IoT

MDPI

Article

Investigating Radio Frequency Vulnerabilities in the Internet of Things (IoT)

Eirini Anthi *, Lowri Williams, Vasilis Ieropoulos and Theodoros Spyridopoulos

School of Computer Science & Informatics, Cardiff University, Cardiff CF24 4AG, UK; williamsl10@cardiff.ac.uk (L.W.); ieropoulosv@cardiff.ac.uk (V.I.); spyridopoulost@cardiff.ac.uk (T.S.)
* Correspondence: anthies@cardiff.ac.uk

Abstract: With the increase in the adoption of Internet of Things (IoT) devices, the security threat they face has become more pervasive. Recent research has demonstrated that most IoT devices are insecure and vulnerable to a range of cyber attacks. The impact of such attacks can vary significantly, from affecting the service of the device itself to putting their owners and their personal information at risk. As a response to improving their security, the focus has been on attacks, specifically on the network layer. However, the importance and impact of other vulnerabilities, such as low-level Radio Frequency (RF) attacks, have been neglected. Such attacks are challenging to detect, and they can be deployed using non-expensive equipment and can cause significant damage. This paper explores security vulnerabilities that target RF communications on popular commercial IoT devices such as Wi-Fi, Zigbee, and 433 Mz. Using software-defined radio, a range of attacks were deployed against the devices, including jamming, replay attacks, packet manipulation, protocol reverse engineering, and harmonic frequency attacks. The results demonstrated that all devices used were susceptible to jamming attacks, and in some cases, they were rendered inoperable and required a hard reset to function correctly again. This finding highlights the lack of protection against both intentional and unintentional jamming. In addition, all devices demonstrated that they were susceptible to replay attacks, which highlights the need for more hardened security measures. Finally, this paper discusses proposals for defence mechanisms for enhancing the security of IoT devices against the aforementioned attacks.

Keywords: internet of things (IoT); smart homes; networking; radio frequency; software-defined radio

Citation: Anthi, E.; Williams, L.; Ieropoulos, V.; Spyridopoulos, T. Investigating Radio Frequency Vulnerabilities in the Internet of Things (IoT). *IoT* **2024**, *5*, 356–380. https://doi.org/10.3390/iot5020018

Academic Editor: Amiya Nayak

Received: 24 April 2024
Revised: 28 May 2024
Accepted: 5 June 2024
Published: 6 June 2024

1. Introduction

The Internet of Things (IoT) refers to the billions of physical devices around the world that are connected to the Internet. Such devices may be connected in a domestic environment, such as a smart home, as well as within Critical National Infrastructures (CNI), which facilitate larger networks, such as smart cities. When connected and combined with sensors, such devices are accompanied by a level of digital intelligence that enables them to communicate data in real time, making the fabric of the world around us smarter and more responsive. However, although the adoption of IoT continues to increase, several studies have reported that such devices introduce tremendous security flaws, and a vast majority are vulnerable to a range of cyber attacks [1].

Current research surrounding the security of IoT focuses on investigating network-based attacks, such as Denial of Service (DoS), Man-In-The-Middle (MITM), malware, and sniffing [2–4]. However, fewer studies have focused on the vulnerability of IoT devices towards Radio Frequency (RF) attacks. Such attacks may be particularly challenging to detect or trace, as adversaries can deploy them and exploit vulnerabilities from a long-range distance [5]. RF-based attacks can be launched using open-source software and affordable hardware, making it possible for anyone to gain the necessary equipment to deploy them [6]. What makes RF-based attacks particularly dangerous is that an adversary

can deploy them from a long range of distances of up to 100 m, allowing malicious actors to exploit vulnerabilities while being undetected and untraceable [5]. Such attacks may lead to the exfiltrating of sensitive user data, such as usernames and passwords, by sniffing the frequency of operation of the targeted device [5] and jamming access to devices, making them inoperable to their users. The subsequent consequences of such attacks range from financial loss to users being unable to gain physical access to their own smart homes.

An example of a popular RF-based attack towards IoT devices is the Zigbee Worm attack, which exploits the Zigbee protocol. This attack targets hard-coded symmetric encryption keys in the Zigbee protocol, which allows the worm to take control of the devices [7]. Infected devices can infect other nearby Zigbee devices, allowing the adversary to take full control of the network. This attack can be achieved as far as 400 m away from the device, increasing the difficulty in tracking and tracing the adversary.

One of the main limitations within the existing literature in this field is that the attacks performed have been deployed using expensive hardware, which requires expensive proprietary software. This means that very few adversaries may be able to gain access to such equipment to leverage such attacks. On the contrary, it is critical to investigate the impact of RF-based attacks that could be deployed using affordable and broadly available hardware.

Given the severity of the impact of RF-based attacks, it is evident that further research is required to understand their behaviours and, subsequently, design defence mechanisms towards mitigating them. To the best of our knowledge, this is the first investigation into how RF-based attacks affect commercially available IoT devices, as well as the proposal of methods towards defending them against such attacks. More importantly, this work considers a realistic attacker model where an adversary uses affordable hardware and open-source tools. The main contributions of the work presented in this paper are the empirical investigations into

1. A detailed experimental setup and methodology for assessing RF vulnerabilities in IoT devices.
2. Launching popular RF-based attacks, including harmonic frequency attacks, using affordable hardware and open-source tools within a realistic attacker model.
3. The behaviour of such attacks towards commercially available smart-home IoT devices.
4. Recommendations for enhancing IoT device security based on our findings, which provide practical guidelines for manufacturers and developers.

The remainder of this paper is divided into the following main sections: Section 2 presents related work, Section 3 introduces the communication protocols used in commercial IoT devices that are relevant and susceptible to RF vulnerabilities, Section 4 introduces RF attacks in ecosystems, Section 5 presents the methodological approach implemented in this paper, including a description of the IoT devices used, as well as their configurations, the RF-based attacks launched on such devices, and a description of the adopted attacker model. Section 6 discusses the findings from deploying RF-based attacks on such devices, Section 8 proposes possible methods towards enhancing the security of IoT devices against RF-based attacks, Section 10 concludes the paper, and finally, Section 11 discusses future work.

2. Related Work

The proliferation of IoT devices for industrial and residential use has increased the significance of network security. Despite notable advancements in this field, research on the RF security of these devices remains relatively scarce.

In many cases, RF investigations have been conducted using very expensive equipment to perform RF-based attacks, which, in most cases, will not be something an average attacker might have access to. In particular, a key study by Bukhari et al. [8] explores innovative solutions to enhance the security of wireless sensor networks, which are integral components of the IoT ecosystem. They propose a federated learning model combined with a Spiking Convolutional Neural Network and Bidirectional Long Short-Term Memory

(SCNN-Bi-LSTM) architecture, aimed at improving the reliability and privacy of intrusion detection systems in these networks. Their findings suggest that this approach not only enhances detection accuracy but also preserves the privacy of the network data, making it particularly valuable in scenarios where sensor nodes handle sensitive information [8]. Moreover, Rugeles Jose de Jesus et al. [9] deployed multiple attacks on both IoT devices and industrial installations using a Universal Software Radio Peripheral (USRP) N210 device. The USRP is an extremely costly piece of hardware that can cover a large range of bandwidth up to 400 MHz. This makes it perfect for performing attacks on devices that utilise frequency hopping, but also unrealistic as these types of equipment are usually owned and operated by state-sponsored attackers [10] like in the case of the Cyprus Surveillance van [11,12]. Although the paper does make valid claims about the types of attacks that can be conducted with different software-defined radios (SDRs), there is no clear indication of what the devices being tested were.

In 2012, Barnickel et al. [13] demonstrated a technique in which a MITM attack could be performed between two devices at the initiation of the Secure Simple Pairing (SSP) process. This vulnerability allowed an adversary to intercept the pairing pin, later allowing them to reconnect to the device without being authorised. Even though the technique is impressive, it requires expensive and sophisticated equipment like the USRP2, which boasts a bandwidth coverage of up to 50 MHz. Notably, Bluetooth is one of the most widely adopted communication protocols due to its low power requirements, making it a staple in many off-the-shelf devices. As the IoT increasingly incorporates battery-less devices, understanding and mitigating energy-related vulnerabilities becomes crucial. A recent study [14] explores the specific challenges posed by energy attacks on these devices, which are typically powered by ambient energy sources. Highlighting the susceptibility to energy depletion attacks, the research emphasises the need for robust security protocols that prevent malicious entities from disrupting device functionality. The paper suggests innovative directions, such as energy-efficient security algorithms and adaptive energy harvesting techniques, to enhance the resilience of IoT systems against such vulnerabilities. These insights are pivotal for developing sustainable and secure IoT architectures in the future.

Furthermore, in the context of real-world scenarios, Karagianis et al. [15] evaluated mitigation strategies against RF. However, the attacker model and assumptions made were not realistic. For instance, during the experimentation phase, the authors evaluated the RF detection model when the cars were next to each other and either at a full stop or at a constant speed. This does not take into account the possibility of frequency drift due to the Doppler effect, which could cause dropped packets and skew the results [16]. In the realm of Industrial IoT (IIoT) security, innovative methodologies are continuously being explored to enhance the robustness of communications under challenging conditions. One such study focuses on the application of RF fingerprinting to secure IIoT systems, even in environments plagued by interference and noise [17]. This approach leverages the unique RF signatures of devices to authenticate and validate communications, effectively mitigating the risks associated with spoofed transmissions and unauthorised access. By adapting to channel impairments and background noise, RF fingerprinting presents a viable security enhancement that ensures the integrity and reliability of critical industrial communications.

Similarly, Arif Mehmood et al. [18] evaluated the security against RF attacks against a smart cart. Although the solution is low cost, using the shelf parts, it lacks any proper filtering from outside interference and also any form of presence check from the receiving device. A simple foil bag would be able to contain the RFID tag's transmission, allowing a malicious actor to leave the store uninterrupted. Phan Duy Hung et al. [19] also created a simplistic IoT device using an Arduino, attacking it using jamming, replay attacks, and sniffing. Although this is a good example of an Arduino-based IoT device, the hardware used in this case lacked any security features by design. In addition, the lack of proper casing and filtering of the device made it susceptible even to the lowest skill level of attack. Finally, the attacks were performed near the device, which could have possibly caused attenuating signals to interfere with the results.

Moreover, Öst Albert [20] attempted to use a HackRF to jam a LORA signal. The attack was unsuccessful as LORA uses spread spectrum signals to communicate and is beyond the HackRF capabilities. This is because LORA devices are used to transmit at a rated 30 dBm or 1 watt, which is 300 times more than the 30 mW transmit power of the HackRF. On the same note, addressing the persistent threat of jamming attacks in IoT environments, recent research [21] introduces a novel approach utilising dynamic channel hopping based on Petri nets formulations. This method enhances the resilience of RF devices by allowing them to dynamically switch communication channels when a jamming attack is detected. The use of Petri nets provides a structured and effective way to model and analyse the state transitions involved in channel hopping, ensuring minimal disruption and maintaining communication integrity.

Finally, Rojas et al. [22] reviewed and summarised the existing literature on RF security techniques for IoT devices. They discussed four specific techniques: beamforming, cooperative jamming, RF fingerprinting, and spread spectrum codes and provided an overview of each technique, including its principles and potential applications in IoT security. However, the study lacked any experimental work, empirical data, or analysis to support their claims.

Consequently, although the existing literature emphasises the critical nature of network security in the IoT, there is a noticeable lack of practical, real-world assessments regarding RF vulnerabilities in these devices. The majority of studies either depend on impractical models, or expensive equipment or fail to incorporate empirical validation. As such, to bridge this gap in the literature, this study conducts an exhaustive evaluation of RF security weaknesses in widely used commercial IoT devices and illustrates how these weaknesses can be easily exploited with inexpensive hardware. This underscores the critical nature of developing resilient defensive systems. Finally, Table 1 summarises the existing literature and its limitations.

Table 1. Limitations in existing literature.

Work	Devices Used	Devices Type	No. of Devices	Attacks
[9]	USRP, UberTooth, RTL-SDR, HackRF	Homebrew, Commercial	N/A	Sniffing, R.E, Replay, DoS
[15]	USRP	Commercial	1	DoS
[18]	Arduino	Homebrew	1	Sniffing, R.E, Replay, DoS
[19]	Arduino, HackRF	Homebrew	1	R.E
[20]	Arduino, HackRF	Homebrew	1	Sniffing, R.E, Replay, DoS
This paper	HackRF, NESDR, CC2531	Commercial	4	Sniffing, R.E, Replay, DoS

3. Radio Frequency Protocols in IoT Ecosystems

This section introduces the communication protocols used in commercial IoT devices that are relevant and susceptible to radio frequency vulnerabilities. These protocols are

widely used in many popular IoT devices; therefore, it is critical to understand how they operate and what attacks they are vulnerable to.

3.1. Wi-Fi

Wi-Fi has been commercially available since 1997, allowing devices to connect to a network wirelessly, with speeds now reaching up to 2 Gbps. The Wi-Fi protocol is a family of wireless network protocols that are based on the IEEE 802.11 standard and are mostly used for LAN (Local Area Network) communication [23]. It is one of the most used protocols for computer networking and is trademarked by the Wi-Fi Alliance, which controls which devices are certified to use the protocol. Wi-Fi uses microwave bands to communicate, specifically 2.4 GHz, 5 GHz, and the up-and-coming 6 GHz, utilising 20 MHz of bandwidth with a limited transmit power of 100 mW or 20 dBm. By using these high frequencies, they allow high data-rate transfers using line of sight [24].

3.2. Zigbee

The Zigbee protocol is an IoT-specific protocol used for the transmission of data from sensors and automation control networks using the IEEE 802.15.4 [25]. It uses very low data rates of around 250 Kpbs and operates on the microwave frequency allocation of 868, 902–928 MHz, and 2.4 GHz frequencies [26]. Usually, higher frequencies are used to transfer data between devices at higher data rates but in a closer range. Zigbee devices have a maximum range of 100 m [27].

Zigbee Infrastructure

The structure of the Zigbee network consists of three main devices: the coordinator, router and end device. A Zigbee network cannot function without a coordinator, as it bridges the device with the network. The coordinator is in charge of storing the information sent by the end device, while the router will re-transmit the data to the appropriate device [28].

3.3. Bluetooth

Bluetooth is a short-range, low-power mode for transferring data between two devices. It is usually used to sync between devices in a mesh network, send files between devices, stream music, etc. It operates on the 2.4 GHz portion of the spectrum, utilising 79 channels of 1 MHz bandwidth each [29]. The Bluetooth network operates using frequency hopping to minimise interference from other sources. When two devices initiate a connection, they first initiate with a handshake signal, which requires authorisation from both parties.

Bluetooth devices are organised with a star topology, meaning that there is one master server that sends data to multiple slave nodes [30]. For example, a master server could be a phone and the slaves are the wireless earphones or speakers. A Bluetooth packet is made up of three parts: the access code, headers, and payload.

4. Radio Frequency Attacks in IoT Ecosystems

Multiple studies have demonstrated that IoT devices are vulnerable to a wide range of attacks, including RF-based and physical attacks (e.g., [9,15,18,20]). Some of the reasons why such devices are insecure include limitations in computational power, lack of transport encryption, insecure web interfaces, lack of authentication and authorisation mechanisms, and their heterogeneity, which makes applying uniform security mechanisms extremely challenging [1]. Consequently, several IoT attack categories have emerged:

- Replay attacks: replay a transmitted signal to the targeted device, causing IoT devices to respond to messages or commands that have already been issued. In the context of RF-based attacks, the adversary needs to be located between the transmitter and the targeted device [31].
- DoS (jamming) attacks: aim to make IoT devices unavailable to their intended users by temporarily or indefinitely disrupting their services [32]. In the context of RF-based

attacks, to produce a jamming signal, the adversary uses a signal generator to produce a continuous wave, which is then sent to the target device. The adversary will also attempt to send out a stronger signal than the target signal, effectively making the target device non-operational.

- Sniffing: utilises specialised software that can be used to intercept and analyse the signal of a device to exploit information such as its modulation, sample rate, and bandwidth, to facilitate other cyber attacks.
- MITM: compromises the communication channel between the IoT device and the intended recipient of its data. Once the connection is compromised, the attacker can act as a proxy, and, therefore, read, insert, and modify the transmitted data [33]. One of the most common forms of MITM attacks is spoofing attacks, where the adversary takes on the identity of an authentic device, trying to lure the victim and intercept their data [34].
- Harmonic attacks: Every electronic device that utilises radio frequencies for communication and operation, such as Wi-Fi, Zigbee, LORA etc., is prone to harmonic frequency emissions [35]. In this case, this could allow attacks to be carried out by using even less sophisticated equipment by transmitting on lower frequencies [35]. More specifically, when a wave oscillates, it produces harmonic waves with a reduction in bandwidth each time. This results in an oscillation on a different frequency. Harmonic Spurious emissions are also problematic in electronic devices as they produce unwanted oscillations close to the fundamental frequency as well as harmonic frequencies, which can cause interference. For example, a device transmitting at exactly 433 MHz with a bandwidth of 25 kHz will also bleed onto other frequencies well over its bandwidth without proper filtering [36]. This is especially prominent in dual-side mode modulations such as Frequency Modulation (FM). Many of these spurious emissions are generated by low-noise amplifiers embedded in the devices, which are used to amplify the low-powered signal transmitted from the device. Using this type of amplification causes not only the desired transmission signal to be amplified but also any harmonics or "noise" it generates [37].

5. Methodology

Figure 1 describes the testing methodology implemented in this paper. Firstly, an extensive analysis of the devices' characteristics was needed to evaluate the type of approach taken. As these devices need to be certified to be sold to consumers, they must be certified by the Federal Communications Commission (FCC) [38] (for the US) and CE [39] (for Europe). A detailed analysis of each device is documented by both organisations, specifying the frequency of operations, their board types, and the protocols used. By referring to these documents, the reconnaissance phase of identifying the frequency of operation could be bypassed. Next, if the frequency of operation was lower than 1.7 GHz, the RTL-SDR would be used to take advantage of its higher sensitivity. If the device used an unencrypted protocol, packets were captured and analysed to determine if they could be reverse-engineered and replayed back to the device while also targeting the first, second, and third harmonics. If this were not the case, jamming attacks would be performed using the HackRF, targeting the frequency of operation and the relevant harmonic frequencies until the impact of the attack was negligible. Lastly, the overall results of the testing were analysed and reported.

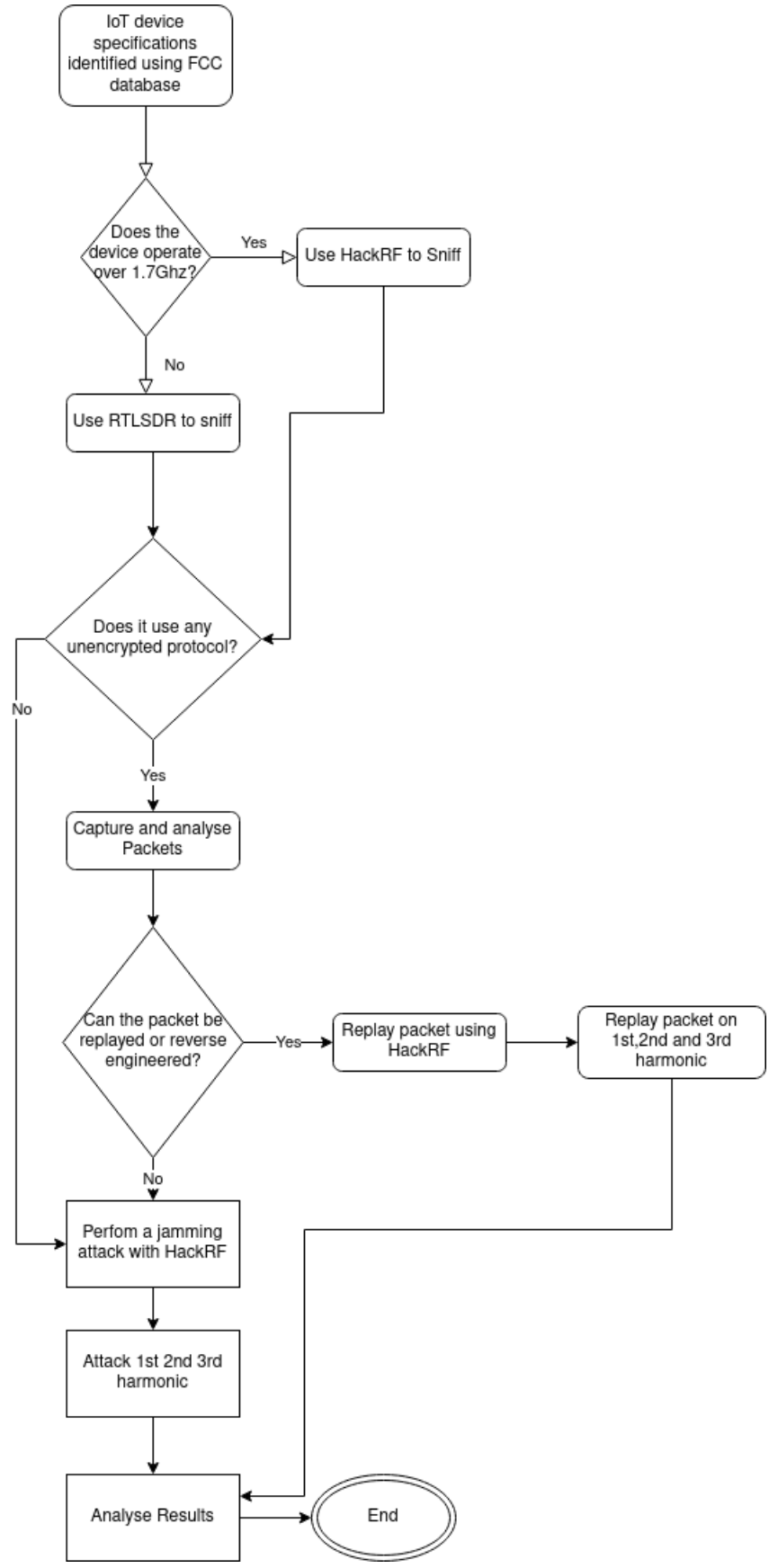

Figure 1. Description of the testing methodology.

5.1. IoT Devices

The testbed used to support the experiments provided in this paper consists of four IoT devices of different types (e.g., multimedia, sensors) that utilise different communication protocols. The rationale behind choosing these devices was due to their popularity [40–42], their large user base, and the diversity in their functionalities and communication protocols.

- The Belkin Nestcam HD is a small, compact wireless camera used for personal surveillance, employing a 720p resolution for video capture. Its maximum transmission power stands at 67.5 mW, which is notably below the 100 mW limit set by the FCC to ensure safety and minimise interference. Given its transmission power and the use of a small PCB antenna, the camera's signals can primarily be received nearby [43]. This design choice emphasises its intended close-range surveillance application while also reducing potential interference.
- The NEST smoke alarm is a smart sensor alarm that uses both Bluetooth and Wi-Fi to connect to the Internet and a Nest Hub. It uses Bluetooth 4.0 with a maximum transfer rate of 1 Mbps as well as a modulation type of GFSK, operating between 2.402 and 2.480 GHz. The channel spacing used by the device is 2 MHz, allowing a HackRF to sniff at least one channel [44].
- inFactory sensors are widely used and can often be found in many weather stations. To send data over the Internet, such stations either have built-in functionality or they use a third-party device. They often operate at 433.920 MHz using FSK modulation, which sends out packets to neighbouring monitoring devices that display the data. These data are often sent using an unencrypted signal, which can be decoded using the rtl_433 software [45] .
- The Sonoff SNZB-02 is a small temperature and humidity sensor that utilises the ZigBee protocol. The device uses a standard 2.4 GHz RF frequency and uses GFSK modulation, which is very low bandwidth and low power modulation. The device itself is housed in a small plastic casing with minimal shielding from other forms of RF interference. The device itself sends out a beaconing packet, which alerts any receiving device of the current temperature and humidity of the area [46].

Table 2 summarises such devices.

Table 2. The IoT devices included in the testbed. All the devices used herein were sourced in the United Kingdom

Manufacturer & IoT Device	Type	Protocol (s)
Belkin—NestCam HD	Multimedia	Wi-Fi
Nest—Smoke Alarm	Sensors	Bluetooth and Wi-Fi
inFactory—Sensors	Sensors	433.920 MHz
Sonoff—SNZB-02 Sensor	Sensors	ZigBee

5.2. Radio Frequency Attack Equipment

This Section describes the tools that were used to deploy RF attacks on the IoT devices described in Section 5.1. The rationale behind selecting such tools was their affordability; this increases the accessibility of such devices to a wider user base, increasing the volume of attacks and, thus, their impact and versatility—their ability to exploit a range of RF protocols. As such, the following devices were used to facilitate the RF attacks launched:

- The HackRF is a half-duplex SDR transceiver designed for RF investigation. It has an operating frequency of 1 MHz to 6 GHz with an 8-bit quadrature sample rate. It has a maximum transmission power of 30 mW, which makes it suitable for transmitting at close ranges. Nonetheless, it is prone to sporadic emissions, which could compromise nearby equipment if used in combination with amplification equipment. In addition, the HackRF is compatible with GNU-Radio, allowing it to be calibrated to

work with multiple other equipment and protocols. One of the main disadvantages of the HackRF is its long-range receiving performance. In this case, a Temperature Compensated Crystal Oscillator (TCXO) was added to the HackRF to increase its reception performance and minimise frequency drift while receiving for long periods. TCXOs are often used to correct the voltage of the tuner chip, which results in variations in frequency over temperature. The voltage correction is applied to a varactor diode located on the crystal circuit, which varies the crystal frequency by a minuscule amount. This results in a more stable frequency [47]. An important factor to consider is that TCXO chips reside when they have experienced extreme temperatures for long periods. This may demonstrate a frequency shift when the chip returns to its normal ambient room temperature.

- The RTL-SDR (NESDR) is a low-cost SDR receiver capable of receiving from 25 MHz up to 1.7 GHz. Its wide popularity has given rise to multiple software packages which support a wide range of decoding protocols, such as Digital mobile radio (DMR), Automatic Packet Reporting System (APRS), Long Range (LORA), Project 25 (P25), and FSK [48]. When compared with the HackRF, the NESDR is only capable of receiving transmissions but has overall better community support. RTL-SDR-based devices are based on two chips: the RTL2832U demodulator chip and the R820T2 tuner chip. It also features a 0.5 PPM (parts per million) low-noise TCXO chip, which is capable of keeping the tuner in sync with the demodulator as it heats up. Since heat is a major factor in the signal stability of the tuner, it is important to keep the device in a stable temperature range [48].
- The CC2531 USB Evaluation Module Kit provides an easy and convenient way to interface with the 802.15.4 ZigBee protocol. It allows the decoding and syncing of ZigBee devices and provides verbose logging data that enable the exploration of any vulnerabilities of the IoT devices [49]. The small PCB antenna allows the testing of IoT devices within a small range without having to expose them to unauthorised third parties. The onboard CC2531 chip is one of the standard chips used for the ZigBee protocol, along with the CC2530, which is a discrete-level chip without the USB interface, and the CC2538, which features an Arm Cortex-M3, which allows it to handle many more simultaneous devices [49]. The small eight-pin headers on the board allow custom firmware to be flashed on the board. In the experiments herein, the CC2531 board was flashed with the sniffing firmware to be compatible with "ZigBee to MQTT", which provided verbose logging of the testing process. This eliminated the need for third-party servers and software, which become unresponsive at any time. After flashing the firmware, a Linux-based (Parrot OS 4.11) [50] device was loaded to act as an MQTT server for connecting to the ZigBee devices via the board.
- GNU Radio is an open-source software development platform used to create virtual signal processing blocks which can be implemented with software-defined radio such as the HackRF. Using GNU Radio creates more complex RF signal simulations without the need for expensive hardware, thus reducing the cost of the testing equipment. Many of the components of GNU Radio are interchangeable and allow it to interface with multiple programs, which expands its functionality. Such software includes SDRangel [51], Sparrow Wi-Fi [52], zigbee2MQTT [53], rtl_433 [54], and Universal Radio Hacker [55,56].

5.3. Identifying the Attack Vector of IoT Devices

Before exploiting the IoT devices discussed in Section 5.1, their characteristics, specifications, and how they operate needed to be analysed. Information such as indications of certifications of specific protocols can be derived from the labels found on the devices. To determine which attacks are relevant to which devices, the following attributes were considered:

1. What protocols does the device use?
   By identifying the device's protocol, an investigation into known vulnerabilities was conducted.
2. What frequencies does the device operate on?
   By using either the FCC listing or software-defined radios, the device's frequency of operation was determined and investigated for harmonic or spurious emissions. More specifically, an investigation into the inner workings of the device and its components was implemented to identify which attributes of the device could be exploited. This was crucial in identifying and launching relevant and effective low-skill and low-effort attacks.
3. Does the device contain sufficient filtering?
   Smaller devices with limited space for hardware often lack sufficient filtering and protection from outside interference. This was the case with the Sonoff SNZB-02 tested in this work, where it was possible to interfere with its normal operation by physically obstructing the device's antenna.
4. Is the device dependent on a third-party application?
   Many IoT devices depend on third-party applications, without which they are not operational. In this case, custom firmware may be installed on the receiving device to make them usable. This was the case with the Sonoff SNZB-02 tested herein, which required the receiver to be flashed with Z-Stack-firmware to allow the receiver to act as a coordinator [57].
5. What other unique characteristics does the device possess?
   Some devices, such as the inFactory sensor, use a specific protocol tailored to the unique characteristics of the device. This meant that a custom attack vector was needed so that it would conform to the specifications of the device.

5.4. Experimental Setup

To conduct the experiments herein, the devices were placed in proximity to the software-defined radios. To eliminate the chances of interference, the IoT devices were placed in a large room of approximately 53.60 m^2, and it was ensured that no other device was present and turned on nearly two meters from each other. As the HackRF has low receiving performance, transmissions between devices present in nearby rooms were not of concern. In addition, to ensure the HackRF transmitted with almost zero loss, an appropriately tuned 2.4 GHz rubber duck antenna was used. To further reduce the chances of interference from other devices, the low-noise amplifier gain and base band gain were reduced until only the signals coming from the tested devices were received. Figure 2 illustrates the experimental set-up.

In more detail, each attack was deployed on the IoT devices at least three times to ensure the reproducibility and effectiveness of the attacks. However, during our experiments, it was noted that certain protocols employed by the IoT devices utilised a feature known as Frequency-Hopping Spread Spectrum (FHSS). FHSS is a method wherein the signal being transmitted jumps between various frequencies in a seemingly random manner. This randomness and unpredictability inherent in FHSS proved to be a challenge, as in some instances, it interfered with the consistent reproduction of our attacks. In essence, because of the dynamic nature of the frequency shifts, not all attacks yielded the same results every time they were executed. This variability underscored the effectiveness of FHSS as a potential defence mechanism against certain types of RF attacks.

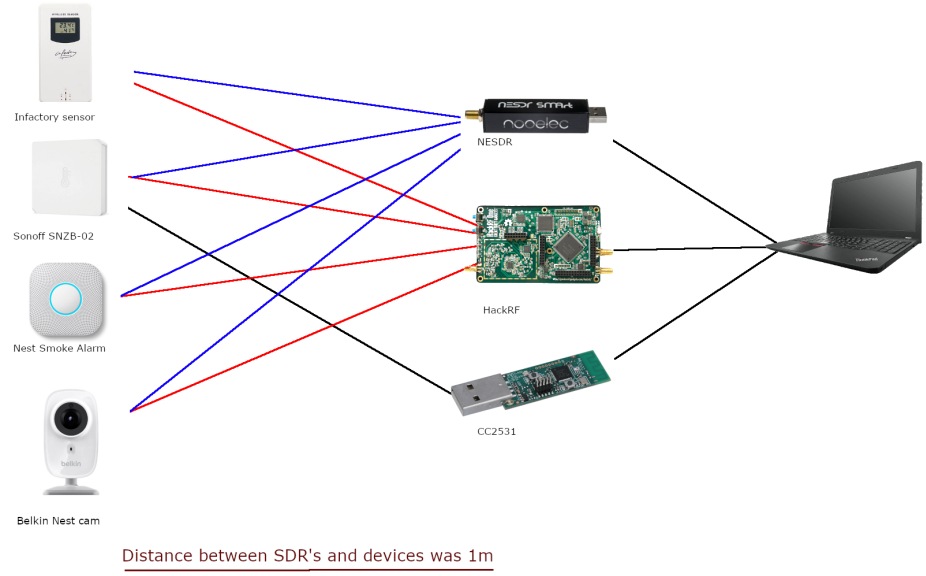

Figure 2. Experimental setup.

6. Experiments and Results

In this section, the findings that have been obtained from a series of experiments are presented. The overall results showed that the devices have varying degrees of fragility to different attacks and, in some cases, could be completely disabled.

6.1. Belkin NestCam HD

With the use of Sparrow Wi-Fi in conjunction with the HackRF, it was deduced that the camera was operating at 2.437 GHz (Channel 6). The traffic from any device trying to connect to the camera could also be viewed if needed.

- Deauthentication—Figure 3 illustrates the interface of the Sparrow Wi-Fi software, which was used to launch a de-authentication attack from both the computer's internal Wi-Fi adaptor and the HackRF. When using the computer's inbuilt Wi-Fi, the graph demonstrates a disconnection of the camera from the network, successfully preventing RF attack equipment from gaining any further connections. The HackRF was also used on its own to perform the attack by monitoring the frequency of ICMP packets. Using "hackrf_transfer", a ".bin" file was saved containing the IQ data and then re-transmitted at 2 Mega samples per second. This resulted in a minor increase in latency. Subsequently, Sparrow Wi-Fi covers the whole channel of operation of the targeted IoT device using the computer's built-in Wi-Fi adaptor. This is due to its larger bandwidth capabilities in comparison to the HackRF.
- Attacking First Harmonic—To evaluate the harmonic attack, ICMP packets were sent on the harmonic frequency of 1.2185 GHz, which also has a second harmonic of 2.437 GHz. The results demonstrated a minor increase in latency, but not enough to cause disruption or any packet loss. These results can be observed from Figures 4 and 5, where high peaks in the latency were observed while attacking via the harmonic, but the attack was not strong enough to cause any disruption. By using the HackRF to send ICMP packets in an attempt to jam the signal, it was observed that there was no impact on the signal quality of the device, indicating its resilience and capability to communicate effectively without any interference-induced degradation. This may be explained by the fact that the HackRF does not transmit at a bandwidth large enough to cover the whole channel on its own; the low power transmitted by the HackRF is not enough to completely overshadow the transmission of the camera; the Wi-Fi changes between modulation types; therefore, if jamming is performed using only one form of modulation, disruption may be temporary.

- Signal Sniffing—The laptop's inbuilt Wi-Fi adaptor was used with monitor mode enabled to capture and inspect the camera's packets. However, no obvious vulnerability to exploits was detected. The HackRF's lack of bandwidth was a limitation for its ability to sniff packets directly.

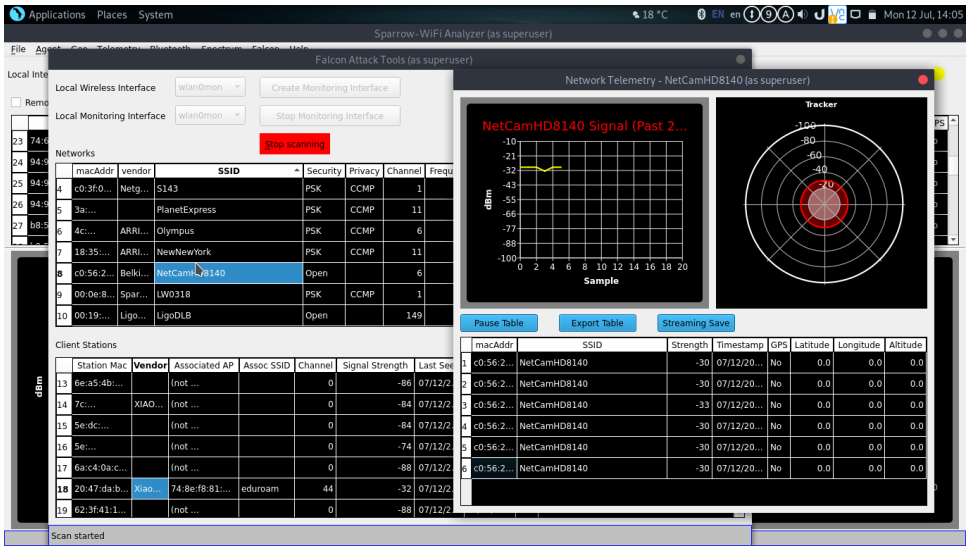

Figure 3. Sparrow Wi-Fi deauthentication.

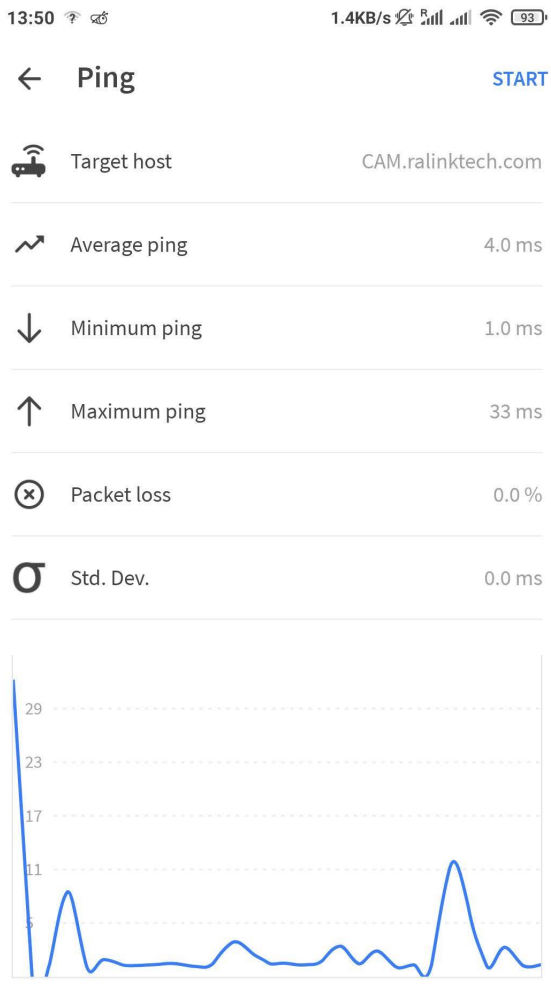

Figure 4. HackRF 2.437 GHz.

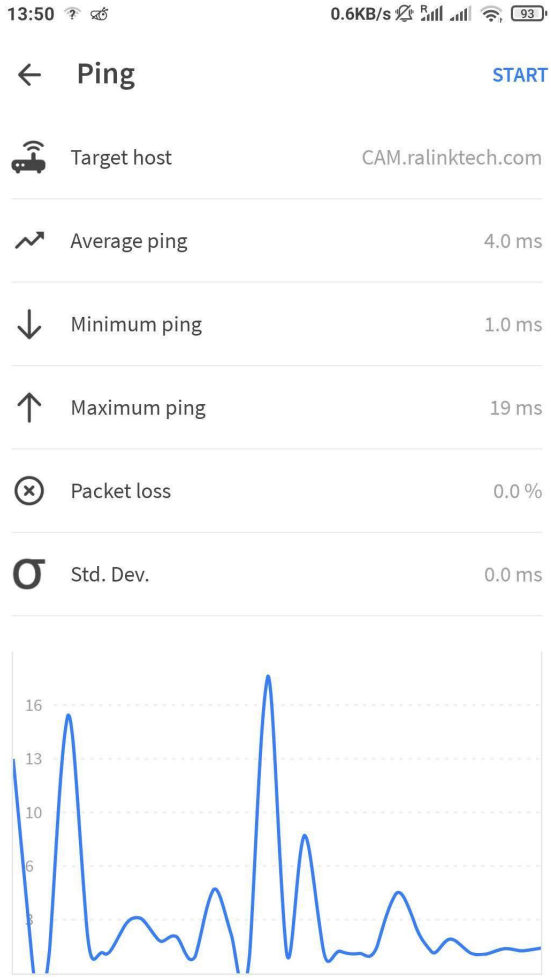

Figure 5. HackRF 1.2185 GHz.

6.2. NEST Smoke Alarm

- Deauthentication—Similar to the Belkin NestCam HD, the Sparrow Wi-Fi software was used to launch a de-authentication attack on the NEST Smoke Alarm from both the computer's internal Wi-Fi adaptor and the HackRF. When using the computer's inbuilt Wi-Fi, no obvious vulnerability to exploit was detected. However, an interesting observation was the large number of beaconing packets being transmitted. The HackRF was also used on its own to perform the attack. The only observation was the increase in latency between the access point and the camera.
- Bluetooth Sniffing—The computer's internal Bluetooth device and the HackRF was used to sniff the Bluetooth spectrum and monitor the packets between the NEST Smoke Alarm. The HackRF's lack of bandwidth did not allow complete packets to be received. In this case, an attempt was made to receive a channel using Channel 37, which is the default starting channel for Bluetooth. This resulted in the reception of multiple malformed packets, with only the NEST's broadcast packets being those that could be decoded successfully. When using the computer's internal Bluetooth device to monitor the packets, it was observed that beaconing packets are mostly transmitted by the NEST device. This demonstrates the limitation of the HackRF's bandwidth and its inability to attack the IoT device using Bluetooth.
- Jamming—The HackRF's lack of bandwidth removes its capability to sniff Wi-fi packets as they are too large. It is also not able to replay the handshake signal for this reason. Using the "hackrf_transfer" function to re-transmit a capture signal did not yield any substantial results.

6.3. inFactory Sensors

To acquire and decode signal transmitted from the inFactory sensors, the "rlt_433" software was used. Such data were sent out using a frequency of 433.920 MHz. To analyse signal data, Universal Radio Hacker was used as it allows signal analysis, packet manipulation, and reverse engineering. The flags in the following command returned the data sent out by the inFactory sensor only, filtering out any unwanted sensors if they were present:

– d: Device number
– R: Device ID
– a: Analyze mode 4
– A: Pulse Analysis
– S: Store all raw signals
– rtl_433 -d 1 -R 91 -a 4 -A -S all

Figure 6 illustrates the data sent out by the sensor in imperial units (Fahrenheit) and converted to Celsius on the receiver. In this case, the temperature is 85.70 F with a humidity of 61%. It can also be observed that the channel being used is listed; in this case, it is Channel 1. This information is useful when reverse engineering packets using Universal Radio Hacker for packet manipulation.

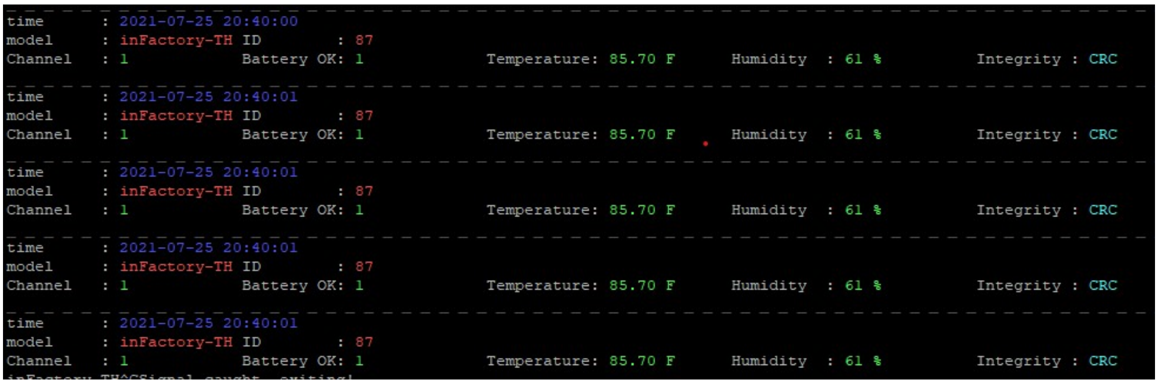

Figure 6. Decoded inFactory sensor data.

- Jamming—Many of these sensors are housed in thin plastic casing with little or no shielding from external RF interference. This makes the device vulnerable to jamming attacks. To investigate the effectiveness of this attack, a signal irrelevant to the modulation mode from the HackRF was sent to the sensor. This jammed the connection between the HackRF and the sensor. This attack was escalated further by utilising a handheld transceiver to transmit a carrier wave on 433.920 MHz with more than 1 watt. This has the potential to jam multiple devices on the same frequency and at a much larger distance.
- Replay Attack—To conduct a replay attack and gauge its success, the sensor was first placed inside a freezer. This chilling process was essential to create an abnormal condition, causing the temperature to drop to an unusually low −9 degrees Celsius, and thereby triggering the sensor to transmit an alert signal. Once the sensor was removed, the Universal Radio Hacker was used to capture this alert signal. Six separate samples were collected for consistency. Leveraging the software's integrated features, the signal was then relayed to the receiver. However, this attack attempt did not succeed. Subsequently, to further examine the feasibility of a replay attack, the following methods—which were designed based on the constraints of the environment and the limitations posed by the physical conditions of the experiment—were attempted:

    1. Removing the batteries from the sensor and sending the signal to the receiver without the sensor present. This method tests the hypothesis that the receiver might recognise a valid signal even in the absence of the actual sensor, offering a basic but essential step in validating the integrity of the receiver.

2. Resetting the receiver and then sending the signal.
3. Sending the signal at 30 s intervals.
4. Capturing the handshake signal sent out by the sensor, disabling the sensor by turning it off, and using the signal we have captured in the hopes that the handshake signal would be enough to trigger the receiver to display an output.

Following our evaluation, all methods were unsuccessful. However, an intriguing observation emerged: after attempting the attack with the sensor turned off, once reactivated, the sensor struggled to re-establish its connection with the receiver. This difficulty persisted even when manually attempting to synchronise using the 'sync' button on the device's rear, as well as by switching the sensor's channels. This suggests that the receiver's internal clock might have been desynchronised, hindering any new device connections. A hard reset of the receiver, achieved by removing its batteries, was the only remedy to re-synchronise the devices.

6.4. Sonoff SNZB-02

- Jamming—Like many devices, the Sonoff SNZB-02 started sending a beaconing packet on Channel 11 (2.405 GHz). This channel is the most frequently used channel by devices that operate in the 2.4 GHz band, as it initiates the time synchronisation of the devices. This channel is also used by the ZigBee protocol to send out the beaconing packets. But, as the Zigbee protocol is designed to use frequency hopping, it becomes difficult to jam the channel. After further investigation, the signal was successfully jammed using the SDRANGEL and its 802.15.4 module. More specifically, by transmitting a random ZigBee packet, it was possible to stop the receiver from gathering telemetry from the device. It is worth noting that this was successful only after synchronising the device with the coordinator. However, as the HackRF has greater transmission power in comparison to the Sonoff device, its signal was overpowering. A bandwidth of 10 MHz was used to jam channels 11 and 12, preventing the device from hopping to other channels. The attack was successful as the device was not responsive and was unable to receive a signal for three minutes. Furthermore, the HackRF's transmission power and LNA gain were set to 61 and 47, respectively, which led to the jamming of both 11 and 12 channels.
- Replay Attack—Using Universal Radio Hacker, it was possible to capture the signal of the device. However, due to the nature of the Zigbee protocol and its use of frequency hopping, it was not possible to successfully replay the initialisation packet. Subsequently, it was not possible to connect to the coordinator using the HackRF. However, although it was possible to capture part of the packet, it was not large enough to be decoded or analysed. With this in mind, it was possible to briefly replay the packet back to the device when it was listening to channel 11. This was only momentary and only happened for a few minutes; after that, we were unable to do the same thing again for some time. This is likely due to the Zigbee protocol using spread spectrum, hence, when the signal was transmitted, only the current channel was monitored for a brief period before it switched to another one. This finding is important, as from an adversary's perspective, being able to capture the full packet in such attacks significantly increases the severity of the possible consequences. For example, if the sensor was connected to a thermostat, the results could be catastrophic as the temperature could be changed uncontrollably, causing bodily harm.
  As shown in Figures 7 and 8, we can deduce:

  1. The replay attack was successful, as the temperature and battery values were the same throughout the attack.
  2. The link quality is calculated by Zigbee2MQTT and not by the sensor.
  3. The values for humidity are inconsistent, which could be due to malformed packets being sent out by HackRF or the use of spread spectrum, which caused only a part of the packet to be received.

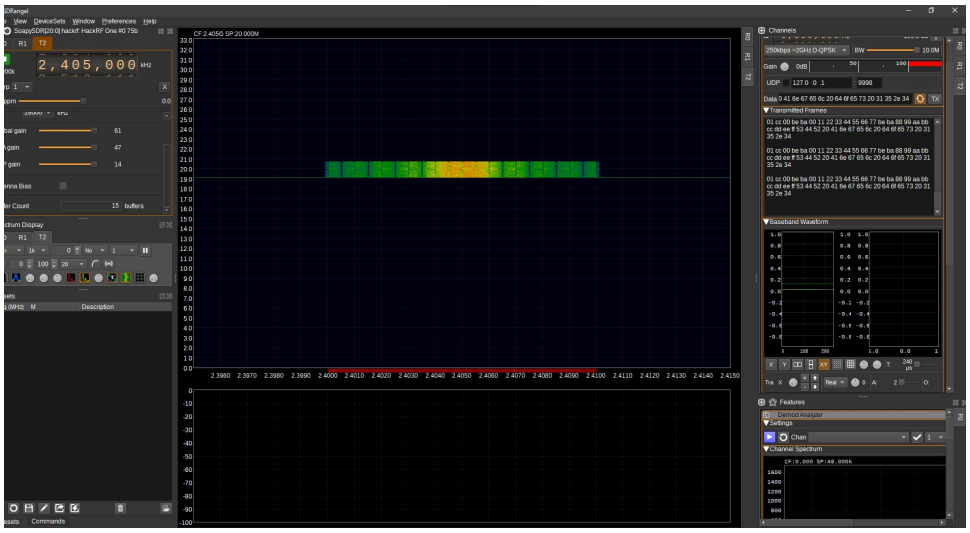

Figure 7. Replaying the signal back using SDR angel.

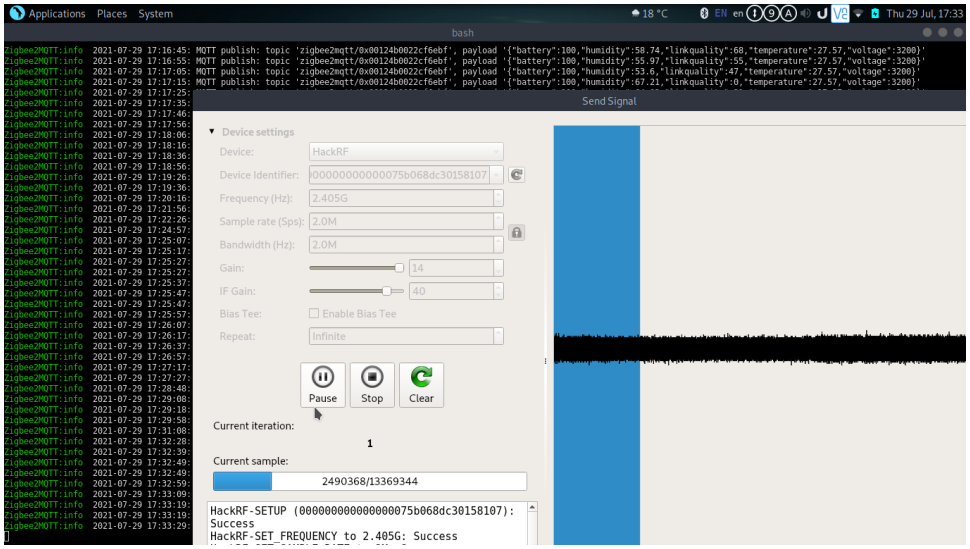

Figure 8. Replay attack.

Table 3 provides a comprehensive overview of the experimental results, detailing the susceptibility of IoT devices to various attacks. It is important to note that all devices, regardless of their communication protocol or the absence of signal hopping, were uniformly vulnerable to sniffing attacks. Devices that operated on a single channel and adhered to one frequency were invariably susceptible to attacks using the first harmonic. However, only devices reliant on a singular communication mode succumbed to jamming attacks. Notably, the Nest smoke alarm showcased resilience against these jamming attempts, as the bandwidth demands of both its Bluetooth and Wi-Fi outpaced the transmission capabilities of the Hack RF. This study underscores the pressing security vulnerabilities present in many IoT devices, highlighting the importance of our research in this realm.

Table 3. Experimental results.

Attack	Belkin NestCam HD	Nest Smoke Alarm	inFactory Sensors	Sonoff SNZB-02 Sensors
Deauthentication	X	X	✓	✓
Attacking first harmonic	✓	✓	✓	X
Signal sniffing	✓	✓	✓	✓
Bluetooth sniffing	N/A	✓	N/A	N/A
Jamming	✓	X	✓	✓
Replay	X	X	✓	X

7. Discussion

The experimental results highlight significant vulnerabilities in IoT devices when subjected to various RF attacks, including jamming and replay attacks. Each device tested—Wi-Fi-enabled security camera, Zigbee-enabled thermostat, and 433 MHz smart garage door opener—demonstrated varying levels of susceptibility, revealing critical insights into their respective security postures.

Our findings indicate that the Wi-Fi-enabled security camera was particularly vulnerable to replay attacks. This can be attributed to insufficient encryption and authentication mechanisms, aligning with some observations by other researchers, who noted similar weaknesses in older IoT devices lacking advanced security protocols. In contrast, the Zigbee-enabled thermostat exhibited resilience under low-intensity jamming but failed under high-power interference. This vulnerability is consistent with [58], which documented comparable weaknesses in Zigbee devices subjected to RF interference. The smart garage door opener operating on the 433 MHz frequency was highly susceptible to both jamming and replay attacks. The lack of robust security features, such as frequency hopping, contributed to its vulnerability.

The success of jamming attacks against the Wi-Fi-enabled security camera and the smart garage door opener underscores the critical importance of signal integrity. The camera's failure to maintain a stable connection during jamming is indicative of its reliance on continuous, unimpeded RF communication, a common weakness in many consumer-grade IoT devices. In contrast, the thermostat's initial resistance to jamming can be attributed to its robust signal modulation techniques, although it ultimately succumbed to sustained high-power attacks.

Replay attacks were particularly effective against devices with static encryption keys or outdated security protocols. The security camera's vulnerability highlights the need for dynamic key management and real-time authentication. These features were absent in the tested devices, rendering them susceptible to unauthorised access through replayed signals.

The practical implications of our findings are significant for the design and development of future IoT devices. Manufacturers must prioritise the integration of advanced security features such as dynamic encryption, frequency hopping, and real-time anomaly detection to mitigate RF attack risks. The insights gained from our experiments can inform the development of more resilient IoT architectures, capable of withstanding the sophisticated nature of modern cyber threats.

8. Towards Defending IoT Devices against Radio Frequency-Based Attacks

This section outlines potential strategies that could be used to protect IoT devices against various RF-based attacks. These techniques are essential for establishing the devices' confidentiality, integrity, and reliability.

8.1. Encryption

Encryption plays a pivotal role in safeguarding data integrity and confidentiality within IoT devices. Among various encryption methods, the Advanced Encryption Standard (AES), particularly AES-256, is highly recommended due to its proven robustness and efficiency [59]. As a symmetric key algorithm, AES-256 encrypts data in 128-bit blocks, providing a high level of security for IoT applications. Symmetric encryption, which uses a single private key for both encryption and decryption processes across devices, is widely utilised in IoT due to these strengths [60]. Furthermore, AES-256 not only secures data against current cryptographic threats but is also considered resistant to future quantum computing attacks, offering a significant level of post-quantum security [61,62]. To address the CPU overhead issues in resource-constrained IoT environments, developments such as cryptographic AES co-processors have been introduced. These co-processors minimise CPU load and enhance key security through advanced isolation mechanisms [61]. Despite the effectiveness of AES, the necessity for a shared private key in large networks introduces risks, such as potential exposure to malicious actors. This vulnerability underscores the importance of also considering asymmetric encryption methods, which use a pair of public and private keys for enhanced security dynamics. Asymmetric encryption is particularly advantageous in scenarios where key distribution poses a security risk. For guidelines on implementing both AES-256 and exploring asymmetric cryptographic options, referring to the latest NIST publications is advised [63].

Thus, asymmetric key encryption was created to solve this problem [59]. One of the most efficient asymmetric-key encryption schemes available is Elliptic Curve Cryptography (ECC), which offers a strong trade-off between security level and resource consumption. ECC is particularly advantageous in environments where computational power, memory, and battery life are limited, providing robust security with smaller key sizes compared to other asymmetric schemes [64,65]. Instead of using the same key to encrypt and decrypt, different private and public keys are used. The public key is shared with everyone, while the private key is used to maintain its function to encrypt confidential data and only allows the owner to access the data. This does come with a performance overhead and requires more processing time and resources, which is exceedingly sparse on low-end devices such as sensors and IoT devices. This is why, even though AES uses a Symmetric Key Encryption standard, due to its low resource requirements, it can work on IoT devices [59]. The effectiveness of ECC in IoT can be explored further in the IETF RFC 7748 [66], which discusses the use of Curve25519 and Curve448 for ECC.

Encryption is one of the best mitigators to sniffing attacks, as it requires a key for the data to be decrypted. In some cases, a dedicated encryption chip is used for handling encryption, which removes the overhead from the CPU.

8.2. Frequency Hopping Spread Spectrum (FHSS)

FHSS has been recognised as a highly effective method for mitigating radio frequency interference and securing wireless transmissions from eavesdropping. FHSS works by rapidly switching the carrier frequency among many frequency channels, using a pseudo-random sequence known to both the transmitter and receiver. This technique significantly enhances resistance to interference and jamming [67]. For a detailed discussion on FHSS applications in IoT security, see IEEE Standard 802.11 [68], which provides guidelines for implementing FHSS in various communication devices.

The implementation of FHSS in IoT devices involves modifying the firmware to support frequency hopping protocols. This includes setting up synchronisation mechanisms that ensure both the transmitter and receiver follow the same frequency hopping sequence without losing alignment. The complexity of the pseudo-random sequence used in FHSS can be adjusted to strike a balance between security and performance, depending on the device's capabilities and the operational environment.

FHSS not only helps reduce the impact of jamming attacks but also minimises the risk of signal interception and unauthorised access. By spreading the signal across multiple

frequencies, FHSS makes it harder for an attacker to maintain continuous interference or eavesdrop on the communication effectively. Additionally, FHSS can be combined with other security measures, such as encryption, to further enhance the overall security of IoT devices.

One of the key advantages of FHSS is its adaptability to various IoT environments. Whether in a densely populated urban area with significant RF interference or in a remote location with minimal interference, FHSS can dynamically adjust its hopping pattern to maintain robust and secure communications. Despite these benefits, implementing FHSS requires compatible hardware and software, which may increase the cost and complexity of IoT devices.

8.3. Scrambling

The oldest method of obfuscating the content of RF transmissions is scrambling. This is achieved by using a set frequency value to invert the frequency of the input signal, which creates a "clone" of the signal on a different frequency range [69]. In other words, the signal is transposed to a different range and can only be decoded by a device using the same frequency offset and settings. Compared to other methods of protection, it is the most primitive, as it can be easily broken and de-scrambled. Even more complex scrambling methods like Split-Band Inversion are also prone to de-scrambling [70].

For example, de-scrambling voice communication can be achieved by recording the signal we want to de-scramble using a program like "SDR#" and an SDR capable of listening on the specified frequency. After recording the IQ data, we play them back using the same software and look for the inverse modulation. Depending on the modulation, we can use either upper- or lower-sideband modulation to demodulate one of the two inverted modulations [69]. We can observe this phenomenon in Figure 9, where the top graph indicates the noise floor and the bottom spectrogram shows the signal, where the redder areas show stronger signals. Scrambling can be implemented through software modifications in the signal processing module of the IoT device. It requires both the transmitter and receiver to be synchronised with the same scrambling sequence.

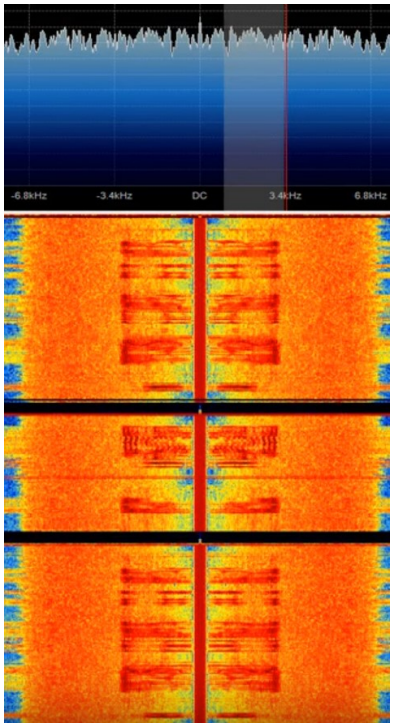

Figure 9. Scrambled signal.

Scrambling adds an extra layer of security by making intercepted signals harder to decode without the correct sequence. It is particularly effective against eavesdropping and simple interception attacks. However, its effectiveness can be limited if the scrambling sequence is discovered by attackers. For that reason, it is recommended that scrambling be combined with other techniques, like encryption for enhanced security.

8.4. Split-Band Inversion

Split-Band Inversion works by adding one more carrier frequency, which divides the spectrum into two equal but opposite parts and then combines them into one signal. This process significantly increases the complexity of the signal, making it difficult for unauthorised receivers to decode. The most challenging aspect of Split-Band Inversion is accurately identifying the two initial frequencies before the combination occurs. These frequencies must be precisely selected and aligned to ensure the integrity of the inverted signal.

Automated tools, such as "deinvert" developed by Oona Raisanen, streamline the process of Split-Band Inversion. Using an RTL-SDR, "deinvert" can autonomously monitor a frequency and perform the necessary operations with minimal manual intervention. This tool simplifies the process of demodulating the inverted signal, making it accessible for practical applications. However, despite its compatibility with Split-Band Inversion, the success rate tends to be marginally lower due to the increased signal complexity and the precision required in frequency alignment [71].

The implementation of Split-Band Inversion involves sophisticated signal processing techniques. It requires digital signal processing (DSP) hardware capable of handling the inversion and recombination of the frequency bands. The DSP algorithms must be meticulously designed to ensure that the signal inversion is both effective and reversible, allowing the intended receiver to reconstruct the original signal accurately.

Split-Band Inversion offers several security benefits. By dividing and inverting the signal spectrum, it creates a more secure communication channel that is resilient to eavesdropping and unauthorised decoding. This technique can be particularly useful in environments where signal interception is a significant concern. Moreover, Split-Band Inversion can be combined with other security measures, such as encryption and FHSS, to further enhance the overall security of IoT devices.

However, the effectiveness of Split-Band Inversion also depends on the precision of the signal processing equipment and the environmental conditions. Variations in the transmission medium, such as multipath propagation and noise, can affect the quality of the inverted signal. Therefore, it is crucial to calibrate the system accurately and ensure that the transmission environment is as stable as possible.

8.5. IoT Security Standards

In IoT security, adherence to established standards is also essential to enhance security. The ISO/IEC 27400:2022 [72] standard provides a comprehensive framework specifically designed for the security needs of IoT devices. This standard emphasises the necessity of systematic security practices, ensuring that protective measures are consistently implemented and maintained across different IoT systems.

Subsequently, employing these standards also enhances the resilience of IoT devices against a spectrum of radio frequency-based threats. By integrating such security frameworks, organisations can safeguard the integrity, availability, and confidentiality of data transmitted between IoT devices. Furthermore, compliance with such standards helps foster trust among users and stakeholders by demonstrating a commitment to security at every level of device operation.

Ultimately, the integration of these standards into the development and deployment of IoT solutions not only mitigates the risk of potential attacks but also increases the robustness and overall reliability of these devices.

9. Limitations

One of the primary limitations of this study is the selection of IoT devices. While we chose a diverse range of devices to represent common categories (Wi-Fi-enabled security camera, Zigbee-enabled thermostat, and 433 MHz smart garage door openers), this selection may not encompass the full spectrum of IoT devices in use today. Future studies should include a broader array of devices, including those with more advanced and varied security features, to provide a more comprehensive analysis of RF vulnerabilities.

Moreover, the experiments were conducted in a controlled laboratory setting, which may not fully replicate real-world environments where IoT devices are typically deployed. Factors such as physical obstructions, varying signal interference from other devices, and environmental noise can significantly affect the performance and security of IoT devices. Conducting similar experiments in more varied environments would help validate the findings and provide more generalisable results.

While the study employed jamming and replay attacks to test the vulnerabilities of IoT devices, there are numerous other RF attack techniques, such as MITM attacks, signal spoofing, and DoS attacks, which were not explored in this work. Including a wider range of attack scenarios could provide a more holistic understanding of the security challenges faced by IoT devices. Additionally, variations in the intensity and frequency of attacks were limited by the capabilities of the SDRs used; more advanced equipment could reveal different outcomes.

Lastly, the IoT devices used in this study were tested with their existing software and firmware versions at the time of the experiment. Updates to software and firmware can significantly alter the security posture of these devices. As such, the results may not be fully applicable to devices that have since received updates addressing the vulnerabilities identified. The ongoing testing and updating of device firmware are necessary to keep pace with evolving threats.

10. Conclusions

The increasing popularity of IoT devices in everyday life highlights their potential to enable seamless connectivity and convenience. However, these benefits come with significant security risks, making IoT devices appealing targets for adversaries. While many solutions have been developed to protect IoT devices from traditional threats such as DoS and MITM attacks, there remains a significant gap in understanding their susceptibility to RF-based attacks. These RF threats are particularly concerning because they can be launched from long distances, allowing attackers to exploit vulnerabilities without detection or tracking.

This research addresses this critical gap by providing a comprehensive analysis of RF-based vulnerabilities in four popular, commercially used IoT devices. By utilising software-defined radio, we uncovered a range of potential risks and demonstrated the susceptibility of these devices using common hardware to perform the attacks. Our results revealed that many IoT devices exhibit a significant vulnerability to jamming attacks, often requiring a comprehensive system reset following a successful hack. The detailed examination of the NEST smoke alarm showed that the use of multiple communication channels, such as Bluetooth and Wi-Fi, increases susceptibility to potential vulnerabilities. Despite the inherent bandwidth limitations of the HackRF, our novel methodology of integrating a laptop's built-in Wi-Fi capability with the HackRF showcased promising prospects for effective exploitation.

In conclusion, this study provides a thorough examination of the vulnerabilities of IoT devices to RF attacks, specifically jamming and replay attacks. Our research is novel in its comprehensive analysis across multiple device types and communication protocols, providing a detailed understanding of the specific weaknesses inherent in each.

These contributions not only fill gaps in the existing literature but also serve as a foundation for future research and development in IoT security. Our findings underscore

the urgent need for more advanced security measures and pave the way for developing IoT devices that are robust against evolving RF threats.

11. Future Work

Given the promising findings from this preliminary study, it is evident that the next logical step is to broaden our experimental scope with the use of more advanced equipment. For instance, during the examination of the Belkin NestCam using software-defined radios, we only managed to achieve a slight disruption to the signal. This suggests the potential for more in-depth assessments, particularly with devices possessing greater bandwidth capabilities, such as the USRP SDR and Ubertooth.

Moreover, considering the vulnerabilities of numerous devices operating on analogue systems, like relays, to magnetic interventions, it becomes imperative to further our research in this direction. Specifically, exploring the application of electromagnetic pulses might offer novel insights into enhancing the range and efficiency of potential disruptions.

Author Contributions: Conceptualization, E.A. and V.I.; Methodology, E.A. and V.I.; Validation, E.A. and V.I.; Investigation, V.I.; Writing—original draft, E.A., L.W. and V.I.; Writing—review & editing, E.A., L.W., V.I. and T.S.; All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The original contributions presented in the study are included in the article, further inquiries can be directed to the corresponding author.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Anthi, E.; Williams, L.; Malgortzata, G.; Theodorakopoulos, G.; Burnap, P. A supervised Intrusion Detection System for Smart Home IoT. *IEEE Internet Things J.* **2018**, *78*, 477–490.
2. Das, R.; Gündüz, M.Z. Analysis of cyber-attacks in IoT-based critical infrastructures. *Int. J. Inf. Secur. Sci.* **2019**, *8*, 122–133.
3. Hamid, H.; Noor, R.M.; Omar, S.N.; Ahmedy, I.; Anjum, S.S.; Shah, S.A.; Kaur, S.; Othman, F.; Tamil, E.M. IOT-based botnet attacks systematic mapping study of literature. *Scientometrics* **2021**, *126*, 2759–2800. https://doi.org/10.1007/s11192-020-03819-5.
4. Sadiq, A.; Anwar, M.; Butt, R.A.; Masud, F.; Shahzad, M.K.; Naseem, S.; Younas, M. A review of phishing attacks and countermeasures for internet of things-based smart business applications in industry 4.0. *Hum. Behav. Emerg. Technol.* **2021**, *3*, 854–864. https://doi.org/10.1002/hbe2.301.
5. Bastille. Mousejack Technical Details. 2016. Available online: https://www.bastille.net/research/vulnerabilities/mousejack/technical-details (accessed on 27 May 2024)
6. Shing, L.; Astacio, J.; Figueroa, A.; Shing, C.C. Vulnerabilities of radio frequencies. In Proceedings of the 2015 12th International Conference on Fuzzy Systems and Knowledge Discovery (FSKD), Zhangjiajie, China, 15–17 August 2015; pp. 2682–2686. https://doi.org/10.1109/FSKD.2015.7382381.
7. Pauli, D. IoT Worm Can Hack Philips Hue Lightbulbs, Spread across Cities. 2016. Available online: https://www.theregister.com/2016/11/10/iot_worm_can_hack_philips_hue_lightbulbs_spread_across_cities/ (accessed on 20 May 2024).
8. Bukhari, S.M.S.; Zafar, M.H.; Abou Houran, M.; Moosavi, S.K.R.; Mansoor, M.; Muaaz, M.; Sanfilippo, F. Secure and privacy-preserving intrusion detection in wireless sensor networks: Federated learning with SCNN-Bi-LSTM for enhanced reliability. *Ad Hoc Netw.* **2024**, *155*, 103407.
9. Rugeles, J.d.J.; Guillen, E.P.; Cardoso, L.S. A Technical Review of Wireless security for the Internet of things: Software Defined Radio perspective. *arXiv* **2020**, arXiv:2009.10171.
10. COX, J. NSO Group Closes Cyprus Office of Spy Firm. 2020. Available online: https://www.vice.com/en/article/ep48kp/nso-group-cyprus-circles-bulgaria-ss7 (accessed on 12 January 2024).
11. AFP. Cyprus Arrests Three in 'Israeli Spy Van' Probe. 2019. Available online: https://www.securityweek.com/cyprus-arrests-three-israeli-spy-van-probe/ (accessed on 12 January 2024).
12. ASP. Cyprus Police Investigate Israeli Owner's 'spy' Van. 2019. Available online: https://apnews.com/general-news-9dd96df20fce433ca86b62228c0016fb (accessed on 12 January 2024).
13. Barnickel, J.; Wang, J.; Meyer, U. Implementing an Attack on Bluetooth 2.1+ Secure Simple Pairing in Passkey Entry Mode. In Proceedings of the 2012 IEEE 11th International Conference on Trust, Security and Privacy in Computing and Communications, TRUSTCOM '12, Liverpool, UK, 25–27 June 2012; pp. 17–24. https://doi.org/10.1109/TrustCom.2012.182.
14. Mottola, L.; Hameed, A.; Voigt, T. Energy Attacks in the Battery-less Internet of Things: Directions for the Future. In Proceedings of the 17th European Workshop on Systems Security, Athens, Greece, 22 April 2024; pp. 29–36.

15. Karagiannis, D.; Argyriou, A. Jamming attack detection in a pair of RF communicating vehicles using unsupervised machine learning. *Veh. Commun.* **2018**, *13*, 56–63. https://doi.org/10.1016/j.vehcom.2018.05.001.
16. Zhang, W.; Yan, X.; Cao, C.; Zeng, X.; Feng, Z.; Ning, J.; Wang, T.; Wu, Z.; Zhang, X. Research on simulation methods for Doppler frequency shift of a coherent inter-satellite laser link in a ground test system. *Infrared Phys. Technol.* **2021**, *113*, 103627. https://doi.org/10.1016/j.infrared.2020.103627.
17. Gul, O.M.; Kulhandjian, M.; Kantarci, B.; Touazi, A.; Ellement, C.; D'amours, C. Secure industrial iot systems via rf fingerprinting under impaired channels with interference and noise. *IEEE Access* **2023**, *11*, 26289–26307.
18. Shahroz, M.; Mushtaq, M.F.; Ahmad, M.; Ullah, S.; Mehmood, A.; Choi, G.S. IoT-Based Smart Shopping Cart Using Radio Frequency Identification. *IEEE Access* **2020**, *8*, 68426–68438. https://doi.org/10.1109/ACCESS.2020.2986681.
19. Hung, P.D.; Vinh, B.T. Vulnerabilities in IoT Devices with Software-Defined Radio. In Proceedings of the 2019 IEEE 4th International Conference on Computer and Communication Systems (ICCCS), Singapore, 23–25 February 2019; pp. 664–668. https://doi.org/10.1109/CCOMS.2019.8821711.
20. Öst, A. Evaluating LoRa and WiFi Jamming. Ph.D. Thesis, Mid Sweden University, Faculty of Science, Technology and Media, Department of Information Systems and Technology. 2018. Available online: https://www.diva-portal.org/smash/record.jsf?pid=diva2%3A1223787&dswid=-4233 (accessed on 12 January 2024).
21. Oukas, N.; Boulif, M.; Abbas, A.; Bahmed, K.; Mosteghanemi, Y. Mitigating Jamming Attacks in IoT RF-Devices through Dynamic Channel Hopping: A Novel Petri-nets Formulation. In Proceedings of the 2023 International Conference on Computer and Applications (ICCA), Yangon, Myanmar, 27–28 February 2023; IEEE: Piscataway, NJ, USA, 2023; pp. 1–6.
22. Rojas, P.; Alahmadi, S.; Bayoumi, M. Physical Layer Security for IoT Communications—A Survey. In Proceedings of the 2021 IEEE 7th World Forum on Internet of Things (WF-IoT), New Orleans, LA, USA, 14 June–31 July 2021; pp. 95–100. https://doi.org/10.1109/WF-IoT51360.2021.9595025.
23. Beal, V. WiFi. 2021. Available online: https://www.webopedia.com/definitions/wifi/ (accessed on 12 January 2024).
24. Electronic Code of Federal Regulations. 2021. Available online: https://www.ecfr.gov/ (accessed on 12 January 2024).
25. IEEE SA—IEEE 802.15.4-2020. Available online: https://standards.ieee.org/ieee/802.15.4/7029/ (accessed on 5 June 2024).
26. Amin, I.; Saeed, A. 5.10 Wireless Technologies in Energy Management. In *Comprehensive Energy Systems*; Dincer, I., Ed.; Elsevier: Oxford, UK, 2018; pp. 389–422. https://doi.org/10.1016/B978-0-12-809597-3.00524-1.
27. Watteyne, T. 4-Lower-power wireless mesh networks for machine-to-machine communications using the IEEE802.15.4 standard. In *Machine-to-machine (M2M) Communications*; Antón-Haro, C., Dohler, M., Eds.; Woodhead Publishing: Oxford, UK, 2015; pp. 63–77. https://doi.org/10.1016/B978-1-78242-102-3.00004-6.
28. Farahani, S. Chapter 2-ZigBee/IEEE 802.15.4 Networking Examples. In *ZigBee Wireless Networks and Transceivers*; Farahani, S., Ed.; Newnes: Burlington, NJ, USA, 2008; pp. 25–32. https://doi.org/10.1016/B978-0-7506-8393-7.00002-9.
29. ScientificAmerican. How Does Bluetooth Work? 2007. Available online: https://www.scientificamerican.com/article/experts-how-does-bluetooth-work/#:~:text=Bluetooth%20technology%20uses%20the%20principles,devices%20that%20are%20actively%20inquiring (accessed on 20 January 2024).
30. Techspirited. A Beginner's Guide to Bluetooth Technology and How Does it Work. 2008. Available online: https://techspirited.com/how-does-bluetooth-work (accessed on 20 January 2024).
31. Chen, J.; Zhang, S.; Wang, H.; Zhang, X. Practicing a record-and-replay system on USRP. In Proceedings of the Second Workshop on Software Radio Implementation Forum-SRIF 13, Hong Kong, China, 12–16 August 2013. https://doi.org/10.1145/2491246.2491257.
32. Andrea, I.; Chrysostomou, C.; Hadjichristofi, G. Internet of Things: Security vulnerabilities and challenges. In Proceedings of the 2015 IEEE symposium on computers and communication (ISCC), Larnaca, Cyprus, 6–9 July 2015; IEEE: Piscataway, NJ, USA, 2015; pp. 180–187.
33. Atzori, L.; Iera, A.; Morabito, G. The internet of things: A survey. *Comput. Netw.* **2010**, *54*, 2787–2805.
34. Mallik, A. Man-In-The-Middle-Attack: Understanding In Simple Words. *Cyberspace J. Pendidik. Teknol. Inf.* **2019**, *2*, 109. https://doi.org/10.22373/cj.v2i2.3453.
35. Punchthrough. Harmonics Part 1-Introduction to Harmonics & BLE. 2020. Available online: https://punchthrough.com/harmonics-part-1-introduction-to-harmonics-ble-2/ (accessed on 12 January 2024).
36. Wi-Fi-Radio Modulation. Tutorials Point. 2011. https://www.tutorialspoint.com/wi-fi/wifi_radio_modulation.htm (accessed on 12 January 2024).
37. RF Wireless World. 2021. Available online: https://www.rfwireless-world.com/Terminology/spurious-vs-harmonics.html (accessed on 20 January 2024).
38. Chairwoman, J.R. 2023. Available online: https://www.fcc.gov/ (accessed on 12 January 2024).
39. CE Marking—Obtaining the Certificate, EU Requirements. Available online: https://europa.eu/youreurope/business/product-requirements/labels-markings/ce-marking/index_en.htm (accessed on 24 May 2024)
40. ecommerceDB. E-Commerce Revenue Analytics belkin.com. Available online: https://ecommercedb.com/store/belkin.com (accessed on 12 January 2024).
41. Alibaba Sonoff Revenue. Available online: https://sonoff.en.alibaba.com/company_profile/transaction_history.html?spm=a2700.shop_cp.13934.2.502d1c9d9tBxHc (accessed on 19 January 2024)

42. Google Nest Protect-Smoke Alarm and Carbon Monoxide Detector (Battery). Available online: https://www.amazon.co.uk/Nest-Protect-Generation-Monoxide-Battery/dp/B00ZC5F9W2 (accessed on 19 December 2023).
43. FCCID. Belkin. NetCam HD F7D7602V2 FCC ID K7SF7D7602V2. 2014. Available online: https://fccid.io/K7SF7D7602V2 (accessed on 19 December 2023).
44. FFCID. Nest Labs Inc Wireless Protect S30. 2014. Available online: https://fccid.io/ZQAS30 (accessed on 19 December 2023).
45. GitHub - merbanan/rtl_433: Program to decode radio transmissions from devices on the ISM bands (and other frequencies). Available online: https://github.com/merbanan/rtl_433 (accessed on 5 June 2024).
46. Shenzhen Sonoff Technologies Co., Ltd. Temperature and Humidity Sensor SNZB-02. 2020. Available online: https://fccid.io/2APN5SNZB-02 (accessed on 19 December 2023).
47. Huang, X.; Liu, D.; Wang, Y.; Chen, P.; Fu, W. 100-MHz Low-Phase-Noise Microprocessor Temperature-Compensated Crystal Oscillator. *IEEE Trans. Circuits Syst. II Express Briefs* **2015**, *62*, 636–640. https://doi.org/10.1109/TCSII.2015.2415652.
48. Nooelec NESDR SMArt v4 SDR-Premium RTL-SDR w/Aluminum Enclosure, 0.5PPM TCXO, SMA Input. RTL2832U & R820T2-Based. Available online: https://www.nooelec.com/store/sdr/sdr-receivers/smart.html (accessed on 19 December 2023).
49. CC2531EMK. Available online: https://www.ti.com/tool/CC2531EMK (accessed on 19 January 2024).
50. How to Sniff Zigbee Traffic. Available online: https://www.zigbee2mqtt.io/advanced/zigbee/04_sniff_zigbee_traffic.html (accessed on 19 January 2024).
51. f4exb. f4exb/Sdrangel: SDR Rx/Tx Software for Airspy, Airspy HF , BladeRF, HackRF, LimeSDR, PlutoSDR, RTL-SDR, SDRplay RSP1 and FunCube. Available online: https://github.com/f4exb/sdrangel (accessed on 19 January 2024).
52. ghostop14. ghostop14/Sparrow-wifi: Next-Gen GUI-based WiFi and Bluetooth Analyzer for Linux. Available online: https://github.com/ghostop14/sparrow-wifi.
53. Koenkk. Koenkk/zigbee2mqtt: Zigbee to MQTT Bridge , Get Rid of Your Proprietary Zigbee Bridges. Available online: https://github.com/Koenkk/zigbee2mqtt (accessed on 19 January 2024).
54. Larsson, B. MerbananRTL_433: Program to Decode Radio Transmissions from Devices on the ISM Bands (and Other Frequencies). 2016. Available online: https://github.com/merbanan/rtl_433 (accessed on 19 January 2024).
55. Pohl, J.; Noack, A. Universal Radio Hacker: A Suite for Analyzing and Attacking Stateful Wireless Protocols. In Proceedings of the 12th USENIX Workshop on Offensive Technologies (WOOT 18), Baltimore, MD, USA, 13–14 August 2018.
56. Universal Radio Hacker (URH): Investigating Wireless Protocols like a Boss. Available online: https://github.com/jopohl/urh (accessed on 5 June 2024).
57. Kanters, K. Z-Stack-Firmware. 2021. Available online: https://github.com/Koenkk/Z-Stack-firmware (accessed on 19 January 2024).
58. Aju, O.G. A survey of zigbee wireless sensor network technology: Topology, applications and challenges. *Int. J. Comput. Appl.* **2015**, *130*, 47–55.
59. Sarker, M.Z.H.; Parvez, M.S. A Cost Effective Symmetric Key Cryptographic Algorithm for Small Amount of Data. In Proceedings of the 2005 Pakistan Section Multitopic Conference, Karachi, Pakistan, 24–25 December 2005. https://doi.org/10.1109/inmic.2005.334435.
60. Advanced Encryption Standard (AES). Available online: https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.197-upd1.pdf (accessed on 27 May 2024).
61. Nannipieri, P.; Di Matteo, S.; Baldanzi, L.; Crocetti, L.; Zulberti, L.; Saponara, S.; Fanucci, L. VLSI design of Advanced-Features AES CryptoProcessor in the framework of the European Processor Initiative. *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.* **2021**, *30*, 177–186.
62. Barker, E.; Barker, W. *Recommendation for Key Management, Part 2: Best Practices for Key Management Organization*; Technical report; National Institute of Standards and Technology: U.S. Department of Commerce, Washington, USA. 2018.
63. National Institute of Standards and Technology. Available online: https://www.nist.gov/ (accessed on 27 May 2024).
64. Marin, L.; Piotr Pawlowski, M.; Jara, A. Optimized ECC implementation for secure communication between heterogeneous IoT devices. *Sensors* **2015**, *15*, 21478–21499.
65. Suárez-Albela, M.; Fernández-Caramés, T.M.; Fraga-Lamas, P.; Castedo, L. A practical performance comparison of ECC and RSA for resource-constrained IoT devices. In Proceedings of the 2018 Global Internet of Things Summit (GIoTS), Bilbao, Spain, 4–7 June 2018; IEEE: Piscataway, NJ, USA, 2018; pp. 1–6.
66. RFC 8031: Curve25519 and Curve448 for the Internet Key Exchange Protocol Version 2 (IKEv2) Key Agreement. Available online: https://www.rfc-editor.org/rfc/rfc8031.html (accessed on 27 May 2024).
67. Torrieri, D. *Principles of Spread-Spectrum Communication Systems*; Springer: Cham, Switzerland, 2005. https://doi.org/10.1007/b99535.
68. IEEE SA-IEEE 802.11-2016. Available online: https://standards.ieee.org/ieee/802.11/5536/ (accessed on 27 May 2024).
69. VOCAL. VOCAL Technologies. 2021. Available online: https://vocal.com/ (accessed on 27 May 2024).
70. Räisänen, O.R. Descrambling Split-Band Voice Inversion with Deinvert. 2017. Available online: http://www.windytan.com/2017/09/descrambling-split-band-voice-inversion.html (accessed on 27 May 2024).

71. Windytan. Windytan/Deinvert: A Voice Inversion Descrambler (and Scrambler). Available online: https://github.com/windytan/deinvert (accessed on 27 May 2024).
72. ISO/IEC 27400:2022-Cybersecurity—IoT Security and Privacy—Guidelines. Available online: https://www.iso.org/standard/44373.html. (accessed on 27 May 2024).