Connectivity and Energy Consumption of Cyber-Physical Systems under Wi-Fi Attack

Pericle Perazzo
Dept. of Information Engineering (DII)
University of Pisa, Italy
pericle.perazzo@unipi.it

Mattia Paladini
Dept. of Information Engineering (DII)
University of Pisa, Italy
m.paladini4@studenti.unipi.it

Alessio Vecchio
Dept. of Information Engineering (DII)
University of Pisa, Italy
alessio.vecchio@unipi.it

***Abstract*—Cyber-Physical Systems (CPSes) represent a groundbreaking evolution in the realm of information technology and engineering, merging the capabilities of computational elements with physical processes. In these systems, the timeliness of knowledge of a process state or a sensor measurement is important, and it can be measured with a metric called Age of Information (AoI). Since CPSes introduce new and significant security challenges, it is interesting to study cyber-attacks aimed at degrading their AoI. Such attacks have been studied mainly with a theoretic-only approach by past literature. In this paper, we study a CPS under attack using an experimental approach, testing a real denial-of-service attack via Wi-Fi against a Raspberry Pi, with various security configurations. In particular, we consider the Wi-Fi deauthentication attack, which is a highly cost-effective attack from the adversarial point of view. As a further contribution we study the impact of such an attack in terms of energy wasted by a battery-powered CPS sensor.**

***Index Terms*—Cyber-Physical Systems, Energy consumption, Age of Information, Wi-Fi Deauthentication.**

I. Introduction

Low-latency Cyber-Physical Systems (CPSes) are finding increasingly significant applications in the world. These applications include smart vehicles that collaborate to improve road capacity [1], cameras that compute point clouds to represent the environment [2], sensor networks that collect and analyze data to timely identify anomalies [3], and remote surgery systems that control and update the position of surgical tools [4]. In these and similar applications, a source produces status update messages conveying timestamps, which are then transmitted via a network to one or more monitors, and it is crucial to keep the remote monitors' knowledge of the system's state up to date. Kaul et al. demonstrated that ensuring timely updates is not the same as maximizing the network's utilization or minimizing the messages' delay [5]. This led to the definition of a new performance metric, called *Age of Information* (AoI), that is used to describe the timeliness of a monitor's understanding of the state of a system or process [6].

Measuring and optimizing the AoI of CPSes in non-adversarial settings is a widely studied field [6]. However, CPSes expose a large attack surface for malicious parties, because they integrate computation, network, and control components. In particular, an attacker could disrupt the timeliness of information by mounting various types of denial-of-service attacks. These attacks are in general the simplest ones to mount, since they often do not need to bypass authentication or similar. Some research approached the problem of AoI-sensitive systems in adversarial environments [7], [8], [9], [10], [11], [12], [13]. However, all of them took a highly theoretical approach, using game theory and formal analysis.

ISBN 978-3-903176-68-3

In this paper, we study an AoI-sensitive system under attack using an experimental approach, testing a real denial-of-service attack via Wi-Fi against a Raspberry Pi, and obtaining real AoI measurements. Moreover, we do not consider jamming like most past literature does, but rather Wi-Fi deauthentication, which is a more clever and realistic attack because it obtains the same effect with far less transmission power. We also consider various security configurations, namely with unencrypted traffic, with traffic encrypted with TLS version 1.2 or 1.3, and with server-only or mutual authentication. Our conclusion is that Wi-Fi deauthentication attacks have a big impact on AoI, therefore a CPS should adopt defenses against this attack at the data-link layer. As a further contribution, we study the impact of Wi-Fi deauthentication attacks also from the point of view of wasted energy on the victim device.

The rest of the paper is organized as follows. Section II reports and compares with related work. Section III introduces some necessary preliminary concepts, namely the AoI, the TLS protocol in its 1.2 and 1.3 versions, and the Wi-Fi deauthentication attack. Section IV describes our experimental testbed. Section V reports and discusses our experimental results. Finally, the paper is concluded in Section VI.

II. Related Work

Nguyen et al. [7] first studied an AoI-sensitive system under attack by a malicious party. The authors considered an adversarial disruption on the data freshness by constructing a two-player non-zero-sum game. In this game, one participant, the honest transmitter, strives to keep the updates it sends to its receiver as fresh as possible, while the other participant, the interferer, seeks to hinder this by jamming the channel. The strategy of each player is determined by the power level they transmit. The authors also calculated the equilibrium points for both Nash and Stackelberg strategies, showing that the Stackelberg strategy dominates the Nash one. The Nguyen et al. paper started a productive research track [8], [9], [10], [11], [12], [13] that studies AoI in adversarial settings, typically (but

not exclusively) considering jamming attacks with a game-theory approach.

Gao et al. [9] examined a dynamic non-zero-sum game on a remote sensing system featuring a sensor, an encoder, a decoder, and an interferer adversary. The adversary generates an additive noise on the channel, and his strategy cost considers the noise power and the error caused on the estimated state.

Xiao et al. [8] examined the real-time delivery of a physical process samples under adversarial conditions. The attacker aims at increasing the age of information by jamming the channel. The authors represented the continuous interaction between the attacker and the system as a dynamic game, in which at each round the attacker chooses a jamming time and the system reacts by postponing the sampling according to a policy.

Garnaev et al. [10] examined an application including a drone and a ground station, in which the success of the mission depends on keeping the received information up-to-date. The authors studied the system using game theory, assuming an adversary that interferes with drone communication.

Banerjee and Ulukus [11] examined a system in which a base station serves multiple users via a wireless channel, and in which the timeliness of information is important. An adversary can jam the channel up to a specified fraction of the time within a specified maximum attack period. The authors do not approach the problem with game theory, but rather they analytically assess the performance of deterministic and probabilistic scheduling algorithms.

Bonagura et al. [13] studied a remote transmitter that dispatches status updates about a process to a receiver. The system can be in two states: the one in which the receiver has accurate knowledge of the process state, and the one in which it does not. The authors do not consider jamming attacks, but rather they study the age of incorrect information (a performance metric derived from AoI) in the presence of an adversary that can inject bogus data.

Virtually all the published research about AoI measurement in adversarial settings takes a highly theoretical approach, using game theory or formal analysis. On the other hand, we use an experimental approach, testing a real denial-of-service attack via Wi-Fi against a Raspberry Pi, and obtaining real AoI measurements of a typical CPS under attack. Moreover, we do not consider jamming but rather Wi-Fi deauthentication, which has the same effect as jamming but with far less transmission power. We also considered the impact of the TLS protocol on the AoI, which represents a novelty in literature to the best of our knowledge. Besides AoI, we also evaluated the impact of the attack from the point of view of wasted energy, on the victim device. Since energy can be a scarce resource in many CPS scenarios (information-producing devices can be battery-operated), denial of service can be achieved by its depletion.

III. Preliminaries

A. Age of Information

The Age of Information (AoI) is an end-to-end metric that measures the "freshness" of information at the observer's side about a specific phenomenon. In our scenario, the phenomenon is sampled by a CPS sensor that transmits the collected data to a receiver (the observer). At the receiver, the AoI at time t is defined as:

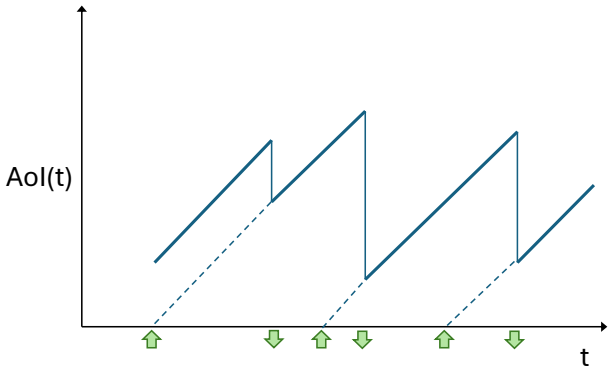


Fig. 1: The typical sawtooth-behavior of AoI. Up-pointing arrows indicate when new information is produced. Down-pointing arrows indicate when information is received.

$$AoI(t) = t - U(t) \tag{1}$$

where $U(t)$ is the generation time of the freshest data received [6]. The AoI increases linearly when no messages are received and drops sharply when new information is obtained. However, the AoI does not drop to zero upon receiving a new message due to the transfer time from the data generation device to the receiver, making network latency the lower bound for AoI. Overall, AoI exhibits an irregular sawtooth pattern, as illustrated in Figure 1. It is well-established that in lightly loaded systems, increasing the message rate improves AoI because fresh information is sent more frequently to the receiver. Conversely, surpassing a certain message rate threshold can worsen AoI due to the formation of communication queues, leading to the reception of outdated information.

The AoI is an instantaneous metric that changes with time. Therefore, what is typically measured is the *average AoI* over a given time interval, which is defined as:

$$\widehat{AoI} = (1/T) \int_{t \in T} AoI(t)\, dt, \tag{2}$$

where T is the time interval over which the average AoI is measured.

B. Transport-Layer Security

Transport Layer Security (TLS) is a cryptographic protocol running over TCP and designed to provide communication confidentiality and authenticity between two peers over a network. Its most visible application in nowadays networks is for implementing HTTPS, but it is also used for emails, instant messaging, M2M communication, etc. The process of creating a secure connection begins with a *handshake*, which establishes a shared session key that is then used to encrypt messages and provide their integrity. Sessions are temporary, and once ended, they must be re-established or resumed.

Typically, only the server owns a digital certificate, so TLS is configured to authenticate only the server. However, in many CPSes, sensors acting as clients can have their certificates, so TLS can perform mutual authentication of server and client. Mutual authentication is often used in CPSes to avoid men in the middle or sensor personification.

The most widespread versions of the TLS protocol nowadays are 1.2 [14] and 1.3 [15], which are both supported by most browsers. Compared to the 1.2 version, TLS 1.3 provides enhanced levels of privacy, security, and performance. From the performance point of view, the TLS v1.2 handshake requires two round trips between client and server, and therefore it requires a non-negligible time to complete. The key idea behind the design of TLS v1.3 is exactly to reduce the number of round trips to one, by drastically narrowing the number of supported cipher suites from 37 to 5. In particular, all the encryption modes that are not based on Authenticated Encryption with Associated Data (AEAD) were dropped from the standard, because considered unsafe or unneeded. The same happened to all the key exchange methods different from Elliptic-Curve Diffie-Hellman (ECDH). Such a simplified handshake results in a significant time saving compared to the TLS v1.2 handshake.

C. Wi-Fi Deauthentication Attack

The Wi-Fi deauthentication attack is an attack that forces a victim station to disconnect from a Wi-Fi network. The attacker can perform the attack by sending a bogus deauthentication frame to the access point. This can be done by simply knowing the victim's MAC address, because in many cases the deauthentication frame is not protected by encryption, even when the protocol is secured with WPA or WPA2. Publicly available tools like Airgrack-ng [16] can easily perform a deauthentication attack. The attack can be aimed at performing a simple denial of service, as well as at forcing the victim station to reconnect in such a way to mount successive attacks, for example authentication sniffing or evil twin connection [17]. In this paper, we are interested in the deauthentication attack as a means for denial of service.

IV. Experimental Testbed

We constructed the experimental setup shown in Figure 2, which includes various hardware and software components. A Raspberry PI serves as a CPS sensor, hosting the client side that produces information. It communicates over Wi-Fi using the 2.4 GHz band and operates as a battery-powered device. We used HTTP as the application protocol, where the Raspberry PI hosts the client, and the web server is executed on a laptop PC connected to the same LAN. The testbed can be configured to use TLS underneath HTTP (HTTPS) or not (classic HTTP). In turn, the TLS protocol can be configured to run the 1.2 version as well as the 1.3 one, and with server-only or mutual authentication. The payload of each HTTP request is 1024 bytes, and the client issues a request every 0.5 seconds. Each experiment is 30 seconds long, so the total number of requests for each experiment is 60. Besides the

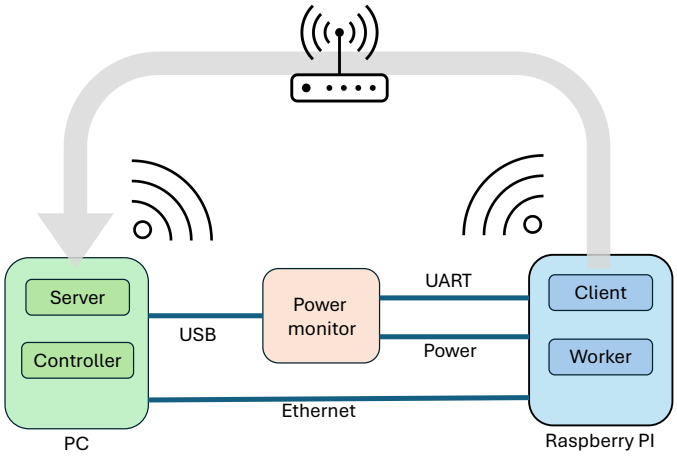


Fig. 2: The experimental setup.

components of the application under test (the HTTP client and server), the two devices also execute two software components (*controller* and *worker*) that are responsible for managing the experiments. The Raspberry PI is also connected to an Otii Arc Pro power monitor [18] to measure its power consumption. The Otii power monitor features a UART interface that can be used to allow the controller to communicate with the worker. In particular, the UART connection is used by the controller running on the PC to send commands about the current experiment to the worker running on the Raspberry PI. The same UART connection is used in the opposite direction by the worker to signal the start and stop times of the significant parts of an experiment execution. This way, the controller can annotate the recorded power trace with accurate timestamps.

The study aims to evaluate the system in terms of the AoI and the energy consumption on the client side. We do not focus on the server's energy consumption, assuming that the alerting/controlling system does not run on a battery-operated device. This setup allows us to measure both the Raspberry PI's energy consumption and the average AoI over the experiment time interval. To ensure precise AoI measurements, the Raspberry PI and the laptop must be adequately synchronized. We achieve this using the NTP protocol, with the two devices connected via a dedicated Ethernet connection, in which the laptop acts as the time-reference server for the Raspberry PI. This Ethernet connection, characterized by an extremely low latency, is used solely for synchronization traffic, and it is turned off just before an experiment run starts, to avoid including its power consumption in the results. All other traffic between the two devices flows via the Wi-Fi connection. From the instantaneous AoI measurements, we compute the average AoI over the experiment time interval. This configuration represents a common scenario in CPSes, where a single board computer acts as a sensor device, typically collecting environmental information that must be quickly transmitted. The laptop server can be seen as a control application managing data collection and applying control logic sensitive to the freshness of the obtained data.

The Web server is Apache 2.4, running a PHP script that receives the client's data and sends back the reception time in the HTTP reply. The client, once the reply is received, can compute the AoI as the clocks are synchronized via NTP. The AoI is computed on the client side just for practical reasons (it is actually the AoI as observed by the server) so to collect the AoI trace together with other collected data: the energy consumption trace produced by the power monitor, a JSON object summarizing the total energy needed, and pcap files of the Wi-Fi traffic. The latter files have been collected using tcpdump on the Raspberry PI and can be useful for detailed analysis. The client is implemented in Python and uses the urllib3.PoolManager to generate the HTTP requests.

Another machine, not shown in Figure 2, is used to carry out the Wi-Fi deauthentication attack. The attacker's machine is connected to the same LAN via a wired connection, to receive input from the controller. Its Wi-Fi interface is in monitoring mode and is used to carry out the attack. The attacker mounts a single attack for each experiment, 15 seconds after the experiment begins (i.e., at its central instant).

V. Results and Discussion

For each security configuration, we carried out 10 independent repetitions of the experiment. Both metrics, average AoI and consumed energy, are characterized by a significant difference when the system is under Wi-Fi deauthentication attack compared to normal operations, as it can be seen in Figure 3. The large difference is due to the loss of connectivity experienced by the client device, which has to reconnect to the Wi-Fi access point before HTTP and HTTPS communication can be restored. The interrupted communication period makes the time needed to complete the set of 60 request-response cycles longer. The longer period obviously also increases the total energy needed. The same applies to the AoI that increases significantly during the disconnection period, as no fresh information can be transferred, and in the end leading to significantly higher average AoI values. The time needed by the Raspberry PI to reconnect to the Wi-Fi access point and then to the server is characterized by high variability. The dispersion can be seen by the amplitude of the boxplots in Figure 3 related to the under-attack experiments compared to the normal ones. On the other hand, no statistically significant difference can be found when comparing TLS version 1.2 with TLS version 1.3, both in terms of average AoI and consumed energy (Figures 3b and 3c). In particular, we carried out a two-tailed t-test, with $p = 0.05$ significance level, and the null hypothesis - the two means are not different - is not rejected. The leaner connection setup of TLS1.3 seems to be negligible for the considered type of attack, where the reconnection time due to the lower layers of the network stack dominates. A similar analysis was carried out to compare HTTPS with both TLS versions against HTTP. Also in this case no statistically significant difference was found.

We also carried out an additional set of experiments where communication takes place via HTTPS with mutual authentication. In this case, both sides are authenticated through a certificate. Results are shown in Figure 4. Also in this case, the difference between the normal and the under-attack scenario is significant. On the contrary, the differences between the two versions of the TLS protocol are not relevant compared to the reconnection time due to the lower layers of the network stack.

Overall, a simple deauthentication attack can significantly degrade the average AoI of a system which, for the considered setup. The median value of $\widehat{AoI}$ across the ten repetitions is approximately 300 ms in the non-adversarial case, and it increases to a value that is in the 6000-7000 ms range, depending on the considered HTTP version and authentication scheme. Similarly, the energy spent increases significantly, albeit in a more limited way. In fact, from a median value that is approximately 60 J for the non-adversarial case, a value of approximately 100 J is reached when under attack.

VI. Conclusions

In this paper, we studied an AoI-sensitive system under attack using an experimental approach, testing a real denial-of-service attack via Wi-Fi against a Raspberry Pi, and obtaining real AoI measurements. We did not consider jamming like most past literature does, but rather Wi-Fi deauthentication, which is a more clever and realistic attack because it obtains the same effect with far less transmission power. We also considered various security configurations, namely without TLS (in-the-clear traffic), with TLS version 1.2 or 1.3, and with server-only or mutual authentication.

Our conclusion is that Wi-Fi deauthentication attacks have a big impact on the average AoI of the CPS and its energy consumption. Therefore, a CPS should adopt defenses against deauthentication at the data-link layer, for example Management Frame Protection [19]. On the other hand, the presence of complex security mechanisms like TLS with mutual authentication, which are often employed in CPSes to defend against men in the middle and sensor personification, does not significantly influence the impact of Wi-Fi deauthentication.

The data collected during the experiments is available on request.

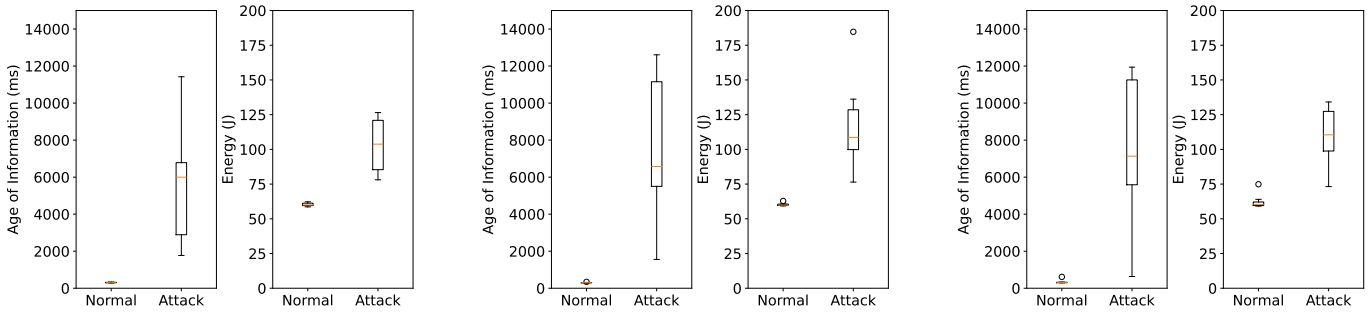
Acknowledgment

This study received funding from the project SERICS (PE00000014) under the MUR National Recovery and Resilience Plan funded by the European Union - NextGenerationEU - UNTWISTER: UsiNg digital TWIns to enable SecuriTy in cybER-physical ecosystems, CUP J33C22002810001, and by the Italian Ministry of Education and Research (MUR) in the framework of the FoReLab project (Departments of Excellence).

References

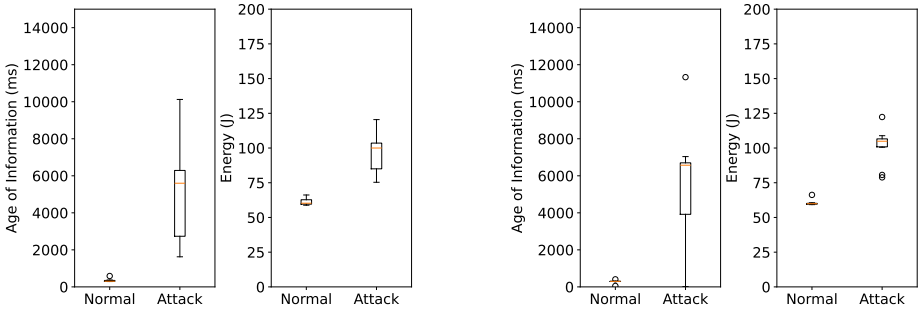
[1] D. Jia, K. Lu, J. Wang, X. Zhang, and X. Shen, "A survey on platoon-based vehicular cyber-physical systems," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 1, pp. 263–284, 2016.

[2] Y. Liu and H. Ma, "Fusion of image and point cloud for accurate obstacle detection in autonomous driving," in *Proceedings of the 6th International Conference on Machine Learning and Machine Intelligence*, ser. MLMI '23. New York, NY, USA: Association for Computing Machinery, 2024, p. 71–76. [Online]. Available: https://doi.org/10.1145/3635638.3635649



(a) no TLS (b) TLS 1.2 (c) TLS 1.3

Fig. 3: Average AoI and consumed energy when using HTTP (no TLS), HTTPS (TLS 1.2), and HTTPS (TLS 1.3).



(a) TLS 1.2 (b) TLS 1.3

Fig. 4: Average AoI and consumed energy when using HTTPS (TLS 1.2) and HTTPS (TLS 1.3) with mutual authentication.

[3] K. S. Adu-Manu, C. Tapparello, W. Heinzelman, F. A. Katsriku, and J.-D. Abdulai, "Water quality monitoring using wireless sensor networks: Current trends and future research directions," *ACM Trans. Sen. Netw.*, vol. 13, no. 1, jan 2017. [Online]. Available: https://doi.org/10.1145/3005719

[4] A. Hentati, A. Ebrahimzadeh, R. H. Glitho, F. Belqasmi, and R. Mizouni, "Remote robotic surgery: Joint placement and scheduling of VNF-FGs," in *Proceedings of the 18th International Conference on Network and Service Management*, ser. CNSM '22. Laxenburg, AUT: International Federation for Information Processing, 2023.

[5] S. Kaul, R. Yates, and M. Gruteser, "Real-time status: How often should one update?" in *2012 Proceedings IEEE INFOCOM*, 2012, pp. 2731–2735.

[6] R. D. Yates, Y. Sun, D. R. Brown, S. K. Kaul, E. Modiano, and S. Ulukus, "Age of information: An introduction and survey," *IEEE Journal on Selected Areas in Communications*, vol. 39, no. 5, pp. 1183–1210, 2021.

[7] G. D. Nguyen, S. Kompella, C. Kam, J. E. Wieselthier, and A. Ephremides, "Impact of hostile interference on information freshness: A game approach," in *2017 15th International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks (WiOpt)*, 2017, pp. 1–7.

[8] Y. Xiao and Y. Sun, "A dynamic jamming game for real-time status updates," in *IEEE INFOCOM 2018 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, 2018, pp. 354–360.

[9] X. Gao, E. Akyol, and T. Başar, "On communication scheduling and remote estimation in the presence of an adversary as a nonzero-sum game," in *2018 IEEE Conference on Decision and Control (CDC)*, 2018, pp. 2710–2715.

[10] A. Garnaev, W. Zhang, J. Zhong, and R. D. Yates, "Maintaining information freshness under jamming," in *IEEE INFOCOM 2019 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, 2019, pp. 90–95.

[11] S. Banerjee and S. Ulukus, "Age of information in the presence of an adversary," in *IEEE INFOCOM 2022 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, 2022, pp. 1–8.

[12] Y. Yang, X. Wei, R. Xu, L. Peng, and L. Liu, "Game-based channel access for AoI-oriented data transmission under dynamic attack," *IEEE Internet of Things Journal*, vol. 9, no. 11, pp. 8820–8837, 2022.

[13] V. Bonagura, S. Panzieri, F. Pascucci, and L. Badia, "A game of age of incorrect information against an adversary injecting false data," in *2023 IEEE International Conference on Cyber Security and Resilience (CSR)*, 2023, pp. 347–352.

[14] E. Rescorla and T. Dierks, "The Transport Layer Security (TLS) Protocol Version 1.2," RFC 5246, Aug. 2008. [Online]. Available: https://www.rfc-editor.org/info/rfc5246

[15] E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.3," RFC 8446, Aug. 2018. [Online]. Available: https://www.rfc-editor.org/info/rfc8446

[16] Aircrack-ng. (2023) Aircrack-ng. [Online]. Available: https://github.com/aircrack-ng/aircrack-ng

[17] Z. Neal and K. Sha, "Analysis of evil twin, deauthentication, and disassociation attacks on Wi-Fi cameras," in *2023 32nd International Conference on Computer Communications and Networks (ICCCN)*, 2023, pp. 1–7.

[18] Qoitech, "Otii Arc Pro by Qoitech," https://www.qoitech.com/otii-arc-pro/, Qoitech AB, accessed on February 21, 2024.

[19] *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 Amendment 6: Medium Access Control (MAC) Security Enhancements*, IEEE Std. IEEE 802.11w-2009, 2009. [Online]. Available: https://standards.ieee.org/standard/802_11w-2009.html