Wireless Communication Attack Using SDR and Low-Cost Devices

Batoul Achaal, Mohamad Rida Mortada, Ali Mansour, Abbass Nasser

Wireless Communication Attack Using SDR and Low-cost Devices

Batoul Achaal[1,2], Mohamad Rida Mortada[2], Ali Mansour[1], and Abbass Nasser[1,2]

[1] Lab-STICC, UMR 6285, ENSTA Bretagne, Brest, France,
[2] American University of Culture and Education, Beirut, Lebanon

Abstract. While Wireless communication (WLC) enhances the user mobility and extends the connection and services to extreme isolated points, is the only way to establish a connection over long distance (mainly earth and space), it has many weak points: Interference, Spectrum limitations, bandwidth cost, various regulations, etc. But the major drawback is the security and risk vulnerabilities. To clarify this idea and highlight several security issues, we are working on the weak points of our wireless networks and communication protocols. In this manuscript, we develop several scenarios of Wireless attacks using simple and low-price equipment. Indeed using Software Defined Radio (SDR), a potential hacker can now access a wide range of wireless-based communication like Keyless entry, GPS and RFID system. It can also interfer and jam several other WLC services and networks.

Keywords: SDR, Wireless communication attack, GPS, RFID

1 Introduction

SDR is a wireless enabling technology that can be used in a variety of applications. Its main goal is to replace analog components and hardwired digital devices -as feasible- in the transceiver, with programmable devices [1].
A potential hacker can now tap into many form of wireless-based communication due to the low cost of SDR [2]. Keyless, Radio frequency identification (RFID) tags, and Global Positioning System (GPS) are all examples under risk. In [3], the authors present the gap in keyless access systems employed by major manufacturers, and their weakness using a variety of equipment, including SDRs and low-cost RF modules. the GPS security remains a critical issue related to the global information infrastructure. Many methods of GPS attack are presented in [4][5][6] based on SDR. Other critical attacks are directed to RFID systems. E-passports, access control, and payment systems are all examples of contactless cards and tags based on RFID systems. Attackers can pick up the radio signals and eavesdrop on the communication between a tag and a reader. With the introduction of SDR, it is now possible to use generic equipment and demodulate in software to jam and attack RFID system. Hereinafter, we mainly focus on WLC attacks that can be conducted using the following platforms:

1. The HackRf One[3] is a SDR peripheral that can transmit and receive radio signals at frequencies ranging from 1 MHz to 6 GHz. The first way to com-

[3] https://www.passion-radio.fr/emetteur-sdr/hackrf-sdr-75.html

municate with the device is by hackrf_transfer program[4]. The second method is using the GNURadio package.

2. Digital Video Broadcasting - Terrestrial (DVB-T) receiver[5] is a SDR radio receiver capable of receiving signals in the frequency band from 24 – 1760 MHz.
3. GNURadio is an open source signal processing software development kit. It can handle a variety of input sources, processing blocks, and output formats.
4. Gqrx[6] is an open source software SDR, powered by the GNURadio. FFT plot, waterfall, and spectrum analyzer modes are among the main features.

This paper discuss the most prevalent WLC attacks, using low-cost SDR devices. The first section "SDR based relay attack" explains how to use SDR to realize replay attacks. The two main categories of GPS-based threats, jamming and spoofing, are demonstrated in the " GPS Jamming & spoofing " section. In the third section, we discuss the ability to jam and eavesdropping RFID using commercial SDR, and finally we proposed an approach for a RFID tag emulator using low-cost devices.

2 SDR based relay attack

This section illustrates the concepts of using SDR in exploitation Cyber Security vulnerabilities, such a: relay attacks when applied to a remote-controlled car for capturing and replaying control signals. The relay attack is a security attack in which a data communication between two parties is intercepted and repeated with or without manipulation regardless the two parties' awareness. In our work, GNURadio and HackRF were used to hack a remote-controlled car. A simple attack may contain two phases. Intercept and record a control signal, and then replaying it later using the HackRF's transmit feature.

2.1 Capturing the control signal

Using a DVB-T & Gqrx or a spectrum analyser, one can analyze the intercepted signals; in our case, we found that the transmission system is using 27.15 MHz as a carrier frequency. GNURadio can be used to intercept the signal and store it in wave format. We identified and stored 4 control signals related to a remote-controlled car, we called these signals: Up - Down, Left and Right.

2.2 Transmission of the control signal

First, a flowgraph for replaying the signal must be created in GNURadio, Figure 1. The "Selector block", and the "QT GUI Range block" allow the transition between commands. "Throttle block" flow of samples such that the average rate does not exceed the specific rate. "Osmocom Sink block" replay the signal

[4] https://github.com/pothosware/PothosCore/wiki/Downloads#windows-installer The Pothos SDR environment includes hackrf_transfer application

[5] https://picclick.fr/DVB-T-Stick-Receiver-DVB-T-FM-DAB-820T2-SDR-Receiver-384147650239.html

[6] https://gqrx.dk/

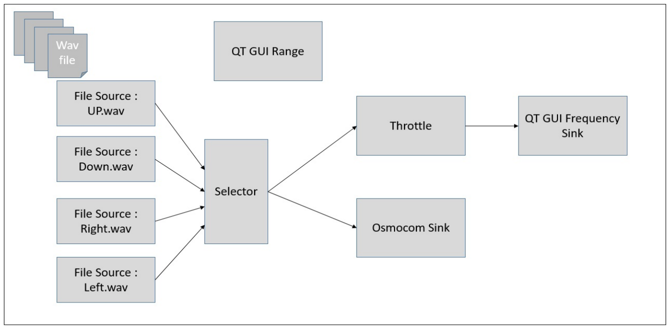


Figure 1: GNURadio flow graph for replaying the signals

through the HackRF. By adjusting appropriate transmission power and other parameters (carrier frequency, bandwidth, etc), we succeeded to overwrite the command of the remote-control and replace it by the signals generated by our GNURadio software and transmitted using the HackRF platform. This kind of scenario can be applied to control a wide variety of drones and remote-controlled devices.

2.3 SDR based relay attack applications

The authors of [7] described replaying attacks on entry keys of cars, the process in there work carried out using SDR tools. The Honda City i-vtec in India was the test subject. The Replay attack was successfully executed in the experiment. We should point out that this type of easy relay attack will only work on older cars that don't have rolling code security. Samy Kamkar, a speaker at the 2015 Def Con conference[7], described how he constructed a device called "RollJam" that can wirelessly break the rolling code protection afforded by wireless entry keys. The basic idea is to jam the signal using a SDR or another RF device, collect the second rolling code after two key pushes, and then replay the first. The attacker now has access to the second unused rolling code, which can be utilized at any time [8].

3 GPS Jamming & spoofing SDR based

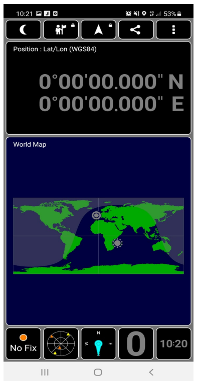
3.1 GPS Background

The global positioning system (GPS) have become ubiquitous in most modern activities, from network time synchronization to locating. GPS was created as a military system. However, due to its broad civilian use, it is subject to jamming and spoofing attacks, which are now possible to anyone with a SDR [4]. Based on data received from several GPS satellites, the GPS receiver estimates its own dimensional position. Each satellite has a precise record of its position and time, which it sends to the receiver [5][6]. The two carrier frequencies are L1 at 1575.42 MHz and L2 at 1227.60 MHz. [9]. In our project, we are mostly concerned with the L1 civilian band.

[7] https://www.youtube.com/watch?v=UNgvShN4USU

3.2 GPS jamming

The intentional interference or jamming GPS is defined in [10] as the release of radio frequency energy of sufficient power and with the correct qualities to prevent receivers in the target area from tracking GPS signals. The implementation of a five GPS jamming techniques using low cost SDR platforms is detailed in [11]. We constructed our platform to test in practical the capacity of SDR in jamming GPS. The HackRF One was used to send a signal developed in GNURadio in order to jam a mobile phone's GPS signal. A "noise source" with Gaussian type, with a sampling frequency of 2 MHz was chosen as the source. The transmission frequency was set at 1.57542 GHz, which is the L1 GPS frequency. The average power spectral density is approximately -50 dBW/Hz .



Position : Lat/Lon (WGS84)

0°00'00.000" N
0°00'00.000" E

Figure 2: As a result of the jamming of initial GPS signal, the "GPS Test app" was unable to identify the location of the target mobile device.

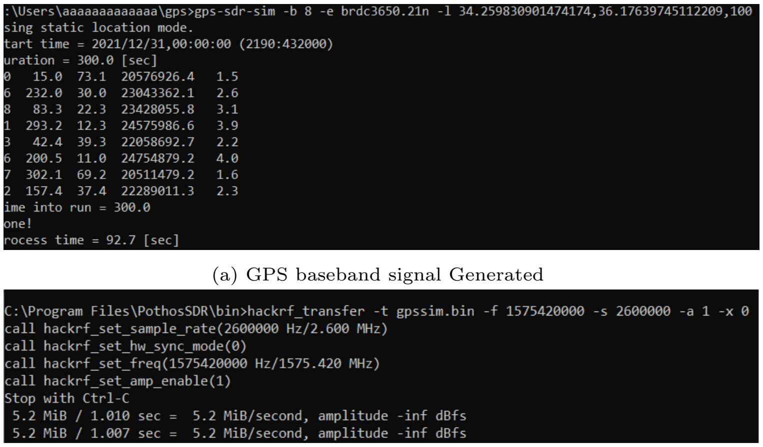
Using the GPS Test app, we test the target phone's GPS location. Figure 2 illustrates the situation in which the app was unable to receives the original GPS signal, as a result, the application became unable to report the location of the mobile device during GNURadio code executing.

3.3 GPS spoofing

While jamming just needs emission of an effective interfering signal and is easily identified by the loss of services, spoofing is more complex since it sends erroneous data to a user expecting to receive reliable data [12][13]. The equipment required to spoof the GPS signal comprises a computer capable of creating the spoofed signal connected to a HackRF. To communicate with the HackRF in this situation, we use the hackrf_transfer[8] utility. A classic GPS spoofing attack consists of two steps: creating a binary transmitting file, and broadcasting a fake GPS signal using the binary file previously created.

[8] We discussed it briefly in the introduction.

GPS-SDR-SIM is an open source projects to create GPS baseband signal, a binary transmission file, that corresponds to the intended fake location, which can be found on GitHub[9]. The code was mainly generated by Takuji Ebinuma [6]. The daily GPS broadcast ephemeris file (BRDC) provides the precise location data of each satellite, allowing receivers to calculate position using prior information in a specific day. BRDC in RINEX format (Receiver Independent Exchange Format), can be downloaded from NASA Archive of Space Geodesy Data[10].



```
:\Users\aaaaaaaaaaaaa\gps>gps-sdr-sim -b 8 -e brdc3650.21n -l 34.259830901474174,36.17639745112209,100
sing static location mode.
tart time = 2021/12/31,00:00:00 (2190:432000)
uration = 300.0 [sec]
0   15.0  73.1  20576926.4   1.5
6  232.0  30.0  23043362.1   2.6
8   83.3  22.3  23428055.8   3.1
1  293.2  12.3  24575986.6   3.9
3   42.4  39.3  22058692.7   2.2
6  200.5  11.0  24754879.2   4.0
7  302.1  69.2  20511479.2   1.6
2  157.4  37.4  22289011.3   2.3
ime into run = 300.0
one!
rocess time = 92.7 [sec]
```

(a) GPS baseband signal Generated

```
C:\Program Files\PothosSDR\bin>hackrf_transfer -t gpssim.bin -f 1575420000 -s 2600000 -a 1 -x 0
call hackrf_set_sample_rate(2600000 Hz/2.600 MHz)
call hackrf_set_hw_sync_mode(0)
call hackrf_set_freq(1575420000 Hz/1575.420 MHz)
call hackrf_set_amp_enable(1)
Stop with Ctrl-C
 5.2 MiB / 1.010 sec =  5.2 MiB/second, amplitude -inf dBfs
 5.2 MiB / 1.007 sec =  5.2 MiB/second, amplitude -inf dBfs
```

(b) Transmitting fake GPS signal

Figure 3: GPS spoofing

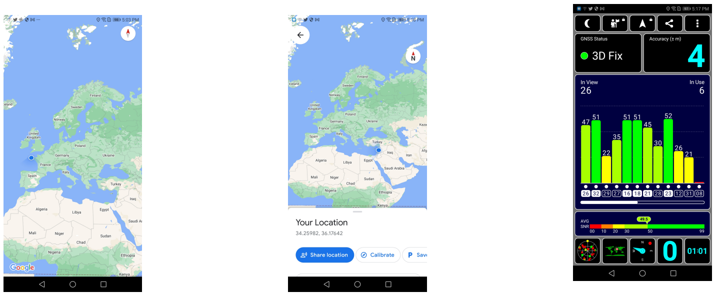
The GPS-SDR-SIM takes the BRDC, and a desired location as input then, it creates a binary file named gpssim.bin 3a, that can be played back on a SDR platform in order to emulate fake GPS signals. For transmitting the signal, we use hack_transfer with gpssim.bin as input data file, carrier frequency of 1575.42 MHz, and a sampling rate = 2.6 MHz, figure3b. The initial position is shown in Figure 4a, which is France. When the HackRF start transmitting, the "Google Map" changed positions point to a fake location Lebanon, Figure 4b. Figure 4c demonstrates the fake satellites viewed by the app.

3.4 GPS attack applications

In [14], the authors apply these methods into DJI Phantom 3 Standard drone. According to the results of their study, GPS jamming and GPS spoofing attacks can cause the drone to lose control by disrupting the GPS signal. In addition, the authors of [15] present a technique for stealing a flying DJI Phantom 3 Standard drone by exploiting civil GPS vulnerabilities. They use the Realtime-generate-

[9] https://github.com/osqzss/gps-sdr-sim

[10] https://cddis.nasa.gov/Data_and_Derived_Products/GNSS/GNSS_data_and_product_archive.html



(a) Initial position, France

(b) Fake position

(c) Satellites viewed by the GPS Test app

Figure 4: Receiving fake GPS signals

fake-GPS-by-joystick[11] project, which allows them to manipulate GPS receivers' reported position in real time using a joystick. Because of its high efficiency and ease of use, they use SDR platforms in all experiments.

4 RFID Jamming and Attacks

4.1 RFID Background

RFID systems work on the principle of using tags to identify items. The tags emit messages that can be read by RFID readers. The majority of RFID tags contain a unique identifier (UID). RFID tags are divided into two types, active and passive based on their electrical power source. Passive tags are introduced in most RFID applications because of their low cost. As a result, we'll concentrate our work on passive tags. In RFID passive applications, the tag (called Proximity IC Card (PICC)) receives the signal from the reader (called Proximity Coupling Device (PCD)) when it falls in its range. Subsequently, PICC stores the extracted energy of the reader PCD transmission. The technique behind is known as inductive coupling [16].

Our goal is to present different ways for RFID jamming and eavesdropping. Then we proposed a method to emulate a tag spoofing the hacked one. We choose to use a RFID-RC522[12] module connected to an Arduino UNO for the demonstration of some part. This module is based on ISO/IEC 14443 Type A 13.56 MHz contactless smart card standard.

4.2 RFID jamming

Reading RFID tag without permission, can lead to privacy issues. To overcome this problem, security-related changes to traditional standards have been made. The authors in [17] suggest a reactive strategy to prohibit unwanted access by artificially interfering with reader commands for a brief period of time to

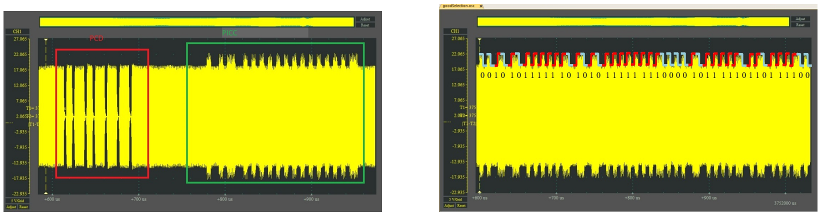
[11] This is a derivative of the GPS-SDR-SIM project.

[12] https://www.gotronic.fr/art-module-rfid-13-56-mhz-tag-rc522-25651.htm#complte_desc

protect such tags. The idea behind their Jamming approach, is to use GNURadio with SDR devices to receive signal from PCD, analyse to detect the RFID activity and then transmit short, arbitrary signal pulses to interfere with the reader command. This approach can be used by attackers in jamming a legitimate passive ultra-high frequency (UHF) tag. Researchers worked on blocking the e-voting system that replaces paper votes with near-field RFID tags, and were successful in prevent the commmunication PCD-PICC[18]. The authors demonstrated that transmitting a powerful signal on 14.408 MHz is enough to block the signal from the PICC. The answer of an ISO14443 PICC is transmitted using load modulation on a subcarrier 847.5 khz. Two side bands at 12.712 MHz and 14.408 MHz are produced. Both sidebands function as data carriers for the tag and are essentially the same, but only the top side band (14.408 MHz) is evaluated by the PCD [18][19].

4.3 RFID Eavesdropping

In [19] the authors describe a platform for recording RFID signals that includes a specific RFID antenna, a DVB-T receiver, an upconverter and a PC with GNURadio software. We rebuilt that platform in order to record the forward and backward signals coming from PCD and PICC respectively. In the next paragraph, we analyzed the signal that has been recorded. ISO/IEC 14443 is a half-duplex block transmission protocol. Request type A & B (REQA, REQB) commands are sent by the PCD to scan the field for tags, while there is no tag. If an Answer To re-Quest (ATQA or ATQB) is received, the procedure to activate card starts. Prior to the transmission of actual data, the standard defines Anticollision phases, which are used to verify that the tag is correctly selected. The PCD activates a specific tag. The PICC responds with a Select AcKnowledge (SAK) instruction, which indicates the tag's type. HALT command is used to deactivate the PICC[20]. The PCD signal and the PICC response, which have a higher amplitude, are shown in Figure 5a.



(a) Part of signal from PCD and PICC

(b) Data transmission (UID)

Figure 5: Recorded Signal

PCD use modified Miller code, while PICC response use Manchester Encoding. Figure 5b shows the modulated carrier for the encoding UID. Manchester encoded signals can be easily decoded. "0" values are represented by falling edges of the signal and "1" values by rising edges. The used binary code is a complement to the Manchester code that has already been read. Each byte is ordered from least significant bit to most significant bit, with a parity bit following each byte (red rectangle in Figure 6). We have successfully extracted the sent UID : 1554C005.

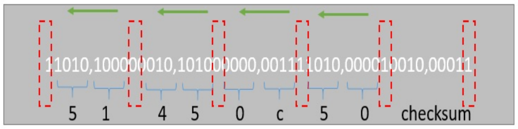


Figure 6: UID extracting : ”1554C005”

4.4 RFID PICC Emulator

Having recovered all the bits of signal including (bits of parity and checksum) and knowing whether the PCD or the PICC is doing the transmission, and understanding the functionality of standard, are all important for tag emulation. The Tag has been programmed to respond to commands based on specific protocol. As a result, simply replaying the recorded signal to make the PCD think the existing PICC in his field is not a choice. Without any external input, the authors of report [21] attempts to simulate both the reader and the tag. The hardware tag emulation implementation was not as successful. In the output, the signal's shape is alternated.

Our goal is to imitate tag in terms of giving his UID and being able to notify his presence in the field of any PCD that works with ISO14443 and communicate with it. Knowing that, the transmission bit-rate of data must be equals to 106 Kbits/s. Therefor, we must modify the load at the receiver coil with a frequency of 106 KHz. Hence, the PIC16F877A microcontroller is used with a clock of 20 MHz which can reach such frequency (106 KHz) by toggling its output pin.

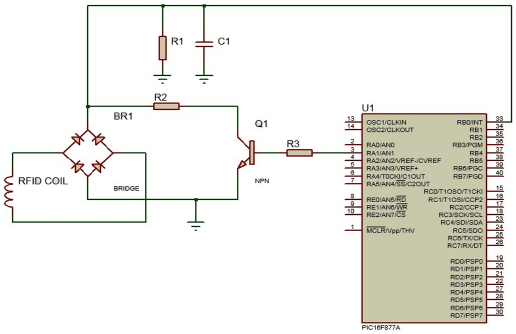


Figure 7: RFID card emulator using PIC16F877A microcontroller

The Figure 7 depicts the simplified circuit of RFID emulator. As we can see, the base of transistor is connected to PORTA 1 of microcontroller, while the coil is connected to a permanent load $R1||C1$ via a full bridge (the bridge is used to match between the AC coming from the coil and the DC coming from the control circuit). When the PORTA 1 is LOW, the transistor is open, thus, the coil faces a load represented by $R1||C1$. Otherwise, if the PORTA 1 is HIGH, then, the transistor is close and the load becomes $R1||R2||C1$ which is less than $R1||C1$. Therefore, in order to represent a digital "1" for example at PORTA 1 in Manchester coding, we must firstly set the PORTA 1 to LOW for $4.7\mu s$ then to HIGH for $4.7\mu s$. In this case, if the coil is at the proximity of a RFID reader, the RFID reader will reveal a change in the electric current due to the change in the load in the secondary coil of the emulator, and thus it will be translated into a digital "1". By reversing the process the digital "0" is represented. In our case, we have succeeded in getting the PCD to read a UID from our emulator as if it were coming from a real PICC. However, the PCD reads a different UID each time, knowing that we always send the same data. Which leads us to suppose that the reader doesn't check the validity of the data received using the checksum field. Some noises and a loss of synchronization between the reader and the emulator leads us to these undesired results.

5 Conclusion

Wireless networks based on radio transmission have a number of vulnerabilities. These can be intercepted, injected, and jammed using low cost devices. In this paper, we gives a quick overview of different WLCs attack scenarios that can be carried out with low-cost hardware. SDR is ideal for a variety of attack and hack operations in different fields, such as keyless entry, GPS, and RFID systems. Finally, We used low-cost hardware to emulate the tag as part of our eavesdropping RFID attack. We were able to get the reader to read a UID from our emulator as though it came from an actual RFID tag. The reader, on the other hand, reads a new UID each time. In the future, we will work to correct the issues.

6 Acknowledgment

The authors are grateful to AID- DGA (l'Agence de l'Innovation de Defense a la Direction Generale de l'Armement – Minitsere des Armees) & ANR (Agence Nationale de le Recherche en France) for supporting our ANRASTRID – Project (ANR-19-ASTR-0005-03).

References

1. Mathew NO Sadiku and Cajetan M Akujuobi. Software-defined radio: a brief overview. *IEEE Potentials*, 23:14–15, 2004.
2. Ali Mansour, Raed Mesleh, and Mohamed Abaza. New challenges in wireless and free space optical communications. *Optics and lasers in engineering*, 89:95–108, 2017.
3. Flavio D Garcia, David Oswald, Timo Kasper, and Pierre Pavlidès. Lock it and still lose it—on the security of automotive remote keyless entry systems. In *25th USENIX Security Symposium*, 2016.

4. Weike Feng, Jean-Michel Friedt, Gwenhaël Goavec-Merou, and François Meyer. Software-defined radio implemented GPS spoofing and its computationally efficient detection and suppression. *IEEE Aerospace and Electronic Systems Magazine*, 36:36–52, 2021.
5. Jonathan A Larcom and Hong Liu. Modeling and characterization of gps spoofing. In *2013 IEEE international conference on technologies for Homeland Security (HST)*, pages 729–734. IEEE, 2013.
6. Kang Wang, Shuhua Chen, and Aimin Pan. Time and position spoofing with open source projects. *black hat Europe*, 148:1–8, 2015.
7. Yash M Kenia. Cyber attacks on smart cars using SDR. 2019.
8. Samy Kamkar. Drive it like you hacked it: New attacks and tools to wirelessly steal cars. *Presentation at DEFCON*, 23, 2015.
9. Shah Zahid Khan, Mujahid Mohsin, and Waseem Iqbal. On GPS spoofing of aerial platforms: a review of threats, challenges, methodologies, and future research directions. *PeerJ Computer Science*, 7:e507, 2021.
10. Transportation Infrastructure. Vulnerability assessment of the transportation infrastructure relying on the global positioning system. *Center, John A. Volpe Nat. Transp. Syst., Tech. Rep*, 2001.
11. Renato Ferreira, João Gaspar, Pedro Sebastião, and Nuno Souto. Effective GPS jamming techniques for UAVs using low-cost SDR platforms. *Wireless Personal Communications*, 115:2705–2727, 2020.
12. Kazim Tugsad Seferoglu and Ahmet Serdar Turk. Review of spoofing and jamming attack on the global navigation systems band and countermeasure. In *2019 9th international conference on recent advances in space technologies (RAST)*, pages 513–520. IEEE, 2019.
13. G Goavec-Merou, JM Friedt, and F Meyer. GPS spoofing using software defined radio. *l'OSU THETA Franche-Comté-Bourgogne*, 2019.
14. Jabang Aru Saputro, Esa Egistian Hartadi, and Mohamad Syahral. Implementation of GPS attacks on DJI phantom 3 standard drone as a security vulnerability test. pages 95–100. IEEE, 2020.
15. Xian-Chun Zheng and Hung-Min Sun. Hijacking unmanned aerial vehicle by exploiting civil GPS vulnerabilities using software-defined radio. *Sensors and Materials*, 32:2729–2743, 2020.
16. Ron Weinstein. RFID: a technical overview and its application to the enterprise. *IT professional*, 7:27–33, 2005.
17. Alexander Bothe, Rene Helmke, and Nils Aschenbruck. Enhancing privacy for passive UHF RFID using an SDR-based reactive jammer. In *2017 IEEE International Conference on RFID Technology & Application (RFID-TA)*, pages 202–207. IEEE, 2017.
18. Yossef Oren, Dvir Schirman, and Avishai Wool. RFID jamming and attacks on israeli e-voting. In *Smart SysTech 2012; European Conference on Smart Objects, Systems and Technologies*, pages 1–7. VDE, 2012.
19. Frédéric Le Roy, Thierry Quiniou, Ali Mansour, Raafat Lababidi, and Denis Le Jeune. RFID eavesdropping using SDR platforms. In *International Conference on Applications in Electronics Pervading Industry, Environment and Society*, pages 208–214. Springer, 2016.
20. NXP Semiconductors. MIFARE ISO/IEC 14443 PICC selection, 2021.
21. Ilias Giechaskiel. Eavesdropping on and emulating MIFARE ultralight and classic cards using software-defined radio, 2015.