IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received April 13, 2020, accepted April 28, 2020, date of publication May 11, 2020, date of current version May 22, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2993553

Attacks and Defenses in Short-Range Wireless Technologies for IoT

KARIM LOUNIS AND MOHAMMAD ZULKERNINE, (Senior Member, IEEE)
Queen's Reliable Software Technology (QRST) Lab, School of Computing, Queen's University, Kingston, ON K7L 3N6, Canada

Corresponding authors: Karim Lounis (lounis@cs.queensu.ca) and Mohammad Zulkernine (mzulker@cs.queensu.ca)

This work was supported in part by the Natural Sciences and Engineering Research Council of Canada (NSERC), and in part by the Canada Research Chairs (CRC) Program.

ABSTRACT The Internet of Things, abbreviated as IoT, is a new networking paradigm composed of wireless and wired networks, geographically distributed and interconnected by a "secured" backbone, essentially, the Internet. It connects billions of heterogeneous devices, called Things, using different communication technologies and provides end-users, all over the world, with a variety of smart applications. IoT constitutes a new evolution for the Internet in terms of diversity, size, and applications. It also invites cybercriminals who exploit IoT infrastructures to conduct large scale, distributed, and devastating cyberattacks that may have serious consequences. The security of IoT infrastructures strongly depends on the security of its wired and wireless infrastructures. Still, the wireless infrastructures are thought to be the most outspread, important, and vulnerable part of IoT. To achieve the security goals in the wireless infrastructures of IoT, it is crucial to have a comprehensive understanding of IoT attacks, their classification, and security solutions in such infrastructures. In this paper, we provide a survey of attacks related to the wireless infrastructures of IoT in general, and to the most used short-range wireless communication technologies in the resource-constrained part of IoT in particular. Namely, we consider Wi-Fi, Bluetooth, ZigBee, and RFID wireless communication technologies. The paper also provides a taxonomy of these attacks based on a security service-based attack classification and discusses existing security defenses and mechanisms that mitigate certain attacks as well as the limitations of these security mechanisms.

INDEX TERMS IoT security, Wi-Fi, Bluetooth, ZigBee, RFID, wireless security, wireless IoT, IoT attack classification, wireless security mechanisms, attack-defense trees, attack countermeasures.

I. INTRODUCTION

Internet of Things (IoT) constitutes a new evolution of the classical Internet network in terms of diversity, size, and applications. This new networking paradigm expands the Internet from a Machine-to-Machine (M2M) communication system to a Things-to-Machine (T2M) and Things-to-Things (T2T) communication system. In other words, IoT aims to transform everything into a computer that can perform computation and communication over a network. IoT is physically perceived as a networking infrastructure composed of wireless and wired telecommunication networks. The wireless part includes, but not limited to, Wi-Fi (Wireless Fidelity), Bluetooth, ZigBee, RFID (Radio Frequency IDentification), Z-Wave, Thread, Wavenis, RuBee, 6LowPan, LoRa, 4G/LTE, NBIoT (Narrowband Internet of Things), RIIoT (Radio Industrial IoT), Tinymesh, Wi-Max (Worldwide Interoperability for Microwave Access), and 5G networks. The wired part includes, but not limited to, Ethernet, ARCNet (Attached Resource Computer NETwork), FDDI (Fiber Distributed Data Interface), PPP (Point-to-Point), Frame Relay, Token Ring, ISDN (Integrated Services Digital Network), X25, and xDSL (x-Digital subscriber Line[1]) networks. These networks are geographically distributed and interconnected through a "secured" backbone, essentially, the Internet. This diversified networking infrastructure allows billions of heterogeneous devices, called Things, to be connected to the same backbone [1], [2] and communicate with each other within the framework of smart applications [3]–[15].

The associate editor coordinating the review of this manuscript and approving it for publication was Sara Pizzi.

[1]x-DSL covers all DSL technologies (digital data over telephone lines), such as ADSL, ADSL2+, RADSL, SDSL, IDSL, VDSL, and HDSL.

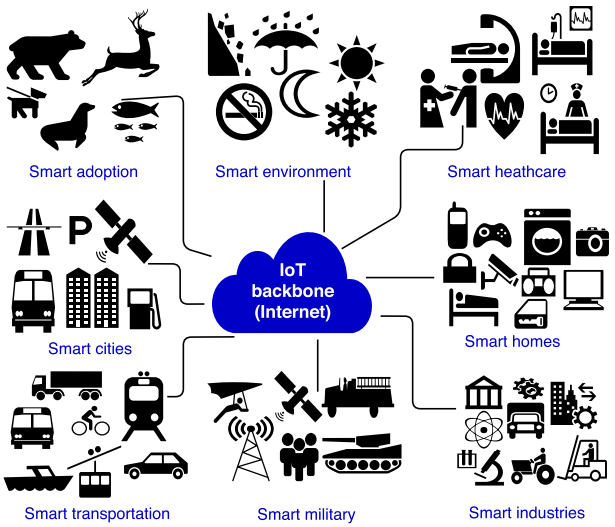

FIGURE 1. General view of Internet of Things.

Figure 1 illustrates a general view of IoT. It shows different possible IoT applications connected trough the same backbone (Internet). IoT allows various smart applications to be accessible to users worldwide. For example, on a local scale, a law enforcement vehicle can communicate with a traffic light system to automatically turn the light into red or green in case of an emergency. In a large scale, a house owner can use a tablet in one country, say Algeria, to connect and communicate with a security camera installed in his own house, located in another country, say Palestine, to monitor and keep a close eye on the location for any possible burglary.

A. IoT SECURITY

IoT brings to our ecosystem a considerable improvement in terms of social development, economic benefits, and governmental activities. On the downside, the IoT can be exploited as a networking platform to conduct large scale, distributed, and devastating cyberattacks that may have serious consequences. In fact, IoT connects an increasing and large number of heterogeneous devices for which security configurations are unknown. If those devices are not secure, they can be compromised and hijacked by cyberattackers to make them their cybersoldiers, also known as bots or zombies.

For instance, in 2015, researchers from a security firm called Rapid7[2] found that a dozen of baby monitoring camera models from eight manufacturers contained critical vulnerabilities. These vulnerabilities could be exploited by hackers to conduct nefarious actions, such as monitoring live video feeds, changing camera settings, harvesting video clips stored online, and even giving remote control to other hackers. Hence, by taking over a considerable number of connected devices (Things), cybercriminals can build large botnets and exploit them to generate attacks that affect the security of information systems at a large scale, generally in the form of a DDoS (Distributed Denial of Service) attack. An example of such an attack is the Mirai attack in 2016 [16]. A malware called Mirai was used to scan the Internet and identify a large number of IoT devices, in particular, IP cameras and routers, that run Linux and use default credentials for telnet[3] connections. The attackers were able to turn those devices into bots and use them to cause a denial of service attack on DNS (Domain Name System) servers. This consequently resulted in cutting off access to some popular websites.

Moreover, attackers can conduct local scale attacks on individual critical devices that may involve human life, such as the Stuxnet attack in 2011 [17], the Ukraine's power-grid blackout in late 2015 [18], the Jeep Cherokee attack in 2015 [19], the Brickerbot attack in 2017 [20], and the Philips lightbulbs attack demonstration in 2018 [21]. These attacks have demonstrated how catastrophic and diversified cybercrimes could be in a world where everything is a computer.

B. MOTIVATION AND OBJECTIVES

Currently, the security of the Internet of Things (IoT) strongly relies on the security mechanisms that are provided by the existing network infrastructures and technologies. We can state that all security issues of the Internet and computers have become security issues of everything as IoT transforms Internet security to everything security. Therefore, there is a serious need for a comprehensive study about IoT security issues in general and IoT attacks in particular.

In this paper, we adopt a security service-based attack classification to review the attacks that occurred in the last two decades on the most used short-range wireless communication technologies of IoT. Namely, we consider Wi-Fi, Bluetooth, ZigBee, and RFID. We also provide possible countermeasures that can be applied to mitigate or at least detect certain attacks. Since IoT is a more recent phenomenon, the reader would be wondering why we are reviewing the attacks that occurred a long time back. There exist multiple reasons: (1) The different versions or standards of each technology are compatible with previous or legacy versions and standards. This allows vulnerable versions to be used. (2) Depending on the situation, person, and country, some users may use old devices or technology standards that contain vulnerabilities. By connecting a vulnerable device to a secured infrastructure, we make the whole infrastructure vulnerable. (3) Certain governmental and provincial infrastructures need much more time than others to upgrade the technologies, devices, or security mechanisms that are used by their citizens and residents. (4) Because of security costs, the latest versions of a technology are not used on certain latest models of devices. For example, you can still find a 2020 high-end car that uses Bluetooth v3.0+HS (High Speed), which appeared in 2009. (5) Some low cost embedded IoT devices have been built offshore by third parties

[2]https://www.rapid7.com/

[3]Telnet is an unencrypted session-layer protocol that allows remote access to connected devices. It operates on the standard network port 23.

which may not be in business for a long time. Hence, any vulnerability that is discovered on those devices will not be fixed. (6) New attacks usually follow an attack methodology similar to the one employed by old attacks. A review of old attacks may help understand new attacks that may occur on the same technologies or on other new technologies, and easily identify countermeasures to mitigate the attacks.

We have limited our study only to Wi-Fi, Bluetooth, ZigBee, and RFID wireless communication technologies for the following main reasons: (1) These technologies are the most used and prominent ones for the future of IoT [22]–[26]. (2) IoT cannot be secure without securing its wireless infrastructures that adopt these technologies. (3) Most of the wireless IoT infrastructures that adopt these technologies are resource-constrained and are subject to different types of attacks.[4] (4) The wireless infrastructure of IoT is the most vulnerable part of IoT. Attackers may exploit this part as an entry point to build their attacking network over the air. (5) These technologies have been around since the late nineties and have a broader background than the new short-range technologies, such as 6LoWPAN, BackFi, Thread, Wavenis, RuBee, and Z-Wave. Hence, we can learn from the current ones to understand and prevent the same attacks from happening on the upcoming technologies. (6) For the sake of the length of the paper and topic-framing, we have not considered mid/long-range wireless communication technologies, such as Wise, M-Bus (Meter-Bus), RIIoT (Radiocrafts Industrial IoT), Tinymesh, WiMax, licensed LPWAN technologies (e.g., LTE-M, EC-GSM, and NB-IoT), unlicensed LPWAN technologies (e.g., MYTHINGS, LoRa, and Sigfox), and cellular technologies (e.g., 3G, 4G/LTE, and 5G).

C. CONTRIBUTIONS

This paper makes the following three major contributions:

– *A generic taxonomy of attacks*. We propose a taxonomy of attacks in IoT Wi-Fi, Bluetooth, ZigBee, and RFID infrastructures. This taxonomy classifies attacks based on the fundamental security services, i.e., authentication, confidentiality, integrity, and availability.

– *An extensive survey*. Following the taxonomy, we review the existing attacks that occurred on Wi-Fi, Bluetooth, ZigBee, and RFID wireless communication technologies in the last twenty years. These attacks might still occur nowadays in Wi-Fi, Bluetooth, ZigBee, and RFID infrastructures of IoT or other new technologies used in IoT infrastructures.

– *Analysis of countermeasures*. For each of the reviewed attacks, we provide the existing countermeasures that are used to mitigate the corresponding attacks and propose new possible security solutions for some of the attacks.

By providing a broader view about possible attacks in the most used short-range wireless communication technologies of IoT, security engineers will have a supportive document to easily answer the following security questions given a Wi-Fi, Bluetooth, ZigBee, or RFID IoT infrastructure: (1) What are the existing attacks? (2) What are the most important security services? (3) What are the most severe attacks? (4) What are the countermeasures to mitigate a given attack? (5) Based on the existing attacks and countermeasures, how can we mitigate new attacks on other IoT wireless communication technologies?.

[4]These attacks are different in terms of the adopted techniques, the impact, the consequences, the protocol layer levels, the type of the exploited vulnerabilities, and the compromised security services.

D. PAPER ORGANIZATION

The remainder of the paper is organized as follows: Section II discusses the existing related IoT attack research directions. Section III presents the most used short-range wireless communication technologies (Wi-Fi, Bluetooth, ZigBee, and RFID) in the resource-constrained parts of IoT as well as their provided security mechanisms. Section IV proposes a generic taxonomy of attacks in wireless IoT infrastructures. Following the proposed taxonomy, Section V, VI, VII. and VIII review the attacks against the considered technologies. Finally, Section IX concludes the paper.

II. EXISTING IoT ATTACK RESEARCH DIRECTIONS

The Internet of Things (IoT) has emerged at a very high speed while its security has been the last matter to consider. This work presents a comprehensive survey on the attacks that took place in the last twenty years in the most used short-range wireless communication technologies in the resource-constrained parts of IoT, following an attack classification. There exists a large number of IoT security research works that have either discussed general IoT security issues, including attacks, or have discussed attacks specifically on different wireless network architectures and wireless communication technologies. As IoT aims to transform everything into a computer, we believe that all research work related to the attacks that have been conducted on a variety of wireless networks and technologies are still valid and applicable to IoT. We have identified the following three IoT attack research directions (**RDs**).

RD1: IoT general security issues. A considerable amount of research work have studied IoT security in general [27]–[56]. These work discussed major security issues such as privacy, authorization, authentication, trust (trustworthiness), accountability, auditability, confidentiality, key management, and attacks at three different levels: application level (e.g., in smart city, smart healthcare, smart transportation, and smart grid), network level (i.e., Internet and cloud computing), and at the perception level (e.g., in WSN, Wi-Fi, Bluetooth, ZigBee, RFID, and Z-wave networks).

RD2: Network architecture-related attacks. Many research works have focused on attacks in particular. They surveyed and classified attacks that are related to the adopted wireless network architecture (mainly Ad hoc) such as WSNs (Wireless Sensor Networks) [57]–[65], MANETs (Mobile Ad hoc Networks) [66]–[78], and VANETs (Vehicular Ad hoc Networks) [79]–[87]. Those types of wireless networks

TABLE 1. IoT attack research directions.

Research Direction	IoT Applications	IoT Wireless Technologies	IoT Network Architectures	IoT Security Services
Notes	Some research work consider security issues within the scope of one or more IoT applications.	Certain research work have emphasized on the considered IoT technology.	Some research work have scoped their work for specific IoT architecture.	Some discuss attacks that breach security services, whereas others discuss the implementation of specific IoT security services.
RD1	– Smart environment [27,28,32,34, 35, 46, 49]. – Smart grid [27, 28, 32, 53]. – Smart healthcare [27, 28, 32, 38, 46, 49, 50]. – Smart transportation [27, 32, 46, 49]. – Others [32, 35, 36]. – General IoT application [29-31, 33, 37, 39-45, 48, 51, 52, 54-56].	– Wi-Fi [34, 36, 39, 41, 47, 54, 55]. – Bluetooth [34, 36, 39, 47, 54, 55]. – ZigBee [27,34, 36,39, 41, 42, 52, 54, 55]. – RFID [27-32, 39-43, 46, 47, 49, 50, 52, 54]. – Others [27,28, 30,34, 39, 41, 47, 54].	– WSNs [27-31,40-43,54, 55]. – Cloud [27,32, 38-40,45, 48, 49, 53, 56]. – MANETs [35]. – VANETs [35, 49].	– Authentication [27-34, 39-47, 49, 51-94, 96, 97-105, 108-120, 122, 123, 125, 127, 128]. – Confidentiality [29-33, 36, 38-43, 45-47, 52, 53-58, 60-63, 65-76, 79, 80-91, 93, 96, 98, 99-107, 109-110, 112, 113, 115-118, 121, 122, 123, 125-131]. – Integrity [29, 30, 32, 38-41, 43, 46, 47, 51, 55, 56, 58, 60, 62, 63, 66, 67, 69-72, 74-76, 78, 79, 80-91, 100-105, 108-110, 112, 113, 115, 117, 119-122, 125-127]. – Availability [29, 31, 32, 39-43, 45-53, 55-76, 78-89, 91, 94, 96, 97, 100-102, 104, 108-110, 113-116, 118, 120, 121, 123-128, 130]. – Access control [27, 33, 34, 40, 42-45, 47, 53, 60, 66, 74, 81, 82, 87-90, 96, 99-101, 105, 112, 113, 122]. – Trust [27, 40, 44, 47, 52, 54, 57, 78, 79, 90, 110, 111, 122]. – Privacy [27, 29, 32, 33, 35-40, 44, 53, 58, 79, 81-84, 87, 109-112, 115, 117, 122, 130, 131]. – Others [32, 33, 38, 40-42, 45, 47, 52, 56, 60, 66, 69, 71, 72, 74-76, 79, 81-84, 86, 88, 96, 100, 108-110, 112, 115, 122, 125].
RD2	– Smart transportation [79-87]. – General applications [57-78, 84].	– Wi-Fi [58, 62, 66, 76]. – Bluetooth [58, 82]. – ZigBee [58, 60, 82]. – RFID [58]. – Others [58, 60, 82].	– WSNs [57-65, 76]. – MANETs [66-78, 84]. – VANETs [79-87].	
RD3	– Smart environment [129]. – Smart transportation [112, 113]. – Smart healthcare [112]. – Others [112, 113]. – General applications [89-131].	– Wi-Fi [89-94]. – Bluetooth [88, 89, 95, 96, 97-102]. – ZigBee [103-107]. – RFID [88, 108-114]. – Others [89, 90, 115-131].	– WSNs [88, 104, 107]. – WLAN [89-94, 97]. – WPAN [88, 89, 95-122]. – WWAN [89,90,123,124, 125-131]. – Cloud [130].	

are also the foundation of many wireless IoT infrastructures. For example, VANETs are the foundation of all IoT smart transportation infrastructures and applications.

RD3: Wireless technology-related attacks. Some research work have surveyed and classified, or reported, attacks that are dependent on the adopted communication technology. The considered technologies are heavily used in IoT. A large majority have studied security attacks considering multiple wireless communication technologies, such as Wi-Fi, Bluetooth, RFID, WiMax, UMTS, LTE, and WSNs[5] [88]–[90], whereas others have chosen to survey attacks in one particular wireless communication technology, e.g., Wi-Fi [91]–[94], Bluetooth [95]–[102], ZigBee (or IEEE 802.15.4) [103]–[107], RFID [108]–[114], 6LoWPAN (Ipv6 over Low-Power Wireless Personal Area Networks) [115]–[118], LoRaWAN [119]–[121], Zwave and Thread [122], WiMax [123]–[128], and 4G (i.e., LTE) or 5G [129]–[131].

Table 1 summarizes the three IoT security attack research directions (i.e., **RD1**, **RD2**, and **RD3**), with respect to the considered IoT applications (Column 2), IoT wireless communication technologies (Column 3), IoT network architectures (Column 4), and IoT security services (Column 5).

The Internet of Things is evolving and spreading very quickly. We believe that it is important to have a comprehensive survey about attacks that are possible on IoT short-range wireless communication technologies. Considering the related surveys, we believe that the reported attacks are not comprehensive and too general compared to what we review in this paper. Also, the way how attacks are generally reviewed in the related work makes the task of security

[5]Note that WSNs (Wireless Sensor Networks) is a resource-constrained type of wireless Ad hoc (Mobile) networks that can adopt different technologies, such as ZigBee, Wi-Fi, Zwave, or Bluetooth wireless communication technology. Thus, we believe that differentiating between a wireless technology and a type of wireless networks is important.

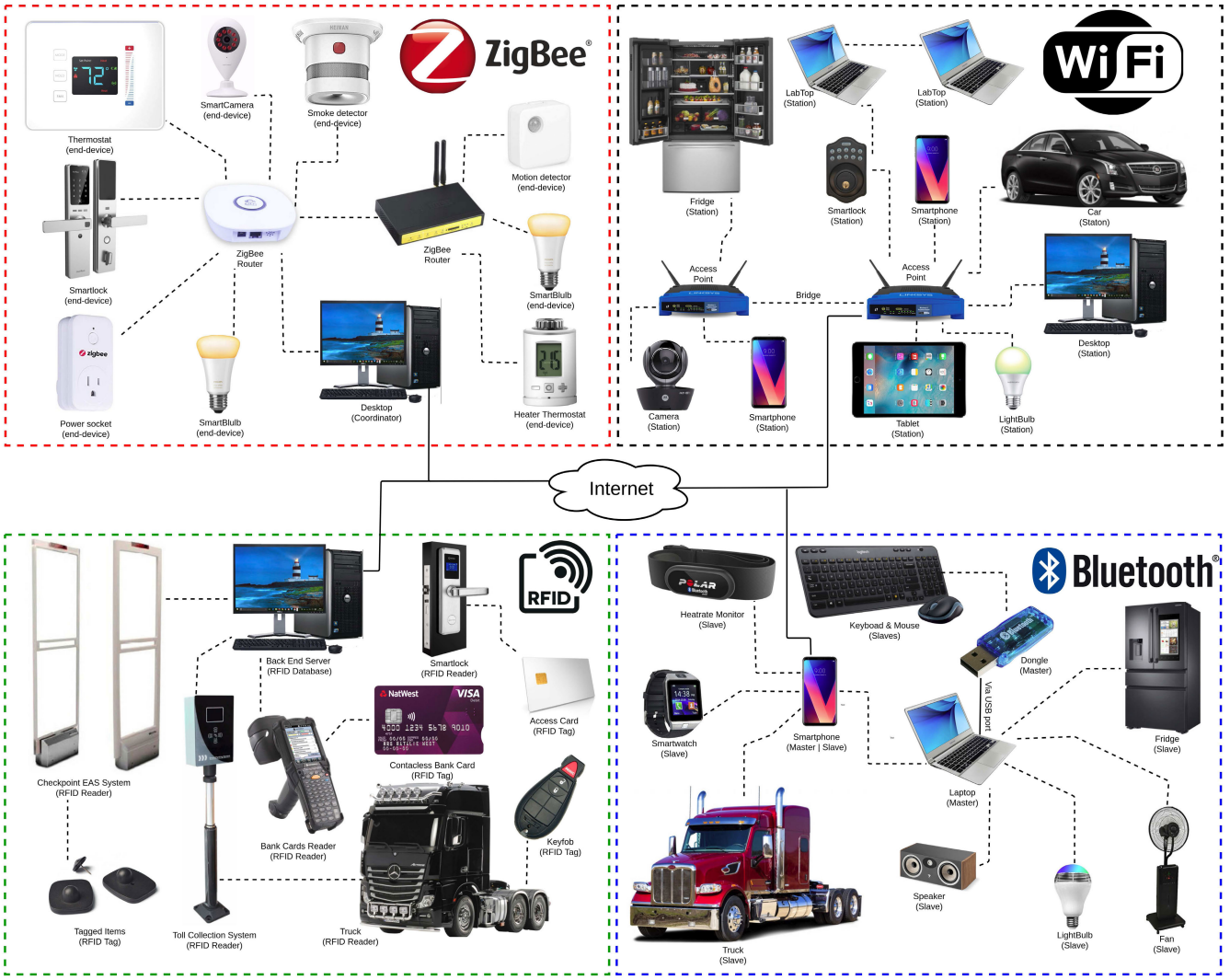

FIGURE 2. Interconnection of Wi-Fi, Bluetooth, ZigBee, and RFID IoT infrastructures.

engineers even harder since the attacks are discussed from a general perspective. Even though some existing surveys have proposed attack classifications, we find those attack classifications not straight-forward. Based on an attack, a security engineer may not directly identify the source of the vulnerability and recommend a countermeasure. We believe that choosing the right attack classification is important and useful in illustrating a holistic and comprehensive picture of possible attacks in Internet of Things.

III. IoT SHORT-RANGE WIRELESS TECHNOLOGIES

This section discusses the most used short-range wireless communication technologies in the resource-constrained part of IoT, namely, Wi-Fi (Wireless Fidelity), Bluetooth, ZigBee, and RFID (Radio Frequency IDentification). We present these technologies and discuss their security mechanisms that are used to provide security services, i.e., authentication, confidentiality, integrity, and availability. Figure 2 illustrates how the four wireless communication technologies are used to interconnect networks in the Internet of Things.

A. WI-FI TECHNOLOGY

1) WI-FI OVERVIEW

Wi-Fi (Wireless Fidelity) is a wireless communication technology based on the IEEE 802.11 standard. It allows the construction of WLANs (Wireless Local Area Networks) over both unlicensed radio bands, the 2.4 GHz ISM (Industrial Scientific and Medical) band and the 5 GHz UNII (Unlicensed National Information Infrastructure) band. It was first introduced in 1999 allowing the implementation of WLANs over a short-range with a basic transmission rate of 2Mbps. Later, Wi-Fi has significantly evolved in many aspects, such as power management, quality of service, data rate, infrastructure modes, and security. Nowadays, a Wi-Fi network can send data at 6.75Gbps [132] and reach a range up to

382km [133]. It is commonly used in domestic places, such as houses, hotels, hospitals, universities, and enterprises.

Wi-Fi allows the construction of WLANs following four different configurations: Infrastructure, Ad Hoc, bridge, and repeater. The first two modes define how Wi-Fi devices can directly or indirectly communicate with each other, whereas, the last two modes define how to extend the range of a Wi-Fi network. In the infrastructure mode, an access point, called coordinator, controls and coordinates a certain number of wireless devices called wireless clients or stations (viz., Wi-Fi in Figure 2). These wireless stations have to be associated and authenticated to the access point to be fully connected to the network. The set of wireless stations along with the access point constitutes a BSS (Basic Service Set) structure which is identified by a BSSID (Basic Service Set Identifier). This BSSID corresponds to the MAC address[6] of the access point. When multiple BSSs are connected, they form an ESS (Extended Service Set) structure identified by an ESSID (Extended Service Set Identifier) or SSID (Service Set Identifier). In an Ad Hoc mode however, wireless devices connect to each other to form different flexible network architectures such as mobile and mesh networks. In such network configurations, each wireless device can be both a wireless station and a wireless coordinator. The set of all connected wireless stations forms the structure of an IBSS (Independent Basic Service Set) identified by an SSID.

Wi-Fi adopts the CSMA/CA (Carrier Sense Multiple Access with Collision Avoidance) protocol to access the radio channel. This protocol allows Wi-Fi devices to send their data while avoiding collisions by applying the binary exponential back-off algorithm. In the exponential back-off algorithm, Wi-Fi devices promptly sense the radio channel for its availability and back off for a random time if the channel is busy. If a Wi-Fi device detects that the radio channel is not busy, it starts transmitting its data (i.e., IEEE 802.11 frames).

2) WI-FI SECURITY

Wi-Fi technology provides a number of security mechanisms. In the following paragraphs, we briefly present these mechanisms that help understand the reviewed Wi-Fi attacks and countermeasures. Interested readers are referred to the IEEE 802.11 specification documents [134] for more details.

a: WEP (WIRED EQUIVALENT PRIVACY)

This security mechanism was introduced as part of the IEEE 802.11 standard in 1997 to provide authentication, encryption, and data integrity. For authentication, WEP provides two security modes, the OSA (Open System Authentication) and the SKA (Shared Key Authentication). In the first mode, any Wi-Fi station can get connected to any access point that adopts this mode. The second mode however, is based on the use of a pre-shared secret key and a challenge-response protocol. If a Wi-Fi station proves to the access point the right possession of the secret key, it gets authenticated. For encryption, WEP applies the RC4 (Ron's Code 4) stream cipher algorithm along with an encryption key and uses the CRC-32 algorithm to generate an ICV (Integrity Check Value) code for data integrity.

b: IEEE 802.11i STANDARD

Few years after WEP was shown to be containing serious vulnerabilities [135]–[139], the IEEE proposed the 802.11i framework [140]. This framework provides stronger security mechanisms for authentication, encryption, and data integrity. Notwithstanding, due to the high demand and pressure for a secure solution to be implemented and released, the Wi-Fi Alliance quickly (in April 2003) started certifying devices based on a draft version of 802.11i under the name of WPA (Wi-Fi Protected Access). In June 2004, the final version implementing the 802.11i specification was ratified under the name of WPA2.

The IEEE 802.11i standard defines two possible authentication modes: enterprise mode, also known as WPA-Enterprise, and personal mode, also known as WPA-PSK. In the first mode, an 802.1X infrastructure is adopted. Such infrastructure consists of an authentication server, e.g., RADUIS (Remote Authentication Dial-in User Service); a network controller (authenticator) usually an access point; and the use of the EAP (Extensible Authentication Protocol). This infrastructure allows any Wi-Fi device, also known as a supplicant, to join the network and to be uniquely identified and authenticated. In the second authentication mode, a pre-shared password is used to derive a cryptographic keychain that is used for authentication, encryption, and data integrity. As the IEEE 802.11i standard was not compatible with WEP, two new encryption mechanisms have been introduced, TKIP (Temporal Key Integrity Protocol) and CCMP (CTR with CBC-MAC Protocol) [141]. A third mode called GCMP (Galois Counter Mode Protocol) was introduced in 2012 [132], [142]. Similar to WEP, TKIP mechanism uses RC4 algorithm but with a longer encryption key. It uses Michael algorithm to compute a code called MIC (Message Integrity Code) for data integrity. CCMP however, is more secure as it uses AES (Advanced Encryption Standard)[7] for encryption and CBC-MAC (Cipher Block Chaining-Message Authentication Code) algorithm for data integrity [141].

c: WPS (WI-FI PROTECTED SETUP)

This security mechanism was introduced by the Wi-Fi Alliance in 2006 to provide an easy and secure procedure to join a Wi-Fi network. Currently, four procedures have been defined: (1) PIN-based procedure, where the user introduces an 8-digit PIN code shown on the new device into the access point memory or vice-versa. (2) PBC (Push Button

[6]MAC (Media Access Control) address is a 48-bit hardware address uniquely associated to the network interface of a device to connect to a network. This address is generally used at the link and MAC protocol-layer.

[7]AES (Advanced Encryption Standard), also known as Rijndael, is a symmetric cipher established by the U.S. National Institute of Standards and Technology (NIST) in 2001 [143].

Configuration), where the user has to simultaneously push a virtual or physical WPS-button on both devices (i.e., access point and the new device). (3) NFC (Near Field Communication), where the user approaches the new device next to the access point so that a near-field contactless authentication can be performed. (4) USB (Universal Serial Bus) mode, where the user needs a USB pendrive to transfer authentication data between Wi-Fi devices.

d: OPPORTUNISTIC WIRELESS ENCRYPTION

This mechanism is defined in the RFC810. It aims to add a security layer for Wi-Fi networks that adopt the open system authentication such as public and guest networks. It uses the Diffie-Hellman key establishment protocol [144] to establish a shared key, known as PMK (Pairwise Master Key). This key is then used to derive other keys to guarantee message authentication, confidentiality, and integrity. Note that this protocol allows a Wi-Fi client and an access point to establish a shared secret key without having shared any credentials a priori.

e: MFP (MANAGEMENT FRAME PROTECTION)

This mechanism is part of the IEEE 802.11w amendment (2009). It aims to increase the security of Wi-Fi management frames by providing data confidentiality, integrity, authenticity, and freshness. It has been optional in WPA and WPA2, but it is mandatory in WPA3 which we discuss in the next paragraph.

f: WPA3 (WI-FI PROTECTED ACCESS 3)

In June 2018, the Wi-Fi Alliance announced WPA3 [145] as the next generation of Wi-Fi security. This new security mechanism aims to completely replace WPA2 mechanism. It provides multiple advantages over WPA2 such as protections against dictionary attacks (through the use of Simultaneous Authentication of Equals protocol [146], also known as dragonfly), forward secrecy, side-channel attacks, and authentication of management frames (through MFP). It allows three possible operational modes: WPA3-SAE (Wi-Fi Protected Access 3-Simultaneous Authentication of Equals), which is used when Wi-Fi devices only support WPA3; WPA3-SAE transition, also known as mixed mode, which allows Wi-Fi devices that only support WPA2 to connect to a WPA3 network; and WPA3-Enterprise 192-bit, which is used in sensitive enterprise environments, such as government and industry. WPA3-Enterprise, in particular v2.0 (December 2019), adds additional security measures to WPA2-Enterprise. For example, in WPA3-Enterprise, supplicants would not have the option of ''skip certificate validation'' or ''accept any certificate'' to complete an authentication with an authentication server, e.g., RADIUS, which was not the case with WPA2-Enterprise. This would mitigate possible evil twin attacks.

B. BLUETOOTH TECHNOLOGY

1) BLUETOOTH OVERVIEW

Bluetooth is a wireless technology based on the IEEE 802.15.1 standard. It is used for exchanging data between fixed and mobile wireless devices within a short-range and building WPANs (Wireless Personal Area Networks). It was originally conceived by the telecommunication vendor Ericsson in 1994, as a wireless alternative to RS-232 cables. It uses the free unlicensed 2.4 GHz ISM (Industrial, Scientific, and Medical) radio band and adopts the FHSS (Frequency Hopping Spread Spectrum) transmission technique to send packets while reducing interference. It employs the master-slave communication mode (viz., Bluetooth in Figure 2). Bluetooth has evolved for the last twenty years, from v1.0 (1999) to Bluetooth 5.2 (2019), coming out with better power consumption, stronger security, higher data rate, and longer range. These improved features made Bluetooth a substantial technology for different IoT applications [4]–[6], [12], [13].

To communicate, Bluetooth devices have to be associated and authenticated to each other. This is performed during an authentication procedure called pairing. The device which initiates the pairing procedure is assigned the role of a master, whereas the other devices which accept the pairing from the master are assigned the role of slaves. When a certain number of slave devices are connected to the same master device, they form a network structure called piconet. The interconnection of at least two piconets forms a scatternet. Each device in a piconet or a scatternet is uniquely identified by a 48-bit Bluetooth device address (BD_ADDR).

a: BLUETOOTH LOW ENERGY

In 2010, the Bluetooth SIG (Special Interest Group) released Bluetooth v4.0+LE (Low Energy), or simply BLE [147]. This new wireless technology includes two sub-specifications: Bluetooth smart, also known as BLE single mode; and the Bluetooth smart ready, also known as BLE dual mode. These two sub-specifications have completely different physical[8] and link layers, which result in two different protocol stacks: BLE dual mode and BLE single mode. BLE dual mode implements the classic Bluetooth stack, which is used in Bluetooth v1.1 to Bluetooth v3.0+HS (High Speed), as well as the Bluetooth smart stack. BLE single mode implements only the Bluetooth smart stack. Bluetooth devices operating over the single mode stack are not compatible with classic Bluetooth devices [147].

2) BLUETOOTH SECURITY

Bluetooth technology provides security services through different security mechanisms. In the following paragraphs, we present the commonly used security mechanisms.

a: DISCOVERABLE MODE

Bluetooth allows devices to be set to a private mode known as non-discoverable mode. This mode hides the presence of a device from other Bluetooth devices. A device in this mode

[8]Classical Bluetooth applies FHSS (Frequency Hopping Spread Spectrum) by hopping over 79 channels at 1600 hops per second, whereas BLE applies FHSS over 40 channels at the same hopping rate.

can only be reached out by other Bluetooth devices that know its Bluetooth device address.

b: PAIRING MECHANISM

Pairing allows two Bluetooth devices to authenticate each other and negotiate on a set of security parameters to derive a master key called link key. This link key is employed further to derive other keys that will be used to guarantee secure communications. Currently, there are three Bluetooth pairing mechanisms: the legacy pairing used in Bluetooth v1.0 to v2.0+EDR (Enhanced Data Rate), the SSP (Secure Simple Pairing) used in v2.1+EDR (Enhanced Data Rate) to v4.1+LE (Low Energy), and the Secure Connections used in Bluetooth v4.2+LE to Bluetooth 5.2:

– *Legacy pairing.* This mechanism is based on the use of a PIN code or a passkey. The same PIN code or passkey must be introduced into the pair of Bluetooth devices to be paired. The PIN code is used as a seed to generate the link key which is employed further to derive keys for mutual authentication and encryption[9] [149].

– *SSP (Secure Simple Pairing).* This mechanism uses the ECDH (Elliptic Curve Diffie-Hellman) key establishment protocol [144] along with the public-private key pairs of both Bluetooth devices to be paired, to generate the link key. The link key is then employed to further derive keys for authentication and encryption. The SSP pairing provides four possible association modes that are flexible in terms of device input/output capabilities: Numeric Comparison, Passkey Entry, Just Works, and Out of Band [149].

– *SC (Secure Connections).* This mechanism upgrades the SSP (Secure Simple Pairing) to utilize longer key sizes and stronger algorithms. For example, the SSP used algorithms that are based on SAFER+ for encryption and authentication, and P-192-ECDH with HMAC-SHA256 for key generation. The Secure Connections however, uses AES-CTR for encryption, HMAC-SHA256 for authentication, and P-256-ECDH with HMAC-SHA256 for key generation. It has also added message integrity service using AES-CCM algorithm.

c: CONFIDENTIALITY MECHANISM

Bluetooth has three encryption modes but only two of them provide confidentiality: encryption mode 1, in which there is no encryption; encryption mode 2, in which only unicast traffic is encrypted; and encryption mode 3, which encrypts all the traffic. From Bluetooth v1.0 to Bluetooth v4.1+LE, the encryption is based on SAFER+, while Bluetooth v4.2+LE to 5.2 uses AES.

d: SECURITY MODES

Bluetooth security modes define when and where security procedures such as authentication and encryption shall be initiated. There are four different security modes [149], security mode 1, 2, 3 and 4. Security mode 1, also known as unprotected mode, provides no security procedures. The security mode 2, also known as service-level enforced security mode, allows the initialization of security procedures after link establishment but before logical channel establishment. Security mode 3, also known as link-level enforced security mode, initiates security procedures before the physical link is fully established. Finally, the security mode 4 allows security procedures to be initiated after the physical and logical link setup. Security modes 1, 2, and 3, use the legacy pairing, whereas security mode 4 uses the SSP or SC pairing. Mode 4 is supposed to be the most secure one.

e: BLE SECURITY

BLE (Bluetooth Low Energy) defines two main pairing modes: legacy pairing and SC (Secure Connections). The BLE legacy pairing applies the SSP (Secure Simple Pairing), which is used in classic Bluetooth (from Bluetooth v2.1+EDR to Bluetooth v4.1+LE) but without the application of ECDH (Elliptic curve Diffie-Hellman). The BLE Secure Connections upgrades SSP pairing in Bluetooth v4.2+LE to Bluetooth 5.2. It uses ECDH, longer keys, and provides data integrity. In BLE legacy pairing, only three association modes are possible: Just Works, Passkey Entry, and Out of Band (i.e., through NFC or Wi-Fi technology). In BLE Secure Connections, Numeric Comparison is added. Besides Numeric Comparison, none of the previous association modes provides protection against passive eavesdroppers. BLE employs AES-CCM for data encryption.

C. ZigBee TECHNOLOGY

1) ZigBee OVERVIEW

ZigBee is a wireless communication technology based on the IEEE 802.15.4 standard. It allows resources-constrained devices, such as power-limited devices, to communicate over the radio and form a WPAN (Wireless Personal Area Network). It is the most used wireless communication technology in home automation and smart lighting [150]. It was initially conceived in 1998 and then standardized in 2003. ZigBee has evolved in the last sixteen years, from the first version of ZigBee (2004) to ZigBee 3.0 (2016) [151], coming out with better power consumption, flexibility, inexpensive deployment [152], security, and new network topology options [150]. ZigBee uses 2.4 GHz ISM (Industrial, Scientific, and Medicine) band but can also operate on other bands [153]. It allows a nominal range of 10 to 100 meters with a data rate varying from 20Kbps to 250Kbps [152], [154]. It uses the DSSS (Direct Sequence Spread Spectrum) technique and adopts CSMA/CA (Carrier Sense Multiple Access Collision Avoidance) to access the radio channel.

In a ZigBee network (viz., ZigBee in Figure 2), a ZigBee device can be either a Zigbee coordinator (ZC), a ZigBee router (ZR), or a ZigBee end-device (ZED). The ZigBee coordinator has the highest capabilities and constitutes the trusted root center of the network. The Zigbee router acts as an intermediate router forwarding and relaying data to

[9] All algorithms used in legacy paring to derive keys are based on SAFER+ (Secure And Fast Encryption Routine +) block cipher [148].

other devices. Finally, the ZigBee end-device generally has sensing capabilities (e.g., detecting motion, smoke, or pressure) and can communicate with its immediate parent device but does not route any data to other devices [155]. ZigBee comes with a number of advantages, such as the provision of long battery lifetime, the support of a large number of nodes in a network (65,000), easy deployment, low costs, and global usage.

2) ZigBee SECURITY

ZigBee technology has the following five main concepts of security [156]:

a: TRUST CENTER

In a ZigBee network, the device that has more physical resources than all other devices is called the network coordinator or trust center. All other devices are called nodes. The network coordinator is assigned the responsibility of managing the security of the whole network by creating and managing three types of security keys: master, link, and network. The master key is used for securely exchanging other secret keys. Link keys are per-link keys used to encrypt messages that are exchanged between two nodes. Finally, the network key is used by new nodes joining the network.

b: AUTHENTICATION AND ENCRYPTION

In ZigBee, the data is encrypted using 128-bit AES algorithm with CCM* mode. This operational mode is a slightly modified version of CBC-MAC (Counter with Cipher Block Chaining Message Authentication Code), allowing authentication and encryption [157].

c: DATA INTEGRITY AND FRESHNESS

ZigBee uses CCM* algorithm to generate the message integrity code. This code ensures that the data has not been altered while being exchanged. It also uses a 32-bit frame counter to distinguish between new and old frames for data freshness [157].

d: SECURITY LEVELS

ZigBee provides two different security levels, namely, high security (commercial security) and the standard security (residential security). The key difference between these two levels resides in how the cryptographic keys, such as the network key, are managed and distributed over the network. The high security provides key confidentiality by allowing the network controller to send the network key in an encrypted format. However, in the standard security level, the network key is sent unencrypted. This makes it easy to be eavesdropped and learned by an attacker as demonstrated in [156]. Meanwhile, the high security mode entropy relies on a pre-installed master key, which is shared among all ZigBee devices. Therefore, the compromise of one single node jeopardizes the entire network.

e: KEY MANAGEMENT

In a ZigBee network, security keys are distributed in three different ways. The first way (in the high-security level) consists of transmitting the keys, such as the network key, in its encrypted form. The second way (in the standard security level) consists of transmitting the network key unencrypted. The last mode, which is a trade-off between usability and security, consists of manually installing the keys, such as the network key, onto each legitimate device.

D. RFID TECHNOLOGY

1) RFID OVERVIEW

RFID (Radio Frequency IDentification) is a wireless technology designed for automatically identifying, tracking, and collecting data from entities such as objects, humans and animals. It can be viewed as ''a means of explicitly labeling objects to facilitate their perception by computing devices'' [110]. It relies on tagging objects to identify, track, and collect data from them using the concept of tag-reader (viz., RFID in Figure 2). An RFID-tag or transponder is the unit (e.g., microchip implant) that stores identification information used to identify and track the object carrying the tag. It is mainly composed of a chip for storage and computation, optional battery for power supply, and an antenna for communication [158]. The RFID-reader however, is a mobile or fixed device that wirelessly interrogates RFID-tags for object identification and tracking. Note that the term ''reader'' purely indicates reading capabilities only. However, in practice, an RFID-reader can also write on the RFID-tag memory. A third party known as backend database is sometimes used in RFID systems. This backend database is requested by the RFID-reader for each tag identification through secure channels.

RFID technology uses different frequencies depending on the type of RFID-tag being used, and the RFID-application. RFID-tags are usually categorized into LF (Low Frequency), HF (High Frequency), UHF (Ultra High Frequency), or SHF (Super High Frequency) transponders. LF-tags (ISO/IEC 18000-2) and HF-tags (ISO/IEC 18000-3) operate on 124 kHz - 135 kHz [110] and 13.56 MHz, respectively, and use the inductive coupling (backscattering) to harvest energy from nearby RFID-readers. They allow a communication range of 0.5m up to 1m. UHF-tags (ISO/IEC 18000-6) and SHF-tags (ISO/IEC 18000-4) operate on 860 MHz - 960 MHz [110], [158] (300 MHz - 928 MHz [159]) and 2.4 GHz - 5.8 GHz, respectively, and use the electromagnetic coupling to harvest energy from nearby readers. UHF-RFID systems allow a range of up to 7m (10m [110]), whereas SHF-RFID systems allow shorter ranges (1-2m). Finally, a ''subtype'' of RFID technology, called NFC (Near Field Communication), is used in most contactless and proximity card (ISO/IEC 14443) applications. It operates on 13.56 MHz band and uses induction coupling. It complies with most RFID standards and adds new features and functions. For example, in NFC technology, a device

(e.g., NFC-capable smartphone) can be set to operate either as an NFC-reader or as an NFC-tag, which is not possible in standard RFID systems.

Commercially, there are two types of RFID-tags, namely, passive and active RFID-tags. Passive RFID-tags are composed of an integrated circuit that mainly holds a processor, limited memory storage, and a radio transceiver. They get power supplied upon the reception of a short-range radio signal from RFID-readers. Active tags however, which are more expensive and less error-prone, have their local power source, e.g., battery, and radio transceiver. They can transmit data in response to a received message from an RFID-reader for a much longer range than a passive RFID-tag.

2) RFID SECURITY

As RFID-tags can be attached to any object or implanted in any living-being, the possibility of reading personally-linked information without consent has raised serious privacy concern. This concern resulted in the development of many security mechanisms to provide security properties as discussed in the following paragraphs.

a: KEY MANAGEMENT

Most RFID applications use symmetric cryptography, e.g., 3DES in ePassports. In such cases, a key management scheme is used. This is because tags and readers share a secret key that is tag-specific, and none of the two parties can start identifying itself to the other party. In fact, on the one hand, if the tag starts identifying itself by sending its identity in plaintext, all other readers operating on the same frequency can read that identity and trace that tag. On the other hand, if the reader starts authenticating itself to the tag, it does not know which secret key to use since it does not know which tag it is interrogating. Some schemes have been proposed in the literature to solve this paradox [160]–[165].

b: AUTHENTICATION

To provide authentication, RFID applies challenge-response-based authentication protocols. Initially, a symmetric key is shared between the tag and reader. The RFID-tag (also known as prover) proves to the RFID-reader (also known as verifier) the right possession of the key without revealing it. This consists of sending a challenge from the reader to the tag. The latter performs some cryptographic operations using the shared key to produce a response and sends it back to the RFID-reader. The RFID-reader performs slightly the same cryptographic operations using the shared key to check whether the results of its computations are equal to the ones received from the RFID-tag. If the results are similar, the RFID-reader authenticates the RFID-tag. If a mutual authentication is required, the protocol runs in reverse. Existing authentication protocols are based on symmetric cryptography [166]–[168]. Asymmetric cryptography however, is less frequently adopted [169]–[171].

c: DISTANCE-BOUNDING PROTOCOL

The distance-bounding protocol is a lightweight authentication protocol that in addition to checking that one communication party (e.g., tag or reader) possesses the correct secret key, checks whether the distance between the reader and tag is below a given threshold [172]. This distance is measured by either the signal strength RSSI (Receiving Signal Strength Indicator) [173] or the RTT (Round Trip Time) that takes for an RFID-reader to send a challenge and receive its response from an RFID-tag. Conceptually, a distance-bounding protocol runs in three phases: (1) The slow phase, also known as initial or setup phase, where the tag and reader agree on session parameters, such as nonces. (2) The fast phase, also known as distance bounding phase, timed phase, or critical phase, where challenge-response rounds occur and the RFID-reader measures the round trips. (3) The verification phase, also known as final signature or authentication phase, where the reader ensures that the fast phase was executed faithfully so that it can use the RTT to calculate the distance. This is done by checking the correctness of all round-trip times and the RFID-tag's proof of knowledge of a valid signature.

d: BACKWARD SECRECY

It consists of protecting the confidentiality of exchanged messages even if an attacker manages to find out the keys that were used to encrypt former messages. This is generally performed by refreshing the key by hashing it along with a timestamp [165].

IV. TAXONOMY OF WIRELESS IoT ATTACKS

In general, Internet of Things (IoT) is subject to two types of security threats, namely, accidental threats and intentional threats. Accidental threats represent the set of threats which are not expected and not involving any intentional parties. It principally targets the physical security of IoT, such as fires, earthquakes, floods, pandemics, explosions and landslides, or software security, such as device failures and software bugs. Intentional threats however, also known as attacks, assume the involvement of an intended party, known as attacker, who undertakes a set of illegal actions to cause harm to the IoT infrastructure. If an attack is successfully conducted, we call it an intrusion. Technically, an attack is an intrusion attempt. If an attack has been performed using only information technology utilities, e.g., computers and software, we call it a cyberattack. In this paper, we limit our research scope to consider only outsider[10] intentional threats or attacks. These attacks aim to compromise security services in IoT. In the following paragraphs, we briefly define the fundamental security services, also known as information assurance pillars, defined in the DoD[11] Information Assurance Certification and Accreditation Process [174]:

[10] Attacks can be classified either as insider or outsider attacks. Insider attacks are generated by a trusted and authorized party within the network (e.g., dishonest employee), whereas outsider attacks are generated by an unauthorized party from outside the network (e.g., a hacker).

[11] U.S. Department of Defense.

Authentication. This service aims to prove that an entity, e.g., an individual, software, or device, is effectively what it claims to be. It is generally set up by proving the possession of a secret (something you know, e.g., password or key), possession of a personal physical item (something you have, e.g., access card), and/or personal features (something you are, e.g., fingerprints, facial, and iris recognition).

Confidentiality. It is also known as secrecy. This service aims to protect the content of the stored and transmitted data from being disclosed to unauthorized parties. It is essentially carried out using encryption or steganography techniques.

Integrity. This service aims to guarantee that the content of stored or transmitted data has not been accidentally or intentionally been modified.

Availability. This service assures that system services and resources are instantly and continuously available for users, when needed.

Non-repudiation. This service prevents any involved party from denying any performed legal or illegal operation (e.g., sent, received, executed, or modified). It is generally provided by the use of digital signature technology which is commonly used in asymmetric cryptography.

As most of the security mechanisms discussed in Section III do not address non-repudiation service, we do not survey this service. Besides the considered fundamental security services, we do not deny the existence of many other overlapping security services, which include but not limited to, auditability, accountability, authorization, trust, traceability, anonymity, liveness, and synchronization. It is not possible to derive a useful orthogonal classification by considering all the above mentioned security services. As a result, we review the existing work only with respect to the fundamental security services, i.e., authentication, confidentiality, integrity, and availability that also cover the other security services.

Classifying attacks has always been an effective practice to help security engineers better understand possible attacks on a given information system. By providing a classification of attacks, one can easily focus on specific types of attacks rather than having a haphazard long list of attacks that would require additional efforts to understand and filter attacks of one's concern. A large number of attack classifications have been proposed in the literature [175]. The most fundamental ones are as follows: the active-passive attack classification proposed by Kent [176], the internal-external attack classification used by McNamara [177], the protocol layer-based attack classification introduced by McHugh [178], and the security service-based attack classification proposed by Stallings [179].

We believe that active-passive [176] and internal-external [177] classifications are too broad and much more abstract. The protocol layer-based classification [178] classifies attacks according to the protocols being exploited to conduct attacks. This classification becomes hard to apply when different protocol stacks (due to the heterogeneity of IoT) are being used. Moreover, as some attacks operate at multiple layers, it becomes hard to determine which layer an attack belongs to, which may lead to different opinions regarding the classification of a given attack.

Stallings' classification [179] appears more general and decisive. It classifies attacks according to which security service is compromised, essentially, authentication (fabrication), confidentiality (interception), integrity (modification), and availability (interruption). We choose to adopt this classification scheme with some modifications. We enrich the classes of this scheme with a new class called domination.[12] This class is used to group attacks that compromise more than one security service at a time. Therefore, the new classification scheme first classifies attacks according to affected wireless communication technology. Second, in each wireless communication technology, attacks are classified into the following five classes:

Fabrication. This class includes attacks that aim to impersonate trusted entities in IoT infrastructures to gain certain privileges and perform illegal actions. For example, an attacker spoofs a master entity in a wireless IoT network and orders slave entities to change their functions.

Interception. This class covers all types of attacks that aim to compromise the confidentiality of IoT wireless infrastructures. For example, an attacker (in this case called eavesdropper) captures wireless traffic over the air and analyzes the traffic to extract confidential and private information.

Modification. This class covers all types of attacks that aim to illegally modify the content of messages and stored data in a wireless IoT infrastructure. As an example, in multihop-based infrastructures, an attacker may intercept network messages, change their contents, and then relay the messages to IoT nodes that have not yet received those messages.

Interruption. This class contains attacks that aim to deny legitimate parties to benefit from a set of services provided by an IoT infrastructure. For example, an attacker may cause a set of nodes in a wireless IoT network to shut down.

Domination. This class comprises attacks that aim to compromise multiple security services at a time. An attack in this class can be a pre-condition for other attacks of other classes. For example, an attacker may crack the password of a Wi-Fi network. This password is then used to generate all cryptographic keys used for authentication (fabrication), encryption (confidentiality), and data integrity (modification). By knowing the network password and impersonating the network access point, the attacker can shut down the network or deprive specific clients for connecting (availability).

We propose the attack classification scheme illustrated in Figure 3 to group the attacks that occur in IoT short-range wireless resource-constrained infrastructures with respect to the considered wireless technologies. In the next sections,

[12] This term is taken from the concept of domination in graph theory. It has been used in [180] to define a set of vital vulnerabilities that can be exploited to generate different types of attacks.

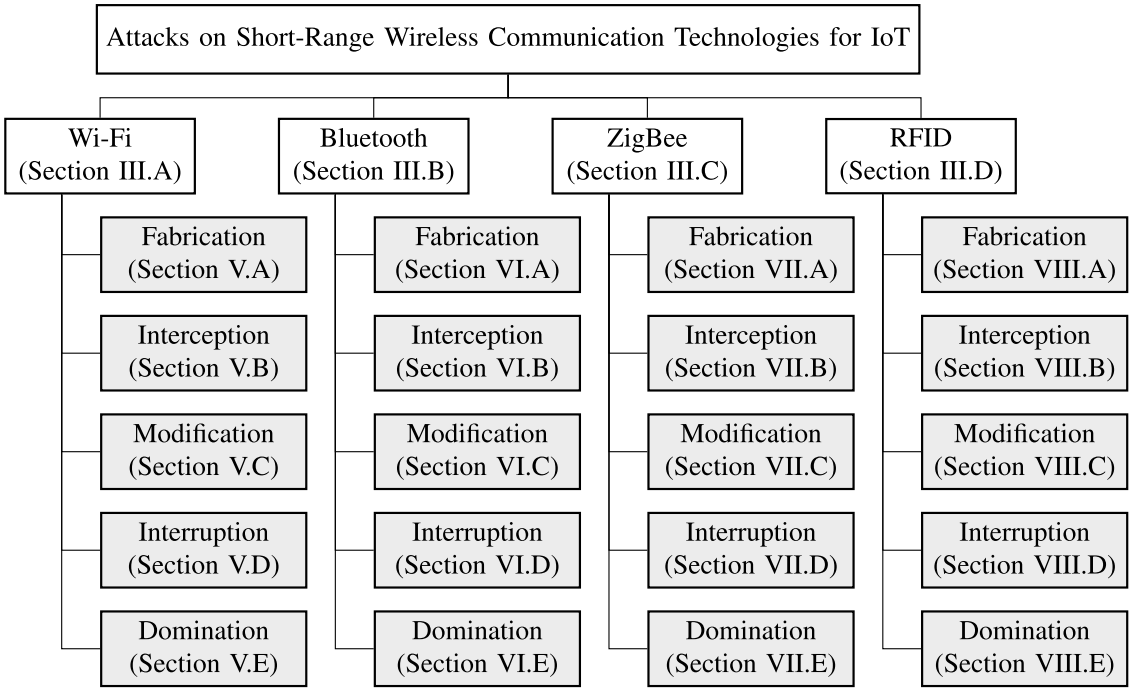

FIGURE 3. Wireless IoT Attack classification scheme w.r.t. Wi-Fi, Bluetooth, ZigBee, and RFID technologies. The corresponding subsection where a technology along with its security mechanisms and attacks are discussed, is indicated in parenthesis.

we review the attacks that occurred in the last two decades on the considered four wireless communication technologies.

V. ATTACKS ON WI-FI TECHNOLOGY

A. FABRICATION ATTACKS ON WI-FI

In the following subsections, we present the attacks that violate the authentication service in Wi-Fi IoT networks. These attacks are mainly due to a partial implementation of authentication on network traffic or due to some flaws in the authentication protocols. Figure 4 illustrates an attack-defense tree[13] [185] based on fabrication attacks on a Wi-Fi infrastructure. Attacks are shown by red circles (◯), whereas defenses are depicted in green squares (□). Attack refinements are depicted by solid lines (◯−◯), whereas attack mitigations are represented by dashed lines (◯. . .□). The root node in an attack tree represents the final goal of the attacker. The intermediate nodes show subgoals. Finally, leaf nodes represent basic (atomic) attacks.

1) ENTITY SPOOFING

Identity spoofing. In this scenario, an attacker spoofs the identity of a Wi-Fi device to impersonate it and gain certain privileges. This can be done by spoofing the MAC address[6] of the target Wi-Fi device, the SSID (i.e., in case of spoofing an access point), or both. This attack is easy to implement since nowadays most Wi-Fi network interfaces support the MAC address changing option as well as the "Master mode" to emulate Wi-Fi access points.

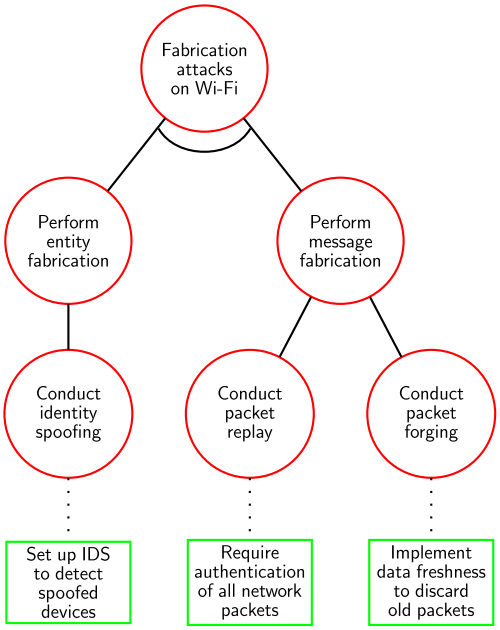

FIGURE 4. An attack-defense tree based on fabrication attacks on a Wi-Fi IoT infrastructure (◯: attacks, □: defenses, ◯−◯: attack refinements, and ◯. . .□: attack mitigations).

Countermeasure. Although it is not that easy to mitigate spoofing, detecting such activity is rather possible.

[13] Attack-Defense Trees are graphical security models used for logically and graphically representing attacks and their defenses. In these trees, conjunctive refinements (AND) are graphically represented by ∧, whereas disjunctive refinements (OR) are represented by ∨ [181]–[184].

Using wireless intrusion detection systems [186], it is possible to detect the presence of two identical devices operating in the network [187]. For instance, a spoofing access point can be localized by analyzing synchronization frames[14] generated by access points and detecting the presence of frames carrying the same BSSID and SSID, but with different timestamps.

2) MESSAGE SPOOFING

a: PACKET FORGING

As authentication is not available in Wi-Fi management and control frames, attackers can easily forge them. In most cases, the attacker creates a frame and indicates the source address as the address of a device which has higher privileges, such as the access point. The devices which receive those forged frames accept them and process them as if they were sent from the true source, i.e., the access point.

Countermeasure. Packet forging can be mitigated by requiring authentication on all types of Wi-Fi frames. For instance, every connected device should be able to verify whether a received frame is coming from a legitimate source or not. To that end, Wi-Fi devices can employ MFP (Management Frames Protection) mechanism, which is optional in WPA and WPA2, but mandatory in WPA3. An alternative consists of using WPA-Enterprise with X509 digital certificates.

b: PACKET REPLAY

Technically, WEP mechanism does not guarantee data freshness. This allows an attacker to capture previously exchanged WEP packets and replay them later on to gain some privileges. For instance, if the attacker captures the challenge-response messages during a previous WEP authentication, the attacker can infer the used keystream. By knowing the IV (Initialization Vector) that was used to generate the keystream, the attacker runs multiple association attempts until the access point asks for a response which uses that known IV. In this case, the attacker responds correctly to the challenge and gets successfully authenticated.

Countermeasure. When data freshness is correctly implemented in an authentication protocol, an attacker will not be able to replay old messages. This countermeasure has been implemented in WPA and WPA2, which aim to replace WEP. Although WPA and WPA2 are relatively more secure than WEP, it is highly recommended to switch to WPA3, which is more secure than WEP, WPA, and WPA2 mechanisms.

B. INTERCEPTION ATTACKS ON WI-FI

Wi-Fi networks have been demonstrated to be vulnerable to interception attacks [135]–[137], [188], [189]. This is fundamentally related to the broadcast nature of the wireless medium along with the implementation flaws discovered in the adopted encryption mechanisms, e.g., RC4. In the following subsections, we review the most known interception attacks on Wi-Fi wireless communication technology. Figure 5 illustrates an attack-defense tree[13] for attacking a Wi-Fi IoT infrastructure through interception attacks.

1) RECONNAISSANCE

a: SNIFFING AND PACKET ANALYSIS

Wi-Fi allows the use of a non-secure mode called open mode. In this mode, no confidentiality is provided and all Wi-Fi frames are sent unencrypted over the radio channel. An attacker can easily capture a number of Wi-Fi frames to analyze them and extract sensitive information such as credentials and private information.

Countermeasure. The most obvious security initiative that can be adopted to mitigate this attack is to use any of the encryption mechanisms provided by either WEP or WPA. However, in some circumstances, certain Wi-Fi networks are intentionally left open for user flexibility, such as the ones provided in supermarkets, large retail shops, or even airports. In such networks, security has to be implemented in the upper layers to use upper-layer security protocols, e.g., TLS (Transport Layer Security). If none of these security measures are used, it is strictly recommended not to use such networks to perform any authentication that involves the use of credentials (e.g., access email account). However, a new alternative consists of using the OWE (Opportunistic Wireless Encryption) to establish an encrypted connection. Even though a password is not shared a priori between a client and an access point, OWE allows them to establish a shared secret key using Diffie-Hellman key establishment protocol.

b: NETWORK DISCOVERY

In this scenario, an attacker uses a network adapter in ''Monitor mode''. The attacker utilizes wireless scanning tools to scan all radio channels to detect and discover nearby Wi-Fi networks. If the attacker is interested in a particular network, it can learn a considerable amount of information related to that network. The information may include BSSID, network SSID, associated stations, approximate location, radio channel, security mechanism, and the brand of the used access points. These information can be exploited for more sophisticated attacks.

Countermeasure. The network administrator should reduce the power transmission of its access points so that it only covers the operational area. It can also set the network configuration so that its SSID is not broadcasted and it is kept hidden. Finally, the use of a discrete SSID name may reduce the chance for attackers to link a particular SSID to a given organization Wi-Fi network and setting it as a target.

c: WARDRIVING

In this attack, attackers collaborate by driving around cities, neighborhoods, and villages, to scan for Wi-Fi networks that use open access mechanism or WEP. They use dedicated tools

[14]Wi-Fi frames are network packets generated at the MAC layer. Synchronization frames are commonly known as beacons. They are periodically broadcasted by access points to indicate their presence in the neighborhood.

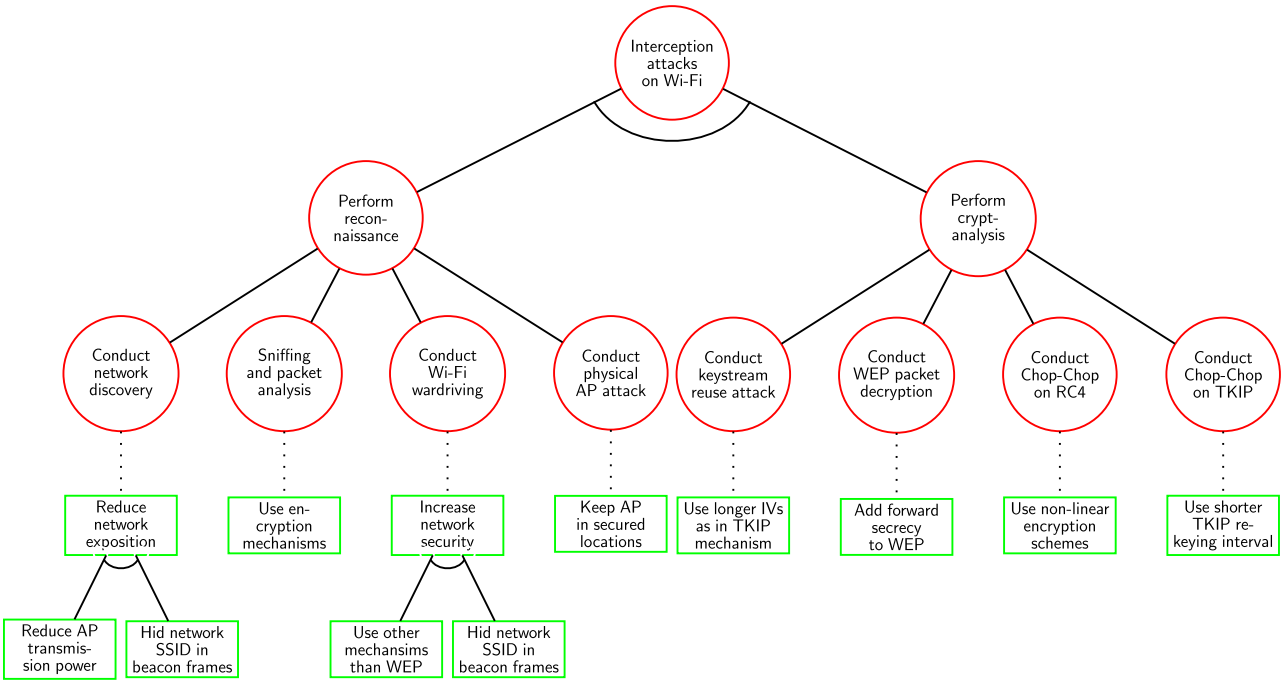

FIGURE 5. An attack-defense tree based on interception attacks on a Wi-Fi IoT infrastructure (○: attacks, □: defenses, ○−○: attack refinements, and ○. . .□: attack mitigations).

and cheap devices along with a GPS (Global Positioning System) device to record or tag the locations of the discovered insecure Wi-Fi networks in a map. The map is then shared among the attackers for future attacks. Other variants of this attack are warcycling or warbiking (using bicycle), wartraining (while inside trains), warwalking, warjogging, and wardroning or warflying (using drones).

Countermeasure. The Wi-Fi network administrator should avoid using the broken security mechanisms such as WEP, or leave the network insecure. This prevents the attackers (wardrivers) from selecting the network as a good target network. Hiding the network SSID is also a good initiative.

d: PHYSICAL ATTACK ON ACCESS POINTS

Many wireless access points have their security information (e.g., logname, password, BSSID, SSID, WPA passphrase, or WPS PIN code) printed on the back or front of the device. Thus, if the access point is not kept in a secure location, an attacker can sneak by the access point and read current credentials (if still not changed) to use them later on. The attacker can also steal devices and gain physical access to their memory to extract important information about the whole network.

Countermeasure. Network access points must be equipped with physical security. These devices should not carry any indication about the network security settings, such as passwords, usernames or IP addresses. It is also recommended to place access points at places which are not easily accessible. This prevents attackers from reaching the device.

2) CRYPTANALYSIS

a: KEYSTREAM REUSE ATTACK

In RC4, the keystream is the concatenation of a 40-bit to 104-bit WEP-key along with an IV (Initialization Vector). The IV changes randomly or incrementally for each packet depending on the implementation. This provides a unique keystream for each packet. Nonetheless, because of the small size of the IV (24-bit), all IV possible combinations (i.e., 2^{24}) are rapidly consumed (few seconds at 5Mbps) [135]. This allows an attacker who eavesdrops an ongoing communication for some time to be able to capture packets encrypted with the same keystream. By having two ciphertexts encrypted with the same keystream, the attacker can compute the xor of the plaintext of the two packets. If the attacker manages to guess at least one plaintext, it will be able to decrypt the remaining plaintexts [190].

Countermeasure. The size of the IV has been increased to 48 bits in TKIP (Temporal Key Integrity Protocol). Also, the way the IV is used in TKIP is more secure than it used to be in WEP. However, it is recommended to use CCMP (Counter Mode CBC-MAC Protocol) encryption mechanism rather than TKIP to avoid dealing with keystream reuse.

b: WEP PACKET DECRYPTION

In this attack, an attacker starts by eavesdropping a WEP authentication and tries to capture the challenge (sent in plaintext) as well as its response. Then, it xors them together to obtain the used keystream. By knowing the IV (sent unencrypted) that was used for generating the keystream,

the attacker would be able to decrypt all packets that were encrypted using the same keystream.

Countermeasure. WEP mechanism does not provide forward secrecy. Encryption algorithms that are based on xoring the plaintext by a keystream (e.g., RC4) should not apply the same keystream twice. This provides forward secrecy.

c: CHOPCHOP ATTACK ON RC4

This attack was posted in the NetStumbler forum by a person under the pseudonym KoreK in 2004 [191]. It allows an attacker to interactively decrypt the last m bytes of an RC4 encrypted packet by sending $m \times 128$ packets to the network. It exploits the linear property of the XOR logical operator used by the RC4 algorithm for encryption, and by the CRC32 algorithm to compute the ICV code for data integrity. The attacker intercepts a target encrypted packet and chops off the last byte which invalidates the ICV code of the packet. Then by assuming the plaintext value of the chopped byte, the attacker adjusts the ICV code so that it becomes valid. Indeed, when the attacker assumes the correct byte, it receives a response from the access point. This response indirectly indicates that the assumption on the last byte was correct. The attacker repeats this process to guess all remaining bytes of the packet.

Countermeasure. RC4 and CRC32 algorithms have serious flaws due to some properties such as the linearity of the XOR logical operator. Algorithms that have this kind of property must be implemented in a very careful manner so that attackers cannot decrypt messages or tamper with messages and adjust their integrity code by flipping some bits. The AES symmetric cipher can be used along with different operational modes to mitigate this attack. This requires the use of WPA2 or WPA3 mechanisms. Nevertheless, the network administrator has to make sure that its Wi-Fi device network cards do not contain the KrØØk vulnerability[15] which allows attackers to decrypt some WPA2 (AES-CCMP) packets.

d: CHOPCHOP ATTACK ON TKIP

This attack [136] allows the attacker to decrypt packets when TKIP is used with a long TKIP re-keying interval. In particular, when the range of IPv4 addresses used in a Wi-Fi network are known and the access point is operating the IEEE 802.11e, the attack becomes easier. The attacker captures encrypted ARP-requests[16] or responses and replays them a number of times in a ChopChop style on different QoS (Quality of Service) channels that still have a lower TSC (TKIP sequence counter). If the access point replies, then the guess was successful and the attacker manages to read the encrypted bytes. More sophisticated variants of this attack were reported in [137] and [189].

[15] KrØØk (CVE-2019-15126), discovered in 2019, is a hardware vulnerability residing in many Wi-Fi chips manufactured by Broadcom and Cypress.

[16] ARP (Address Resolution Protocol) is a link layer protocol that translates a logical 32-bit IPv4 address of a connected device into its physical 48-bit MAC address.

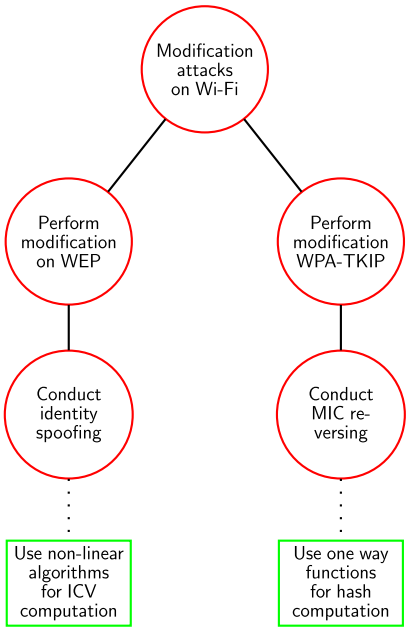

FIGURE 6. An attack-defense tree based on modification attacks on a Wi-Fi IoT infrastructure (○: attacks, □: defenses, ○−○: attack refinements, and ○...□: attack mitigations).

Countermeasure. Considering the network configurations that are exploited by this attack, an obvious solution consists of using a shorter TKIP re-keying interval.

C. MODIFICATION ATTACKS ON WI-FI

The following attacks allow attackers to modify the content of transmitted packets and to adjust their integrity code in such a way so that the packets look as if they were sent from a trusted source. Victim devices receive the packets and process them. Figure 6 illustrates an attack-defense tree[13] based on modification attacks on Wi-Fi IoT infrastructures.

1) MODIFICATION ON WEP

ICV tampering. WEP mechanism adopts the CRC32 algorithm to generate an ICV (Integrity Check Value) to guarantee data integrity. It has been demonstrated in [188] that an attacker can modify the content of a message and adjust the IVC value accordingly to make it valid. The CRC method used to compute the ICV is called a linear method (or affine) in which an attacker can predict which bits in the ICV will be flipped if the attacker changes a single bit in the message.

Countermeasure. The CRC algorithm is usually used for error detection and correction. It is not an adequate algorithm for integrity protection, in particular, to protect against intentional tampering. We highly recommend to use WPA2 or WPA3 where data integrity codes are generated using AES-Cipher Bloc Chaining-Message Authentication Code.

2) MODIFICATION ON WPA-TKIP

Micheal algorithm attack. This attack is a consequence of the ChopChop attack on TKIP described in Section V.B.2.

When the ChopChop attack is performed on TKIP, an attacker manages to get a plaintext along with its corresponding MIC (Message Integrity Check) code. The attacker would be able to reverse the Micheal algorithm (as it is not a one way function [192]) and recover the MIC key that was used to compute the MIC. This allows the attacker to modify the contents and regenerate the MIC using the disclosed key.

Countermeasure. The Micheal algorithm has been shown to contain many security flaws [192], [193]. It is not a one way function and hence can be reversed. Thus, data integrity functions that do have such properties should not be used to preserve data integrity. WPA2 or WPA3 would be a better alternative as data integrity codes are generated using AES-CBC-MAC, which is thus far considered secure.

D. INTERRUPTION ATTACKS ON WI-FI

Wi-Fi is entirely vulnerable to attacks on network availability. Practically, we emphasize on denial of service attacks. We have noticed that almost all attacks on Wi-Fi availability are due to a partial implementation of authentication in Wi-Fi. Figure 7 illustrates an attack-defense tree[13] based on interruption attacks on a Wi-Fi IoT infrastructure.

1) FRAME SPOOFING

a: DEVICE DEAUTHENTICATION

The IEEE 802.11 management frames (e.g., disassociation request/response and deauthentication request/response frames) are not authenticated when WEP, WPA-PSK, and WPA2-PSK mechanisms are used. This allows an attacker to spoof any Wi-Fi device and send forged frames over the network. In the deauthentication attack, the attacker spoofs the access point and repeatedly sends forged deauthentication frames to connected devices and cause their permanent disconnection [194]. Another way of performing this attack on certain access points was discussed in [195]. It consists of establishing a connection using OSA (Open System Authentication) with an access point using the access point's MAC address (self-connection). As a consequence, certain access points react to such authentication attempt by sending a deauthentication frame to the entire network. This would deauthenticate all connected stations.

b: DEVICE DISASSOCIATION

Similar to the deauthentication attack, an attacker spoofs the access point and sends forged disassociation requests to connected Wi-Fi devices and causes their disassociation from the network. The target devices get disassociated but not deauthenticated. They just have to re-associate to join the network again [194].

c: DEVICE REASSOCIATION

In this scenario, the attacker spoofs a legitimate Wi-Fi device which is associated with a given BSS and tries to reassociate it with a second BSS without any disassociation from the first one. In this way, the attacker creates inconsistencies in the network configuration causing several network protocol execution failures [135].

d: PACKET WASTING

The IEEE 802.11 defines a power saving mode that allows Wi-Fi devices with limited power supply to switch into sleep mode to save some energy. During the power saving period, the access point buffers all packets destined to devices in sleep mode. This requires all Wi-Fi devices to be synchronized with the access point to wake up at the right time to retrieve their respective buffered packets. The key synchronization information are periodically broadcasted by the access point using the TIM (Traffic Indication Map) field of the beacon management frame. When a Wi-Fi device wakes up from the power saving mode, it requests its buffered packets if there are any from the access point. The access point delivers the packets to its destination and cleans its buffer to save memory space. In such circumstances, an attacker spoofs a legitimate Wi-Fi device while it is sleeping and causes the access point to deliver the packets and clean its buffer. Thus, when the legitimate Wi-Fi device wakes up and requests for its packets, the access point informs that device that there is nothing buffered for it [194].

e: DEVICE DESYNCHRONIZATION

This attack scenario aims to cause disturbance on the power saving mode. The attacker spoofs the access point and sends forged beacon management frames that contain wrong synchronization information. This would cause Wi-Fi stations to wake up from the power saving mode at the wrong time [194].

f: TRAFFIC FREEZING

In this scenario, the attacker spoofs a legitimate Wi-Fi device and sends forged management frames informing the access point that the device is switching into power saving mode. This will considerably drain real-time traffic sent to the legitimate Wi-Fi device [135].

g: SLEEP DEPRIVATION

In this scenario, the attacker spoofs the access point and sends forged beacon frames containing information that indicates the presence of buffered packets for devices in power saving mode. The devices in the power saving mode send a request to retrieve their packets and stay awake for the entire beacon interval if a response is not received. By repeating this process, the attacker prevents the legitimate Wi-Fi devices from using the power saving mode and thereby drains their batteries [135], [194].

Countermeasure. To mitigate the previous attacks, the 802.11 management frames must be authenticated. Originally, WEP and WPA-PSK did not provide any authentication for management frames. However, since the IEEE 802.11w amendment, it has become possible to use the MFP (Management Frame Protection) and mitigate all previous attacks. An alternative consists of using WPA-Enterprise with X509 digital certificates to provide frame authentication.

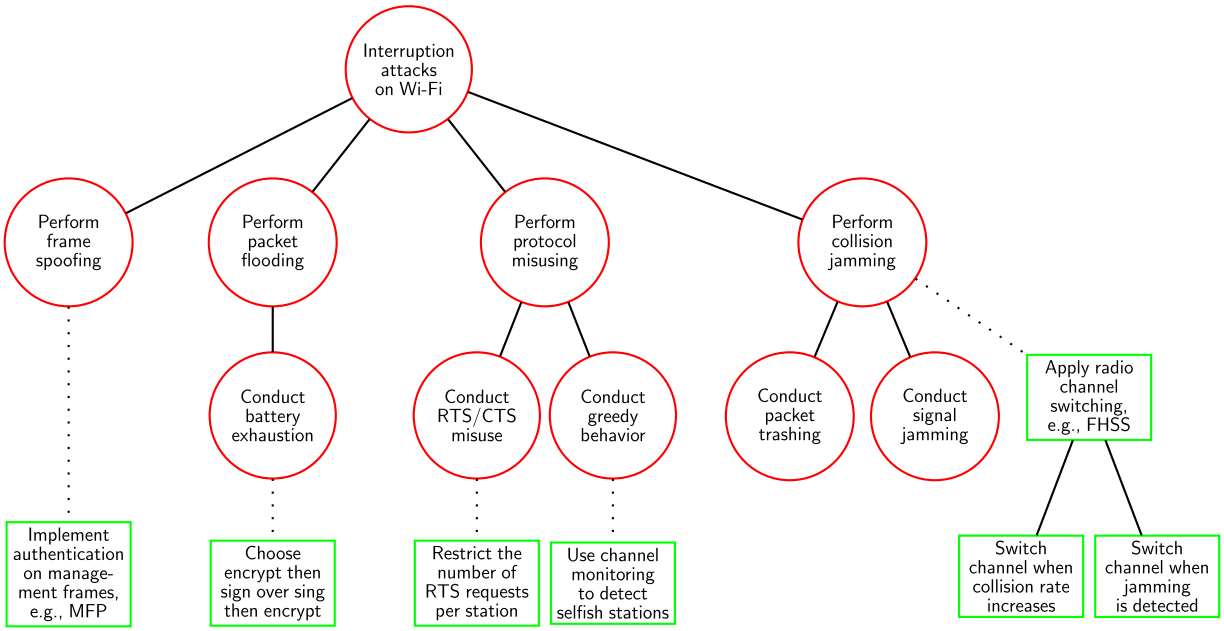

FIGURE 7. An attack-defense tree based on interruption attacks on a Wi-Fi IoT infrastructure (○: attacks, □: defenses, ○–○: attack refinements, and ○. . .□: attack mitigations).

h: CONNECTION DEPRIVATION ON WPA3

It is possible to deprive legitimate Wi-Fi supplicants that attempt to get authenticated and connected to a WPA3-configured access point. As discussed in [195] and [196], an attacker can spoof a legitimate access point and then, in a race condition, reply negatively to any connection attempt from a legitimate supplicant to repeatedly cause an authentication failure. For instance, during a WPA3-SAE (Wi-Fi Protected Access 3-Simultanious Authentication of Equals) authentication, the supplicant proposes to use a Diffie-Hellman group, e.g., group 19. The access point (authenticator) checks whether the proposed group is supported. If the proposed group is supported, the authentication goes on. However, if it is not supported by the authenticator, the latter replies to the supplicant with a negative message causing the authentication to stop. An attacker can send crafted negative replies each time the supplicant proposes a DH-group. The supplicant will be forced to abort the authentication at each attempt.

Countermeasure. As recommended in [195] and [196], future Wi-Fi supplicants and access points must be designed in such a way so that they take decisions based on a group of unauthenticated messages instead of the first unauthenticated message that is received. In this way, supplicants and access points become smarter during an authentication. This would mitigate the discussed connection deprivation attacks.

2) FLOODING

Battery exhaustion. In this attack, the attacker sends a flood of encrypted and meaningless traffic to Wi-Fi devices with limited resources (e.g., Wi-Fi sensors). Those devices consume a large amount of energy by processing that network traffic before dropping them off.

Countermeasure. This attack is effective when the target device cannot distinguish whether the incoming traffic is bogus or legitimate. Also, if the target device has to perform many cryptographic operations before concluding whether to drop or not a given packet, the attack will have a significant negative impact. If data freshness is considered, an attacker cannot flood old messages or predict future messages by spoofing devices. Moreover, the encryption algorithm should be implemented in such a way so that the target device can perform some lightweight pre-checking on the received packets before performing any expensive cryptographic operation.

3) PROTOCOL MISUSING

a: RTS REQUEST MISUSE

The IEEE 802.11 standard specifies a four-way packet transmission protocol called virtual carrier-sense or RTS/CTS (Request To Send/Clear To Send). This protocol allows a Wi-Fi device to allocate the radio channel to reliably send its packets. In this scenario, an attacker repeatedly sends RTS requests asking to allocate the radio channel for a long period. If the radio channel is granted to the attacker, all connected Wi-Fi devices are then denied from accessing the radio channel to send their packets [194].

Countermeasure. The network administrator must ensure that the radio channel is fairly shared and used among the associated Wi-Fi devices. For example, it can configure the access points to accept a limited number of RTS-requests per hour and per Wi-Fi device.

b: GREEDY BEHAVIOR

To access the radio channel using the CSMA/CA protocol, all connected Wi-Fi devices sense the radio channel for its availability. If the radio channel is found to be clear, all

Wi-Fi devices wait for a certain amount of time known as DIFS (DCF[17] Interframe Space) before starting the transmission of their packets. If the channel is found to be busy, before or after waiting for DIFS, all Wi-Fi devices wait till the radio channel becomes clear. Once it becomes clear, all Wi-Fi devices wait for another DIFS and compute a random timer (uniformly chosen in between 0 and CW-1, where CW is the contention window, usually set to 15). The timer is then decremented while the radio channel is clear and the timer is greater than 0. The first Wi-Fi device whose timer expires, starts transmitting its packets. Meanwhile, all other Wi-Fi devices abstain from decreasing their timer as long as the channel is busy. Under these circumstances, an attacker violates the rules and starts transmitting before the expiry of the shortest possible timer. This will have two disproportional impacts. First, the data rate of the attacker will increase considerably as it is taking the whole network bandwidth. Second, the data rate of the other devices will slow down and may get nullified [194].

Countermeasure. The greedy behavior can be detected using an intrusion detection system. The system monitors how the radio channel is shared and used among a certain number of Wi-Fi devices. If a device unfairly uses the radio channel, the network administrator may suspend that device from the network for sometime or disconnect it. However, such an aggressive countermeasure can be exploited by an attacker to disconnect legitimate devices by spoofing the latter and conducting a greedy behavior attack.

4) COLLISIONS AND JAMMING

a: PACKETS TRASHING

In this scenario, the attacker sends random packets exactly at the same time where a legitimate Wi-Fi device is transmitting its packets. This causes a collision of packets which results in a wrong integrity code or FCS (Frame Check Sequence). These corrupted packets are automatically discarded upon their reception due to FCS verification error [135], [197].

b: CHANNEL JAMMING

Usually, in a Wi-Fi network, communications occur on a fixed radio channel on the 2.4 GHz band. In this attack, an attacker generates random signals (noise) on the operational radio channel and causes the connected Wi-Fi devices to believe that the radio channel is busy. This drains the network performance and denies legitimate devices from accessing the radio channel to send their packets.

Countermeasure. The above two attacks can be detected by analyzing the radio channels but cannot be mitigated. One of the techniques that can be employed is to automatically switch to another radio channel when the collision or data rate goes down below a certain threshold. Also, the network administrator can set up a mechanism that can localize from where a specific network traffic or radio signal is coming from and hence may try to localize the source, i.e., attacker.

[17]DCF (Distributed Coordination Function) is a concurrent-based access mode where all Wi-Fi devices have the same chance to access the radio channel. The other mode is PCF (Point Coordination Function), where the access to the radio channel is controlled by the access point.

E. DOMINATION ATTACKS ON WI-FI

In the following subsections, we enumerate Wi-Fi attacks which compromise more than one security services. Figure 8 illustrates an attack-defense tree[13] based on domination attacks on a Wi-Fi IoT infrastructure.

1) SOCIAL ENGINEERING

Access point cloning. This attack is also known as Evil twin. In this scenario, the attacker sets its Wi-Fi adapter into master mode (i.e., access point mode) and adapts its network settings to be similar to a target access point settings (i.e., same MAC address, SSID, and radio channel). The attacker then boosts the signal strength to monopolize the radio channel and leaves the network with no security mechanism. This attracts careless Wi-Fi users to connect to the attacker's access point and use free Internet. Since no security is setup, the attacker analyzes the network traffic to extract any credentials. A more interesting scenario occurs when WPA-Enterprise is used with one-way authentication, where Wi-Fi supplicants do not have to authenticate the WPA authenticator (server). The supplicants would have the option of ''skip certificate validation'' or ''accept any certificate'' to complete the authentication. The attacker may mislead supplicants to connect to the attacker's access point instead of the legitimate one.

Countermeasure. This attack can be detected by setting a wireless IDS (Intrusion Detection System), such as Kismet [186], that can detect the presence of identical access points within the same area [187]. The IDS captures and analyzes the network traffic to detect access points with the same SSID, same MAC address, same (or different) security mechanism, but with different beacon timestamps. When WPA-Enterprise is used, mutual authentication must be established. Supplicants should not have the choice of ''skipping certificate validation'' or ''accepting any certificate''. Such a policy is enforced in WPA3-Enterprise.

2) OUT OF BAND ATTACKS

Wi-Fi backdoor. Most access points and routers, with wireless capabilities, either bought from a retail shop or offered by an ISP (Internet Service Provider), come with default security settings (e.g., logname=admin, password=admin or logname=" " and password=admin). It is the responsibility of the subscriber to change the default settings. An attacker, who is subscribed to an ISP, tests the connectivity with all possible IP addresses that are in its network subnet. For example, if its IP address is 67.193.191.125, the attacker pings all IP addresses from 67.193.191.1 to 67.193.191.254. If an IP address replies to the ping, the attacker web-browses the IP address for the login page of the remote router. If the credentials of that router are left to default and that device allows connections from outside (i.e., Internet), the attacker will be able to login into the subscriber's router and learn a

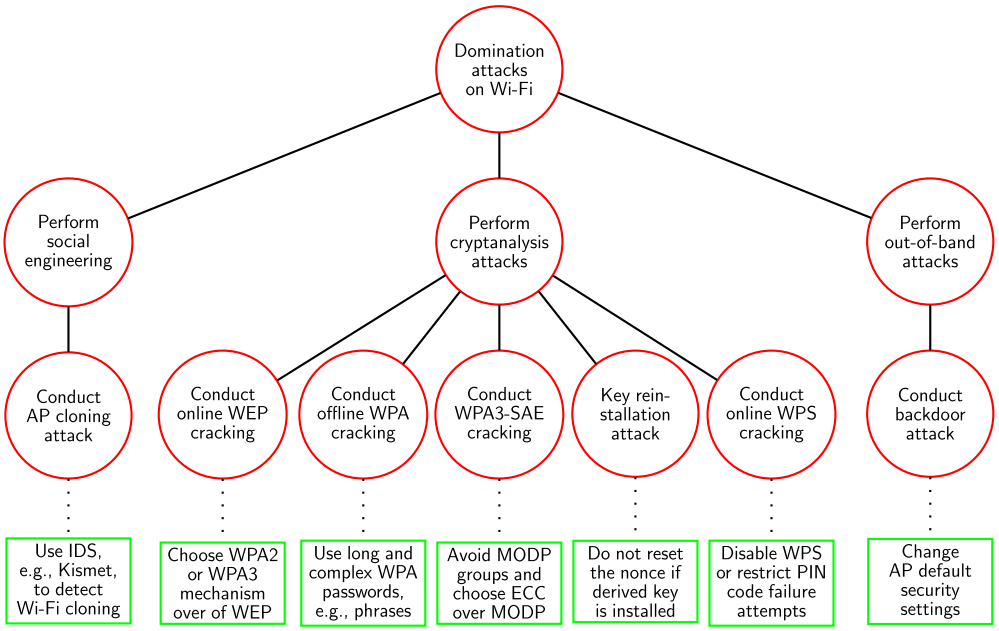

FIGURE 8. An attack-defense tree based on domination attacks on a Wi-Fi IoT infrastructure (○: attacks, □: defenses, ○–○: attack refinements, and ○. . .□: attack mitigations).

number of sensitive information related to the subscriber's itself or the Wi-Fi network, such as WEP/WPA key, SSID, connected clients, phone number, email address, subscriber's address, and subscriber's name.

Countermeasure. The network administrator has to change the default network and security configurations, such as the network SSID (changed to a discrete name), the IP address range, user names and passwords. It should also disable non secure mechanisms, such as WEP and WPS, on both frequency bands, i.e., 2.4GH and 5Ghz.

3) CRYPTANALYSIS

a: ONLINE WEP KEY CRACKING

In 2001, the key scheduling algorithm of RC4 used in the WEP mechanism was shown to contain severe design flaws [135]–[139], [188], [198]–[200]. These flaws can be exploited by attackers to recover the WEP key and decrypt all network communications. Few years later, in 2004, researchers [199] demonstrated that an attacker equipped with an ordinary computer can gradually reconstruct the WEP key in less than 2 hours. If an attacker passively eavesdrops a large number of WEP-encrypted packets (around 4,000,000 to 6,000,000 packets), it will be able to perform a byte by byte keystream recovery till recovering the whole WEP key. Interestingly, in the same year, a person under the pseudonym KoreK [191] posted on the NetStumbler forum an improved version of the technique [199]. Its technique reduces the required number of packets for cracking the WEP key to 700,000 packets [136] (500,000 packets [200]). Three years later (2007), researchers [200] demonstrated that a 104-bit WEP key can be cracked in 60 seconds using 35,000 to 40,000 packets (with 0.5 probability of success) and using 85,000 packets (with 0.95 probability of success). Once the key is disclosed, the attacker can fabricate, decrypt, and/or modify the content of Wi-Fi packets.

Countermeasure. The user should not use WEP security mechanism as well as the devices that only support WEP. The WEP key can be cracked easily using modern computers. As Wi-Fi Alliance recommends, we also suggest the use of WPA2-PSK or WPA3-SAE instead of WEP.

b: OFFLINE WPA KEY CRACKING

This attack aims to find the WPA password of a given Wi-Fi network. An attacker starts by eavesdropping a communication between a Wi-Fi station and an access point and tries to capture the four-way-handshake messages (by forcing a re-authentication). This handshake consists of four EAPoL[18] messages containing values generated by both parties to prove to each other the knowledge of the correct password. Upon capturing the four EAPoL messages, the attacker operates a brute force procedure or uses a dictionary of words to find out the right password that was used during the four-way-handshake. This attack may take decades to succeed on ordinary computers if the password is strong enough. However, it may also take less than a second if the password is in the attacker's dictionary. There are some cheap online cloud services, such as WPACracker.com [201], that can be used to crack a WPA key in a shorter time. The attacker just has to capture the handshake and upload it to the cloud service.

Countermeasure. The network administrator has to make sure that the used WPA passwords in its Wi-Fi network fulfill

[18]EAPoL: Extensible Authentication Protocol over LAN.

certain password security patterns. These patterns include the length of the password (e.g., must be at least 6 characters) and the used letters (e.g., mixture of uppercase, lowercase, special characters, and numbers). The password should also be updated regularly and kept secret.

c: WPA3 KEY CRACKING

In April 2019, researchers [202] discovered a set of vulnerabilities named Dragonblood. These vulnerabilities were discovered in the SAE (Simultaneous Authentication of Equals) handshake (a.k.a., dragonfly) used in WPA3-SAE. They demonstrated that by abusing timing or cache-based side-channel leaks (from the password encoding method[19]), it is possible to recover the WPA3 password using password partitioning attacks. The same work showed that it is possible to trick a Wi-Fi client into downgrading from WPA3-SAE to WPA2-PSK. This would allow an attacker perform offline WPA2 key cracking attack.

Countermeasure. It is recommended [202] not to use a set of multiplicative groups such as group 22, 23, and 24. Also, it is recommended to use ECC DH-groups over MODP and exclude MAC addresses during password encoding. This would decrease side-channel leaks. Furthermore, to mitigate the downgrading attack, Wi-Fi clients should remember if a network supports WPA3-SAE. Wi-Fi clients should not connect to a Wi-Fi access point that indicates the support of only WPA2-PSK if the same access point has been previously saved as a WPA3-capable access point.

d: KEY RE-INSTALLATION

This set of attacks were introduced in 2017 under the name of KRACKs (Key Reinstallation Attacks) [203]. It exploits the fact that some WPA implementations allow the retransmission of the third EAPoL message of the WPA four-way-handshake if an acknowledgment is not received. By doing so, the receiver reinstalls a previously installed keychain each time it receives this third EAPoL message. In addition to that, it resets the transmit packet counter as well as the receive replay counter. This forces the receiver (usually the supplicant) to reuse the same key twice (i.e., data is encrypted using the same key twice). The attacker exploits this to generate multiple attacks. To that end, the attacker first sets up a man-in-the middle scenario between the supplicant and the access point during a four-way-handshake and prevents the supplicant acknowledgment message (i.e., the fourth EAPoL message) from reaching the access point. This consequently induces the access point to resend the third EAPoL message again to the supplicant. The latter reinstalls the derived PTK keychain and resets the nonces used by the encryption mechanism. This allows the attacker to replay and decrypt certain messages (in case of TKIP, CCMP, and GCMP) and/or forge packets (in case of TKIP and GCMP). Furthermore, if the packets can be decrypted, the attacker can perform higher level attacks.

Countermeasure. The network administrator must ensure that the WPA implementation used in its network meets the following criteria: (1) Does not allow the retransmission of the third EAPoL message during the four-way-handshake. (2) Does not reset the nonce if the key is reinstalled [203].

[19]WPA3 applies two password encoding methods: (1) hash-to-curve is used when ECC (Elliptic Curve Cryptography) is adopted to encode the password into an elliptic curve point. (2) hash-to-element is used when MODP (Multiplicative groups modulo a prime) is adopted to encode the password into a group element.

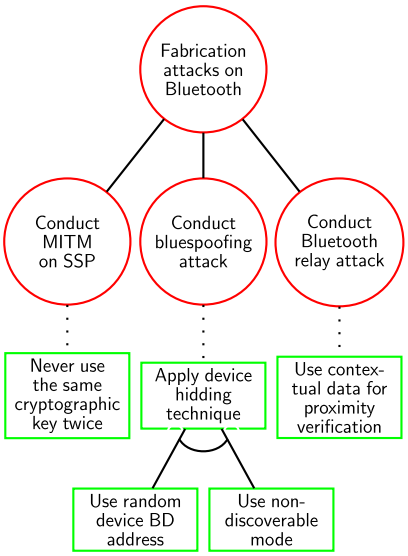

FIGURE 9. An attack-defense tree based on fabrication attacks on a Bluetooth IoT infrastructure (○: attacks, □: defenses, ○–○: attack refinements, and ○. . .□: attack mitigations).

e: WPS ONLINE CRACKING.

In 2011, WPS (Wi-Fi Protected Setup) was discovered to have a serious design flaw which can easily be exploited to brute force the PIN code and retrieve the WPA passphrase. Tools, such as pixiewps [204] and Reaver [205], can be used for this purpose.

Countermeasure. To mitigate this attack, the administrator can perform one of the following: (1) Disable the WPS mechanism on both radio bands, the 2.4 GHz and the 5GHz. (2) Restrict the number of WPS PIN code failure attempts to 3 and delay the next attempt by 30 minutes.

VI. ATTACKS ON BLUETOOTH TECHNOLOGY

A. FABRICATION ATTACKS ON BLUETOOTH

In the following paragraphs, we review the existing attacks on Bluetooth authentication. These attacks allow an attacker to impersonate a legitimate Bluetooth user to benefit from certain privileges and cause harm to the network. Figure 9 illustrates an attack-defense tree[13] based on fabrication attacks on a Bluetooth IoT infrastructure.

1) ENTITY SPOOFING

a: BLUESPOOFING

Each Bluetooth device is identified by its 48-bit long unique BD_ADDR (Bluetooth Device Address) and a

UTF-8 encoded user-friendly name of 248-byte maximum length. In Bluespoofing, an attacker spoofs the identity of a target Bluetooth device and impersonates it to gain unauthorized access to certain services.

Countermeasure. Spoofing a Bluetooth device is relatively easy. However, the attacker first needs to know the device address and user-friendly name of its target device. If the target device uses the non-discoverable mode along with the anonymity mode [95], the attacker will have difficulty to learn the information needed for spoofing. Also, disabling Bluetooth when not needed is a good security practice.

b: MAN IN THE MIDDLE ATTACK

In [206]–[211], the authors demonstrated how a MITM attack is possible when using SSP (Secure Simple Pairing) with passkey-entry association mode. If an attacker guesses the passkey that was previously used and knows that it will be reused in a future pairing, it will be able to impersonate both trusted parties during the future pairing. This is possible because there is no way to authenticate and check whether the exchanged public keys during the second phase of the SSP belong to the right entities or not since there is no certification authority.

Countermeasure. A lightweight certification authority must be implemented in such a way so that it certifies the ownership of a public key by a given Bluetooth device. Furthermore, it is highly recommended not to use the same SSP-passkey twice. A better alternative consists of upgrading to Secure Connections pairing, where MITM is hard to perform.

c: RELAY ATTACK

In this attack, the attacker stands in the middle of two legitimate Bluetooth devices and tricks them to get them connected and believe that they are in close proximity. The attacker manages to set up this scenario by just relaying messages throughout a built tunnel. The attacker does not modify the content of the messages, but just relays them. The purpose is to establish a connection from a further distance in the same way as it occurs when both devices are close to each other. The tunnel can be implemented in different ways. Researchers [212] have demonstrated relay attacks on Bluetooth legacy (i.e., Bluetooth versions before v2.1+EDR) by implementing the tunnel with a Bluetooth device that can impersonate one or both legitimate devices.

Countermeasure. One of the techniques that can be applied against relay attacks is to impose the devices to use contextual information extracted from their immediate environment [213]. Their close proximity may be verified based on the similarity between the contextual information. Such information can include temperature, humidity, radio signals, distance between the two devices, and geographic location.

B. INTERCEPTION ATTACKS ON BLUETOOTH

In this section, we present attacks that aim to affect data confidentiality in Bluetooth communications. Figure 10 illustrates an attack-defense tree[13] based on interception attacks on a Bluetooth IoT infrastructure.

1) EXPLOIT PROTOCOL VULNERABILITIES

a: BLUESNARFING

This attack is also known as Bluestumbling [214]. It consists of exploiting a security vulnerability in the OBEX (OBject EXchange) protocol to gain unauthorized access to a Bluetooth device and copy sensitive information from the device. Such information include people addresses, calendar, contact list, call/message history, files, and other device specific information.

Countermeasure. This attack is due to a vulnerability in the OBEX protocol. This vulnerability is fixed and the network administrator must ensure that all Bluetooth devices have the updated version of the OBEX protocol.

b: CAR WHISPERING

It was discovered by Trifinite Group in 2005 [215]. Car whispering is a technique used by attackers to hack a car's hands-free Bluetooth system. It exploits the fact that a car hands-free system uses a 4-digit PIN code which is in most cases set by the manufacturer to ''0000'' or ''1234''. Once connected, the attacker can insert or record audio and interact with other drivers on the move.

Countermeasure. Bluetooth devices that use PIN code-based authentication (i.e., legacy pairing) must use complex passphrases instead of PIN codes. If the use of PIN code cannot be avoided, the Bluetooth user must ensure that its devices are not using default values such as ''0000'' or ''1234''.

2) RECONNAISSANCE

a: BLUESNIPING

As Bluetooth was designed to be used for short-range, attackers are constrained to be located within the radio range of their targets. In 2004, a group of hackers conceived a hardware device called Bluesniper. This device, made essentially of a Yagi-antenna, allows an attacker to send and receive Bluetooth signals one mile away from its target [216]. This allows the attacker to be more discrete. In [217], it was shown, using a 2.4 GHz Yagi-antenna, that it is possible to intercept BLE traffic from 425 meters away.

Countermeasure. Bluetooth users should reduce the transmission power to only cover the needed range. However, as in most cases, reducing the transmission power of a Bluetooth device is not possible (e.g., on a smartphone), it is highly recommended to use the Bluetooth non-discoverable mode and switch Bluetooth off when not needed. Anonymity is also an effective measure against traceability.

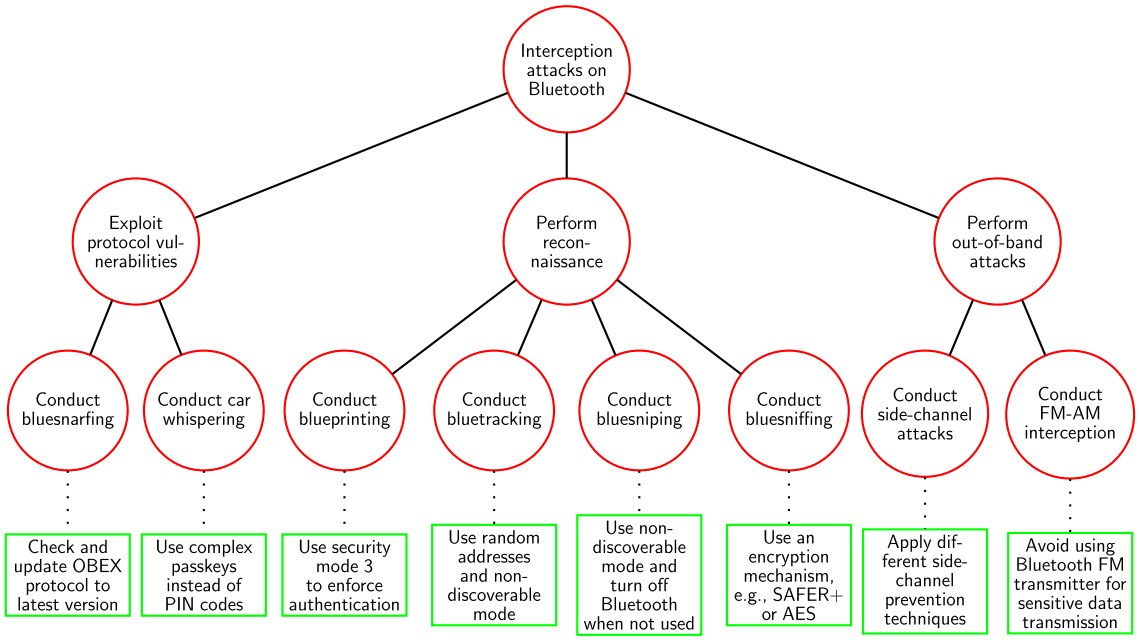

FIGURE 10. An attack-defense tree based on interception attacks on a Bluetooth IoT infrastructure (○: attacks, □: defenses, ○–○: attack refinements, and ○. . .□: attack mitigations).

b: BLUEPRINTING

In this scenario, attackers try to find out the details about nearby Bluetooth devices. Information such as Bluetooth device address, make, model, firmware version, provided services, and channels, are collected for a malicious future use. For example, an attacker can use SDP (Service Discovery Protocol) to collect necessary information and exploit them to generate other attacks [218].

Countermeasure. In Bluetooth security mode 1, 2, and 4, any Bluetooth user can perform service discovery as well as other operations (e.g., echo-request/echo-reply using L2PAC protocol) on remote Bluetooth devices using SDP and learn useful information. This operation does not need any credentials (paring-free connection [219]). However, in Bluetooth security mode 3, SDP can only be used if the user knows the credentials (pairing-based connection [219]). In this case, the Blueprinting will be made harder.

c: BLUETRACKING

In this scenario, the attacker tracks a Bluetooth device address along with its user-friendly name and follows its movements to learn sensitive and private information, such as the house address, the workplace address, the frequently visited places, and the current location [220].

Countermeasure. A Bluetooth user can adopt the non-discoverable mode along with the anonymity mode [95] to hide its presence and become untraceable. It is also recommended to switch off Bluetooth when not needed.

d: BLUESNIFFING

This attack consists of eavesdropping ongoing Bluetooth communications to capture Bluetooth packets and extract sensitive information, such as voice (e.g., during a conversation), files, or even passwords (e.g., when using a non secure Bluetooth keyboard). For transmitting data, Bluetooth technology uses FHSS (Frequency-Hopping Spread Spectrum) to minimize interferences. It also uses packet whitening (or scrambling) for improving error resilience and security. Therefore, for the attacker to be able to capture and correctly interpret all packets being exchanged, it should know the frequency hopping sequence used between two devices as well as how to unwhiten the packets. In [221], it was demonstrated that the frequency hopping sequence can be determined using both the address and the clock of the master device. They also managed to unwhite packets using the lower six bits of the clock. Nowadays, it is easy to sniff an existing Bluetooth communication using dedicated hardware tools such as Ubertooth One and Sniffle [217].

Countermeasure. Bluetooth users must use the available encryption mechanisms in Bluetooth. If the devices run Bluetooth v4.1+LE or earlier versions, then SAFER+ must be used. If the devices run Bluetooth v4.2+LE or newer, the AES encryption mechanism is employed.

3) OUT OF BAND ATTACKS

a: SIDE CHANNEL ATTACKS

As the latest versions of Bluetooth use AES encryption mechanism, it is possible to conduct different techniques of side-channel attacks. Techniques, such as DPA (Differential Power Analysis) and SPA (Simple Power Analysis) [222], can be applied to infer information that can be used to disclose the secret keys [223].

Countermeasure. To mitigate side-channel attacks, researchers have proposed multiple approaches that could

be used as a protection against these attacks. These approaches include, but not limited to, masking [224], cross-copying [225], conditional assignment [226], bucketing [227], and predictive timing mitigation [228].

b: OUT-OF-BAND INTERCEPTION

Some cars do not have any Bluetooth interface. However, this does not prevent certain drivers from using a Bluetooth radio transceiver (e.g., T10 FM transmitter) that is plugged into the vehicle power outlet socket. The Bluetooth transceiver receives audio data from a paired Bluetooth device (e.g., driver's phone) and transmits the audio data as AM (Amplitude Modulation) or FM (Frequency Modulation) radio signals on an arbitrary frequency (i.e., chosen by the driver). The driver sets its car radio filter (or car stereo) on the same frequency to receive those signals and play them on the car speakers. This creates a vulnerability since the driver is downgrading from an encrypted communication (i.e., Bluetooth) to an unencrypted communication (Radio AM or FM). Attackers just have to eavesdrop on AM and FM channels and listen to what the driver is listening.

Countermeasure. This attack concerns cars that do not have a Bluetooth interface. Thus, to mitigate this attack, drivers should not use a Bluetooth transceiver if the data to be broadcasted on the car speakers (e.g., phone call) contain any confidential, private, or sensitive information.

C. MODIFICATION ATTACKS ON BLUETOOTH

We could not identify a particular scenario that contributes to only modification in Bluetooth communications. However, there exist attacks on Bluetooth that aim to modify the configuration (integrity) of BLE smart devices, such as heart rate monitors, smart lock, lightbulb, smart padlocks, blood GMS (Glucose Monitoring System), wristbands [217], [229]–[236], and the configuration of Bluetooth networks.

1) DEVICE INTEGRITY MODIFICATION

This attack consists of sending forged write-commands to a Bluetooth smart device in such a way so that the execution of the commands on the smart device modifies its security configurations (e.g., password) and parameters settings (e.g., lightbulb brightness). For example, We demonstrated how we can remotely (from 100 meters away) modify the authentication password stored on a bicycle lock and use the new password to unlock the bicycle. We also showed how the brightness of a smart lightbulb can be boosted to 255% and turned off [217].

Countermeasure. The attack on BLE device integrity is due to the "Just Works" association mode. In this association mode, any Bluetooth device can connect to a Bluetooth smart device (e.g., a home smartlock) without authentication and send unauthenticated write-commands to the smart device. The latter blindly executes the commands that allow the attacker to modify the integrity of the device (e.g., unlock the smartlock). To thwart this attack, it is highly recommended to use the other association modes. A combination of the "Just Works" and the Out-of-Band association mode can be adopted to implement a stronger authentication.

2) NETWORK INTEGRITY MODIFICATION

It is possible to switch the roles of two Bluetooth devices from "master" to "slave" or vice-versa [219]. In fact, certain devices are vulnerable to bluecutting attack (discussed in Subsection VI.D.3) where an attacker can disconnect a device from another by forcing the establishment of a new connection. The attacker d_0 spoofs a slave device, say d_1, that is connected to a master device, say d_2, and initiates a new connection with device d_2. This would disconnect d_1 from d_2 and establish a new connection with d_2. However, this time the attacker d_0 holds the role of the master and device d_2 holds the role of the slave. This would consequently modify the configuration and behavior of a Bluetooth piconet or scatternet.

Countermeasure. As discussed in [219], Bluetooth needs to adopt a security mode that is similar to security mode 3 used in earlier Bluetooth versions. This attack is possible due to the possibility of creating a pairing-free connection when the Bluetooth security mode 4 is used. Security mode 4 allows attackers to initiate a connection with any device without any authentication (i.e., pairing-free connection).

D. INTERRUPTION ATTACKS ON BLUETOOTH

In this section, we present Bluetooth attacks that affect the availability of Bluetooth networks. By generating such attacks, an attacker can cause an IoT Bluetooth network to go down and make all provided Bluetooth services unavailable. Figure 11 illustrates an attack-defense tree[13] based on interruption attacks on a Bluetooth IoT infrastructure.

1) SOCIAL ENGINEERING

a: BLUEJACKING

This attack consists of sending anonymous and unsolicited messages, e.g., business cards, with an offensive content using OBEX (OBject EXchange) protocol. The attacker creates a new contact on its device and assigns the offensive content as a name to that contact and then sends that contact card to the target. When the target receives the business card, it displays the message "Would you like to add *offensive content* to your address book?". We consider this attack as a denial of service as it interrupts a user from doing something useful and forces the user to do something with a view to wasting his or her time, money, and energy.

b: BLUETOOTHING

This attack appeared in 2004 as a hoax for arranging dates. It consists of sending messages containing the word "toothing?" to nearby Bluetooth devices asking them for a date. Such messages may be considered as a harassment for some people. We consider this as a denial of service as the victim is forced to stop doing something useful.

Countermeasure. The administrator has to make sure that all of its devices are running the updated version of the OBEX protocol. This would mitigate the previous two attacks.

2) FLOODING

a: BLUESPAMMING

In this scenario, the attacker exploits a vulnerability in the OBEX protocol to spam a target Bluetooth device with a large amount of crafted files [237], [238].

Countermeasure. Similar to the two previous attacks, Bluetooth devices should be running the latest version of the OBEX protocol, where the exploited vulnerability is patched.

b: BATTERY EXHAUSTION

Bluetooth networks are resource-constrained networks where devices run dedicated algorithms to moderately use and conserve their limited batteries while performing their tasks. In such conditions, an attacker exploits this energy-related weakness to exhaust batteries of those devices. The attacker repeatedly sends unsolicited and encrypted messages to target devices. These devices run cryptographic algorithms and consume a huge amount of energy before ignoring and dropping those nonsense messages.

Countermeasure. Bluetooth protocol must be implemented in such a way so that any device can differentiate an authentic message from a crafted one before any expensive operation. Bluetooth devices can then drop and ignore unsolicited or replayed messages without consuming much energy and without performing expensive cryptographic operations.

3) DEVICE DISRUPTION

a: BLUECHOPPING

The purpose of this attack is to disrupt an established Bluetooth piconet. The attacker spoofs the identity of a connected Bluetooth device in a piconet and tries to establish a connection with a master device that manages another piconet. In this case, the network will consider the spoofed device to be linked to both piconets. Consequently, it will disturb the network configuration.

Countermeasure. Spoofing must be made hard to perform by using the non-discoverable mode along with the anonymity mode [95]. The administrator can also setup an IDS (Intrusion Detection System) [239]–[241] to detect the presence of duplicated Bluetooth devices on a piconet or scatternet.

b: BLUESMACKING

The L2CAP (Logical Link Control and Adaption Protocol) protocol allows devices to send Echo-request and receive Echo-reply messages from remote devices to check the connectivity (round-trip time). In this scenario, an attacker sends large-size Echo-requests to its target. Upon the reception of the requests, certain device's Bluetooth stack crashes driving the system into a livelock. Recently, a set of twelve vulnerabilities, called SweynTooth (ICS-ALERT-20-063-01), were discovered on many BLE devices. These vulnerabilities allow attackers to remotely crash a BLE device by sending non-standardized packets.

Countermeasure. The network administrator has to make sure that all of its Bluetooth devices implement a Bluetooth stack that rejects packets that have a non-standardized size and format (e.g., contain empty fields) or the stack should be able to handle large size packets.

c: BLUEDUMPING

This attack occurs when two Bluetooth devices have already paired in the past and generated the shared link key for possible future communications [242]. The attacker spoofs one of the two devices and requests the other one to re-perform the pairing from the beginning by claiming the loss of the link key. The other device accepts the request and discards the stored link key. The attacker then aborts the connection. In this way, the spoofed device cannot automatically connect to the other device since the latter no longer has the link key. The pairing should be re-performed.

Countermeasure. This attack remains possible as long as the Bluetooth protocol allows a Bluetooth device to forget its link key. A network administrator can set its devices for non-discoverable mode to make the spoofing harder for the attacker as the latter primarily needs its target Bluetooth device's address and its Bluetooth user-friendly name.

d: BLUECUTTING

This attack is also known as connection dumping [219]. The Bluetooth implementation on certain devices allows the establishment of more than one connection at the same time with the same remote Bluetooth device. However, the termination of one of these multiple connections terminates all the other remaining connections. An attacker spoofs a legitimate Bluetooth device to establish a pairing-free connection (i.e., a connection that does not need any authentication). The attacker uses SDP (Service Discovery Protocol) for instance, with another Bluetooth device which is currently connected to the spoofed device and causes the termination of its pairing-free connection. This results in the disconnection of the legitimate Bluetooth device.

e: BLUEDEPRIVING

Some modern Bluetooth devices, including BLE smart devices, do not allow the establishment of multiple connections at the same time with the same remote Bluetooth device. An attacker exploits this fact to spoof a legitimate device to establish a pairing-free connection with this type of devices before the spoofed device makes the connection. Once the connection is established, the spoofed ''legitimate'' Bluetooth device cannot connect to those devices since the attacker has already occupied the connection. This results in a connection deprivation [217], [219].

Countermeasure. Bluetooth should implement a new security mode that operates in the same way as security mode 3 (in v2.0+EDR and earlier). In security mode 3, any device

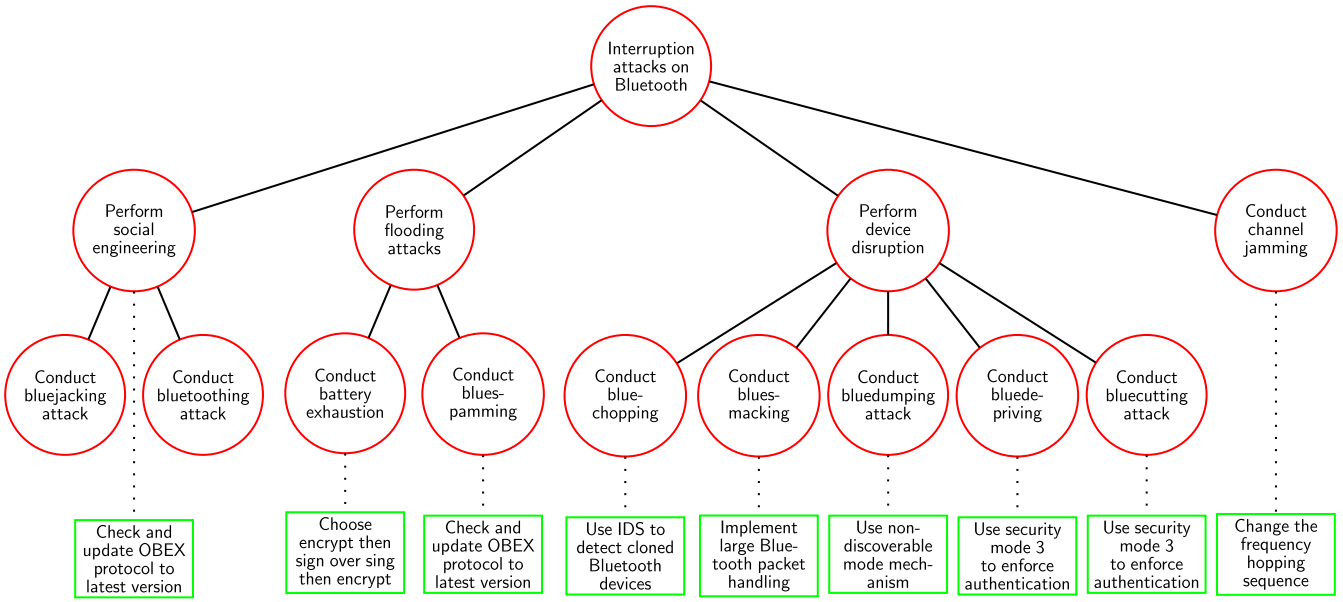

FIGURE 11. An attack-defense tree based on interruption attacks on a Bluetooth IoT infrastructure (○: attacks, □: defenses, ○–○: attack refinements, and ○. . .□: attack mitigations).

has to authenticate for every Bluetooth service it requests. In fact, the notion of pairing-free connection does not exist in this mode since all connections must be authenticated. Thus, an attacker cannot establish a spoofed connection.

f: BLUETOOTH SPEAKER HIJACKING

In this attack, the attacker uses a Bluetooth device, e.g., smartphone, and tries to establish a connection with a Bluetooth speaker that is already connected to another Bluetooth device using the "Just Works" association mode. Unfortunately, certain Bluetooth speakers, such as the Amazon Alexa (running Bluetooth 5), drop the existing connection and accept the new one. In this way, the attacker can take over the speaker and play its own audio, which may contain offensive and threatening content.

Countermeasure. To thward this attack, Bluetooth device vendors have to control how their Bluetooth devices handle connection requests when using the "Just Works" association mode before releasing those devices into the markets. Certain Bluetooth devices, such as BLE smart devices, mitigate this attack by restricting the device to only one connection and denying any second connection request. However, the latter "security measure" constitutes a vulnerability that can be exploited to generate the Bluedepriving attack.

4) CHANNEL JAMMING

Bluejamming. In this attack, the attacker continually sends random radio signals all over the used communication channels. This denies legitimate Bluetooth devices from accessing the radio channel and sending their data. An alternative approach consists of setting up a device (known as Bug) that has the same identity as a legitimate one which is currently connected to a piconet. When a device communicates with the legitimate device, the legitimate device and the "Bug" device will simultaneously respond and jam each other.

Countermeasure. From a practical point of view, jamming Bluetooth communications implies generating noise signals all over 79 channels in the Bluetooth band (40 channels in BLE), which is not practical for attackers. Such jamming can be detected by analyzing the Bluetooth band and localizing from where specific disruptive signals are generated. Moreover, the network administrator can configure its Bluetooth devices to frequently change the hopping sequence.

E. DOMINATION ATTACKS ON BLUETOOTH

In this section, we present Bluetooth attacks in which an attacker can perform a number of security breaches that affect multiple security services in an IoT Bluetooth infrastructure. Figure 12 illustrates an attack-defense tree[13] based on domination attacks on a Bluetooth IoT infrastructure.

1) CRYPTANALYSIS

a: OFFLINE/ONLINE PIN CRACKING

The offline PIN cracking attack is also known as PIN crunching. The attacker eavesdrops a pairing between two devices then uses the captured packets to brute force the PIN code that was used during that pairing. The time it takes to crack such a PIN code depends on its length. It has been demonstrated that a 4-digit PIN code can be cracked in less than 0.06 sec on an old Pentium IV 3GHz HT computer [242]. Once the PIN code is cracked, all secret keys can be generated. The attacker can intercept, decrypt, fabricate, and modify packets, and may cause interruption as well. In online

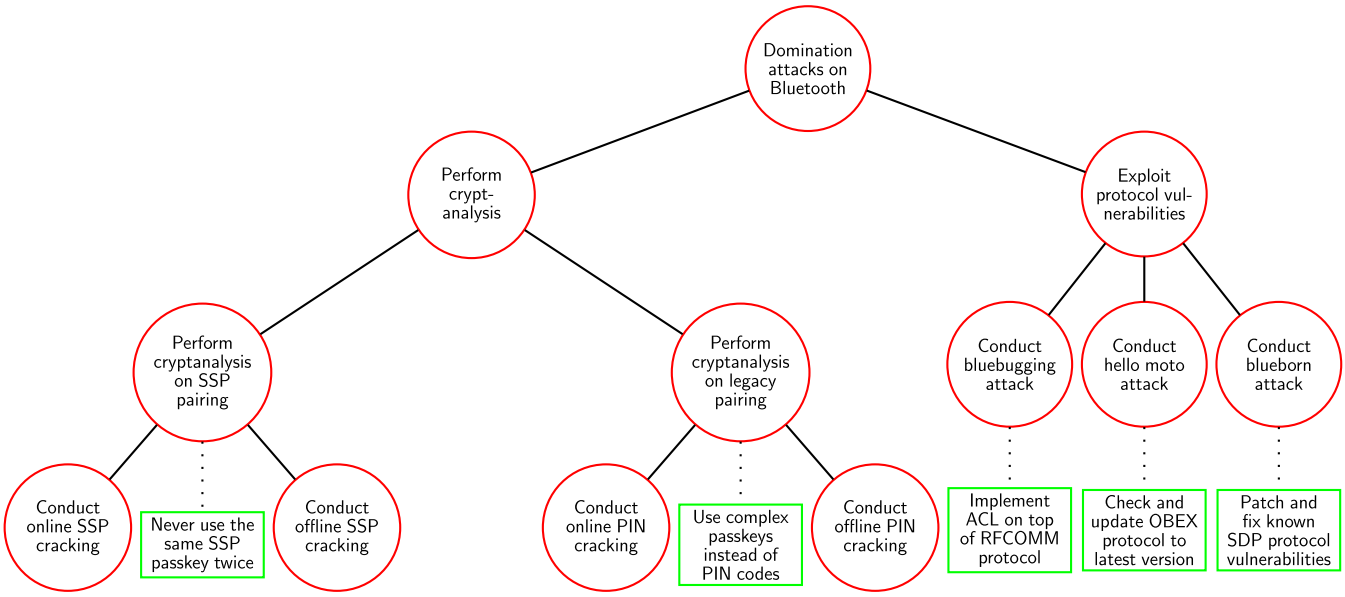

FIGURE 12. An attack-defense tree based on domination attacks on a Bluetooth IoT infrastructure (○: attacks, □: defenses, ○–○: attack refinements, and ○. . .□: attack mitigations).

PIN cracking, the attacker tries to connect with the target device by guessing different PIN code values. The attacker changes its BD_ADDR address every time a PIN guess fails. The attacker bypasses the ever-increasing delay between retries. This attack works well if a fixed or short PIN code is used.

Countermeasure. The network administrator has to make sure that its network uses simple secure pairing and not legacy pairing. If the legacy pairing cannot be avoided, then the administrator has to ensure that its devices use complex passphrases instead of simple and short PIN codes.

b: OFFLINE/ONLINE SSP PASSKEY CRACKING

The offline SSP passkey cracking attack concerns the secure simple pairing when used with passkey-entry association mode. The attacker first captures all messages exchanged during an SSP pairing (in passkey-entry mode). Then it runs around 20 tests before figuring out the passkey that was used. Interestingly, if the same passkey is used in the later sessions, the attacker can decrypt and read the messages. Also, if the key is unchanged, the attacker can fabricate packets and cause interruption as well. In the online SSP passkey cracking attack, the attacker establishes a man-in-the-middle scenario by spoofing both communicating devices and performs a bit-by-bit test to determine the passkey. During a secure simple pairing with passkey-entry mode, in authentication stage 1, both parties authenticate and prove to each other the possession of a 20-bit passkey (e.g., $p_k{=}b_0, \ldots b_{19}$). Each device, in each i^{th} round proves to the other device that it possesses the right bit b_i. If a party gets the wrong bit, the pairing is aborted by the other party. The attacker exploits this abortion mechanism to brute force the passkey as follows: In a given round i, the attacker assumes that $b_i = 0$. Then, if the round ends successfully, the attacker concludes that $b_i = 0$. Otherwise $b_i = 1$. If b_i is wrong, the other party aborts the communication. The attacker repeats the pairing using the previously learned b_i bits until it figures out the whole passkey [211].

Countermeasure. The user has to make sure that if SSP pairing is used along with the passkey-entry mode, the passkey has to be changed for every session and should not be used twice. Also, the use of non-discoverable mode reduces the possibility of being spoofed during an online passkey cracking attack. Turning Bluetooth off when not needed is also a good security practice.

2) EXPLOIT VULNERABILITIES IN PROTOCOLS

a: BLUEBUGGING

In this attack [243], the attacker uses the RFCOMM protocol over the serial port channel to establish a connection with the target device without any pairing. Once connected, the attacker runs on the target device a set of commands (e.g., AT commands) to perform the following: sending messages (SMS or MMS), making phone calls, reading contacts, and changing the phone configurations.

Countermeasure. The network administrator has to make sure that its Bluetooth devices do not accept any serial connection through RFCOMM without asking for an authorization. Devices that do not have any high-level authorization mechanism must not be used.

b: HELOMOTO ATTACK

This attack exploits incorrect processing of "trusted device" handling on certain Motorola devices [244]. The attacker initiates a connection using the OBEX push profile and pretends sending a vCard. The sending process is interrupted by the

attacker whose profile is stored on the trusted device list of the target device. By taking advantage of this entry on that list, the attacker connects to the headset profile of the target device without any authentication and uses AT commands to control it.

Countermeasure. The network administrator has to make sure that all its Bluetooth devices, in particular, Motorola devices, are running the updated version of OBEX. This will prevent attackers from generating the previous attack.

c: BLUEBORNING

It was discovered by Armis Labs [245] in 2017. This attack exploits a set of vulnerabilities in the implementation of Bluetooth stack in various operating systems such as Android, Linux, iOS and Windows. When these vulnerabilities are exploited, an attacker can remotely hijack a Bluetooth device and cause serious breaches. The attack does not require the target device to be paired to the attacker's device. Also, it does not need the target device to download any crafted files or click on any phishing URL.

Countermeasure. Several patches have been developed to address the Blueborne vulnerabilities. Microsoft, Google, and Apple have already fixed the flaws on their devices.

VII. ATTACKS ON ZigBee TECHNOLOGY

A. FABRICATION ATTACK ON ZigBee

In the following paragraphs, we enumerate different attacks that aim to bypass the authentication mechanisms used in ZigBee. Figure 13 illustrates an attack-defense tree[13] based on fabrication attacks on a ZigBee IoT infrastructure.

1) MESSAGE SPOOFING

a: ROGUE ACKNOWLEDGMENT

The IEEE 802.15.4 specification does not provide any authentication, confidentiality, or data integrity protection for the acknowledgment frames. An attacker can spoof any ZigBee device and send acknowledgment frames to cause another ZigBee device believe that its frames have been correctly received by the destination. At the same time, the attacker ensures that it intercepts frames sent by a ZigBee device before sending any spoofed acknowledgment to the other party [105].

Countermeasure. The IEEE 802.15.4 specification must provide authentication for all management frames to prevent attackers from spoofing the network coordinator or ZigBee devices and forging spoofed packets. ZigBee Alliance may implement the MFP (Management Frame Protection) specified in the IEEE 802.11w amendment to solve the problem.

b: PACKET INJECTION

This attack is also known as PIP (Packet In Packet) attack [246]. It allows an attacker to insert (hide) a malicious packet inside a normal packet payload that is permitted onto the network. By exploiting a bit error in the original packet (i.e., outer frame), the attacker can force its malicious packet (i.e., inner frame) to be interpreted instead of the original packet. This attack is generally used to bypass any firewall or intrusion detection system.

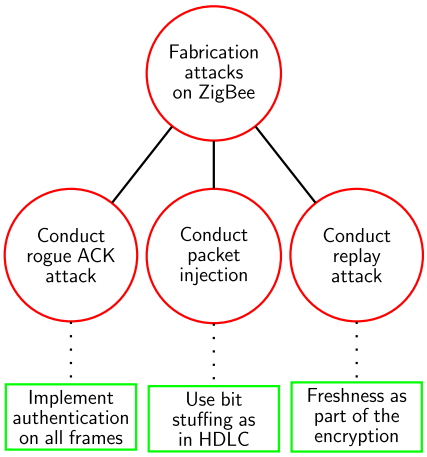

FIGURE 13. An attack-defense tree based on fabrication attacks on a ZigBee IoT infrastructure (○: attacks, □: defenses, ○–○: attack refinements, and ○. . .□: attack mitigations).

Countermeasure. A solution to this attack was discussed in [247]. It uses bit-stuffing which is an error detection mechanism used in HDLC[20] protocol to inhibit control information to appear in the payload of a frame.

c: REPLAY ATTACK

ZigBee uses a 32-bit frame counter to differentiate old frames from new ones. Old frames are discarded. In [248], the authors have demonstrated that a replay attack is still possible. Other researchers [249] have demonstrated the attack using a software tool called KillerBee [248].

Countermeasure. Some researchers [103] have suggested that ZigBee Alliance should integrate a timestamp within the encryption mechanism to mitigate replay attacks. Also, data freshness can be implemented at a higher level protocol, e.g., included in the message authentication protocol [250].

B. INTERCEPTION ATTACKS ON ZigBee

In the following paragraphs, we enumerate various passive and active attacks that could be launched by an attacker to intercept, extract, and reveal sensitive information about a ZigBee network. Figure 14 illustrates an attack-defense tree[13] based on interception attacks on a ZigBee IoT infrastructure.

1) OUT OF BAND ATTACKS

a: PHYSICAL ATTACK

In this scenario, an attacker physically gains access to a ZigBee device (e.g., by stealing it). The attacker uses a set of sophisticated hardware and software tools to extract sensitive information such as security keys, which are generally stored in an unencrypted format in a flash memory [251]–[253].

[20]HDLC (High-Level Data Link) is a link layer protocol developed by the ISO (International Organization for Standardization) in 1979.

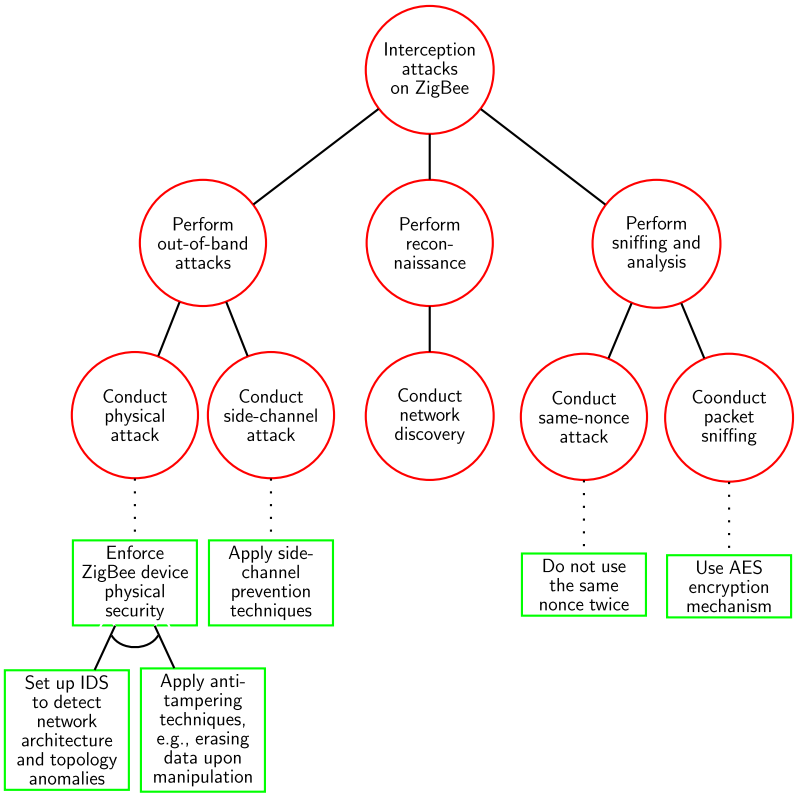

FIGURE 14. An attack-defense tree for attacking a ZigBee IoT infrastructure through interception (○: attacks, □: defenses, ○–○: attack refinements, ○ . . . □: attack mitigations, and : conjunction refinement).

This attack is made easy as almost all ZigBee devices are not built tamper-resistant.

Countermeasure. The administrator can set up an intrusion detection system which upon detecting an unplanned removal of a ZigBee device from the network, invalidates the keys and generates new ones. Another approach consists of augmenting the ZigBee devices with anti-tamper measures in such a way so that the devices start erasing their content upon detecting any physical tampering [254]. However, the ZigBee network administrator has to be aware that such countermeasure can be exploited by an attacker to cause a denial of service attack by intentionally provoking the ZigBee devices to erase their contents and become unavailable.

b: SIDE CHANNEL ATTACKS

ZigBee uses the AES symmetric encryption algorithm along with the CCM* mode to encrypt the data and to generate the message integrity code. Researchers [106], [107] have demonstrated that it is possible to retrieve the secret keys by conducting a side-channel analysis on a ZigBee device. This attack assumes that the attacker has full (physical) control over the device.

Countermeasure. There exist many techniques to prevent side-channel attacks. These techniques include masking [224], cross-copying [225], conditional assignment [226], bucketing [227], and predictive timing mitigation [228]. Implementing one of the previous techniques in ZigBee will certainly mitigate this type of attacks.

2) RECONNAISSANCE

a: NETWORK DISCOVERY

In this scenario, the attacker eavesdrops the radio channel and tries to discover available ZigBee networks. If a network is detected, the attacker can (using dedicated tools [248]) inject packets to make the network react and disclose information related to its configuration. For instance, as part of a network discovery process, ZigBee devices send a beacon request frame on each radio channel to discover ZigBee routers or ZigBee coordinators. If a ZigBee router or a ZigBee coordinator receives the request frame, it responds by disclosing its PAN ID (Personal Area Network IDentifier), source address, and other useful information [249]. The attacker just has to mimic a network discovery process.

Countermeasure. Thus far, there is no clear or simple countermeasure for this attack as beacon request frames are essential to ZigBee network discovery process as discussed in [103]. Implementing a comprehensive countermeasure for this attack requires changing the whole ZigBee protocol.

3) SNIFFING AND ANALYSIS

a: SAME-NONCE ATTACK

ZigBee uses the AES-CBC-MAC algorithm to provide encryption and data integrity. The algorithm uses a nonce

along with the encryption key to produce a unique output. Similar to the key-reuse attack in Wi-Fi, if the same nonce was used along with the same key to encrypt two successive messages, an attacker would be able to recover partial information about the plaintext [255]. A ZigBee device can be forced to reuse the same nonce with the same key by causing a power failure on the device.

Countermeasure. The network administrator has to make sure that the nonces are not used twice with the same key. One practical solution is to refresh the key after all possible values of a nonce have been used [255].

b: PACKET SNIFFING

In this scenario, an attacker uses a ZigBee network sniffer, such as KisBee [256] or KillerBee [248], to capture exchanged packets over the radio. The packets are analyzed later on using a protocol analyzer, such as Wireshark, to extract sensitive information. This is possible as in most cases, ZigBee networks (e.g., in a Wireless Sensor Network) do not apply encryption just to save some energy. Also, if the standard security level is used and the network key has not been pre-installed onto the ZigBee devices, then there is a chance to capture the network key. Indeed, the latter will be sent unencrypted by the network coordinator to every ZigBee device joining the ZigBee network [173].

Countermeasure. The network administrator has to make sure that its network uses AES to encrypt the data and the new cryptographic keys are distributed securely. No secret information should be sent unencrypted (assumption set by the ZigBee Alliance [157]). Also, it is a good initiative to preload ZigBee devices with cryptographic keys using an out-of-band channel to prevent their interception over the air.

C. MODIFICATION ATTACKS ON ZigBee

Thus far, no attack was reported on breaching data integrity when AES-CBC-MAC is used. The only situation where this may happen is when an attacker manages to learn the key that is used to compute the MIC (Message Integrity Code). Data integrity is not restricted to packets being sent over the radio. It also covers the protocols and programs running on a device. If an attacker gets physical access to a ZigBee device, it connects to the device and modifies internal data structures, such as static routing tables and application protocols. Then, the device behaves differently in the network. For instance, compromised devices can be used to generate Ad hoc architecture-related attacks that affect availability.

In the case of a sybil attack, the compromised node absorbs all packets by declaring itself as having multiple identities [257]. Through a wormhole attack, compromised nodes are used to establish a hidden tunnel to communicate and trick other nodes that are far from each other to make them believe that they are close to each other [249]. In the case of a black-hole attack, the compromised node drops all packets that it is supposed to forward [258]. In a selective forwarding attack, the compromised nodes select which packet to drop or forward. Finally, in a sinkhole attack, the compromised nodes are positioned next to the network coordinator (often called sink) and drop all packets coming for the sink [259].

D. INTERRUPTION ATTACKS ON ZigBee

In the following paragraphs, we enumerate different attacks that aim to make a ZigBee network partially or completely unavailable. Figure 15 illustrates an attack-defense tree[13] based on interruption attacks on a ZigBee IoT infrastructure.

1) PROTOCOL MISUSING

a: FRAME TRASHING

As a protection against replay attacks, ZigBee technology uses a 32-bit frame counter at the network layer to distinguish between fresh and old frames. This frame counter is not encrypted and it is reset to zero after updating the network key by the coordinator or the network administrator. If for any reason, the frame counter has been reset and the network key has not been updated, an attacker can inject a forged frame (by copying an old encrypted frame payload) and set the frame counter to its maximum (i.e., `0xFFFFFFFF`). This enforces all ZigBee devices to drop all future frames upon their reception [105].

Countermeasure. The ZigBee protocol must ensure that the network key is updated and not used twice after the expiring of the frame counter [105]. This prevents an attacker from using an old encrypted message and replaying it back with manipulated frame counter.

b: GREEDY BEHAVIOR ATTACK

Similar to IEEE 802.11, an attacker violates the CSMA/CA protocol and applies the binary back-off algorithm to his or her benefits. The attacker does not wait for a random timer and interframe intervals but monopolizes the access to the radio channel.

Countermeasure. The network administrator can set up a network watcher to monitor device behaviors and how the shared radio channel is used. If a device is suspected for behaving maliciously, the administrator can, for example, suspend that device for the time being. However, such an aggressive countermeasure can be exploited by an attacker to suspend legitimate ZigBee devices by spoofing the latter and conducting a greedy behavior attack.

2) POWER DRAINING

a: BATTERY EXHAUSTION

In most cases, ZigBee devices are powered by a 3A battery and equipped with a sensor unit (e.g., MicaZ of CrossBow). These devices sleep most of the time and wake up when an event occurs. An attacker can generate a large amount of bogus and encrypted traffic to be processed by the ZigBee devices and prevent them from going to sleep mode. This considerably drains their power [260].

Countermeasure. One approach to deal with this attack is to set up an intrusion detection system to detect abnormal behaviors. An alternative is to structure the encryption and

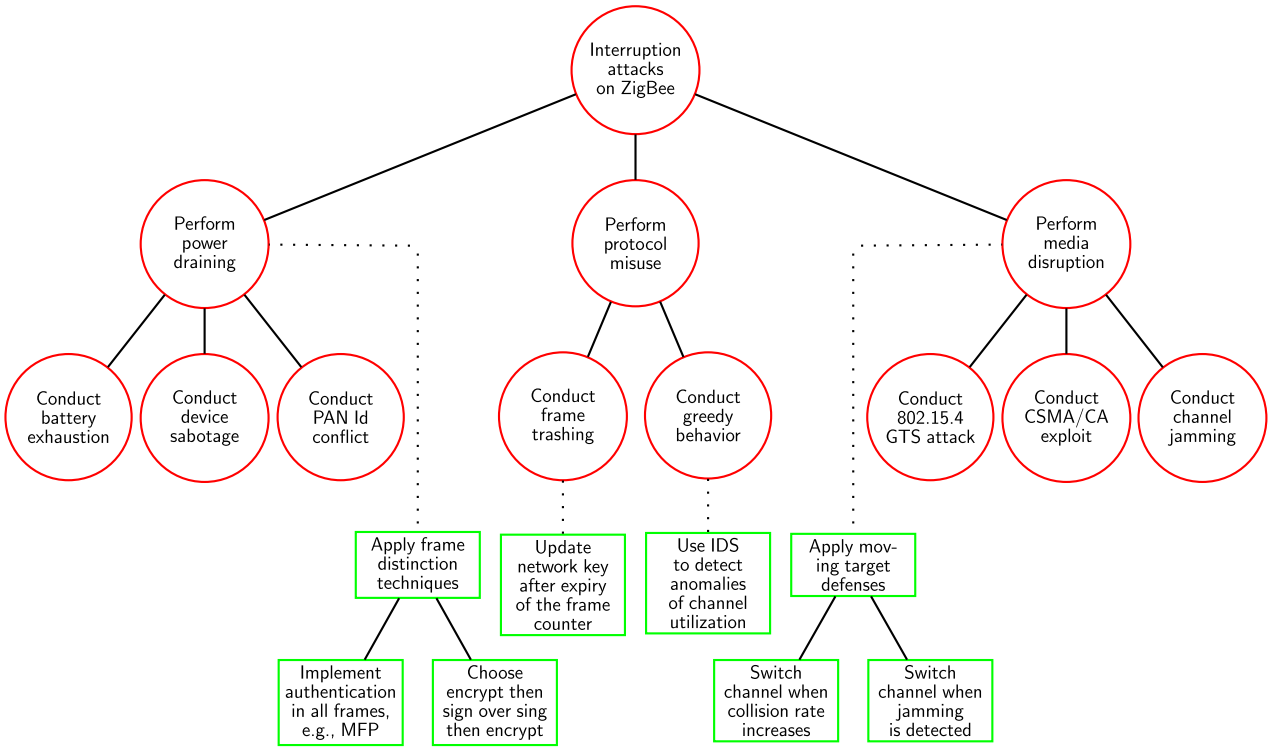

FIGURE 15. An attack-defense tree based on interruption attacks in a ZigBee IoT infrastructure (○: attacks, □: defenses, ○−○: attack refinements, and ○...□: attack mitigations).

authentication of packets in such a way so that devices can pre-distinguish legitimate traffic from illegitimate traffic before processing them to avoid unnecessary draining of power.

b: END-DEVICE SABOTAGE

Depending on the application, ZigBee end-devices can spend most of their time in a power-saving mode while sensing other measures such as temperature, humidity and pressure, using dedicated sensors. During this phase, the power consumption is very low. In the wake-up period of the duty-cycle, those devices consume a considerable amount of power by sending poll requests to retrieve their data (from the network coordinator). The data was sent to those devices while they were in power-saving mode. An attacker abuses this mechanism by spoofing the network coordinator and sending broadcast or multi-cast poll replies to all poll requests to keep the devices awake. This would considerably drain their battery [156].

Countermeasure. The ZigBee protocol must ensure that poll requests and responses are authenticated and refreshed [250] to prevent attackers from spoofing any ZigBee device and replaying the old messages. Also, applying a mechanism that is similar to MFP (Management Frame Protection), which is used in Wi-Fi, would be a perfect option.

c: PAN-ID CONFLICT ATTACK

Usually, a ZigBee network adopts the infrastructure mode. In this mode, a set of resource-constrained devices are associated with a network coordinator. Each device has its own unique PAN-ID (Personal Area Network IDentifier) and is aware of its coordinator's PAN-ID. The existence of more than one coordinator's PAN-ID in the same network causes a conflict which is automatically detected and reported to the coordinator. This initiates a conflict resolution procedure by generating and sharing a new PAN-ID. An attacker exploits this procedure to continually send fake conflict notification messages obliging the coordinator to initiate the resolution procedure. This will consequently drain the power source of the resource-constrained devices and delay their communications as well [261].

Countermeasure. The user must ensure that its network provides authentication for the conflict resolution messages so that attackers cannot spoof and replay those messages [250].

3) MEDIA DISRUPTION

a: GUARANTEED TIME SLOTS ATTACK

Similar to the RTS/CTS (Request To Send/Clear To Send) mechanism used in the IEEE 802.11, the IEEE 802.15.4 standard uses the GTS/ACK (Guaranteed Time Slots/Acknowledgments) mechanism to allocate the channel and guarantee a collision-free transmission [262]. A ZigBee device sends a GTS-request to the network coordinator which acknowledges its reception and takes a decision on whether it accepts or rejects the request. If the request is accepted, the device is notified in the next beacon management frame. In this circumstance, an attacker intercepts the beacon frames

to learn when the GTS transmissions will take place and plans to perform random jamming. This would disrupt the transmission and cause collisions which are not supposed to happen in GTS/ACK [261].

b: CSMA/CA EXPLOIT

It is also known as link-layer jamming. In this scenario, an attacker floods a radio channel with bogus frames to unnecessarily occupy the channel. This will prevent legitimate ZigBee devices from accessing the radio channel to send their data as long as the channel is occupied [263].

c: ZIGBEE RADIO JAMMING

In this scenario, the attacker generates random signals over the radio channel and causes interference. This will paralyze the network and prevent legitimate Zigbee devices from accessing the radio channel.

Countermeasure. Media disruption has always been a difficult class of attacks to mitigate. Nowadays, there exist some techniques that are based on radio monitoring to detect and localize unusual radio signals. This may be applied to detect jamming and collision when they occur and take appropriate action such as switching radio channels.

E. DOMINATION ATTACKS ON ZigBee

Zigbee defines a certain number of application profiles, such as HAPAP (Home Automation Public Application Profile) and ZLL (ZigBee Light Link Profile). These profiles define how messages are formatted, sent, and processed. This allows ZigBee devices from different vendors to properly communicate with each other within the framework of a particular application (e.g., home automation). To be compatible with other devices of different manufacturers, ZigBee devices have to implement a standard interface which subsequently implies the use of standard cryptographic keys. For example, the default trust center link key defined by ZigBee Alliance is `0×5A 0×69 0×67 0×42 0×65 0×65 0×41 0×6c 0×6C 0×69 0×61 0×6E 0×63 0×65 0×30 0×39` [264]. This key is used by the trust center to encrypt the network key and send it to the devices joining the network. If an attacker can capture the encrypted network key during joining, it will decrypt the key using the standard trust center link key. This may compromise the confidentiality of the whole network as well as its availability.

Countermeasure. The default trust center link key should not be used since the key is considered as public knowledge and thus provides the same level of security as in the unencrypted scheme. In many cases, this default and standard key has been removed from ZigBee v3.0. For personal ZigBee applications, it is highly recommended to create the cryptographic keys and physically upload them into the devices rather than sending them over the radio [250].

VIII. ATTACKS ON RFID TECHNOLOGY

A. FABRICATION ATTACKS ON RFID

In this section, we review the attacks that affect RFID authentication. These attacks allow an attacker to impersonate an RFID-tag and bypass RFID-based authentication systems, such as keyless entry systems and contactless authentication systems. Figure 16 illustrates an attack-defense tree[13] based on fabrication attacks on an RFID IoT infrastructure.

1) FABRICATION ON ENTITIES

a: SHOPLIFTING

It is also known as boosting or five-fingers-discount. In many retail shops, at the shop entrance, washroom entrance, or the exit doors of the shop, an EAS (Electronic Article Surveillance) system is installed. This system detects EAS-tagged items that are sold in the retail shop and have not been disabled [110]. For example, items that are being intentionally or accidentally taken away from the shop without paying their price will trigger the EAS alarm. Shoplifting is considered an RFID attack not based on the fact of stealing items from a shop but based on the fact of stealing the RFID-tag for further reverse-engineering. Nowadays, attackers commonly bypass EAS systems by applying different techniques, such as hiding the item in cheap foil-lined bags or using an expensive RFID EAS-jammer.

Countermeasure. Nowadays, EAS systems are equipped with the ability to detect foiled-lined bags and magnetic items and the customers are informed not to enter with such items into the shop. They can leave them at the entrance and collect those back before leaving the shop.

b: LOCATION-BASED ATTACKS

This type of attacks have two main features: (1) In a normal circumstance, an RFID-tag is instantaneously activated within the range of an RFID-reader. As a result, the RFID-reader makes the wrong assumption that the RFID-tag is in its close proximity. (2) These attacks operate at the physical-layer which make them difficult, if not impossible, to mitigate using upper-layer security protocols.

– *Distance fraud.* The distance fraud attack allows an RFID-tag operating outside the authorized range to convince an RFID-reader that it is within the authorized range [138]. The RFID-tag uses either a crafted antenna or starts sending the responses before the challenges are received to reduce the delays that may result from being outside the authorized range. The latter case can be prevented by sending multiple challenges with a strict condition that the responses must be dependent on the challenges. This attack has more effects on RFID applications where the access rights change according to the physical location.

– *Mafia fraud.* This is also known as a relay attack [265]. This attack can be performed regardless of which cryptographic system is being used and how powerful it is. It is a man or men-in-the-middle attack (depending on the number of relays). It takes place when an RFID-reader unawarely interacts with a rogue RFID-tag that manages to fool the reader into thinking that it is directly communicating with the legitimate RFID-tag. The rogue RFID-tag relays the challenges sent from the RFID-reader to the legitimate

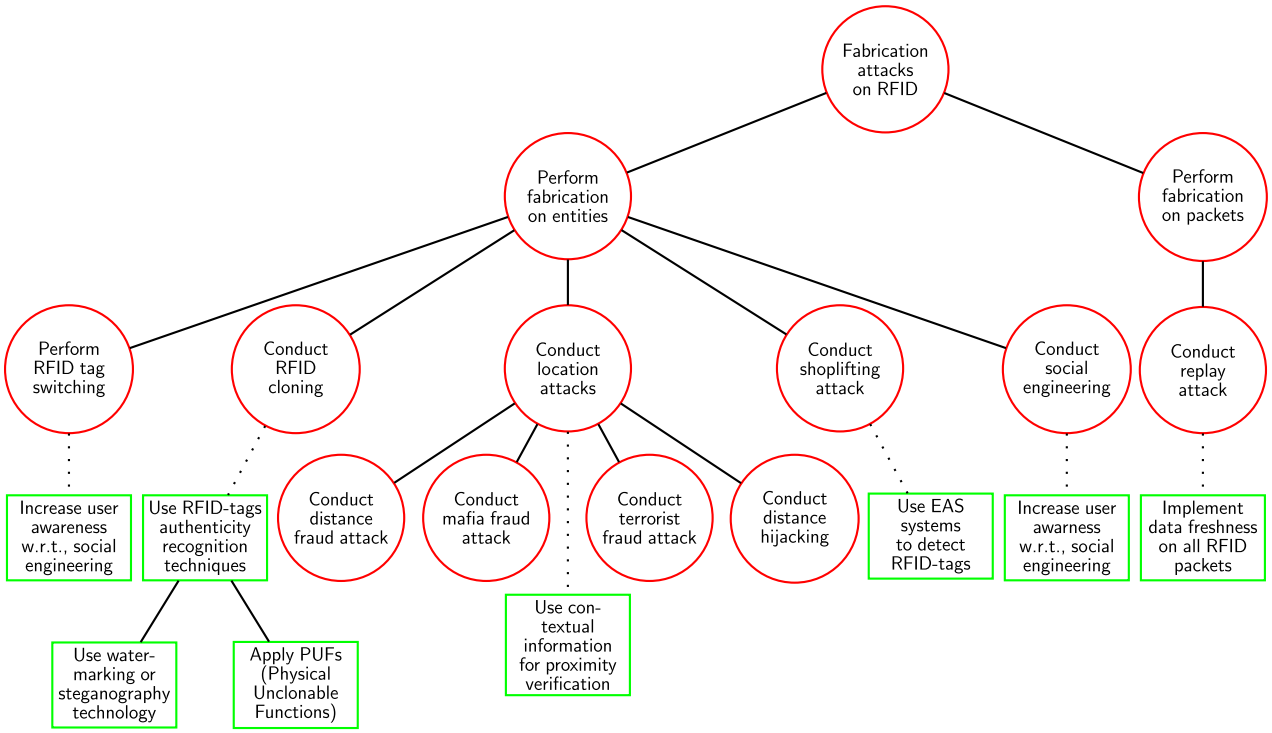

FIGURE 16. An attack-defense tree based on fabrication attacks on an RFID IoT infrastructure (○: attacks, □: defenses, ○–○: attack refinements, and ○. . .□: attack mitigations).

RFID-tag as well as the responses sent from the legitimate RFID-tag to RFID-reader. It has been shown that contactless smart cards (i.e., ISO-14443 standard) are vulnerable to relay attacks [266]. Similarly, in [267], the authors have presented a system to carry out relay attacks on ISO-14443A (e.g., digital passport, Atmel AT88SC153 smart card, and Ticket for FIFA World cup 2006 [268]). In [269], the authors demonstrated how to use a relay attack to break the passive keyless entry system of various modern cars. Moreover, hundreds of high-end cars were stolen using this attack all over the world in 2019. Note that the terms mafia-fraud attack and relay attack are interchangeable. However, some authors consider mafia-fraud attack more sophisticated and active than relay attack by assuming that in a mafia-fraud attack, attackers can manipulate and modify the messages rather than simply relaying them as in a relay attack.

– *Terrorist fraud.* In this attack [270], the adversary receives some support from a legitimate RFID-tag, e.g., with necessary information to impersonate the latter. This information does not contain any clue about the security parameters, such as the secret key. Also, this information allows the adversary to pass only a single run of the protocol.

– *Distance Hijacking.* In this attack [271], a rogue RFID-tag convinces an RFID-reader that it is at a distance which is different from the actual distance. This is done by making use of a legitimate RFID-tag to provide the tag with a false upper bound on the distance between the reader and the tag.

Countermeasure. To mitigate location-based attacks, RFID protocols apply different techniques to ensure that an RFID-tag is inside the operational range of an RFID-reader. Classical approaches are based on measuring the round trip time of the messages. Other approaches are based on the measurement of RSSI (Receiving Signal Strength Indicator) indicator, GPS (Global Positioning System) location, temperature, light intensity, and voice recognition. A practical technique, called Faraday cage, consists of using dedicated gadgets that protect the RFID-tag from being interrogated by unauthorized RFID-readers. Passive gadgets, such as metal-shielding, cover RFID-tags and prevent radio signals from reaching them. The tags remain inactive until the owner performs an action, such as pressing a button, opening a cover, or entering biometrics or password. Reactive gadgets however, such as the Vaultcard RFID blocking card [272], send strong jamming signals upon detecting a reading signal.

c: RFID CLONING

This attack consists of replicating an authentic RFID-tag to create a rogue RFID-tag that is used for impersonating the authentic RFID-tag and gaining access to certain privileges. Such attack has been performed to introduce bogus counterfeit pharmaceuticals and medications tagged with authentic cloned RFID-tags of legitimate medicals [273]. Another proof of concept was demonstrated in DEFCON 2015 [274], by creating an identical copy of the German passport using cheap off the shelf hardware.

Countermeasure. RFID-tags must be augmented with a technology that prevents cloning (e.g., HID iClass

RFID-cards), or at least, allows detecting a forged RFID-tag from an authentic one. Steganography or watermarking can be used to hide information inside authentic RFID-tags. A better alternative consists of using PUFs (Physical Unclonable Function) to implement security protocols on RFID systems.

d: SOCIAL ENGINEERING

An attacker employs social engineering techniques to compromise an RFID authentication system and gain unauthorized access to restricted locations. For example, an attacker may conduct a tailgating attack over any person entering an access restricted building that requires a badge or access card.
Countermeasure. RFID users should be aware of their surroundings. Attackers use different smart social engineering techniques to distract users, gain their trust, to perform unauthorized access to certain services or physical locations. In certain circumstances, security officers are employed to secure access to critical locations.

e: RFID-TAG SWITCHING

In this scenario, an attacker targets a RFID-tag which is tagged to a valuable object (e.g., items in the supermarket). Since RFID-tags present poor physical security, the tags that are not protected from external trespassers and can easily be captured, removed or swapped. In this attack, the attacker switches the tag of an expensive RFID-item with the one of a cheaper item to pay less at the supermarket checkout. Such an attack is possible because certain back-end servers cannot check and establish the correct association between the RFID-tag and the item.
Countermeasure. The cashiers should be aware of the approximate price of the items in the supermarket so that it can detect whether an RFID-tag has been switched on or not.

2) FABRICATION ON MESSAGES

Replay attack. If the messages that are exchanged between an RFID-tag and an RFID-reader do not contain any fresh nonces, an attacker can reuse old messages and replay them again to gain similar access or privileges.
Countermeasure. Data freshness must be provided by the authentication protocol used by the RFID application to prevent attackers from replaying old messages and gaining unauthorized access to restricted services.

B. INTERCEPTION ATTACKS ON RFID

Some sophisticated RFID-tags store not only an identification number but also other personal information which may be strictly private. For example, the VeriChip-tag is a human-implantable RFID-tag designed especially for medical-record indexing. By scanning a patient's tag, a hospital can easily locate the patient's medical record [4]. If this information is not secured, any passive or active eavesdropper can extract and learn sensitive information and then use those to perform further attacks. For example, the eavesdropper may threaten victims to publish their private data if they do not pay a certain amount of money. In the following paragraphs, we enumerate different attacks on RFID confidentiality. Figure 17 illustrates an attack-defense tree[13] based on interception attacks on an RFID IoT infrastructure.

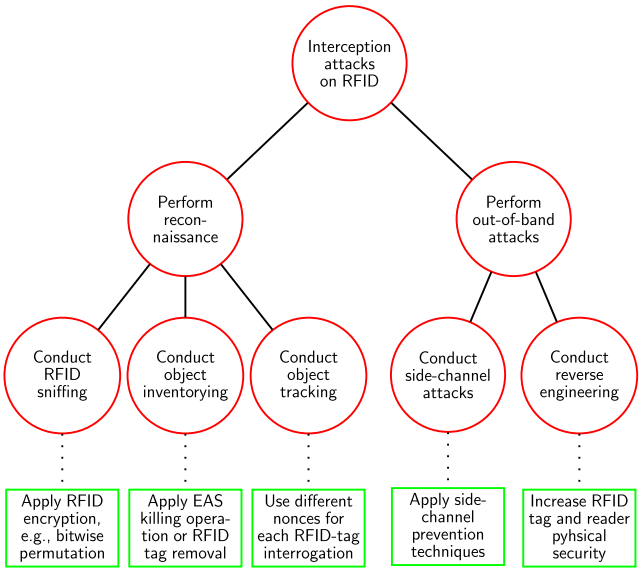

FIGURE 17. An attack-defense tree based on interception attacks on an RFID IoT infrastructure (○: attacks, □: defenses, ○–○: attack refinements, and ○. . .□: attack mitigations).

1) RFID EAVESDROPPING

Similar to other wireless technologies, RFID is also subject to eavesdropping. In this attack, the attacker uses a high-gain antenna in order to capture ongoing communications between two legitimate RFID devices, e.g., tag and reader. The attacker will be able to learn information such as the protocol being used (i.e., its message chart) or the attacker may perform traffic analysis to extract sensitive information.

2) OBJECT INVENTORYING

RFID-tags store in their internal memory a unique serial number which they use for identification. Certain tags, e.g., EPC (Electronic Product Code) tags, carry other information about the item to which they are attached. The other information may include the manufacturer of the object and product code, also known as stock keeping unit. Thus, a person carrying an EPC-tag is subject to object inventorying. An attacker can silently read what objects the person is carrying and harvest important personal information.
Countermeasure. The RFID protocol must apply lightweight (ultra-lightweight) symmetric cryptography or ECC (Elliptic Curve Cryptography) to provide data confidentiality and privacy. Sensitive information must be kept secret while being stored and processed. Any information that can be used to identify a given entity must not be revealed to unauthorized parties. Moreover, to preserve the privacy of their customers, retail shops apply EAS-killing that consists of deactivating all associated tags of the purchased items

upon payment. Thus, an eavesdropper would not be able to capture information about customer's shopping list and infer private information.

3) OBJECT TRACKING

RFID-tags generally start transmitting first after being power-supplied by a nearby RFID-reader. They send their unique identification number over the air, in most cases unencrypted, in order to identify themselves to RFID-readers. A passive eavesdropper can easily use a rogue RFID-reader with a powerful antenna to detect RFID-tags in the neighborhood and track particular tagged-objects.

Countermeasure. The static identifier of a given RFID-tag may be traced when being transmitted over the air. Applying only encryption on the identifier will just transform the identifier into a meta-identifier which remains static and traceable. Thus, it is recommended to use a new nonce whenever the RFID-tag is requested by an RFID-reader. The use of the nonce along with encryption makes the transmitted information useless and unique for each session which harden the tracking process. Other techniques can also be used as well to make a RFID-tag untraceable [160], [173], [275], [276].

4) SIDE-CHANNEL ATTACKS

Currently, there are two forms of side-channel attacks on RFID, timing-based and power-based. A timing-based attack consists of extracting information (e.g., secret keys) from the variations of the processing times. A power-based attack consists of extracting information from the variations of the power consumption [267], [277].

Countermeasure. The RFID protocol must be implemented in such a way so that side-channel attacks become very hard to realize. Masking [224], cross-copying [225], conditional assignment [226], bucketing [227], and predictive timing mitigation [228] can be adopted while implementing the protocol. These techniques make it harder for an attacker to perform side-channel attacks.

5) REVERSE ENGINEERING

RFID-tags and RFID-readers are subject to physical attacks. In this scenario, an attacker captures an RFID-tag or RFID-reader, and applies reverse-engineering to extract information such as the used protocol, cryptographic keys, and other confidential information.

Countermeasure. To physically secure RFID devices, additional measures should be taken. Traditional security measures can be used such as cameras, guards, and misuse detectors (e.g., an alarm is triggered upon RFID-tag removal).

C. MODIFICATION ATTACKS ON RFID

An attacker may take over an RFID-tag or reader and try to modify the internal protocol in order to adapt it to its needs. For instance, the attacker can modify the reader's functions in such a way so that it authenticates the attacker RFID-tag as a legitimate tag to bypass certain authentication systems.

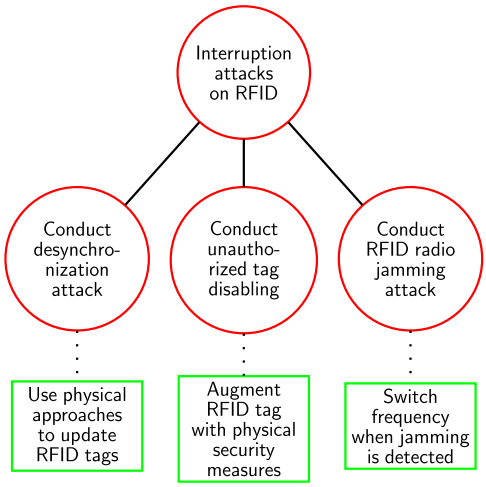

FIGURE 18. An attack-defense tree based on interruption attacks on an RFID IoT infrastructure (○: attacks, □: defenses, ○−○: attack refinements, and ○. . .□: attack mitigations).

D. INTERRUPTION ATTACKS ON RFID

Due to the small size and limited resource capacity, RFID-tags are attractive target devices for denial of service attacks. In the following paragraphs, we enumerate RFID attacks on availability. Figure 18 illustrates an attack-defense tree[13] based on interruption attacks on an RFID IoT infrastructure.

1) RFID DESYNCHRONIZATION

In some applications, the RFID-tag security parameters, e.g., secret key, need to be updated. The attacker sets a man-in-the-middle scenario and prevents the RFID-tag from being synchronized and updated with the RFID-reader. Thus, the tag will be containing old security parameters and will fail in all later authentication challenges.

Countermeasure. The network administrator must ensure that its RFID devices are kept updated. A more secure and not scalable way to do that is to physically update the RFID-tags and RFID-readers using dedicated devices.

2) UNAUTHORIZED TAG-DISABLING

In this scenario, an attacker uses a rogue RFID-reader to manipulate an RFID-tag so that it becomes permanently or temporarily unavailable. This can be achieved by removing or destroying a physical tag (e.g., applying pressure, chemical exposure, or trimming off any visible antenna), misusing the kill command, or using a dedicated device (e.g., RFID-Zapper) to disable the RFID-tag. This will prevent all legitimate RFID-readers from communicating with the vandalized RFID-tag [109], [278].

Countermeasure. Most sensitive RFID-tags are equipped with an alarm (e.g., alarm with noise level $\geq$110dB) that triggers when the tag is undergoing an abnormal pressure.

3) RFID SIGNALS JAMMING

In this scenario, an attacker uses an RFID-device to generate random signals over the used RFID frequencies

(e.g., LF, HF, or UHF) to cram the radio channel and disrupt the correct function of RFID-tags and readers.

Countermeasure. RFID jamming can be detected by setting an RFID radio scanner that triggers an alarm when it receives useless and unexpected RFID signals. Some avoidance techniques consist of switching the operational frequency between the RFID-tag and the RFID-reader. For example, certain UHF-RFID readers in the US (e.g., those operating between 902.0 MHz and 928.0 MHz), employ frequency hopping spread spectrum (FHSS) to avoid interferences. These RFID-readers change their operational frequency from time to time. However, this only makes sense if the bandwidth of the used frequency band is wide. Otherwise, switching the frequency in a narrow band is not useful.

E. DOMINATION ATTACKS ON RFID

In this section, we review RFID attacks that affect multiple security services at a time.

Offline RFID-tag key cracking. Most RFID systems use a challenge-response mechanism along with a shared secret key between an RFID-tag and an RFID-reader for authentication. Due to the resource-constrained nature of some RFID-tags, very short keys are used (e.g., 40-bit keys in a Digital Signature Transponders or DST). This makes the brute force attack possible to crack the secret key and clone the RFID-tag [279]. Some researchers cracked a car DST-key in less than 30 minutes and stole their own car as well as purchased gas using a cloned SpeedPass [110].

Countermeasure. When short keys cannot be replaced by longer keys, then it is recommended to limit the use of a short key for a limited time. The key should be changed frequently.

IX. CONCLUSION

The Internet of Things (IoT) connects billions of heterogeneous devices, called Things, using different communication technologies and protocols to provide end-users, all over the world, with access to a variety of smart applications. It also invites cybercriminals who exploit the IoT infrastructures to conduct large scale, distributed, and devastating cyberattacks. The security of IoT infrastructures strongly depends on the security of its wired and wireless infrastructures. While the wireless infrastructure is thought to be the most outspread part in IoT, it is at the same time the most vulnerable and accessible for attackers. Hence, more focus should be placed on the security of wireless infrastructures of IoT.

In this paper, we have introduced an attack classification for wireless IoT attacks. This classification categorizes an attack based on which security service is compromised by the attack. We have adopted the classification to review the attacks that occurred in the last two decades on Wi-Fi, Bluetooth, ZigBee, and RFID wireless communication technologies. These wireless communication technologies are considered to be the most used for short-range wireless communications in IoT. We have also discussed possible countermeasures that can be applied to mitigate, or at least detect, certain attacks. In summary, the paper makes the following main contributions: (1) Present a generic taxonomy of attacks in Wi-Fi, Bluetooth, ZigBee, and RFID IoT infrastructures. (2) Survey the attacks on Wi-Fi, Bluetooth, ZigBee, and RFID technologies. (3) Analyze and recommend possible countermeasures to mitigate the reviewed attacks.

Considering the reviewed attacks and the existing security mechanisms, we have observed that most attacks were due to the flaws left on the authentication protocol. We claim that authentication is the most important and critical security service, in the sense where compromising authentication would in most cases lead to the compromising of the remaining security services, such as confidentiality, integrity, and availability. Therefore, we claim that if authentication is vigorously considered and perfectly implemented, a large number of attacks will be completely mitigated. Although the existing authentication mechanisms in the considered wireless communication technologies provide a certain level of security, we believe that the application of these mechanisms will not last for too long. In fact, as IoT is rapidly transforming the Internet into a Thing to Thing communication system, the need for new authentication protocols, mainly thing-to-thing authentication protocols, is rising. Also, besides authentication, we believe that in most cases the reviewed attacks that are related to compromising data integrity and system availability have a bigger impact than the attacks that are related to breaching IoT data confidentiality.

In the future, we will survey and classify mid and long-range IoT wireless communication technologies, such as LoRa, Sigfox, NB-IoT, WiMax, UMTS, 4G/LTE, and 5G, which are largely used in large-scale IoT applications.

REFERENCES

[1] Statista. (2018). *Internet of Things-Number of Connected Devices Worldwide 2015–2025*. Accessed: Jan. 19, 2020. [Online]. Available: https://www.statista.com/statistics/471264/iot-number-of-connected-devices-worldwide/

[2] Gartner. (2018). *Gartner Says 8.4 Billion Connected Things Will Be in Use in 2017, Up 31 Percent From 2016*. Accessed: Jan. 19, 2020. [Online]. Available: https://www.gartner.com/en/newsroom/press-releases/2017-02-07-gartner-says-8-billion-connected-things-will-be-in-use-in-2017-up-31-percent-from-2016

[3] T. Igoe, *Getting Started With RFID: Identify Objects in the Physical World With Arduino*. Newton, MA, USA: O'Reilly Media, 2012.

[4] J. Halamka, "Straight from the shoulder," *New England J. Med.*, vol. 353, no. 4, pp. 331–333, Jul. 2005.

[5] CTV-Calgary-News. (2012). *Wireless Waves Used to Track Travel Times*. Accessed: Jan. 19, 2020. [Online]. Available: https://calgary.ctvnews.ca/wireless-waves-used-to-track-travel-times-1.1054731

[6] Tenbu. (2009). *Tenbu's Nio Is Kind of Like a Car Alarm for Your Cellphone*. Accessed: Jan. 19, 2020. [Online]. Available: http://www.ohgizmo.com/2009/03/30/tenbu-nio-is-kind-of-like-a-car-alarm-for-your-cellphone/

[7] M. Jin, N. Bekiaris-Liberis, K. Weekly, C. J. Spanos, and A. M. Bayen, "Occupancy detection via environmental sensing," *IEEE Trans. Autom. Sci. Eng.*, vol. 15, no. 2, pp. 443–455, Apr. 2018.

[8] ACHRNews. (2013). *Control Your Castle: The Latest in HVAC Home Automation*. Accessed: Jan. 19, 2020. [Online]. Available: https://www.achrnews.com/articles/124160-control-your-castle-the-latest-in-hvac-home-automation

[9] Consumer-Reports-News. (2016). *Nest Protect: Smoke and CO Alarms*. Accessed: Jan. 19, 2020. [Online]. Available: https://www.consumerreports.org/cro/news/2014/02/consumer-reports-review-of-nest-protect-smoke-and-co-alarm/index.htm

[10] M. N. K. Boulos and N. M. Al-Shorbaji, "On the Internet of Things, smart cities and the WHO healthy cities," *Int. J. Health Geogr.*, vol. 13, no. 1, p. 10, 2014.
[11] Smart-Home-Geeks. (2017). *Sure Flap-Smart Cat Flap Coming Soon!* Accessed: Jan. 19, 2020. [Online]. Available: https://www.smarthomegeeks.co.uk/news/smart-cat-flap/
[12] The-Reporter. (2017). *First Smart Parking Goes Operational in Ethiopia*. Accessed: Jan. 19, 2020. [Online]. Available: https://www.thereporterethiopia.com/content/first-smart-parking-goes-operational
[13] Orthogonal. (2018). *The Growing Significance of Bluetooth BTLE in Healthcare*. Accessed: Jan. 19, 2020. [Online]. Available: https://www.thereporterethiopia.com/content/first-smart-parking-goes-operational
[14] EECatalog. (2017). *Bluetooth 5 Expands into the Smart Grid*. Accessed: Jan. 19, 2020. [Online]. Available: http://eecatalog.com/wireless/2017/09/07/Bluetooth-5-expands-into-the-smart-grid/
[15] T. Kalaivani, A. Allirani, and P. Priya, "A survey on Zigbee based wireless sensor networks in agriculture," in *Proc. 3rd Int. Conf. Trendz Inf. Sci. Comput.*, Dec. 2011, pp. 85–89.
[16] The-Guardian. (2016). *DDoS Attack that Disrupted Internet Was Largest of Its Kind in History, Experts Say*. Accessed: Jan. 19, 2020. [Online]. Available: https://www.theguardian.com/technology/2016/oct/26/ddos-attack-dyn-mirai-botnet
[17] NewYork-Times. (2011). *Stuxnet Worm Attack on Iranian Nuclear Facilities*. Accessed: Jan. 19, 2020. [Online]. Available: https://www.nytimes.com/2011/01/16/world/middleeast/16stuxnet.html
[18] K. Zetter. (2016). *Inside The Cunning, Unprecedented Hack of Ukraine's Power Grid*. Accessed: Jan. 19, 2020. [Online]. Available: https://www.wired.com/2016/03/inside-cunning-unprecedented-hack-ukraines-power-grid/
[19] The-Guardian. (2015). *Fiat Chrysler Recalls 1.4m Vehicles in Wake of Jeep Hacking Revelation*. Accessed: Jan. 19, 2020. [Online]. Available: https://www.theguardian.com/business/2015/jul/24/fiat-chrysler-recall-jeep-hacking
[20] BleepingComputer. (2017). *BrickerBot Dev Claims Cyber-Attack That Affected Over 60,000 Indian Modems*. Accessed: Jan. 19, 2020. [Online]. Available: https://www.bleepingcomputer.com/news/security/brickerbot-dev-claims-cyber-attack-that-affected-over-60-000-indian-modems/
[21] E. Ronen, A. Shamir, A.-O. Weingarten, and C. O'Flynn, "IoT goes nuclear: Creating a zigbee chain reaction," *IEEE Secur. Privacy*, vol. 16, no. 1, pp. 54–62, Jan. 2018.
[22] LinkLabs. (2015). *Five Types of Wireless Technology for the IoT*. [Online]. Available: https://www.link-labs.com/blog/types-of-wireless-technology
[23] Cognixia. (2017). *Wireless Communication Technologies for IoT*. Accessed: Jan. 19, 2020. [Online]. Available: https://www.cognixia.com/wireless-communication-technologies-for-IoT
[24] A. Rathore. (2018). *Wireless Technologies for IoT*. Accessed: Jan. 19, 2020. [Online]. Available: https://iot.electronicsforu.com/research-articles/wireless-technologies-iot/
[25] S. Al-Sarawi, M. Anbar, K. Alieyan, and M. Alzubaidi, "Internet of Things communication protocols: Review," in *Proc. 8th Int. Conf. Inf. Technol.*, 2017, pp. 685–690.
[26] N. Lethaby. (2018). *Wireless Connectivity for The Internet of Things: One Size Does Not Fit All*. Accessed: Jan. 19, 2020. [Online]. Available: http://www.ti.com/lit/wp/swry010a/swry010a.pdf
[27] F. A. Alaba, M. Othman, I. A. T. Hashem, and F. Alotaibi, "Internet of Things security: A survey," *J. Netw. Comput. Appl.*, vol. 88, pp. 10–28, Jun. 2017.
[28] Y. Yang, L. Wu, G. Yin, L. Li, and H. Zhao, "A survey on security and privacy issues in Internet-of-Things," *IEEE Internet Things J.*, vol. 4, no. 5, pp. 1250–1258, Oct. 2017.
[29] T. Borgohain, U. Kumar, and S. Sanyal, "Survey of security and privacy issues of Internet of Things," *CoRR*, vol. abs/1501.02211, pp. 1–7, Jan. 2015.
[30] A. Oracevic, S. Dilek, and S. Ozdemir, "Security in Internet of Things: A survey," in *Proc. Int. Symp. Netw., Comput. Commun.*, May 2017, pp. 1–6.
[31] J. Deogirikar and A. Vidhate, "Security attacks in IoT: A survey," in *Proc. Int. Conf. I-SMAC (IoT Social, Mobile, Anal. Cloud) (I-SMAC)*, Feb. 2017, pp. 32–37.
[32] D. E. Kouicem, A. Bouabdallah, and H. Lakhlef, "Internet of Things security: A top-down survey," *Comput. Netw.*, vol. 141, pp. 199–221, Aug. 2018.
[33] M. Conti, A. Dehghantanha, K. Franke, and S. Watson, "Internet of Things security and forensics: Challenges and opportunities," *Future Gener. Comput. Syst.*, vol. 78, pp. 544–546, Jan. 2018.
[34] M. Ammar, G. Russello, and B. Crispo, "Internet of Things: A survey on the security of IoT frameworks," *J. Inf. Secur. Appl.*, vol. 38, pp. 8–27, Feb. 2018.
[35] J. M. De-Funtes, L. Gonzalez-Manzano, J. Lopez, and P. Peris-Lopez, "Editorial: Security and privacy in Internet of Things," in *Mobile Networks and Applications*. Cham, Switzerland: Springer, 2018, pp. 1–3.
[36] G. Chu, N. Apthorpe, and N. Feamster, "Security and privacy analyses of Internet of Things Children's toys," *IEEE Internet Things J.*, vol. 6, no. 1, pp. 978–985, Feb. 2019.
[37] M. Dabbagh and A. Rayes, "Internet of Things security and privacy," in *Internet Things From Hype to Reality*. Cham, Switzerland: Springer, 2019, pp. 211–238.
[38] W. Sun, Z. Cai, Y. Li, F. Liu, S. Fang, and G. Wang, "Security and privacy in the medical Internet of Things," in *Security and Communication Networks*. Cham, Switzerland: Springer, 2018.
[39] Z. Ren, X. Liu, R. Ye, and T. Zhang, "Security and privacy on Internet of Things," in *Proc. IEEE Int. Conf. Electron. Inf. Emergency Commun.*, Jul. 2017, pp. 140–144.
[40] I. Andrea, C. Chrysostomou, and G. Hadjichristofi, "Internet of Things: Security vulnerabilities and challenges," in *Proc. IEEE Symp. Comput. Commun. (ISCC)*, Jul. 2015, pp. 180–187.
[41] S. Vashi, J. Ram, J. Modi, S. Verma, and C. Prakash, "Internet of Things (IoT): A vision, architectural elements, and security issues," in *Proc. Int. Conf. I-SMAC (IoT Social, Mobile, Anal. Cloud) (I-SMAC)*, Feb. 2017, pp. 492–496.
[42] M. M. Ahemd, M. A. Shah, and A. Wahid, "IoT security: A layered approach for attacks & defenses," in *Proc. Int. Conf. Commun. Technol. (ComTech)*, Apr. 2017, pp. 104–110.
[43] M. Daud, Q. Khan, and Y. Saleem, "A study of key technologies for IoT and associated security challenges," in *Proc. Int. Symp. Wireless Syst. Netw. (ISWSN)*, Nov. 2017, pp. 1–6.
[44] I. R. Waz, M. A. Sobh, and A. M. Bahaa-Eldin, "Internet of Things security platforms," in *Proc. 12th Int. Conf. Comput. Eng. Syst.*, 2017, pp. 500–507.
[45] Z. Bakhshi, A. Balador, and J. Mustafa, "Industrial IoT security threats and concerns by considering cisco and microsoft IoT reference models," in *Proc. IEEE Wireless Commun. Netw. Conf. Workshops (WCNCW)*, Apr. 2018, pp. 173–178.
[46] M. Nawir, A. Amir, N. Yaakob, and O. B. Lynn, "Internet of Things (IoT): Taxonomy of security attacks," in *Proc. 3rd Int. Conf. Electron. Design (ICED)*, Aug. 2016, pp. 321–326.
[47] M. M. Hossain, M. Fotouhi, and R. Hasan, "Towards an analysis of security issues, challenges, and open problems in the Internet of Things," in *Proc. IEEE World Congr. Services*, Jun. 2015, pp. 21–28.
[48] K. Sonar and H. Upadhyay, "A survey: DDOS attack on Internet of Things," *Int. J. Eng. Res. Develop.*, vol. 10, no. 11, pp. 58–63, 2014.
[49] A. Alsaidi and F. Kausar, "Security attacks and countermeasures on cloud assisted IoT applications," in *Proc. IEEE Int. Conf. Smart Cloud (SmartCloud)*, Sep. 2018, pp. 213–217.
[50] A. Djenna and D. Eddine Saidouni, "Cyber attacks classification in IoT-based-healthcare infrastructure," in *Proc. 2nd Cyber Secur. Netw. Conf. (CSNet)*, Oct. 2018, pp. 1–4.
[51] P. P. Lokulwar and H. R. Deshmukh, "Threat analysis and attacks modelling in routing towards IoT," in *Proc. Int. Conf. I-SMAC (IoT Social, Mobile, Anal. Cloud) (I-SMAC)*, Feb. 2017, pp. 721–726.
[52] S. Rizvi, A. Kurtz, J. Pfeffer, and M. Rizvi, "Securing the Internet of Things (IoT): A security taxonomy for IoT," in *Proc. 17th IEEE Int. Conf. Trust, Secur. Privacy Comput. Commun./ 12th IEEE Int. Conf. Big Data Sci. Eng. (TrustCom/BigDataSE)*, Aug. 2018, pp. 163–168.
[53] K. Kimani, V. Oduol, and K. Langat, "Cyber security challenges for IoT-based smart grid networks," *Int. J. Crit. Infrastruct. Protection*, vol. 25, pp. 36–49, Jun. 2019.
[54] M. B. Mohamad Noor and W. H. Hassan, "Current research on Internet of Things (IoT) security: A survey," *Comput. Netw.*, vol. 148, pp. 283–294, Jan. 2019.
[55] A. Soni, R. Upadhyay, and A. Jain, "Internet of Things and wireless physical layer security: A survey," in *Computer Communication, Networking and Internet Security*. Cham, Switzerland: Springer, 2017, pp. 115–123.

[56] N. Neshenko, E. Bou-Harb, J. Crichigno, G. Kaddoum, and N. Ghani, "Demystifying IoT security: An exhaustive survey on IoT vulnerabilities and a first empirical look on Internet-scale IoT exploitations," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 3, pp. 2702–2733, 3rd Quart., 2019.

[57] C. Karlof and D. Wagner, "Secure routing in wireless sensor networks: Attacks and countermeasures," in *Proc. 1st IEEE Int. Workshop Sensor Netw. Protocols Appl.*, May 2003, pp. 113–127.

[58] D. M. Mendez, I. Papapanagiotou, and B. Yang, "Internet of Things: Survey on security and privacy," *CoRR*, vol. abs/1707.01879, pp. 1–16, Jul. 2017.

[59] V. Gharu, M. Pawar, and J. Agarwal, "A literature survey on security issues of WSN and different types of attacks in network," *Indian J. Comput. Sci. Eng.*, vol. 8, no. 2, pp. 80–83, 2017.

[60] I. Tomic and J. A. McCann, "A survey of potential security issues in existing wireless sensor network protocols," *IEEE Internet Things J.*, vol. 4, no. 6, pp. 1910–1923, Dec. 2017.

[61] A. Rani and S. Kumar, "A survey of security in wireless sensor networks," in *Proc. 3rd Int. Conf. Comput. Intell. Commun. Technol.*, Feb. 2017, pp. 1–5.

[62] P. Sinha, V. K. Jha, A. K. Rai, and B. Bhushan, "Security vulnerabilities, attacks and countermeasures in wireless sensor networks at various layers of OSI reference model: A survey," in *Proc. Int. Conf. Signal Process. Commun. (ICSPC)*, Jul. 2017, pp. 288–293.

[63] T. Azzabi, H. Farhat, and N. Sahli, "A survey on wireless sensor networks security issues and military specificities," in *Proc. Int. Conf. Adv. Syst. Electric Technol. (IC_ASET)*, Jan. 2017, pp. 66–72.

[64] A. S. Naik and R. Murugan, "Security attacks and energy efficiency in wireless sensor networks: A survey," *Int. J. Appl. Eng. Res.*, vol. 13, no. 1, pp. 107–112, 2018.

[65] I. Butun, P. Österberg, and H. Song, "Security of the Internet of Things: Vulnerabilities, attacks, and countermeasures," *IEEE Commun. Surveys Tuts.*, vol. 22, no. 1, pp. 616–644, 1st Quart., 2020.

[66] B. Wu, J. Chen, J. Wu, and M. Cardei, "A survey of attacks and countermeasures in mobile ad hoc networks," in *Wireless Network Security*. Cham, Switzerland: Springer, 2007, pp. 103–135.

[67] R. KumarSingh, R. Joshi, and M. Singhal, "Analysis of security threats and vulnerabilities in mobile ad hoc network (MANET)," *Int. J. Comput. Appl.*, vol. 68, no. 4, pp. 25–29, 2013.

[68] P. Rajakumar, V. T. Prasanna, and A. Pitchaikkannu, "Security attacks and detection schemes in MANET," in *Proc. Int. Conf. Electron. Commun. Syst. (ICECS)*, Feb. 2014, pp. 1–6.

[69] J. G. Ponsam and R. Srinivasan, "A survey on MANET security challenges, attacks and its countermeasures," *Int. J. Emerg. Trends Technol. Comput. Sci.*, vol. 3, no. 1, pp. 274–279, 2014.

[70] U. K. Singh, D. N. Goswami, K. C. Phuleria, and S. Sharma, "An analysis of security attacks found in mobile ad-hoc network," *Int. J. Sci. Eng. Res.*, vol. 5, no. 5, pp. 1586–1592, 2014.

[71] N. Raj, P. Bharti, and S. Thakur, "Vulnerabilities, challenges and threats in securing mobile ad-hoc network," in *Proc. 5th Int. Conf. Commun. Syst. Netw. Technol.*, Apr. 2015, pp. 771–775.

[72] A. Dorri, S. R. Kamel, and E. Kheyrkhah, "Security challenges in mobile ad hoc networks: A survey," *Int. J. Comput. Sci. Eng. Survey*, vol. 6, no. 1, pp. 15–29, 2015.

[73] P. Chahal, G. Kumar Tak, and A. Singh Tomar, "Comparative analysis of various attacks on MANET," *Int. J. Comput. Appl.*, vol. 111, no. 12, pp. 42–46, 2015.

[74] N. Zanoon, N. Albdour, H. S. A. Hamatta, and R. M. Al-Tarawneh, "Security challenges as a factor affecting the security of manet: Attacks and security solutions," *Int. J. Netw. Secur. Appl.*, vol. 7, no. 3, pp. 01–13, May 2015.

[75] R. Meddeb, B. Triki, F. Jemili, and O. Korbaa, "A survey of attacks in mobile ad hoc networks," in *Proc. Int. Conf. Eng. MIS (ICEMIS)*, May 2017, pp. 1–7.

[76] D. Djenouri, L. Khelladi, and A. N. Badache, "A survey of security issues in mobile ad hoc and sensor networks," *IEEE Commun. Surveys Tuts.*, vol. 7, no. 4, pp. 2–28, 4th Quart., 2005.

[77] B. Kannhavong, H. Nakayama, Y. Nemoto, N. Kato, and A. Jamalipour, "A survey of routing attacks in mobile ad hoc networks," *IEEE Wireless Commun.*, vol. 14, no. 5, pp. 85–91, Oct. 2007.

[78] S. Agrawal, S. Jain, and S. Sharma, "A survey of routing attacks and security measures in mobile ad-hoc networks," *CoRR*, vol. abs/1105.5623, pp. 41–48, May 2011.

[79] J. M. De-Fuentes, A. I. Gonzalez-Tablas, and A. Ribagorda, "Overview of security issues in vehicular ad-hoc networks, in *Handbook of Research on Mobility and Computing: Evolving Technologies and Ubiquitous Impacts*. Hershey, PA, USA: IGI Global, 2011.

[80] A. Rawat, S. Sharma, and R. Sushil, "VANET: Security attacks and its possible solutions," in *J. Inf. Oper. Manage.*, vol. 3, no. 1, pp. 301–303, 2012.

[81] H. Hasrouny, A. Samhat, C. Bassil, and A. Laouiti, "VANet security challenges and solutions: A survey," *Veh. Commun.*, vol. 7, pp. 7–20, 2017.

[82] P. Agarwal, "Technical review on different applications, challenges and security in VANET," *J. Multimedia Technol. Recent Adv.*, vol. 4, no. 3, pp. 21–30, 2017.

[83] M. A. H. Al Junaid, A. A. Syed, M. N. M. Warip, K. N. F. K. Azi, and N. H. Romli, "Classification of security attacks in VANET: A review of requirements and perspectives," in *Proc. Malaysian Tech. Univ. Conf. Eng. Technol.*, vol. 150, pp. 1–7, Feb. 2018.

[84] Sheikh, Liang, and Wang, "A survey of security services, attacks, and applications for vehicular ad hoc networks (VANETs)," *Sensors*, vol. 19, no. 16, p. 3589, 2019.

[85] M. Jain and R. Saxena, "VANET: Security attacks, solution and simulation," in *Proc. 2nd Int. Conf. Comput. Intell. Informat.*, vol. 712, 2018, pp. 457–466.

[86] M. N. Mejri, J. Ben-Othman, and M. Hamdi, "Survey on VANET security challenges and possible cryptographic solutions," *Veh. Commun.*, vol. 1, no. 2, pp. 53–66, Apr. 2014.

[87] R. G. Engoulou, M. Bellaïche, S. Pierre, and A. Quintero, "VANET security surveys," *Comput. Commun.*, vol. 44, pp. 1–13, May 2014.

[88] S. K. Panigrahy, S. K. Jena, and A. K. Turuk, "Security in Bluetooth, RFID and wireless sensor networks," in *Proc. Int. Conf. Commun., Comput. Secur. (ICCCS)*, 2011, pp. 628–633.

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

[90] M. Shin, J. Ma, A. Mishra, and W. A. Arbaugh, "Wireless network security and interworking," *Proc. IEEE*, vol. 94, no. 2, pp. 455–466, Feb. 2006.

[91] C. Kolias, G. Kambourakis, A. Stavrou, and S. Gritzalis, "Intrusion detection in 802.11 networks: Empirical evaluation of threats and a public dataset," *IEEE Commun. Surveys Tuts.*, vol. 18, no. 1, pp. 184–208, 1st Quart., 2016.

[92] H. Berghel and J. Uecker, "WiFi attack vectors," *Commun. ACM*, vol. 48, no. 8, pp. 21–28, 2005.

[93] C. Sudar, S. K. Arjun, and L. R. Deepthi, "Time-based one-time password for Wi-Fi authentication and security," in *Proc. Int. Conf. Adv. Comput., Commun. Informat. (ICACCI)*, Sep. 2017, pp. 1212–1216.

[94] K. Bicakci and B. Tavli, "Denial-of-service attacks and countermeasures in IEEE 802.11 wireless networks," *Comput. Standards Interfaces*, vol. 31, no. 5, pp. 931–941, Sep. 2009.

[95] C. Gehrmann, J. Persson, and B. Smeets, *Bluetooth Security*. Norwood, MA, USA: Artech House, 2004.

[96] N. B. I. Minar and M. Tarique, "Bluetooth security threats and solutions: A survey," *J. Distrib. Parallel Syst.*, vol. 3, no. 1, pp. 127–148, 2012.

[97] M. M. W. Iqbal, F. Kausar, and M. A. Wahla, "Attacks on Bluetooth security architecture and its countermeasures," in *Proc. 4th Int. Conf. Inf. Secur. Assurance*, 2010, pp. 190–197.

[98] J. D. Padgette, "Bluetooth security in the DoD," in *Proc. IEEE Mil. Commun. Conf. (MILCOM)*, Oct. 2009, pp. 1–6.

[99] P. Cope, J. Campbell, and T. Hayajneh, "An investigation of Bluetooth security vulnerabilities," in *Proc. IEEE 7th Annu. Comput. Commun. Workshop Conf. (CCWC)*, Jan. 2017, pp. 1–7.

[100] S. S. Hassan, S. D. Bibon, M. S. Hossain, and M. Atiquzzaman, "Security threats in Bluetooth technology," *Comput. Secur.*, vol. 74, pp. 308–322, May 2018.

[101] L. Chen, P. Cooper, and Q. Liu, "Security in Bluetooth networks and communications," in *Wireless Network Security*. Cham, Switzerland: Springer, 2013, pp. 77–94.

[102] V. K. Dubey, K. Vaishali, N. Behar, and M. Shrivastava, "A review on Bluetooth security vulnerabilities and a proposed prototype model for enhancing security against MITM attack," *Int. J. Res. Stud. Comput. Sci. Eng.*, 2015, pp. 69–75.

[103] O. Olawumi, K. Haataja, M. Asikainen, N. Vidgren, and P. Toivanen, "Three practical attacks against ZigBee security: Attack scenario definitions, practical experiments, countermeasures, and lessons learned," in *Proc. 14th Int. Conf. Hybrid Intell. Syst.*, Dec. 2014, pp. 199–206.

[104] M. Sharma, A. Tandon, S. Narayan, and B. Bhushan, "Classification and analysis of security attacks in WSNs and IEEE 802.15.4 standards: A survey," in *Proc. 3rd Int. Conf. Adv. Comput., Commun. Autom. (ICACCA) (Fall)*, Sep. 2017, pp. 1–5.

[105] N. Sastry and D. Wagner, "Security considerations for IEEE 802.15.4 networks," in *Proc. ACM Workshop Wireless Secur. (WiSe)*, 2004, pp. 32–42.

[106] C. O'Flynn and Z. Chen, "Power analysis attacks against IEEE 802.15.4 nodes," in *Proc. 7th Int. Workshop Constructive Side-Channel Anal. Secure Design*, 2016, pp. 55–70.

[107] G. de Meulenaer and F.-X. Standaert, "Stealthy compromise of wireless sensor nodes with power analysis attacks," in *Proc. 2nd Int. Conf. Mobile Lightweight Wireless Syst.*, 2010, pp. 229–242.

[108] H. Li, Y. Chen, and Z. He, "The survey of RFID attacks and defenses," in *Proc. 8th Int. Conf. Wireless Commun., Netw. Mobile Comput.*, Sep. 2012, pp. 1–4.

[109] A. Mitrokotsa, M. R. Rieback, and A. S. Tanenbaum, "Classifying RFID attacks and defenses," *Inf. Syst. Frontiers*, vol. 12, no. 5, pp. 491–505, Nov. 2010.

[110] A. Juels, "RFID security and privacy: A research survey," *IEEE J. Sel. Areas Commun.*, vol. 24, no. 2, pp. 381–394, Feb. 2006.

[111] K. Bu, M. Weng, Y. Zheng, B. Xiao, and X. Liu, "You can clone but you cannot hide: A survey of clone prevention and detection for RFID," *IEEE Commun. Surveys Tuts.*, vol. 19, no. 3, pp. 1682–1700, 3rd Quart., 2017.

[112] M. El Beqqal and M. Azizi, "Classification of major security attacks against RFID systems," in *Proc. Int. Conf. Wireless Technol., Embedded Intell. Syst. (WITS)*, Apr. 2017, pp. 1–6.

[113] M. R. Rieback, B. Crispo, and A. S. Tanenbaum, "The evolution of RFID security," *IEEE Pervas. Comput.*, vol. 5, no. 1, pp. 62–69, Mar. 2006.

[114] T. Van Dursen and S. Radomirovic, "Attacks on RFID protocols," in *Proc. IACR Cryptol. ePrint Arch.*, 2008, pp. 1–56.

[115] C. Hennebert and J. D. Santos, "Security protocols and privacy issues into 6LoWPAN stack: A synthesis," *IEEE Internet Things J.*, vol. 1, no. 5, pp. 384–398, Oct. 2014.

[116] A. Mayzaud, R. Badonnel, I. Chrisment, and I. G. Est-Nancy, "A taxonomy of attacks in RPL-based Internet of Things," *Int. J. Netw. Secur.*, vol. 18, no. 3, pp. 459–473, 2016.

[117] S. Vohra and R. Srivastava, "A survey on techniques for securing 6LoWPAN," in *Proc. 5th Int. Conf. Commun. Syst. Netw. Technol.*, Apr. 2015, pp. 643–647.

[118] P. Pongle and G. Chavan, "A survey: Attacks on RPL and 6LoWPAN in IoT," in *Proc. Int. Conf. Pervas. Comput. (ICPC)*, Jan. 2015, pp. 1–6.

[119] S. Na, D. Hwang, W. Shin, and K.-H. Kim, "Scenario and countermeasure for replay attack using join request messages in LoRaWAN," in *Proc. Int. Conf. Inf. Netw. (ICOIN)*, 2017, pp. 718–720.

[120] I. Butun, N. Pereira, and M. Gidlund, "Analysis of LoRaWAN v1.1 security: Research paper," in *Proc. 4th ACM MobiHoc Workshop Exper. Design Implement. Smart Objects (SMARTOBJECTS)*, 2018, pp. 1–6.

[121] X. Yang, E. Karampatzakis, C. Doerr, and F. Kuipers, "Security vulnerabilities in LoRaWAN," in *Proc. IEEE/ACM 3rd Int. Conf. Internet-of-Things Design Implement. (IoTDI)*, Apr. 2018, pp. 129–140.

[122] I. Unwala, Z. Taqvi, and J. Lu, "IoT security: ZWave and thread," in *Proc. IEEE Green Technol. Conf. (GreenTech)*, Apr. 2018, pp. 176–182.

[123] C. Kumar, Y. Arya, and G. Agarwal, "A review report on WiMAX vulnerabilities, security threats and their solutions," in *Proc. 2nd Int. Conf. Inventive Commun. Comput. Technol. (ICICCT)*, Apr. 2018, pp. 1963–1967.

[124] S. N. Ghormare, S. Sorte, and S. S. Dorle, "Detection and prevention of wormhole attack in WiMAX based mobile adhoc network," in *Proc. 2nd Int. Conf. Electron., Commun. Aerosp. Technol. (ICECA)*, Mar. 2018, pp. 1097–1101.

[125] M. Nasreldin, H. Aslan, M. El-Hennawy, and A. El-Hennawy, "WiMax security," in *Proc. 22nd Int. Conf. Adv. Inf. Netw. Appl.*, Mar. 2008, pp. 1335–1340.

[126] V. K. Jatav and V. Singh, "Mobile WiMAX network security threats and solutions: A survey," in *Proc. Int. Conf. Comput. Commun. Technol. (ICCCT)*, Sep. 2014, pp. 135–140.

[127] B. Bhargava, Y. Zhang, N. Idika, L. Lilien, and M. Azarmi, "Collaborative attacks in WiMAX networks," *Secur. Commun. Netw.*, vol. 2, no. 5, pp. 373–391, Sep. 2009.

[128] M. A. Hasnat, S. T. A. Rumee, M. A. Razzaque, and M. Mamun-Or-Rashid, "Security study of 5G heterogeneous network: Current solutions, limitations & future direction," in *Proc. Int. Conf. Electr., Comput. Commun. Eng. (ECCE)*, Feb. 2019, pp. 1–4.

[129] N. Wang, L. Jiao, P. Wang, M. Dabaghchian, and K. Zeng, "Efficient identity spoofing attack detection for IoT in mm-wave and massive MIMO 5G communication," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2018, pp. 1–6.

[130] I. Ahmad, T. Kumar, M. Liyanage, J. Okwuibe, M. Ylianttila, and A. Gurtov, "Overview of 5G security challenges and solutions," *IEEE Commun. Standards Mag.*, vol. 2, no. 1, pp. 36–43, Mar. 2018.

[131] S. R. Hussain, M. Echeverria, O. Chowdhury, N. Li, and E. Bertino, "Privacy attacks to the 4G and 5G cellular paging protocols using side channel information," in *Proc. Netw. Distrib. Syst. Secur. Symp.*, 2019, pp. 1–15.

[132] *Amendment 4: Enhancements for Very High Throughput for Operation in Bands below 6 GHz*, IEEE Standard 802.11ac, 2013.

[133] E. Pietrosemoli, "Long distance Wifi trial," in *Proc. Int. Summit Community Wireless Netw.*, 2007, pp. pp. 1–21. [Online]. Available: https://lafibre.info/testdebit/wifi/200705_long_distance_wiFi_trial.pdf

[134] Wi-Fi-Alliance. (2018). *Wi-Fi Specifications*. Accessed: Jan. 19, 2020. [Online]. Available: https://www.wi-fi.org/discover-wi-fi/specifications

[135] B. Jerman-Blažič, W. Schneider, and T. Klobucar, *Security and Privacy in Advanced Networking Technologies* (NATO Science Series: Computer and Systems Sciences). Amsterdam, The Netherlands: IOS Press, 2004.

[136] E. Tews and M. Beck, "Practical attacks against WEP and WPA," in *Proc. 2nd ACM Conf. Wireless Netw. Secur. (WiSec)*, 2009, pp. 79–86.

[137] N. AlFardan, D. J. Bernstein, K. G. Paterson, B. Poettering, and J. C. N. Schuldt, "On the security of RC4 in TLS," in *Proc. 22nd USENIX Secur. Symp. (USENIX)*, 2013, pp. 305–320.

[138] A. Stubblefield, J. Ioannidis, and A. D. Rubin, "Using the Fluhrer, Mantin, and Shamir attack to break WEP," in *Proc. Netw. Distrib. Syst. Secur. Symp.*, 2002, pp. 1–13.

[139] S. Fluhrer, I. Mantin, and A. Shamir, "Weaknesses in the key scheduling algorithm of RC4," in *Proc. 8th Annu. Int. Workshop Sel. Areas Cryptogr.* New York, NY, USA: Springer-Verlag, 2001, pp. 1–24.

[140] *Amendment 6: Medium Access Control (MAC) Security Enhancements*, IEEE Standard 802.11i, 2004.

[141] *Wireless LAN Medium Access Control and Physical Layer Specification*, IEEE Standard 802.11, 2016.

[142] *Amendment 3: Enhancement for Very High Troughput 60GHz Band*, IEEE Standard 802.11ad, 2012.

[143] NIST. (2001). *Advanced Encryption Standard (AES)*. Accessed: Jan. 19, 2020. [Online]. Available: http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf

[144] W. Diffie and M. Hellman, "New directions in cryptography," *IEEE Trans. Inf. Theory*, vol. 22, no. 6, pp. 644–654, Nov. 1976.

[145] Wi-Fi-Alliance. (2018). *WPA3 Specification Version 1.0*. Accessed: Jan. 19, 2020. [Online]. Available: https://www.wi-fi.org

[146] D. Harkins, "Simultaneous authentication of equals: A secure, password-based key exchange for mesh networks," in *Proc. 2nd Int. Conf. Sensor Technol. Appl. (Sensorcomm)*, 2008, pp. 839–844.

[147] *Bluetooth Core Specification Version 5.0*, Bluetooth-SIG, Kirkland, WA, USA, 2018.

[148] J. Massey, G. Khachatrian, and M. Kuregian, "Secure and fast encryption routine+," in *Proc. 1st Adv. Encryption Standard (AES) Candidate Conf. Rep.*, vol. 104, no. 1. Gaithersburg, MD, USA: NIST, 1998. [Online]. Available: https://www.ieee-security.org/Cipher/ConfReports/conf-rep-aes.html

[149] *Bluetooth Core Specification Version 5.0*, Bluetooth-SIG, Bluetooth Spec document, 2018.

[150] R. Milman. (2011). *Internet of Business: Bluetooth and ZigBee to Dominate Wireless IoT Connectivity*. Accessed: Jan. 19, 2020. [Online]. Available: https://internetofbusiness.com/iotdriving-wireless-connectivity/

[151] Texas-Instruments, "What's New ZigBee 3.0," White Paper SWRA615, 2018.

[152] ZigBee Alliance. (2018). *ZigBee Specification: ZigBee and ZigBee Pro*. Accessed: Jan. 19, 2020. [Online]. Available: http://zigbee.org

[153] D. Gislason, *Zigbee Wireless Networking*. Amsterdam, The Netherlands: Elsevier, 2008.

[154] M. U. B. Aftab, *Building Bluetooth Low Energy Systems*. Birmingham, U.K.: Packt, 2017.

[155] F. Eady, *Hands-On ZigBee: Implementing 802.15.4 with Microcontrollers*. Amsterdam, The Netherlands: Elsevier, 2010.

[156] N. Vidgren, K. Haataja, J. L. Patino-Andres, J. J. Ramirez-Sanchis, and P. Toivanen, "Security threats in ZigBee-enabled systems: Vulnerability evaluation, practical experiments, countermeasures, and lessons learned," in *Proc. 46th Hawaii Int. Conf. Syst. Sci.*, Jan. 2013, pp. 5132–5138.

[157] *ZigBee Specification*, ZigBee-Alliance, document 053474r17, 2008.
[158] S. Evdokimov, B. Fabian, O. Günther, L. Ivantysynova, and H. Ziekow, "RFID and the Internet of Things: Technology, applications, and security challenges," in *Foundations and Trends in Technology, Information and Operations Management*. Boston, MA, USA: Now, 2011.
[159] N. Bartneck, V. Klaas, and H. Schönherr, *Optimizing Processes With RFID and Auto ID*. Hoboken, NJ, USA: Wiley, 2009.
[160] A. Juels, "Minimalist cryptography for low-cost RFID tags," in *Proc. 4th Int. Conf. Secur. Commun. Netw.*, 2004, pp. 149–164.
[161] M. Ohkubo, "Efficient hash-chain based RFID privacy protection scheme," in *Proc. Int. Conf. Ubiquitous Comput.-Ubicomp, Workshop Privacy, Current Status Future Directions*, 2004, pp. 1–4.
[162] D. Henrici and P. Müller, "Hash-based enhancement of location privacy for radio-frequency identification devices using varying identifiers," in *Proc. 2nd IEEE Annu. Conf. Pervas. Comput. Commun. Workshops*, 2004, pp. 149–153.
[163] D. Molnar and D. Wagner, "Privacy and security in library RFID: Issues, practices, and architectures," in *Proc. 11th ACM Conf. Comput. Commun. Secur. (CCS)*, 2004, pp. 210–219.
[164] S. A. Weis, S. E. Sarma, R. L. Rivest, and D. W. Engels, "Security and privacy aspects of low-cost radio frequency identification systems," in *Proc. 1st Int. Conf. Secur. Pervas. Comput.*, 2003, pp. 201–212.
[165] T. Dimitriou, "A lightweight RFID protocol to protect against traceability and cloning attacks," in *Proc. 1st Int. Conf. Secur. Privacy Emerg. Areas Commun. Netw. (SECURECOMM)*, 2005, pp. 59–66.
[166] M. Feldhofer, J. Wolkerstorfer, and V. Rijmen, "AES implementation on a grain of sand," *IEE Proc. Inf. Secur.*, vol. 152, pp. 13–20, Oct. 2005.
[167] S. Piramuthu, "HB and related lightweight authentication protocols for secure RFID tag-reader authentication," Decis. Inf. Sci., Univ. Florida, Gainesville, FL, USA, Tech. Rep., 2006.
[168] M. Aigner and M. Feldhofer, "Secure symmetric authentication for RFID tags," Graz Univ. Technol., Graz, Austria, Tech. Rep., 2005.
[169] M. Girault, L. Juniot, and M. Robshaw, "The feasibility of on-the-tag public key cryptography," in *Proc. Conf. RFID Secur.*, 2007, p. 68.
[170] J. Wolkerstorfer, "Is elliptic-curve cryptography suitable to secure RFID tags?" in *Proc. Workshop RFID Light-Weight Cryptogr.*, 2005, pp. 1–816.
[171] A. J. Menezes, P. C. van Oorschot, and S. A. Vanstone, *Handbook of Applied Cryptography* (Discrete Mathematics and Its Applications). Boca Raton, FL, USA: CRC Press, 1996.
[172] S. Brands and D. Chaum, "Distance-bounding protocols," in *Proc. Workshop Theory Appl. Cryptograph. Techn.*, 1993, pp. 344–359.
[173] K. P. Fishkin and S. Roy, "Enhancing RFID privacy via antenna energy analysis," in *Proc. RFID Privacy Workshop*, 2003, pp. 1–3.
[174] *Information Assurance: Instruction-8500.1*, Assistant Secretary Defense Netw. Inf. Integr., DoD, Jones & Bartlett Learn., Arlington, VA, USA, 2002.
[175] V. Igure and R. Williams, "Taxonomies of attacks and vulnerabilities in computer systems," *IEEE Commun. Surveys Tuts.*, vol. 10, no. 1, pp. 6–19, 1st Quart., 2008.
[176] S. T. Kent, "Encryption-based protection for interactive user/computer communication," in *Proc. 5th Symp. Data Commun. (SIGCOMM)*, vol. 5, 1977, pp. 7–13.
[177] R. McNamara, "Networks—Where does the real threat lie?" *Inf. Secur. Tech. Rep.*, vol. 3, no. 4, pp. 65–74, 1998.
[178] J. McHugh, "The 1998 Lincoln laboratory IDS evaluation," in *Recent Advances in Intrusion Detection*. Berlin, Germany: Springer, 2000, pp. 145–161.
[179] W. Stallings, *Cryptography and Network Security: Principles and Practice* (Instructor's Manual). Upper Saddle River, NJ, USA: Prentice-Hall, 1998.
[180] G. S. Bopche and B. M. Mehtre, "Exploiting domination in attack graph for enterprise network hardening," in *Proc. Int. Symp. Secur. Comput. Commun.*, vol. 536, 2015, pp. 342–353.
[181] K. Lounis, "Stochastic-based semantics of attack-defense trees for security assessment," in *Proc. 9th Int. Workshop Practical Appl. Stochastic Modeling (PASM)*, vol. 337. Amsterdam, The Netherlands: Elsevier, 2018, pp. 135–154.
[182] R. Jhawar, K. Lounis, and S. Mauw, "A stochastic framework for quantitative analysis of attack-defense trees," in *Proc. 12th Int. Workshop Secur. Trust Manage.*, vol. 9871. Cham, Switzerland: Springer, 2016, pp. 138–153.
[183] R. Jhawar, K. Lounis, S. Mauw, and Y. Ramírez-Cruz, "Semi-automatically augmenting attack trees using an annotated attack tree library," in *Proc. 14th Int. Workshop Secur. Trust Manage.*, vol. 11091. Cham, Switzerland: Springer, 2018, pp. 85–101.
[184] O. Gadyatskaya, R. Jhawar, P. Kordy, K. Lounis, S. Mauw, and R. Trujillo-Rasúa, "Attack trees for practical security assessment: Ranking of attack scenarios with ADTool 2.0," in *Proc. 13th Int. Conf. Quant. Eval. Syst.*, vol. 9826. Cham, Switzerland: Springer, 2016, pp. 159–162.
[185] B. Kordy, S. Mauw, S. Radomirović, and P. Schweitzer, "Foundations of attack–defense trees," in *Formal Aspects of Security and Trust*. Cham, Switzerland: Springer, 2011, pp. 80–95.
[186] M. Kershaw. (2008). *Kismet: A Wireless Network and Device Detector, Sniffer, Wardriving Tool, and WIDS (Wireless Intrusion Detection) Framework*. Accessed: Jan. 19, 2020. [Online]. Available: https://www.kismetwireless.net/
[187] K. Lounis, A. Babakhouya, and N. Taboudjemat, "Setting up a wireless intrusion detection solution based on kismet," CERIST (Centre Recherche sur l'Inf. Sci. Techn.), Algiers, Algeria, Tech. Rep. CERIST-DTISI/RT-11-000000023-dz, 2011.
[188] N. Borisov, I. Goldberg, and D. Wagner, "Intercepting mobile communications: The insecurity of 802.11," in *Proc. 7th Annu. Int. Conf. Mobile Comput. Netw. (MobiCom)*, 2001, pp. 180–189.
[189] K. G. Paterson, B. Poettering, and J. C. N. Schuldt, "Plaintext recovery attacks against WPA/TKIP," in *Fast Software Encryption*. Cham, Switzerland: Springer, 2015, pp. 325–349.
[190] E. Dawson and L. Nielsen, "Automated cryptanalysis of XOR plaintext strings," *Cryptologia*, vol. 20, no. 2, pp. 165–181, Apr. 1996.
[191] KoreK. (2004). *Next Generation of WEP Attacks?* Accessed: Jan. 19, 2020. [Online]. Available: http://www.netstumbler.org/news/next-generation-of-wep-attacks-t12277.html
[192] A. Wool, "A note on the fragility of the Michael message integrity code," *IEEE Trans. Wireless Commun.*, vol. 3, no. 5, pp. 1459–1462, Sep. 2004.
[193] N. Ferguson. (2002). *Michael: An Improved MIC for 802.11 WEP*. Accessed: Jan. 19, 2020. [Online]. Available: http://grouper.ieee.org/groups/802/11/Documents/DocumentHolder/2-020.zip
[194] J. Bellardo and S. Savage, "802.11 denial-of-service attacks: Real vulnerabilities and practical solutions," in *Proc. 12th Conf. USENIX Secur. Symp.*, vol. 12. Berkeley, CA, USA: USENIX Assoc., 2003, pp. 15–27.
[195] K. Lounis and M. Zulkernine, "Bad-token: Denial of service attacks on WPA3," in *Proc. ACM 12th Int. Conf. Secur. Inf. Netw.*, vol. 15, 2019, pp. 1–8.
[196] K. Lounis and M. Zulkernine, "Connection deprivation attacks on WPA3," in *Proc. 14th Int. Conf. Risks Secur. Internet Syst.*, vol. 12026, 2019, pp. 164–176.
[197] G. Lin and N. Guevara, "On link layer denial of service in data wireless LANs," *Wireless Commun. Mobile Comput.*, vol. 5, no. 3, pp. 273–284, 2005.
[198] W. A. Arbaugh, N. Shankar, Y. C. J. Wan, and K. Zhang, "Your 80211 wireless network has no clothes," *IEEE Wireless Commun.*, vol. 9, no. 6, pp. 44–51, Dec. 2002.
[199] A. Stubblefield, J. Ioannidis, and A. D. Rubin, "A key recovery attack on the 802.11b wired equivalent privacy protocol (WEP)," in *Proc. ACM Trans. Inf. Syst. Secur.*, vol. 7, no. 2, pp. 319–332, 2004.
[200] E. Tews, R.-P. Weinmann, and A. Pyshkin, "Breaking 104 Bit WEP in less than 60 seconds," in *Information Security Applications*. Berlin, Germany: Springer, 2007, pp. 188–202.
[201] WPACracker. (2009). *WPACracker.net*. Accessed: Jan. 19, 2020. [Online]. Available: http://www.wpacrack.net/index.html
[202] M. Vanhoef and E. Ronen. (2019). *Dragonblood: A Security Analysis of WPA3's SAE Handshake*. Accessed: Apr. 29, 2019. [Online]. Available: https://papers.mathyvanhoef.com/dragonblood.pdf
[203] M. Vanhoef and F. Piessens, "Key reinstallation attacks: Forcing nonce reuse in WPA2," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, Oct. 2017, pp. 1313–1328.
[204] Pixiewps. (2014). *An Offline Wi-Fi Protected Setup Brute-Force Utility*. Accessed: Jan. 19, 2020. [Online]. Available: https://github.com/wiire-a/pixiewps
[205] Reaver. (2011). *Brute Force Attack Utility Against Wifi Protected Setup Registrar PINs*. Accessed: Jan. 19, 2020. [Online]. Available: https://github.com/t6x/reaver-wps-fork-t6x
[206] K. Hypponen and K. M. J. Haataja, "'Nino' man-in-the-middle attack on Bluetooth secure simple pairing," in *Proc. 3rd IEEE/IFIP Int. Conf. Central Asia Internet*, Sep. 2007, pp. 1–5.
[207] K. M. J. Haataja and K. Hypponen, "Man-In-The-Middle attacks on Bluetooth: A comparative analysis, a novel attack, and countermeasures," in *Proc. 3rd Int. Symp. Commun., Control Signal Process.*, Mar. 2008, pp. 1096–1102.

[208] K. Haataja and P. Toivanen, "Practical Man-in-the-Middle attacks against Bluetooth secure simple pairing," in *Proc. 4th Int. Conf. Wireless Commun., Netw. Mobile Comput.*, Oct. 2008, pp. 1–5.

[209] K. Haataja and P. Toivanen, "Two practical man-in-the-middle attacks on Bluetooth secure simple pairing and countermeasures," *IEEE Trans. Wireless Commun.*, vol. 9, no. 1, pp. 384–392, Jan. 2010.

[210] J. Barnickel, J. Wang, and U. Meyer, "Implementing an attack on Bluetooth 2.1+ secure simple pairing in passkey entry mode," in *Proc. IEEE 11th Int. Conf. Trust, Secur. Privacy Comput. Commun.*, Jun. 2012, pp. 17–24.

[211] D.-Z. Sun, Y. Mu, and W. Susilo, "Man-in-the-middle attacks on secure simple pairing in Bluetooth standard V5.0 and its countermeasure," *Pers. Ubiquitous Comput.*, vol. 22, no. 1, pp. 55–67, Feb. 2018.

[212] A. Levi, E. Çetintaş, M. Aydos, C. K. Koç, and M. U. Çağlayan, "Relay attacks on Bluetooth authentication and solutions," in *Proc. Int. Symp. Comput. Inf. Sci.* 2004, pp. 278–288.

[213] J. Wang, K. Lounis, and M. Zulkernine, "CSKES: A context-based secure keyless entry system," in *Proc. IEEE 43rd Annu. Comput. Softw. Appl. Conf. (COMPSAC)*, Jul. 2019, pp. 817–822.

[214] A. Laurie, M. Holtmann, and M. Herfurt. (2007). *Hacking Bluetooth Enabled Mobile Phones and Beyond*. Accessed: Jan. 19, 2020. [Online]. Available: http://www.blackhat.com/html/bh-europe-05/bh-eu-05-speakers.html

[215] M. Herfurt. (2005). *Introducing the Car Whisperer at What The Hack*. Accessed: Jan. 19, 2020. [Online]. Available: https://trifinite.org/trifinite_stuff_carwhisperer.html

[216] J. Hering. (2004). *Bluetooth Cracking Gun: BlueSniper*. Accessed: Jan. 19, 2020. [Online]. Available: https://www.defcon.org/html/links/dc-archives/dc-12-archive.html

[217] K. Lounis and M. Zulkernine, "Bluetooth low energy makes 'just works' not work," in *Proc. Cyber Secur. Netw. Conf.*, Quito, Ecuador, 2019, pp. 1–9.

[218] C. Miller and M. Herfurt. (2004). *Blueprinting*. Accessed: Jan. 19, 2020. [Online]. Available: https://trifinite.org/trifinite_stuff_blueprinting.html

[219] K. Lounis and M. Zulkernine, "Connection dumping vulnerability affecting Bluetooth availability," in *Proc. 13th Int. Conf. Risks Secur. Internet Syst.*, 2018, pp. 188–204.

[220] M. Jakobsson and S. Wetzel, "Security weaknesses in Bluetooth," in *Proc. Topics Cryptol., Cryptogr. Track RSA Conf.*, 2001, pp. 176–191.

[221] D. Spill and A. Bittau, "BlueSniff: Eve meets alice and Bluetooth," in *Proc. 1st USENIX Workshop Offensive Technol.*, 2007, pp. 1–10.

[222] P. Kocher, J. Jaffe, and B. Jun, "Differential power analysis," in *Proc. 19th Annu. Int. Cryptol. Conf. Adv. Cryptol. (CRYPTO)*, 1999, pp. 388–397.

[223] A. Adomnicai, J. J. A. Fournier, and L. Masson, "Hardware security threats against Bluetooth mesh networks," in *Proc. IEEE Conf. Commun. Netw. Secur. (CNS)*, May 2018, pp. 1–9.

[224] F.-X. Standaert, E. Peeters, and J.-J. Quisquater, "On the masking countermeasure and higher-order power analysis attacks," in *Proc. Int. Conf. Inf. Technol., Coding Comput. (ITCC)*, vol. 2, 2005, pp. 562–567.

[225] J. Agat, "Transforming out timing leaks," in *Proc. 27th ACM SIGPLAN-SIGACT Symp. Princ. Program. Lang. (POPL)*, 2000, pp. 40–53.

[226] J. Agat, "Transforming out timing leaks," in *Proc. 27th ACM SIGPLAN-SIGACT Symp. Princ. Program. Lang. (POPL)*. Berlin, Germany: Springer, 2000, pp. 156–168.

[227] B. Köpf and M. Dürmuth, "A provably secure and efficient countermeasure against timing attacks," in *Proc. 22nd IEEE Comput. Secur. Found. Symp.*, Jul. 2009, pp. 324–335.

[228] D. Zhang, A. Askarov, and A. C. Myers, "Predictive mitigation of timing channels in interactive systems," in *Proc. 18th ACM Conf. Comput. Commun. Secur. (CCS)*, 2011, pp. 563–574.

[229] M. Ryan. (2012). *I Am Jack's Heart Monitor*. Accessed: Apr. 29, 2019. [Online]. Available: http://lacklustre.net/Bluetooth/hacking_btle-i_am_jacks_heart_monitor-mikeryan-toorcon

[230] A. Rose and B. Ramsey, "Picking Bluetooth low energy locks from a quarter mile away," Paris Bally's Conv. Centers, Las Vegas, NV, USA, Tech. Rep. DEF CON 24, 2016, Accessed: Apr. 29, 2019.

[231] O. Arias, J. Wurm, K. Hoang, and Y. Jin, "Privacy and security in Internet of Things and wearable devices," *IEEE Trans. Multi-Scale Comput. Syst.*, vol. 1, no. 2, pp. 99–109, Apr. 2015.

[232] D. Cauquil, "BtleJuice: The Bluetooth smart man in the middle framework," GreHack, Grenoble, France, Tech. Rep., 2016.

[233] S. Jasek, "Blue picking: Hacking Bluetooth smart locks," HackInTheBox, Amsterdam, The Netherlands, Tech. Rep., 2017.

[234] V. Tan, "Hacking BLE bicycle locks for fun and a small profit," Paris Bally's Conv. Centers, Las Vegas, NV, USA, Tech. Rep. DEF CON 26, 2018.

[235] B. Cyr, W. Horn, D. Miao, and M. Specter, "Security analysis of wearable fitness devices (Fitbit)," Massachusetts Inst. Technol., Cambridge, MA, USA, Tech. Rep., 2014.

[236] Q. Zhang and Z. Liang, "Security analysis of Bluetooth low energy based smart wristbands," in *Proc. 2nd Int. Conf. Frontiers Sensors Technol. (ICFST)*, Apr. 2017.

[237] C. Mulliner. (2013). *BlueSpam*. Accessed: Jan. 19, 2020. [Online]. Available: http://www.mulliner.org/palm/bluespam.php

[238] K. Haataja, "Bluetooth network vulnerability to disclosure, integrity and denial-of-service attacks," in *Proc. Annu. Finnish Data Process. Week Univ. Petrozavodsk Adv. Methods Mod. Inf. Technol.*, 2006, pp. 63–103.

[239] T. Oconnor and D. Reeves, "Bluetooth network-based misuse detection," in *Proc. Annu. Comput. Secur. Appl. Conf. (ACSAC)*, Dec. 2008, pp. 377–391.

[240] P. Satam, S. Satam, and S. Hariri, "Bluetooth intrusion detection system (BIDS)," in *Proc. IEEE/ACS 15th Int. Conf. Comput. Syst. Appl. (AICCSA)*, Oct. 2018, pp. 1–7.

[241] K. M. J. Haataja, "New efficient intrusion detection and prevention system for Bluetooth networks," in *Proc. 1st Int. ICST Conf. Mobile Wireless Middleware, Oper. Syst. Appl.*, in Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering, vol. 16, 2007, pp. 1–6.

[242] Y. Shaked and A. Wool, "Cracking the Bluetooth PIN," in *Proc. 3rd Int. Conf. Mobile Syst., Appl., Services (MobiSys)*, 2005, pp. 39–50.

[243] M. Herfurt. (2004). *BluBug*. Accessed: Jan. 19, 2020. [Online]. Available: https://trifinite.org/trifinite_stuff_bluebug.html

[244] A. Laurie. (2013). *HeloMoto Bluetooth Device Planter*. [Online]. Available: https://trifinite.org/trifinite_stuff_helomoto.html

[245] Armis-Company. (2017). *BlueBorne Cyber Threat Impacts Amazon Echo and Google Home*. [Online]. Available: https://www.armis.com/blueborne/

[246] T. Goodspeed, S. Bratus, R. Melgares, R. Shapiro, and R. Speers, "Packets in packets: Orson Welles' in-band signaling attacks for modern radios," in *Proc. 5th USENIX Conf. Offensive Technol.*, 2011, pp. 1–8.

[247] A. Biswas, A. Alkhalid, T. Kunz, and C.-H. Lung, "A lightweight defence against the packet in packet attack in ZigBee networks," in *Proc. IFIP Wireless Days*, Nov. 2012, pp. 1–3.

[248] J. Wright. (2011). *KillerBee: Practical ZigBee Exploitation Framework or Wireless Hacking and the Kinetic World*. Accessed: Jan. 19, 2020. [Online]. Available: http://www.willhackforsushi.com/presentations/toorcon11-wright.pdf

[249] J. Cache, J. Wright, and V. Liu, *Hacking Exposed Wireless: Wireless Security Secrets & Solutions*. New York, NY, USA: McGraw-Hill, 2010, pp. 1–513.

[250] C. Benzaid, K. Lounis, A. Al-Nemrat, N. Badache, and M. Alazab, "Fast authentication in wireless sensor networks," *Future Gener. Comput. Syst.*, vol. 55, pp. 362–375, Feb. 2016.

[251] J. Durech and M. Franekova, "Security attacks to ZigBee technology and their practical realization," in *Proc. IEEE 12th Int. Symp. Appl. Mach. Intell. Informat. (SAMI)*, Jan. 2014, pp. 345–349.

[252] J. Li and Q. Yang. (2015). *I'm a Newbie Yet I Can Hack ZigBee*. Accessed: Jan. 19, 2020. [Online]. Available: https://www.defcon.org/html/defcon-23/dc-23-speakers.html

[253] T. Goodspeed. (2009). *Extracting Keys From Second Generation ZigBee Chips, in Black Hat Briefing*. Accessed: Jan. 19, 2020. [Online]. Available: http://www.blackhat.com/presentations/bh-usa-09/GOODSPEED/BHUSA09-Goodspeed-ZigbeeChips-PAPER.pdf

[254] T. Zillner, *ZigBee Exploited: The Good, the Bad and the Ugly*. Wien, Austria: Congnosec Gmbh, 2015, pp. 1–8. [Online]. Available: https://media.kasperskycontenthub.com/wp-content/uploads/sites/43/2015/11/20081735/us-15-Zillner-ZigBee-Exploited-The-Good-The-Bad-And-The-Ugly-wp.pdf

[255] L. N. Whitehurst, T. R. Andel, and J. T. McDonald, "Exploring security in ZigBee networks," in *Proc. 9th Annu. Cyber Inf. Secur. Res. Conf. (CISR)*, 2014, pp. 25–28.

[256] M. Kershaw. (2013). *KisBee: A Small, Battery Powered, Open Source Hardware Device for Capturing 802.15.4*. Accessed: Jan. 19, 2020. [Online]. Available: https://www.kismetwireless.net/kisbee/

[257] J. R. Douceur, "The Sybil attack," in *Peer-to-Peer System*. Berlin, Germany: Springer, 2002, pp. 251–260.

[258] P. Papadimitratos and Z. J. Haas, "Secure link state routing for mobile ad hoc networks," in *Proc. Symp. Appl. Internet Workshops*, IEEE Computer Society, 2003, pp. 379–383.
[259] Y. Zhang, J. Zheng, and M. Ma, *Handbook of Research on Wireless Security*. Hershey, PA, USA: IGI Global Research Collection, Information Science Reference, 2008.
[260] M. Pirretti, S. Zhu, N. Vijaykrishnan, P. McDaniel, M. Kandemir, and R. Brooks, "The sleep deprivation attack in sensor networks: Analysis and methods of defense," *Int. J. Distrib. Sensor Netw.*, vol. 2, no. 3, pp. 267–287, Jul. 2006.
[261] R. Sokullu, O. Dagdeviren, and I. Korkmaz, "On the IEEE 802.15.4 MAC layer attacks: GTS attack," in *Proc. 2nd Int. Conf. Sensor Technol. Appl. (Sensorcomm)*, 2008, pp. 673–678.
[262] *Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks*, IEEE Standard 802.15.4, 2003.
[263] Y. W. Law, P. Hartel, J. den Hartog, and P. Havinga, "Link-layer jamming attacks on S-MAC," in *Proc. 2nd Eur. Workshop Wireless Sensor Netw.*, 2005, pp. 217–225.
[264] *ZigBee Specification*, ZigBee-Alliance, document 053474r20, 2009.
[265] Y. Desmedt, C. Goutier, and S. Bengio, "Special uses and abuses of the Fiat-Shamir passport protocol," in in *Proc. Conf. Theory Appl. Cryptograph. Techn.*, 1987, pp. 21–39.
[266] Z. Kfir and A. Wool, "Picking virtual pockets using relay attacks on contactless smartcard," in *Proc. 1st Int. Conf. Secur. Privacy Emerg. Areas Commun. Netw. (SECURECOMM)*, 2005, pp. 47–58.
[267] D. Carluccio, K. Lemke, and C. Paar, "Electromagnetic side channel analysis of a contactless smart card: First results," in *Proc. ECrypt Workshop RFID Lightweight Crypto*, 2005.
[268] P. Kitsos, *Security RFID Sensor Network* (Wireless Networks and Mobile Communications). Boca Raton, FL, USA: CRC Press, 2016.
[269] A. Francillon, B. Danev, and S. Capkun, "Relay attacks on passive keyless entry and start systems in modern cars," in *Proc. IACR Cryptol. ePrint Arch.*, vol. 2010, 2010, p. 332.
[270] Y. Desmedt, "Major security problems with unforgeable(ferge)-Fiat-Shamir proofs of identity and how to overcome them," in *Proc. Worldwide Congr. Comput. Commun. Secur. Protection*, 1988, pp. 15–17.
[271] C. J. F. Cremers, K. B. Rasmussen, and S. Capkun, "Distance hijacking attacks on distance bounding protocols," in *Proc. IACR Cryptol. ePrint Arch.*, 2011.
[272] Vaultskin. (2018). *RFID Blocking Card, Anti-Skimming Protection*. [Online]. Available: https://www.vaultskin.com/vaultcard.html
[273] A. Juels, "Strengthening EPC tags against cloning," in *Proc. 4th ACM Workshop Wireless Secur. (WiSe)*, 2005, pp. 67–76.
[274] L. Grunwald, "Security by politics–why it will never work," Tech. Rep., 2007.
[275] P. Golle, M. Jakobsson, A. Juels, and P. Syverson, "Universal re-encryption for Mixnets," in *Topics in Cryptology—CT-RSA*. Cham, Switzerland: Springer, 2004, pp. 163–178.
[276] A. Juels and R. Pappu, "Squealing euros: Privacy protection in RFID-enabled banknotes," in *Proc. 7th Int. Conf Financial Cryptogr.*, 2003, pp. 103–121.
[277] Y. Oren and A. Shamir, "Power analysis of RFID tags," in *Advance Cryptology*. 2006.
[278] M. Burmester and B. D. Medeiros, "RFID academic convocation, the RFID journal," in *Proc. 8th Int. Conf. Wireless Commun., Netw. Mobile Comput.*, 2017, pp. 1–10.
[279] S. Bono, M. Green, A. Stubblefield, A. Juels, A. Rubin, and M. Szydlo, "Security analysis of a cryptographically-enabled RFID device," in *Proc. 14th USENIX Secur. Symp.*, 2005, pp. 1–16.

KARIM LOUNIS received the master's degree in networks and distributed systems from the University of Science and Technology Houari Boumedienne (USTHB), Algeria, in 2013, and the second master's degree in security of information systems from the University of East-Paris Creteil (UPEC), France, in 2014. He is currently pursuing the Ph.D. degree in cybersecurity with the Queen's Reliable Software Technology Group, School of Computing, Queen's University, Kingston, Canada. He worked as a Graduate Research Assistant in information security with the system security group (CISPA: Center for IT-Security, Privacy and Accountability), Saarland University, Germany, from 2014 to 2015, and then with the security and trust of software systems group (SnT: Interdisciplinary Centre for Security, Reliability and Trust), University of Luxembourg, Luxembourg, from 2015 to 2017. His research interests include information security, networks security, and the IoT security.

MOHAMMAD ZULKERNINE (Senior Member, IEEE) received the Ph.D. degree from the University of Waterloo, Canada. He joined Queen's University, in 2003, and spent his sabbatical as a Visiting Professor at the University of Trento, Italy, and as a Researcher at Irdeto, Canada. He is currently a Professor and a Canada Research Chair with the School of Computing, Queen's University, Canada, where he leads the Queen's Reliable Software Technology (QRST) research group. His current research focuses on building reliable and secures software systems and he has extensive publications in this area. He has led major research projects supported by a number of provincial and federal agencies and industry partners. He is a Senior Member of ACM, and a licensed Professional Engineer in the Province of Ontario, Canada. He has been holding the leadership positions, such as the general chair, the organizing chair, and the program chair of many major research conferences and workshops.