (SDR) in case of analyzing the physical layer of any wireless technology, including Wi-Fi. This paper focuses on the PHY layer of IEEE 802.11 in the sense of capturing the traffic and understanding the captured data by decoding them on the MAC layer. This is mostly important if considering any cybersecurity solution, which does not change the standard, but uses no external methods such as intrusion detection, intrusion prevention, or any analytical methods, which might use PHY layer information as an additional parametric source of data.

The rest of the paper is organized as follows. The Sections II-A and II-B provide basic information about the experimental environment, including parametric description of used hardware and software. Section II-C includes results from physical layer capturing via software defined radio together with analyzing the data on the MAC layer followed by Section II-D, which provides an example of possible visualization of the captured traffic for possible traffic analytical tools and operators. Finally, Section III summarized the findings.

II. Results

A. Experimental environment

The research was focused on measurements of the most commonly used frequency – 2.4 GHz. This frequency is used regularly for wireless networks at home and work. All measurements were conducted in everyday conditions, i.e., office or home. The assumption is that there is an average level of wireless interference, e.g., other wireless networks, mobile devices, computers, and other similar devices. Regarding other wireless networks, there was a desire to use the least occupied frequency spectrum to avoid unnecessary interference, so the

The research described in this paper was financed by the Technology Agency of the Czech Republic under grant no. TJ02000332.

measured spectrum was initially scanned by mobile application and the least occupied frequency band was selected.

B. HW – MAC Layer Analysis Enablers

Several different hardware can analyze the MAC layer. Any wireless access card is usually capable of capturing raw wireless data. After that, it depends whether the firmware of the device allows capturing, manipulation, and accessing the data for further analysis. Not all off-the-shelf devices are capable of doing that. In this research, affordable devices under 1500 $ were selected. Also, the diversity of utilized HW was necessary, and the final selection of HW reflects on that. During our research, these data capturing enablers were used: USB Wi-Fi card, LimeSDR, and BladeRF A9. The selected HW is depicted in Fig. 1.

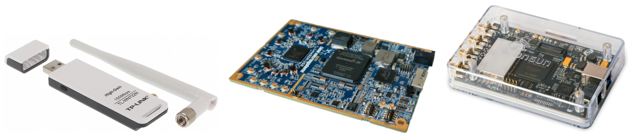

Fig. 1. Selected HW enablers for wireless network analysis (from left: TP-LINK, LimeSDR and BladeRF)

The main parameters of the selected HW are shown in Tab. I.

TABLE I
PARAMETERS OF SELECTED HW

	USB Wi-Fi	LimeSDR	BladeRF
Version	TP-LINK TL-WN722N	LimeSDR	Blade RF micro A9 rev1.3
Frequency	2.4 GHz	100 kHz to 3.8 GHz	47 MHz to 6 GHz
Bandwidth	22	61.44 MHz	56 MHz
No. of antennas	1	6 RX, 4 TX	4
Interface	USB 2.0	USB 3.0	USB 3.0
External power	NO	NO	YES, 9 V
FPGA Chip	-	Altera Cyclone IV EP4CE40F23	Altera Cyclone V 5CEBA9F23C8N
MIMO	NO	2x2	2x2

The Wi-Fi cards are typically used for regular wireless connectivity to a wireless network. The SDRs are more suitable for research and experiments to capture and analyze data since they provide higher performance, higher bandwidth, more antennas, and other benefits. Also, they can transmit and receive on a specific frequency.

C. MAC Layer Capture Results

The results of captured data by different devices are discussed in this section. The setup in measurements with USB Wi-Fi card is shown in Fig. 2. The capturing Wi-Fi card used was TP-LINK TL-WN722N with chip Atheros AR9271O. The card was connected to Kali linux version 2019.4 running as a virtual machine. Program Aircrack-ng, specifically the script Airmon-ng was used for switching the card into monitoring mode. This is necessary for capturing the capabilities of the card. Not all wireless cards can be switched to the monitoring mode and in that case, it is not possible to use that device.

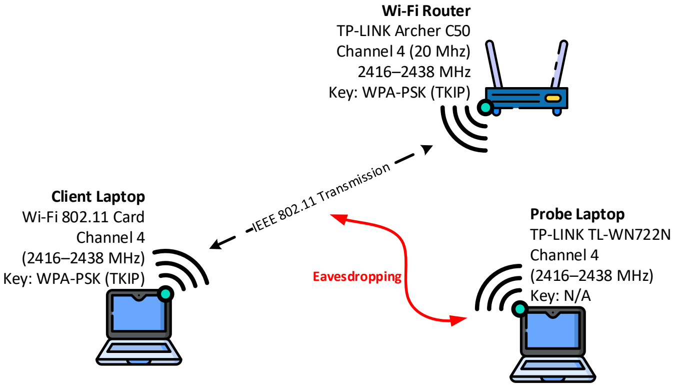

Fig. 2. Measurement topology for wireless network analysis

Simulation of traffic was done by a laptop connected to router TP-LINK Archer C50. The ping tool was used to send data for simply identifying the captured traffic. The data of one sent message was set to 1000 B and the transmit sequence contained five messages. This sequence was constantly repeating. The results were visualized in Wireshark and are shown in Fig. 3 and in Fig. 4.

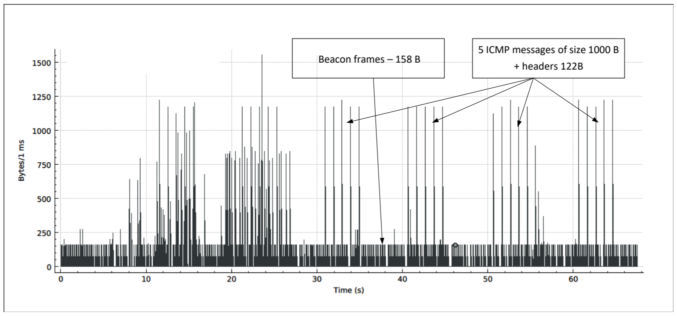

Fig. 3. Traffic captured from outside of the network with USB Wi-Fi card

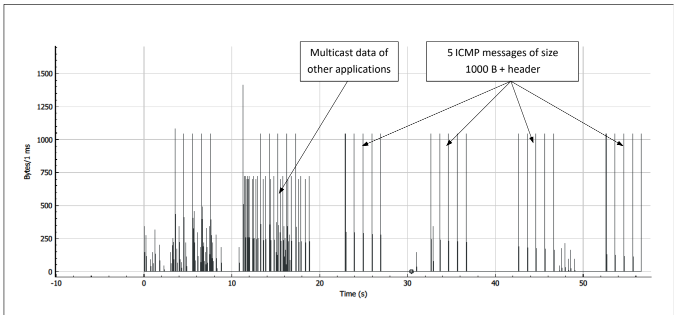

Fig. 4. Traffic captured from inside of the network with USB Wi-Fi card

It is clear from the results that capture from within the network and from outside of the network are almost equal. The ICMP packets are visible clearly on both captured files. The capture from outside the network contains beacon packets from all networks in the area, unlike the second from within the network. After this, the intention was to capture the same traffic with the two SDRs. Controlling the SDR is not as straight forward as a Wi-Fi card, mainly because of the number of parameters that need to be set. Also, it is usually necessary to calibrate an SDR as it was with these two devices. Because controlling of SDRs is more complex than with Wi-Fi cards, it requires more complex software. The program that supports selected two SDRs is GNU radio [32]. It is an open-source designed specifically to control SDRs. The user has to create a project by connecting multiple function blocks. In our case, the blocks to capture and demodulate the wireless signal and then decode the packets were needed. For easy analysis, the data was exported to a PCAP file. During this project, the versions 3.7.11 and 3.8.1 of GNU Radio were used. As a starting point, the example wifi_rx_rftap from [33] was used. This example is compatible with various SDRs, including LimeSDR and BladeRF, but usually, at least one component – in this case, the source – has to be switched according to the device. The measurements were conducted on the same topology from Fig. 2 with two exceptions. One, the capturing device was changed from USB Wi-Fi card to LimeSDR, and two, the USB Wi-Fi card was used as a verification capture device. The results are shown in Fig. 5 and Fig. 6.

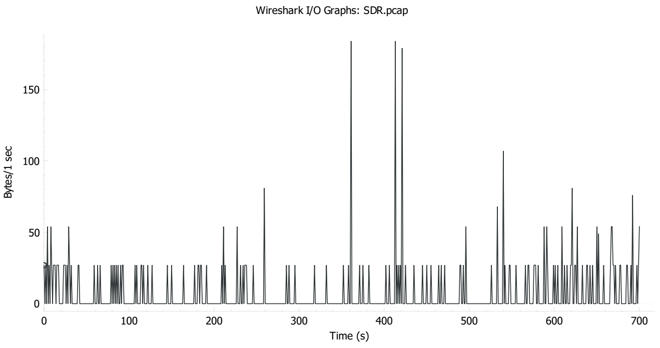

Fig. 5. Captured traffic with LimeSDR

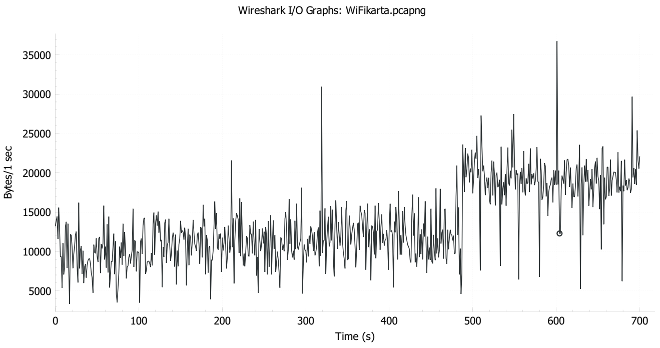

Fig. 6. Captured traffic with verification USB Wi-Fi card

They show that the amount of captured traffic with LimeSDR was much lower than with the Wi-Fi card. This was the reason to repeat the measurements on different versions of Linux OS with different versions of GNU Radio. However, the changes appeared not to have any effect on the measurements. BladeRF was verified on the same GNU Radio example as LimeSDR, except the source block was changed to OsmoSDR from the project Osmocom [34]. The resulting captured traffic was very similar to the one with LimeSDR from Fig 5. For that reason, another way of controlling BladeRF was explored. BladeRF features drivers for MS Windows to run in MATLAB. After installation and properly setting up the Windows PATH variable, running the RX GUI Demo (file: $bladeRF\_rx\_gui.m$). The result is shown in Fig. 7.

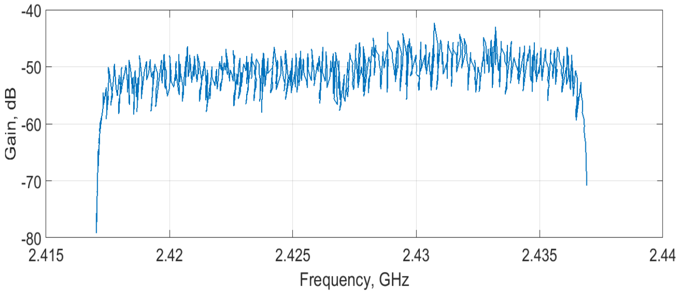

Fig. 7. Tested MATLAB script with BladeRF

In this demo, it is possible to set numerous parameters, e.g., transceiver gain, frequency bandwidth, sample rate, frequency. The demo worked as expected, except for a few programs crashes due to changing the parameters while the program was running. It is recommended first to stop real-time capture, change the parameters and finally start the program again. The use of MATLAB with SDR is not a very common thing; however, to our experience, it seems like an appropriate alternative to GNU Radio. The drawback, in this case, is that MATLAB lacks complete implementations for capture and analysis of traffic and that would require further work for finish decoding and exporting into the PCAP file. These steps are needed for full implementation:

- Decoder – To decode the MAC layer frames. A way of implementing this is described in [35].
- PCAP export – To analyze captured data with available tools [36].

D. Divided Source-based Traffic Visualization

There are existing tools to visualize network traffic stream from the PCAP file. Wireshark is one of the well-known tools for this purpose. However, the goal of this research is to be able to visualize network traffic in real-time. Also, possible implementation in other software tools is required. For that reason, a tool for visualizing the network traffic from the PCAP file was developed in the python programming language. A detailed description of the tool is out of scope. However, the list of utilized packages is the following: numpy, scipy, sklearn, pyshark, and matplotlib. Captured, analyzed, and visualized traffic is shown in Fig. 8. It is possible to identify the ICMP packets in the second and third columns. Through this visualization, it might be possible to identify attacks and anomalies on a wireless network.

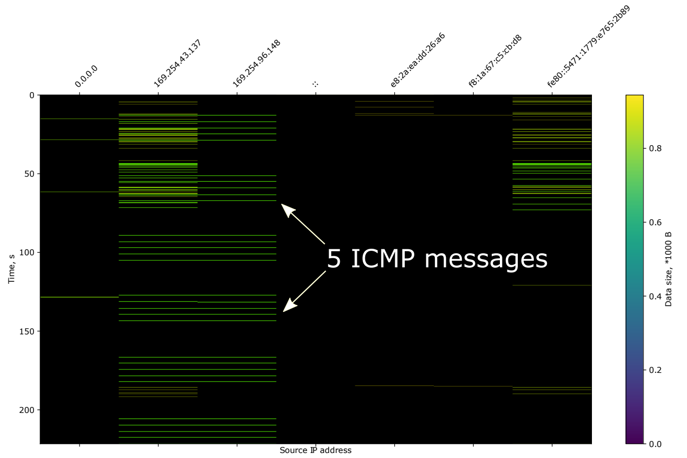

Fig. 8. PCAP traffic visualizer, Wi-Fi card captured traffic within the network

III. Conclusion

The paper summarized the cybersecurity issues in the IEEE 802.11, including the PHY layer. The theoretical sources introduce the challenges of current Wi-Fi technologies, where practical results show the approach, which might help develop the solution for introduced challenges. SDR is a powerful tool and PHY layer contains essential information, which might be used for early incident detection before the wireless network becomes compromised. Future work should focus on using these in detection and prevention systems.

References

[1] Y. Wu, A. Khisti, C. Xiao, G. Caire, K.-K. Wong, and X. Gao, "A survey of physical layer security techniques for 5G wireless networks and challenges ahead," *IEEE Journal on Selected Areas in Communications*, vol. 36, no. 4, pp. 679–695, 2018.

[2] J. Xiong and K. Jamieson, "Securearray: Improving wifi security with fine-grained physical-layer information," in *Proceedings of the 19th annual international conference on Mobile computing & networking*, 2013, pp. 441–452.

[3] Y. Zou, J. Zhu, X. Wang, and V. C. Leung, "Improving physical-layer security in wireless communications using diversity techniques," *IEEE Network*, vol. 29, no. 1, pp. 42–48, 2015.

[4] M. E. Hellman, "An overview of public key cryptography," *IEEE Communications Magazine*, vol. 40, no. 5, pp. 42–49, 2002.

[5] S. V. Kartalopoulos, "A primer on cryptography in communications," *IEEE Communications Magazine*, vol. 44, no. 4, pp. 146–151, 2006.

[6] E. Tews and M. Beck, "Practical attacks against WEP and WPA," in *Proceedings of the second ACM conference on Wireless network security*, 2009, pp. 79–86.

[7] B. Bertka, "802.11 w security: DoS attacks and vulnerability controls," in *Proc. of Infocom*, 2012.

[8] M. Eian and S. F. Mjølsnes, "The modeling and comparison of wireless network denial of service attacks," in *Proceedings of the 3rd ACM SOSP workshop on networking, systems, and applications on mobile handhelds*, 2011, pp. 1–6.

[9] W.-F. Alliance, "WPA3 specification version 1.0," 2019.

[10] M. Vanhoef and E. Ronen, "Dragonblood: Analyzing the Dragonfly Handshake of WPA3 and EAP-pwd," in *Proceedings of the IEEE Symposium on Security and Privacy-S&P)*. IEEE, 2020.

[11] K. Lounis and M. Zulkernine, "Bad-token: denial of service attacks on WPA3," in *Proceedings of the 12th International Conference on Security of Information and Networks*, 2019, pp. 1–8.

[12] ——, "WPA3 Connection Deprivation Attacks," in *International Conference on Risks and Security of Internet and Systems*, 2019, pp. 164–176.

[13] C. P. Kohlios and T. Hayajneh, "A comprehensive attack flow model and security analysis for Wi-Fi and WPA3," *Electronics*, vol. 7, no. 11, p. 284, 2018.

[14] B. I. Reddy and V. Srikanth, "Review on wireless security protocols (WEP, WPA, WPA2 & WPA3)," *International Journal of Scientific Research in Computer Science, Engineering and Information Technology*, 2019.

[15] A. Bittau, M. Handley, and J. Lackey, "The final nail in WEP's coffin," in *2006 IEEE Symposium on Security and Privacy (S&P'06)*. IEEE, 2006, pp. 15–pp.

[16] H. Wu, Y. Peng, K. Long, and S. Cheng, "A simple model of IEEE 802.11 wireless LAN," in *2001 International Conferences on Info-Tech and Info-Net. Proceedings (Cat. No. 01EX479)*, vol. 2. IEEE, 2001, pp. 514–519.

[17] A. D. Wyner, "The wire-tap channel," *Bell system technical journal*, vol. 54, no. 8, pp. 1355–1387, 1975.

[18] E. Tekin and A. Yener, "The general gaussian multiple-access and two-way wiretap channels: Achievable rates and cooperative jamming," *IEEE Transactions on Information Theory*, vol. 54, no. 6, pp. 2735–2751, 2008.

[19] A. Khisti and G. W. Wornell, "Secure transmission with multiple antennas—Part II: The MIMOME wiretap channel," *IEEE Transactions on Information Theory*, vol. 56, no. 11, pp. 5515–5532, 2010.

[20] F. Oggier and B. Hassibi, "The secrecy capacity of the MIMO wiretap channel," *IEEE Transactions on Information Theory*, vol. 57, no. 8, pp. 4961–4972, 2011.

[21] L. Dong, Z. Han, A. P. Petropulu, and H. V. Poor, "Improving wireless physical layer security via cooperating relays," *IEEE transactions on signal processing*, vol. 58, no. 3, pp. 1875–1888, 2009.

[22] S. A. A. Fakoorian and A. L. Swindlehurst, "Solutions for the MIMO Gaussian wiretap channel with a cooperative jammer," *IEEE Transactions on signal Processing*, vol. 59, no. 10, pp. 5013–5022, 2011.

[23] G. Zheng, L.-C. Choo, and K.-K. Wong, "Optimal cooperative jamming to enhance physical layer security using relays," *IEEE Transactions on Signal Processing*, vol. 59, no. 3, pp. 1317–1322, 2010.

[24] X. He and A. Yener, "Providing secrecy with structured codes: Two-user Gaussian channels," *IEEE Transactions on Information Theory*, vol. 60, no. 4, pp. 2121–2138, 2014.

[25] A. Mukherjee, S. A. A. Fakoorian, J. Huang, and A. L. Swindlehurst, "Principles of physical layer security in multiuser wireless networks: A survey," *IEEE Communications Surveys & Tutorials*, vol. 16, no. 3, pp. 1550–1573, 2014.

[26] N. Yang, L. Wang, G. Geraci, M. Elkashlan, J. Yuan, and M. Di Renzo, "Safeguarding 5G wireless communication networks using physical layer security," *IEEE Communications Magazine*, vol. 53, no. 4, pp. 20–27, 2015.

[27] Y. Zou, J. Zhu, X. Wang, and L. Hanzo, "A survey on wireless security: Technical challenges, recent advances, and future trends," *Proceedings of the IEEE*, vol. 104, no. 9, pp. 1727–1765, 2016.

[28] X. Chen, D. W. K. Ng, W. H. Gerstacker, and H.-H. Chen, "A survey on multiple-antenna techniques for physical layer security," *IEEE Communications Surveys & Tutorials*, vol. 19, no. 2, pp. 1027–1053, 2016.

[29] G. Li, J. Yu, Y. Xing, and A. Hu, "Location-invariant physical layer identification approach for WiFi devices," *IEEE Access*, vol. 7, pp. 106 974–106 986, 2019.

[30] P. K. Harmer, D. R. Reising, and M. A. Temple, "Classifier selection for physical layer security augmentation in cognitive radio networks," in *2013 IEEE International Conference on Communications (ICC)*. IEEE, 2013, pp. 2846–2851.

[31] F. Zhu, F. Gao, T. Zhang, K. Sun, and M. Yao, "Physical-layer security for full duplex communications with self-interference mitigation," *IEEE Transactions on Wireless Communications*, vol. 15, no. 1, pp. 329–340, 2015.

[32] GNU Radio project. GNU Radio. Accessed June 2020.

[33] B. Bloessl. IEEE 802.11 a/g/p transceiver for GNU Radio. Accessed June 2020.

[34] J.-F. Lang. Open Source Mobile Communications. Accessed June 2020.

[35] Mathworks. 802.11 MAC Frame Decoding. Accessed June 2020.

[36] ——. 802.11 MAC Frame Decoding. Accessed June 2020.