RÉPUBLIQUE FRANÇAISE
Liberté Égalité Fraternité

Université de Toulouse

Doctorat de l'Université de Toulouse

préparé à l'INSA Toulouse

Sécurité pour l'Internet des objets: mécanismes matériels d'authentification pour objets communicants

---

Thèse présentée et soutenue, le 28 novembre 2025 par

Lamoussa SANOGO

École doctorale
GEETS - Génie Electrique Electronique,Télécommunications et Santé : du système au nanosystème

Spécialité
MicroNano Systèmes

Unité de recherche
LAAS - Laboratoire d'Analyse et d'Architecture des Systèmes

Thèse dirigée par
Daniela DRAGOMIRESCU et Eric ALATA

Composition du jury
M. Mohamed KAANICHE, Président, CNRS Occitanie Ouest
M. Guy GOGNIAT, Rapporteur, Université de Bretagne Sud
M. Aurélien FRANCILLON, Rapporteur, EURECOM
M. Arnaud VENA, Examinateur, Université de Montpellier
M. Franck COURBON, Examinateur, Ethicronics
M. Gaël LOUBET, Examinateur, INSA Toulouse
Mme Daniela DRAGOMIRESCU, Directrice de thèse, INSA Toulouse
M. Eric ALATA, Co-directeur de thèse, INSA Toulouse

Membres invités
M. Alexandru TAKACS, Co-encadrant, Université Toulouse
M. Nicolas RIVIÈRE, Université de Toulouse

Acknowledgements

First of all, I would like to thank my thesis supervisors, Prof. Daniela Dragomirescu, Dr. Eric Alata, and Dr. Alexandru Takacs, for the trust they have placed in me among the other applicants for this very interesting thesis. Doing a thesis was my first option after my master's degree, so thank you for have given me this incredible opportunity to make this dream come true. As a science enthusiast, this thesis journey has been definitely my greatest learning and professional experience so far. Indeed, this journey has made me grow even more, both on a professional and personal point of view. Our results have led me to present at international scientific conferences in France and abroad, as well as, at numerous national scientific conferences. I have had the chance to meet researchers from all over the world, and have further grown my professional network. Through my work on this exciting thesis topic, I have learned a lot, both on a technical and scientific point of view, and further developed my professional skills. None of this would have been possible if you had not trusted me. THANK YOU.

I would also like to thank you for your excellent supervision throughout this thesis. Thank you for your invaluable contribution. Thank you for sharing with me your many years of experience in scientific research.

I would also like to thank my dear colleague Gaël Loubet for his invaluable contribution to this work. My thanks also go to Taki Eddine Djidjekh, my colleague and PhD student in the same office throughout these years. Thank you for your contribution especially to the design of the single-piece 2-port circular polarization antenna. My acknowledgements also go to Alexandre Monti for his contribution to the design of the SPDT switch, the power detector and the dual linearly polarized antenna.

Obviously, I would like to thank the funding institutions of this thesis, the French National Center for Scientific Research (CNRS), the French Ministry of Higher Education, Research and Innovation (MESRI) and the National Institute for Applied Sciences (INSA) based in Toulouse. I would also like to thank the Laboratory of Systems Analysis and Architecture of the CNRS (LAAS-CNRS), my host laboratory, for providing me all the facilities and expertise required for my research work.

I would like to thank all my colleagues of the LAAS-CNRS and especially all my colleagues of the MINC team and the former TSF team (Now part of the recently-created TRUST team). Considering the amount of work required for successfully completing a thesis, it is a great chance to be conducting it in such wonderful teams.

I couldn't forget to thank my wonderful family I love immensely and whose constant support has always been a huge strength for me. Thank you so much for everything. Grateful, forever!

Sécurité pour l'Internet des objets: mécanismes matériels d'authentification pour objets communicants

Résumé :

Les objets connectés sont très vulnérables d'un point de vue cybersécurité. Les facteurs de vulnérabilité incluent « (1)-une conception fragile d'un point de vue sécurité, les concepteurs ayant tendance à se focaliser davantage sur les services que sur la sécurité, (2)-le mode de communication sans fil, facilite les attaques de type rejoue, imitation, homme du milieu et usurpation d'identité, des attaques devenues trop accessibles grâce à la radiologicielle, (3)-les interactions avec les appareils à travers des mises à jour logiciel, ainsi que la collecte et le partage d'informations à des fins économiques, ces interactions ouvrent la porte à des attaques par firmware corrompu, par exemple ». Cette vulnérabilité fait des objets connectés des cibles privilégiées pour les attaquants qui, en plus, peuvent les utiliser comme point d'entrée vers des infrastructures informatiques plus importantes. Par ailleurs, les attaques visant l'internet des objets vont sans doute se multiplier avec l'arrivé de l'intelligence artificielle qui peut être un puissant assistant pour les attaquants.

Alors que, contrairement aux systèmes informatiques classiques comme les ordinateurs personnels, sécuriser les objets connectés reste un grand défis à cause de leur caractère d'objet à ressources limitées (puissance de calcul, mémoire, énergie), de la forte décentralisation des réseaux engendrés par le mode de communication direct machine-à-machine qui rend difficile, par exemple, le déploiement de solution de type pare-feu, et de l'utilisation d'une multitude de protocoles, entrainant la contrainte d'indépendance aux protocoles pour les solutions de sécurité, d'où le grand intérêt pour la couche physique (PHY), plus précisément les ondes radio dans ce cas.
Ces contraintes liées à l'internet des objets font que les solutions de sécurité traditionnelles utilisées par les systèmes informatiques classiques ne sont pas adaptées aux objets connectés.

Les objectifs de cette thèse visent à proposer des solutions de sécurité originales au niveau de la couche PHY pour permettre aux objets connectés de découvrir leur environnement, de s'y intégrer en toute confiance et de s'y adapter en fonction du nombre d'objets, du débit de données requis, de la portée et de la puissance de transmission nécessaires, ainsi que de l'énergie disponible. Cette adaptabilité à l'environnement doit tenir compte de la mobilité de ces objets, qui doivent pouvoir appartenir à différents réseaux à différents moments, ainsi que de la confiance qu'on peut ou pas avoir dans ces différents réseaux. Il s'agit de trouver de nouveaux mécanismes de sécurité qui tiennent compte des contraintes liées à l'internet des objets : des solutions de sécurité à faible empreinte mémoire, à faible charge de calcul, économes en énergie, génériques et indépendantes des protocoles. Il s'agit également de proposer, en cas d'attaques réussies, des contre-mesures par le biais de techniques de reconfiguration et de réadaptation matériel / logiciel.

Ainsi, ces travaux s'intéressent dans un premier temps à une approche de sécurité proposée dans l'état de l'art, à savoir le fingerprinting dont l'objectif est d'associer à chaque appareil électronique une signature, unique et constante dans le temps, conçue à partir des imperfections inhérentes introduites par les processus de fabrication.
Après avoir mis en évidence les limites et les possibilités du fingerprinting, à travers un cas d'étude pratique utilisant la Densité Spectrale de Puissance (DSP) comme caractéristique de

fingerprinting, ces travaux proposent dans un second temps, une nouvelle approche de sécurité innovante, à travers l'authentification dynamique de messages avec une technique basée sur la Polarization Shift Keying (modulation par déplacement de polarisation). Cette approche de sécurité - à empreinte mémoire légère, indépendante des protocoles et non invasive - consiste à contrôler cryptographiquement la polarisation de l'onde radio émise par un appareil dans le but de transmettre, parallèlement au message principal, des données d'authentification de ce message principal. Les résultats expérimentaux de cette nouvelle méthode d'authentification montrent que cette technique est une approche prometteuse pour sécuriser les communications dans l'internet des objets en se basant sur la couche PHY.

En effet, on peut remarquer que ces deux approches de sécurité - à savoir l'identification par fingerprinting ou l'authentification basé sur la modulation par déplacement de polarisation – s'intéressent principalement au support de transmission c'est-à-dire les ondes électromagnétiques dans ce cas, c'est pourquoi on les qualifie d'approches de sécurité au niveau de la couche physique (PHY-Layer) du modèle OSI (Open Systems Interconnection). Indépendamment du protocole utilisé par un objet communicant, au bout de la chaine, la transmission s'effectue sur les ondes électromagnétiques. Cela fait de la couche PHY une couche très intéressante dans la recherche de solutions de sécurité qui répondent aux exigences de l'internet des objets, en particulier la généricité et l'indépendance aux protocoles. Les solutions proposées sont génériques et indépendantes des protocoles.

Security for Internet of Things: hardware authentication mechanisms for communicating objects

Abstract:

The Internet of Things (IoT) devices are highly vulnerable in terms of cybersecurity. Vulnerability factors include "(1)-weak security design, which focuses more on services than on security, (2)-wireless communication makes attacks such as replay, mimicking, man-in-the-middle and spoofing easier, furthermore these attacks have become too accessible thanks to Software-Defined Radio (SDR), (3)-interaction with devices through firmware updates, as well as collecting and sharing information for economic purposes, these interactions open the possibility for corrupted firmware attack, for example". This vulnerability makes IoT devices prime targets for attackers, who can use them as entry points to larger IT infrastructures. Furthermore, attacks targeting IoT will certainly increase with the emergence of Artificial Intelligence (AI), which can be a powerful assistant for attackers.

While, unlike traditional IT systems - e.g., personal computers - securing IoT devices remains a major challenge, because of their resource-constrained nature, the high decentralization of IoT networks as a result of direct machine-to-machine communication, making it difficult, for example, the deployment of firewall-like solutions, and the use of a multitude of protocols in IoT, making protocol independency a requirement for security solutions, hence the high interest in the physical (PHY) layer, more specifically radio waves in this case.
These IoT-related constraints make the traditional security solutions used by traditional IT systems unsuitable for IoT devices.

The objectives of this thesis are to propose original security solutions at the PHY layer to allow IoT devices to discover their environment, integrate into it with confidence, and adapt to it depending on the number of devices, the required data rate, the required range and transmission power, and the available energy. This adaptability to the environment must take into account the mobility of these devices, which must be able to belong to different networks at different moments, as well as the trust that can or cannot be placed in these different networks. New security mechanisms that take into account IoT-related constraints need to be found: security solutions with a lightweight memory footprint, low computational workload, energy-efficient, generic and protocol-independent. In the case of successful attacks, countermeasures through hardware/software reconfiguration and readaptation techniques are also needed.

Thus, this work focuses firstly on a security approach proposed in the literature, known as fingerprinting, aiming to associate each electronic device with a unique and time-constant signature, built from the inherent imperfections introduced by manufacturing processes.
After highlighting the limitations and opportunities of fingerprinting, through a hands-on application using Power Spectral Density (PSD) as fingerprinting feature, this work proposes a new and innovative security approach, through dynamic authentication of messages using a technique based on Polarization Shift Keying. This security approach - lightweight, protocol-independent and non-invasive - consists in cryptographically controlling the polarization of the radio wave emitted by a device in order to transmit, in parallel with the main message, authentication data for this main message. The experimental results of this new authentication method show that this technique is a promising approach for securing communications in IoT based on the PHY layer.

Indeed, it can be noticed that both of these security approaches - fingerprinting-based identification and polarization shift keying-based authentication - are mainly based on the transmission media, i.e., electromagnetic waves in this case. This is why they are referred to as security approaches at the physical layer (PHY-Layer) of the OSI (Open Systems Interconnection) model. Regardless of the protocol used by the IoT device, at the end of the chain, transmission takes place over electromagnetic waves. This makes the PHY layer a very interesting one in the research for security solutions that meet the requirements of IoT, in particular genericity and protocol independence.
The proposed solutions are generic and protocol-independent.

Table of contents

Acknowledgements 1

Abstract (French) 3

Abstract (English) 5

Table of contents 7

Acronyms 10

List of figures 14

List of tables 18

General Introduction 19

1. Radiocommunication 20
2. From communicating objects to Internet of Things (IoT) 20
3. Why this thesis? 22
4. Organization of the Manuscript 22

Chapter 1: IoT Landscape: Overview and Security 25

1. IoT Market 26
2. IoT Cyberthreats Landscape 28
3. IoT Devices Vulnerability and Security Challenges 30
   3.1. Vulnerability Factors 30
   3.2. Security Challenges 30
   3.3. SHM device: A good example 31
4. Conclusion 32

Chapter 2: Context of the work and State-of-art 33

1. Context 34
2. Objective of this thesis 34
3. State-of-the-art 35
4. Threat Model 38
5. Conclusion 40

Chapter 3: Robustness of the PSD-based Devices Fingerprinting 41

1. Introduction 42
2. Power Spectral Density (PSD)-based Fingerprinting 43
   2.1. Related Work 43
   2.2. Analytic Development 43
      2.2.1. PSD and Correlation Function 43
      2.2.2. PSD through a channel 43
   2.3. Methodology 45
      2.3.1. Aims of This Work 45
      2.3.2. Experiments Overview 45
      2.3.3. PSDs Normalization and Comparison 47

2.4. Experiments and Results 49
2.4.1. Experimental Setup 49
2.4.2. Results 54
2.5. Discussion 63
3. Conclusion 65

Chapter 4: Polarization Shift Keying (PoSK)-based Message Authentication System 67

1. Introduction 68
2. PoSK-based Message Authentication System 68
2.1. Overview 68
2.2. Architecture design of Linear Polarizations Shift Keying 69
2.3. Architecture design of Circular Polarizations Shift Keying 71
3. Authentication Approach 73
3.1. Authentication Approach in Synchronization-required architectures 73
3.2. Time Synchronization Technique between PoSK and Modulation symbols in order to mitigate PoSK impact on the Bit Error Rate (BER) 73
3.3. Authentication Approach in Synchronization-free architectures 75
4. Experimentation 75
4.1. Software-Defined Radio (SDR) 75
4.2. Experimentation tools 79
4.2.1. Wireless Sensor and Security Gateway circuits 79
4.2.2. Power Detection 79
4.2.3. Single Pole Double Throw (SPDT) switch 80
4.2.4. Splitter 81
4.2.5. Phase Shifter 81
4.2.6. Cryptographic Polarization Selector 82
4.2.7. Linear Polarization Antenna 82
4.2.8. Circular Polarization Antenna 83
4.2.9. Universal Radio Hacker (URH) 84
4.2.10. GNU Radio 85
4.2.10.1. 2-ASK demodulation 85
4.2.10.2. 2-FSK demodulation 86
4.2.10.3. BPSK demodulation 87
4.3. Experimental setup 88
4.4. Cryptographic polarization selection algorithm 90
4.5. Linear Polarizations Shift Keying Experimentation 91
4.5.1. Synchronization-required architecture 91
4.5.2. Synchronization-free architecture 96
4.5.3. Software-Defined Cryptographic Polarization Controller (CPC) 100
4.6. Circular Polarizations Shift Keying Experimentation 104
4.6.1. Experimentation with Hardware CPC 104
4.6.2. Experimentation with Software-Defined CPC 108
4.7. Assessment of the robustness of the PoSK-based message authentication system ... 115
5. PoSK-based Message Authentication System Experimentation in real data transmission: Image, Video and File transmission 118
6. Impact of PoSK on the communication quality: Bit Error Rate (BER) / Packet Error Rate (PER) 120

6.1. Experimental Setup 121
6.2. Results 122
6.2.1. BPSK 122
6.2.2. Improved PER using Power Squelch 125
6.2.3. 2-ASK 126
6.2.4. 2-FSK 126
6.3. Criteria for choosing polarization keying frequency 130
7. Energy Consumption 131
8. Discussion 133
8.1. Linear vs Circular Polarizations Shift Keying 133
8.2. Synchronization-required vs Synchronization-free Architectures 133
8.3. Software-Defined Cryptographic Polarization Controller 133
8.4. Undesired Polarization Sensitivity 135
8.5. Polarization Switching Time 136
8.6. Inter-Polarization Cross-talk 137
9. Conclusion 137

General Conclusion and Perspectives 141

List of contributions 143

References 145

Appendix 1: Theoretical Model of PoSK 151

Appendix 2: BER as a Function of Polarization Keying Frequency 152

Acronyms

A

AAS – Active Antenna System
ABP – Activation by Personalization
ADC – Analog-to-Digital Converter
AES – Advanced Encryption Standard
AGC – Automatic Gain Control
AI – Artificial Intelligence
AN – Actuator Node
AR – Axial Ratio
ASCON – A new standard for lightweight authenticated encryption and hashing, announced in 2023 by the National Institute of Standards and Technology (NIST).
ASIC – Application-Specific Integrated Circuit
ASK – Amplitude Shift Keying

B

BER – Bit Error Rate
BLE – Bluetooth Low Energy
BPSK – Binary Phase Shift Keying
BYOD – Bring Your Own Device

C

CAGR – Compound Annual Growth Rate
COTS – Commercial-Off-The-Shelf
CPC – Cryptographic Polarization Controller
CRM – Cardiac Rhythm Management

D

DAC – Digital-to-Analog Converter
DBSCAN – Density-Based Spatial Clustering of Applications with Noise
DC – Direct Current
DDoS – Distributed Denial of Service
DNS – Domain Name System
DoS – Denial of Service
DR – Detection Rate
DSP – Digital Signal Processor

E

EEG – ElectroEncephaloGraph

F

FAR – False-Alarm Rate
FIR – Finite Impulse Response
FM – Frequency Modulation

FPGA – Field Programmable Gate Arrays
FSK – Frequency Shift Keying

G

GaAs – Gallium arsenide
GD – Gateway Device
GMSK – Gaussian Minimum-Shift Keying
GNSS – Global Navigation Satellite System
GPIO – General Purpose Input / Output
GPS – Global Positioning System
GSM – Global System for Mobile Communications

H

HFSS – High Frequency Structure Simulator (A software from Ansys)
HIDS – Host-based Intrusion Detection System

I

IC – Integrated Circuit
IDS – Intrusion Detection System
IIoT – Industrial IoT
IIR – Infinite Impulse Response
IoT – Internet of Things
IP – Internet Protocol
IPS – Intrusion Prevention System
I/Q – In-phase / Quadrature
ISM – Industrial, Scientific, Medical
IT – Information Technology

K

kbps – kilobits per second

L

LAAS-CNRS – Laboratory of Systems Analysis and Architecture (LAAS) of the National Center for Scientific Research (CNRS)
LAN – Local Area Network
LFSR – Linear Feedback Shift Register
LHCP – Left-Hand Circular Polarization
LoRa – Long Range
LoRaWAN – Long Range Wide Area Network
LOS – Line Of Sight

M

M2M – Machine to Machine
MAC – Media Access Control
MD – Malicious Device
MitM – Man-in-the-Middle

MMIC – Monolithic Microwave Integrated Circuit

N

NB-IoT – Narrowband Internet of Things
NIDS – Network-based Intrusion Detection System

O

OOT – Out-Of-Tree
OpenBTS – Open Base Transceiver Station (An open-source software-based GSM access point, i.e., GSM base station)
OS – Operating System
OSI – Open Systems Interconnection
OTAA – Over The Air Activation

P

P2P – Peer-to-Peer
PC – Personal Computer
PCB – Printed Circuit Board
PHY – Physical
PHY-Layer – Physical Layer
PIFA – Planar Inverted-F Antenna
PIN – Positive Intrinsic Negative
PMU – Power Management Unit
PoSK – Polarization Shift Keying
PSD – Power Spectral Density
PSK – Phase Shift Keying

Q

QAM – Quadrature Amplitude Modulation
QPSK – Quadrature Phase Shift Keying

R

RAM – Random-Access Memory
RF – Radio-Frequency
RFIC – Radio Frequency Integrated Circuit
RF-DNA – Radio Frequency - Distinct Native Attribute
RF-PUF – Radio Frequency - Physical Unclonable Functions
RHCP – Right-Hand Circular Polarization
RIS – Reconfigurable Intelligent Surface
RMS – Root Mean Square
RTM – Remote Tank Monitoring
Rx – Reception

S

SCADA – Supervisory Control And Data Acquisition
SDA – Software-Defined Antenna

SDR – Software-Defined Radio
SHM – Structural Health Monitoring
Sigfox – A global Low-Power Wide-Area (LPWA) networking protocol
SMA – SubMiniature version A (A category of RF connector)
SN – Sensing Node
SoC – System-on-Chip
SPDT – Single Pole Double Throw

T

Tx – Transmission

U

UHD – USRP Hardware Driver
ULD – Unit Load Device
URH – Universal Radio Hacker
USB – Universal Serial Bus
USRP – Universal Software Radio Peripheral

V

VGA – Variable Gain Amplifier
VNA – Vector Network Analyzer

W

Wi-Fi – Wireless Fidelity
WPT – Wireless Power Transfer
WSN – Wireless Sensor Network

X

XOR – Exclusive OR logic operation

Z

ZigBee – A low-power, low-data-rate, IEEE 802.15.4-based technology for wireless Personal Area Networks (PAN).
Z-Wave – A low-power, interoperable, wireless communications technology designed specifically for residential and commercial building automation.

List of figures

Figure 1. An illustrative image of an IoT environment. Source: https://www.indiamart.com/proddetail/iot-wireless-sensor-network-solution-service-26086465291.html?mTd=1. ……..……..……..…. ***26***

Figure 2. (A) – Replay attack experimental setup: **(a)** setup with RN2483 LoRa transceiver as attacker device, **(b)** setup with HackRF One as attacker device, **(c)** diagram of replay attack process. **(B)** - LoRa transceiver serial interface displaying captured frame (left) and gateway ChirpStack network application with original and duplicated frames (right). …… ***32***

Figure 3. Threat model illustration in an IoT environment. SN = Sensing Node; MD = Malicious Device; GD = Gateway Device; AN = Actuator Node; PC = Personal Computer; cx = Channel x (x=1, 2, 3, 4, 5, 6, 7, 8, 9)…….. ***39***

Figure 4. Illustration of in-situ and ex-situ fingerprintings. …… ***42***

Figure 5. Comparison of two different PSDs before and after amplitude normalization. Parameter d is frequency-dependent and is given by $d = |PSD_1^*(f) - PSD_2^*(f)|$, where * denotes the normalized amplitude. In this figure, d is shown at $f = 2.4255$ GHz as an example. …… ***47***

Figure 6. (a) The different blocks making up the BLE module (source: [72]). **(b)** The BLE devices used in the PSD-based fingerprinting experiments. These devices were designed in the context of the same project; they are different versions of the same product whose hardware and software architectures have evolved over time. Thus, v1 (the board at the left end) refers to the first ever version; v2 (the two boards in the middle) and v3 (the two boards at the right end) refer to the second and third versions, respectively. …….. ***51***

Figure 7. PSD-based fingerprinting experimental setup in the anechoic chamber. The BLE device emits the signal at 2 m from the RSA306B. The latter is equipped with a BLE antenna and driven by the Tektronix SignalVu-PC software. This way, we can capture the BLE signal in real-time and record it as IQ samples on the PC in order to be used later in a python script for estimating the power spectral density using Welch's average periodogram method. …… ***52***

Figure 8. Schematic of the PSD-based fingerprinting experimental setup. …… ***52***

Figure 9. Profile of the power consumption of one of our BLE devices during advertising, i.e., the broadcast of the advertising packet on each of the three primary advertising channels, one after another. Each peak corresponds to the advertising on a channel. ……..…….. ***54***

Figure 10. Twenty PSDs of a same BLE device in a static experimental setting. The region between the red lines is the band B where the PSDs are compared; we chose B = 2 MHz, which is the channel bandwidth of the BLE. So, we ignored the ambient noise to have a more relevant analysis. ……. ***55***

Figure 11. (a) PSDs of anechoic chamber experiment **(b)** PSDs of open-space experiment. …… ***57***

Figure 12. PSDs of the same BLE device but measured with two different identifiers. …… ***59***

Figure 13. PSDs of four different BLE devices using the same identifier (the one of dev48) and, thus, always transmitting exactly the same data. Graph (**a**) plots are an overlapping of graph (**b**) four groups of plots……... ***61***

Figure 14. PSDs of dev48 (green plots) and devCA (brown plots) BLE devices using the same identifier (the one of dev48) and, thus, always transmitting exactly the same data. …….. ***62***

Figure 15. Overview of the PoSK-based message authentication system. …… ***69***

Figure 16. Architectures of the linear PoSK-based message authentication system: **(a)** - Synchronization-required architecture; **(b)** - Synchronization-free architecture. ……..…… ***69-70***

Figure 17. Architectures of the circular PoSK-based message authentication system: **(a)** - Synchronization-required architecture; **(b)** - Synchronization-free architecture. ……. ***71***

Figure 18. Illustration of the time offset between the beginning of transmission and the beginning of PoSK……..…….……..... ***74***

Figure 19. Software radio architecture. …….. ***76***

Figure 20. USRP B210. ……..…….. ***77***

Figure 21. HackRF One. …….. ***78***

Figure 22. 2.45 GHz RF Power detector. ……..…… ***79***

Figure 23. Single Pole Double Throw (SPDT) Switch. ……..…….. ***80***

Figure 24. (a): Power Splitter/Combiner ZX10R-14-S+; **(b)**: Q-053-90, a 90 Degree Hybrid Coupler from ET Industries. ……..…..…….. ***81***

Figure 25. The evaluation board EV1HMC647ALP6 of the Analog Devices HMC647ALP6E phase shifter. The diagrams are from the HMC647ALP6E datasheet. ……..……..……. ***81***

Figure 26. 2.45 GHz dual-linearly polarized antenna. ……..…….. ***82***

Figure 27. 2-port 2.45 GHz RHCP and LHCP -enable antenna. **(a)**: Image of one of the fabricated antennas. **(b)**: Return Loss of the 2 ports of the antenna, measured using a VNA. **(c)**: Realized Gain in LHCP mode, from Ansys HFSS simulation of the antenna. **(d)**: Realized Gain in RHCP mode, from Ansys HFSS simulation of the antenna. NOTE: The gap between the m1 and m2 markers is actually almost identical in both cases (c) and (d), at just over 11 dB. ……..……..…… ***83***

Figure 28. URH emission (Tx) and reception (Rx) processes. …… ***85***

Figure 29. 2-ASK demodulator GNU Radio flowgraph. …….. ***86***

Figure 30. 2-FSK demodulator GNU Radio flowgraph. ……..…… ***87***

Figure 31. BPSK demodulator GNU Radio flowgraph. …… ***88***

Figure 32. PoSK-based message authentication system experimental setup overview. ……..…… ***89***

Figure 33. Architecture of the LFSR used in the PoSK experimentation in this work. …… ***90***

Figure 34. Linear PoSK; Synchronization-required hardware-oriented architecture; Transmission with PoSK; Signals received by the Security Gateway: **[a]** – Received 64 kHz / 128 kHz FSK-modulated analog signal; **[b]** – The demodulation of the 64 kHz / 128 kHz FSK-modulated analog signal shown in [a]; **[c]** – The instantaneous power of the 64 kHz / 128 kHz FSK-modulated analog signal shown in [a]. ……..…… ***93***

Figure 35. Linear PoSK; Synchronization-required hardware-oriented architecture; Transmission in cross-polarization V to H; Signals received by the Security Gateway: **[a]** - Received 64 kHz / 128 kHz FSK-modulated analog signal; **[b]** - The instantaneous power of the 64 kHz / 128 kHz FSK-modulated analog signal shown in [a]……..…… ***94***

Figure 36. Linear PoSK; Synchronization-required hardware-oriented architecture; Reception with a circular polarization antenna (shown on Figure 37) of the signal emitted with and without PoSK; Signals received by the Security Gateway: **[a]** - Received 64 kHz / 128 kHz FSK-modulated analog signals; **[b]** - The demodulations of the 64 kHz / 128 kHz FSK-modulated analog signals shown in [a]; **[c]** - Post-filtering version of the signals shown on [b]: to reduce the risk of bit errors due to PoSK, a low-pass filter with a cut-off frequency at least equal to the main message symbols rate (F_S) can be used to clean the signal before binary slicing. …… ***95***

Figure 37. Linear PoSK experimentation: the circular (LHCP) polarization antenna used by the Security Gateway, which led to the results shown on Figure 34(c). For the green plot of the radiation pattern, the reference emitting antenna is at 0° angle relative to the circularly-polarized antenna, and 90° angle for the brown plot. ……. ***96***

Figure 38. Linear PoSK experimentation: Synchronization-free hardware-oriented architecture (Figure 16(b)) Security Gateway. The port P2 was connected to another SDR device, an USRP B210, as the HackRF One has only one port. …… ***97***

Figure 39. Linear PoSK, Synchronization-free hardware-oriented architecture, Signals received by the Security Gateway, Transmission with PoSK, Main message received on port P1: **[a]** - Received 64 kHz / 128 kHz FSK - modulated analog signal; **[b]** - The demodulation of the 64 kHz / 128 kHz FSK-modulated analog signal shown in [a]; **[c]** - The instantaneous power of the 64 kHz / 128 kHz FSK-modulated analog signal shown in [a]. …….. ***98***

Figure 40. Linear PoSK, Synchronization-free hardware-oriented architecture, Signals received by the Security Gateway, Transmission with PoSK, Authentication code received on port P2 (see Figure 16(b)): **[a]** - Analog raw authentication signal; **[b]** - The instantaneous power of the analog authentication signal shown in [a]. ……. ***99***

Figure 41. Linear PoSK, Synchronization-free hardware-oriented architecture, Signals received by the Security Gateway, Transmission with PoSK, Main message received on port P1: **[a]** - Received 50% / 100% ASK - modulated analog signal; **[b]** - The demodulation of the 50% / 100% ASK-modulated analog signal shown in [a]…… ***99***

Figure 42. Linear PoSK, Synchronization-free hardware-oriented architecture, Signals received by the Security Gateway, Transmission with PoSK, Main message received on port P1: **[a]** - Received -90° / +90° PSK -modulated analog signal; **[b]** - The demodulation of the -90° / +90° PSK-modulated analog signal shown in [a]. …… ***100***

Figure 43. Linear PoSK-based message authentication system, Synchronization-free architecture, Experimentation with Software-Defined Cryptographic Polarization Controller. ……..…… ***101***

Figure 44. GNU Radio flowgraphs for linear PoSK-based message authentication system, Synchronization-free architecture, Experimentation with Software-Defined Cryptographic Polarization Controller. **(a)**: Wireless Sensor flowgraph. **(b)**: Security Gateway flowgraph. …… ***102***

Figure 45. Linear PoSK, Synchronization-free architecture experimentation with Software-Defined Cryptographic Polarization Controller, Authentication code and Main message signals. …….. ***104***

Figure 46. Circular PoSK experimentation, Synchronization-free hardware-oriented architecture: **(a)**: The Wireless Sensor; **(b)**: The Security Gateway configured to receive on only one of the two polarizations, the LHCP (port P2 on Figure 17(b)), i.e., the Security Gateway of Figure 17(b) without the RHCP part, and without the port P1. ……...……. ***106***

Figure 47. Circular PoSK, Synchronization-free hardware-oriented architecture, Signal received by the Security Gateway on its port P2, while the Wireless Sensor, when transmitting the message, was performing PoSK using the evaluation boards of the HMC647ALP6E phase shifter (EV1HMC647ALP6). **[a]** - Received 64 kHz / 128 kHz FSK-modulated analog signal; **[b]** - The instantaneous power of the 64 kHz / 128 kHz FSK-modulated analog signal shown in [a]. The experimental setup of this experiment is shown on Figure 46. ……..…… ***107***

Figure 48. Circular PoSK-based message authentication system, Synchronization-free architecture, Experimentation with Software-Defined Cryptographic Polarization Controller. ……..…… ***108***

Figure 49. Circular PoSK experimentation, Synchronization-free software-oriented architecture: **(a)** - the single-piece 2-port circular polarization antenna; **(b)** - Security Gateway experimental setup in the synchronization-free software-oriented architecture (the Cryptographic Polarization Controller is software-defined). The two cables connecting the antenna to the USRP B210 must be the same length to avoid an undesired offset in the phase shift between the two ports. ……..…… ***109***

Figure 50. GNU Radio flowgraphs for circular PoSK-based message authentication system, Synchronization-free architecture, Experimentation with Software-Defined Cryptographic Polarization Controller. **(a)**: Wireless Sensor flowgraph. **(b)**: Security Gateway flowgraph. ……..…… ***109-110***

Figure 51. Circular PoSK, Synchronization-free software-oriented architecture, Signals received by the Security Gateway, Transmission with PoSK, Main message signal received on port P1: **[a]** - Received 64 kHz / 128 kHz FSK -modulated analog signal; **[b]** - The demodulation of the 64 kHz / 128 kHz FSK-modulated analog signal shown in [a]; **[c]** - The instantaneous power of the 64 kHz / 128 kHz FSK-modulated analog signal shown in [a]……..…… ***114***

Figure 52. Circular PoSK, Synchronization-free software-oriented architecture, Signals received by the Security Gateway, Transmission with PoSK, Authentication code received on port P2 (see Figure 17(b)): **[a]** - Analog raw authentication signal; **[b]** - The instantaneous power of the analog authentication signal shown in [a]. …… ***115***

Figure 53. Linear PoSK experimentation; Assessment of the robustness of the PoSK-based message authentication system, Wireless Sensor experimental setup. ……..…… ***116***

Figure 54. First 10 bits of the demodulation of the main message for different values of the polarization keying frequency (F_{PoSK}), The demodulations of the 500 kHz / 1 MHz FSK-modulated analog signals. The symbol rate is 100 kbaud. The bit rate is 100 kbps. ……..…… ***117***

Figure 55. First 10 bits of the demodulation of the main message for different values of the polarization keying frequency (F_{PoSK}), Post-filtering version of the signals shown on Figure 54: to reduce the risk of bit errors due to PoSK, a low-pass filter with a cut-off frequency at least equal to the main message symbols rate (F_S) can be used to clean the signal before binary slicing, here a low-pass filter with 400 kHz of cut-off frequency was applied to them. The symbol rate is 100 kbaud. The bit rate is 100 kbps. ……..…… ***118***

Figure 56. Linear PoSK; Synchronization-free software-oriented architecture; GNU Radio flowgraphs structure for real data transmission. **(a)** – Wireless Sensor; **(b)** – Security Gateway. …… ***119-120***

Figure 57. Experimental setup for studies of the impact of PoSK on the PER. ……..…… ***121***

Figure 58. The PER as a function of polarization keying frequency, in BPSK-modulated message transmission, experimented for three different BPSK modulation symbol rates. The PER falls to 0% definitely at the frequencies highlighted in color. ……..……..…… ***123***

Figure 59. BPSK-modulated message reception at 31.25 kbaud symbol rate, when it was transmitted with **(a)**: a prohibited polarization keying frequency (10 kHz) related to the symbol rate and **(b)**: an authorized polarization keying frequency (100 kHz) related to the symbol rate. ……..…… ***124***

Figure 60. The PER as a function of polarization keying frequency, in BPSK-modulated message transmission, experimented for three different BPSK modulation symbol rates. Power Squelch was added to improve the PER. The PER falls to 0% definitely at the frequencies highlighted in color. ……..…… ***125***

Figure 61. The PER as a function of polarization keying frequency, in 2-FSK-modulated message transmission, experimented **(a)**: for three different FSK modulation symbol rates, with a unique FSK deviation (50 kHz); and **(b)**: for five different FSK deviations, with a unique symbol rate (125 kbaud). The PER falls to 0% definitely at the frequencies highlighted in color. …… ***127***

Figure 62. (a) – Power consumption of the custom-made SPDT switch shown on Figure 23; **(b)** – Power consumption of the evaluation board EV1HMC647ALP6 shown on Figure 25. ……..…… ***132***

Figure 63. The BER as a function of polarization keying frequency F_{PoSK}; **(a)** BPSK-modulated message transmission. **(b)** 2-FSK-modulated message transmission. …… ***152***

List of tables

Table 1. Summary of the PSD-based fingerprinting experiments. ……. ***46***

Table 2. Configuration parameters of the RSA306B real-time spectrum analyzer. ……..…… ***54***

Table 3. Results of the comparison of the 20 PSDs. ……. ***56***

Table 4. Results of the comparison of the average PSDs of anechoic chamber experiments with the ones of open space experiments for each BLE device. ……..……...…… ***58***

Table 5. Results of the comparison of the PSDs of the same device transmitting two different identifiers. …..... ***60***

Table 6. Results of the comparison of the PSDs of four BLE devices transmitting the same data. …….. ***62***

Table 7. Comparative table of hardware-oriented and software-oriented PoSK-based message authentication system architectures. ……...…...…….. ***135***

General Introduction

1. Radiocommunication

Radiocommunication is a communication technique that uses hertzian waves (also known as radio waves) as a transmission medium. A complete history of the advent of radiocommunications would be a long story to tell. But we cannot skip over mentioning a few major scientific figures whose discoveries led to the emergence of radiocommunications. In 1865, physicist James Clerk Maxwell published the final form of his famous equations [1], through which he mathematically demonstrated the existence of electromagnetic waves, thereby establishing the theoretical foundation for all radio technology. Later, in 1887, physicist Heinrich Hertz provided the first experimental proof of Maxwell's theories by setting up an experimental setup to generate and detect hertzian waves. Thus, he experimentally proved the existence of these waves and their ability to travel through the air. In honor of Heinrich Hertz, radio waves are called "hertzian waves".

In 1901, physicist and electrical engineer Guglielmo Marconi achieved the first transatlantic radio signal transmission. He installed a transmitting antenna in Poldhu, Cornwall, England, then traveled to Signal Hill, St. John's Newfoundland, Canada to install a receiving station: the first signals across the Atlantic (3400 km) were successfully received. This also demonstrated that radio waves can follow the curvature of the Earth, which makes global radiocommunication possible.

Another major contribution to radiocommunications is Friis' well-known transmission equation, one of the most fundamental equations in radiocommunications, presented first by radio engineer Harald T. Friis in 1946. The Friis Transmission Equation is used to calculate the power received from one antenna (with gain G_r), when transmitted from another antenna (with gain G_t), separated by a distance d, and operating at frequency f or wavelength λ. The Friis equation is defined by:

$$P_r = P_t G_t G_r (\frac{\lambda}{4\pi d})^2 \qquad (1)$$

Where P_r is the available power at the receive antenna terminals and P_t is the power delivered to the terminals of an isotropic transmit antenna. In relation (1), these two powers are expressed in Watts.

2. From communicating objects to Internet of Things (IoT)

A communicating object refers to any electronic object, other than traditional computing systems, capable of processing and transmitting information. A connected oximeter is a good example of a communicating object. This small device measures the level of oxygen in the blood, processes this information, and transmits it to a phone for example. This phone can then send this information to a server in the cloud and/or display it on a local mobile application.
Unlike a traditional computing system such as a laptop, a communicating object is a low-resource system, i.e., it has limited computation power, limited memory capacity, and a limited amount of available energy.
Generally speaking, a communicating object is specialized for a specific task (e.g., measuring and transmitting physiological data) with a small amount of information to process. Another

example of a communicating object is an electronic car key that locks and unlocks the car doors at the simple press of a button.

Sometimes, these are ordinary objects that exist since a long time ago, to which manufacturers now add communication capabilities to make them even more convenient to use. The oximeter has been around for several decades now, while the connected oximeter is more recent.

When it comes to communication modes, the most commonly used by communicating objects is the wireless communication, more specifically radiocommunication through radio waves, introduced in the previous section. The choice of this communication medium is mainly motivated by the comfort offered by this way of communicating.

The Internet of Things (IoT) is basically the internet of communicating objects. The IoT aims to make objects communicating so they can exchange information between themselves. This allows for the creation of local networks of communicating objects. These local networks can then be connected to the global internet.

A pertinent example of an IoT environment is a smart home. Consider a residence illuminated with light bulbs that can be turned on and off via Wi-Fi or Bluetooth from a mobile application. The thermostat of this residence is connected, allowing you to set the temperature from the same mobile application, which also allows you to visualize the current temperature of the residence in real time. The laundry machine is connected, so you can access all its operating options from the mobile application, it also alerts you through the mobile application when the wash cycle is complete. The oven is connected, so you can start it, stop it, adjust the power, set the timer, and so on from the mobile application. The fridge is connected, so you can adjust its temperature from the mobile application. The residence is equipped with smoke and motion detectors that send their information to the mobile application. Connected surveillance cameras allow you to watch a live video of the residence on demand via the application. Access to the residence is possible via the application. The residents are equipped with connected medical devices that measure their physiological parameters and send this data to the mobile application. And so on ... The list of things that can be made communicating in a residence is still long. Nowadays, we are trying to make everything communicating, the term "Internet of Everything (IoE)" is already in use.

In the case of the smart home example, by making the mobile application accessible via the global internet, it is possible to access and control all of the connected objects in the residence from anywhere in the world with a simple internet access. This is the idea behind IoT.

The benefits of IoT are numerous. It adds a significant level of comfort to our lifestyle by making it easier to interact with everyday objects that have become communicating objects. Monitoring our health is also one of the major benefits of IoT. It has positively impacted many sectors, including security, agriculture, industry, healthcare, and more. However, this comes at a cost, not only in terms of energy and environmental footprint, but also in terms of security. Indeed, IoT faces serious data and communications security issues. IoT is highly vulnerable from a cybersecurity perspective, which is very problematic. Imagine hackers taking control of the application used in the smart home example, giving them full control over the home appliances. These hackers will have a wide range of possibilities, such as locking residents in or spying on them through connected cameras. Imagine a hacker impersonating your connected medical device and reporting false information about your health.

3. Why this thesis?

The vulnerability of communicating objects from a cybersecurity point of view is a serious concern, as cyberattacks on these objects can lead to disastrous consequences, as mentioned in the previous section. This is a thesis on the security of radiocommunications between communicating objects in the context of the Internet of Things.

Entitled "Security for Internet of Things: hardware authentication mechanisms for communicating objects", this thesis was initiated thanks to a funding from the French Ministry of Higher Education and Scientific Research, through the following institutions:

- The "Institut National des Sciences Appliquées de Toulouse (INSA Toulouse)" as the host institution. INSA Toulouse is part of the prestigious "Groupe INSA", a national engineering school that also has a strong research activity in addition to its high-level training programs.
- The "Laboratoire d'Analyse et d'Architecture des Systèmes (LAAS)" of the "Centre National de la Recherche Scientifique (CNRS)" as a host research laboratory. Specialized in systems analysis and architecture, the LAAS-CNRS is one of France's major scientific research laboratories, with over 500 members working in various fields including computer science, robotics, automatic control, and micro- and nano- systems.

One of the projects that motivated the initiation of this thesis is the McBIM (Material communicating with the BIM - Building Information Modeling) project, in which our team, the MINC (Micro- and Nano- systems for Communications) team at LAAS-CNRS, is strongly involved. The McBIM project aims to monitor the structural health of civil engineering structures throughout their lifetime with the objective of taking preventive actions in order to avoid unexpected collapses. To do this, a number of communicating objects are buried in the concrete. These objects measure certain parameters such as moisture in the concrete and radio-communicate this information to a data collector located outside the concrete. The data collector establishes a health status report of the structure based on data gathered from communicating objects located in the concrete. As this project could be implemented in critical infrastructure such as nuclear power plants or bridges, the authenticity of the data transmitted to the collector must be guaranteed. In other words, before accepting the received data, the collector must be absolutely sure that this data truly comes from the expected communicating objects. An attacker must not be able to spoof the identity of these communicating objects and send false information to the data collector on behalf of these objects. Yet nowadays, thanks to scientific and technological advances in radiocommunications, such attacks have become accessible even to non-experts in the domain.

This work is in line with the research for security mechanisms to counter, among others, these types of attacks targeting communicating objects.

4. Organization of the Manuscript

The manuscript is organized as follows:

Chapter 1: IoT Landscape: Overview and Security: In this chapter, we discuss the ever-growing IoT market despite cybersecurity concerns. We then address the landscape of cyberthreats targeting IoT. Finally, we conclude with the IoT devices vulnerability factors and security challenges, taking as an example the communicating object that must be buried in

concrete to monitor the health of civil engineering structures as part of the McBIM project, briefly outlined in the previous section.

Chapter 2: Context of the work and State-of-art: In this chapter, we first present the context of this work. Next, we present the objectives of this thesis and what motivated us to undertake this work. We then position our work in relation to the state-of-the-art. Next, we present the types of attacks that our approach can counter. Finally, we conclude with the contributions of this work to the research in security mechanisms for IoT.

Chapter 3: Robustness of the PSD-based Devices Fingerprinting: In this chapter, we discuss one of the security approaches proposed in the literature: device identification using Fingerprinting technique, which aims to leverage manufacturing imperfections to assign each device a unique and time-constant signature. Through a practical case study using Power Spectral Density (PSD) as a Fingerprinting feature, we highlight the challenges that have to be overcome for Fingerprinting technique to be effective in device identification.

Chapter 4: Polarization Shift Keying (PoSK)-based Message Authentication System: In this chapter, we propose an innovative security approach based on radio wave polarization. This consists of transmitting in parallel a message and an authentication code for this message, using the radio wave polarization as a carrier for the authentication code. The experimental results are very promising.

After this final chapter, the manuscript's general conclusion revisits the two security approaches: identification using fingerprinting technology and messages authentication using radio wave polarization, as well as the perspectives of the latter approach.

Chapter 1: IoT Landscape: Overview and Security

1. IoT Market

The number of connected devices continues to grow. IoT Analytics, a leading global provider of market insights and strategic business intelligence for the IoT, AI, Cloud, Edge, and Industry 4.0, reported in September 2024 that there were 16.6 billion connected IoT devices by the end of 2023 (a growth of 15% over 2022). IoT Analytics expects this to grow 13% to 18.8 billion by the end of 2024. The number of connected IoT devices is estimated to grow to 40 billion by 2030, according to IoT Analytics [2]. The image shown on Figure 1 illustrates a modern IoT environment.



Figure 1. An illustrative image of an IoT environment. Source: https://www.indiamart.com/proddetail/iot-wireless-sensor-network-solution-service-26086465291.html?mTd=1.

As illustrated by the image shown in Figure 1, Internet of Things (IoT) devices have invaded every sector of the modern society, including:

Healthcare and Wearables: Berg Insight, the world's leading IoT market research provider, reported in September 2024 that the number of remotely monitored patients reached 76.7 million worldwide in 2023. Berg Insight estimates that the number of remotely monitored patients will grow at a CAGR of 12.8 percent to reach 140.1 million by 2028 [3]. The report says that the three main applications are monitoring of patients with sleep therapy devices, glucose level monitoring of patients with diabetes and monitoring of patients with implantable CRM devices.

To collect vital signs data and health metrics in real-time, digital healthcare often uses devices commonly known as wearables, because their user can wear them on. These wearables then send the data they collect to other types of devices, *e.g.*, a smartphone or to the cloud, for processing, analysis and information. The smartwatch is one of the well-known smart wearable

devices used in digital health. Most smartwatches can, for example, measure heart rate, steps taken, sleep patterns, physical activity tracking and monitoring, even more. There are also wearables for measuring blood pressure, blood glucose, oxygen levels, etc... For the wearer's everyday safety and security, some smartwatches provide features such as accident detection, fall detection, emergency calls, GPS location and navigation, etc...

Smart Home and Smart City: By adding connectivity to devices that historically were without it and by creating new possibilities, IoT is transforming homes and cities by making them more and more smart. Home appliances are becoming smart devices thanks to IoT. Smart refrigerators can keep track of food inventory and suggest recipes based on available ingredients. Smart washing machines can be controlled remotely, etc… Smart home devices also include smart thermostats, lighting systems, security cameras, and voice-activated assistants like Amazon's Alexa and Google Home. Smart home devices allow users to control and automate various aspects of their home environment, from adjusting the temperature to turning on lights with a voice command. Statista, a global data and business intelligence platform, reported that revenue in the Smart Home market is projected to reach US$154.4bn in 2024 and it is expected to show a CAGR 2024-2028 of 10.67%, resulting in a projected market volume of US$231.6bn by 2028 [4]. According to Transforma Insights, a research firm focused on the world of Digital Transformation, nine key domains of change in the Smart Cites sector that are enabled by digital transformation are: Smart Traffic Management, Smart Streetlights, Public Safety, Waste Management, Environment Monitoring, Smart Mobility, Integrated Transport Systems, Smart Data Management, Wearable Cameras for Personal Protection [5]. One of the IoT applications for Smart Cities is smart metering of electricity, water, gas, etc... IoT Business News, a premier online platform delivering the latest updates and insights on the IoT industry, reported in August 2024 that the total number of smart meters globally will double over the next decade, from 1.7 billion at the end of 2023 to 3.4 billion at the end of 2033. The total revenue set to be generated by smart metering globally was USD17 billion in 2023, a figure which will grow to USD40 billion in 2033 [6].

Industrial IoT (IIoT): IoT is also revolutionizing industry with massive integration of intelligent digital technologies into manufacturing and industrial processes. We are talking about the fourth industrial revolution or Industry 4.0, whose technological pillars include IIoT. Industry 4.0 allows for smart manufacturing and the creation of intelligent factories. It aims to enhance productivity, efficiency, and flexibility while enabling more intelligent decision-making and customization in manufacturing and supply chain operations. In September 2024, a research report from the IoT analyst firm Berg Insight says that annual shipments of wireless devices for industrial automation applications reached 10.7 million units worldwide in 2023, accounting for approximately 9 percent of all new connected nodes. Growing at a CAGR of 12.3 percent, annual shipments are expected to reach 19.1 million by 2028. The installed base of wireless devices in industrial automation reached at the same time 56.5 million in 2023 [7]. According to the Industrial Connectivity Market Report 2024–2028 by IoT Analytics, the global industrial connectivity market is projected to grow from US$89 billion in 2023 to US$104 billion by 2028 at a CAGR of 5% over the next 4 years [8].

Remote Monitoring and Tracking: IoT is also used for real-time monitoring and tracking, *e.g.*, weather monitoring. Berg Insight reported, in December 2023, that the installed base of remote tank monitoring (RTM) systems is forecasted to grow at a compound annual growth rate of 25.0 percent from 7.7 million units at the end of 2022 to 23.5 million units by 2027 [9]. New IoT

applications have emerged, such as Structural Health Monitoring (SHM) [10] which consists in monitoring the evolution of reinforced concrete parameters in civil engineering structures, *e.g.*, nuclear power plants, bridges, buildings, to prevent irreversible failures, avoid collapse, and allow to take preventive actions. Berg Insight reported, in November 2023, that the number of active tracking devices deployed for cargo carrying units (*i.e.*, trailers, intermodal containers, rail freight wagons, air freight ULDs and pallets) reached 8.8 million worldwide in 2022. Growing at a CAGR of 18.2 percent, this number is expected to reach 20.3 million by 2027 [11]. Tracking luggage, pets and children is another IoT use case. Transforma Insights reported, in November 2022, that the number of connected devices for Child and Pet Tracking is expected to reach 19.5 million by 2030, generating USD 1.2 billion [12].

Agriculture: IoT devices can be used to monitor soil moisture, sunlight, humidity and temperature. These data can be used in decision-aid systems for farmers in order to help them make the best choices in crop selection and treatment to keep plants healthy and thus increase productivity. Improving the efficiency of farming methods also result in better utilization of energy and resources, *e.g.*, water. According to Transforma Insights, nine key domains of change in the agriculture sector that are enabled by digital transformation are: Crop Management, Automatic Irrigation Systems, Livestock Management, Indoor Vertical Farming and Smart Greenhouses, Drone Farming, Farm Automation and Robots, Farm Management, Produce Monitoring, Supply Monitoring [13].

From this far-from-exhaustive overview of the IoT market, it is clear that IoT is present in every sector of activity; making IoT the largest data generator, we are talking about Zettabytes (10^{12} Gigabytes) of data. This huge amount of data generated by the IoT continues to grow as the number of connected devices increases, while the security of this sensitive data remains a major concern.

2. IoT Cyberthreats Landscape

The 2024 IoT Security Landscape Report [14] of NETGEAR and Bitdefender, the global leader in cybersecurity software, reveals that; Every 24 hours, home network devices see an average of 10 attacks; Every 24 hours, Bitdefender smart home security solutions block an average of 2.5 million threats, or roughly 1,736 threats per minute. According to the report, in 2023, the highest number of vulnerabilities were discovered in TV sets (34%), smart plugs (18%), digital video recorders (13%), and routers (12%). Of the vulnerabilities discovered in the study, the two most common types are buffer overflow (28.25%) and denial of service (27.20%). Next in line is code execution, gain privilege, memory corruption, and information leak. While less common than overflow and denial of service, code execution and privilege vulnerabilities pose significant risks due to their potential to give attackers access and privileges on compromised devices. In other words, this leads to unauthorized access, data breaches, and system compromise.

One of the major events, which drew the world's attention to the high vulnerability of IoT devices, was the advent of the Mirai malware in September 2016. The attacker uses Mirai to gain access and privileges to a connected device. The infected device, which is then under the attacker's control, is called a *bot*, as it is now similar to a robot serving the attacker. By creating thousands of *bots* all over the world, the attacker builds up a *botnet*, an army of *bots* at his disposal, waiting for his instructions. The attacker can then use his *botnet* to carry out what are

known as DDoS attacks, by ordering all his *bots* to make requests collectively to a targeted server, in order to saturate the latter and bring it down, thus causing a service outage. On September 20th, 2016, the website of the well-known security journalist Brian Krebs, *KrebsOnSecurity*, was one of the first victims of a Mirai botnet-based DDoS, with a magnitude of about 620 Gbps [15]. A few weeks later, the largest DDoS attack of its kind was targeted at a popular Dynamic DNS provider, Dyn. On October 21st, 2016, an attack with a magnitude of 1.2 Tbps was directed towards Dyn over the DNS port, 53. This attack was not only the largest of its kind, but it was also extremely effective; bringing down a significant number of websites and services who used Dyn as their DNS provider, including GitHub, Twitter, Reddit, Netflix and many others. French web hosting services provider, the giant OVHCloud (formerly OVH), was also among the early victims of a Mirai botnet-based DDoS. The literature provides numerous resources on Mirai [16], [17], [18].

Another malware that has raised global awareness of cybersecurity issues is the *Stuxnet* worm. Discovered in 2010, the *Stuxnet* worm mainly targeted SCADA systems in industrial controllers. According to [19], it was later found that the malware was not designed to spy, but rather to sabotage centrifuges in the power facilities of Natanz in Iran. It is believed that the USA built *Stuxnet* with the support of Israel with the goal of stopping or delaying the Iranian nuclear program. The worm was probably implanted in the Natanz power plant's network by using a compromised USB drive. The *Stuxnet* worm showed that industrial facilities, like any other sector, are also exposed to cyberthreats. Actually, with the massive integration of IoT devices into industrial facilities, the latter are even more vulnerable nowadays.

Recently, another malware drew the world's attention, the *Pegasus* spyware, discovered for the first time in August 2016. *Pegasus* is a spyware developed by the Israeli cyber-arms company NSO Group that is designed to be covertly and remotely installed on mobile phones running iOS and Android. *Pegasus'* attack methodology consists of the following steps: (1)- Gaining access to victim device, (2)- Malware propagation, (3)- Privilege Escalation, and (4)- Exfiltration of Data [20]. Nowadays, mobile phones are used for applications that go far beyond their historical functions (*e.g.*, making calls, exchanging messages). For example, they are used to control IoT devices and to collect information from IoT devices. Mobile phones have become our closest friends, and are very likely the most sensitive devices in everyone's device collection. This makes *Pegasus* one of the most dangerous spyware in the world. *Pegasus* can spy through social media applications like WhatsApp, Viber, Facebook, and messaging apps, as well as continuous access to the phone's camera and microphone. Although only discovered in 2016, after further investigation it was found that Pegasus had been operating in the United Arab Emirates (UAE) as far back as 2013. Among the numerous *Pegasus*'s victims are prominent political figures, and its users include many governments around the world.

In addition to the less frequent world-class attacks, IoT devices are the targets of thousands of more or less significant-scale attacks in different regions around the world every day and in every sector. Targeted devices include video surveillance cameras, secure access devices, smart thermostats, connected medical devices, audiovisual devices (e.g., Smart TVs), smart meters (electricity, gas, water), identification devices, payment devices, and so on.

For example, the impersonation of car keys has become a trend in recent years and allows thieves to unlock car doors using low-cost and easy-to-purchase radio devices.

3. IoT Devices Vulnerability and Security Challenges

3.1. Vulnerability Factors

IoT devices are highly vulnerable from a cybersecurity point of view. Their vulnerability factors include:

(a) - Weak Security Design: Young companies, because of limited expertise and experience, often focus more on the services provided by the IoT devices than on their security. Consequently, a great number of IoT devices are built with only cost as the driving design tenet and, as a result, these devices are often designed from COTS components and many of them are having poor configurations and open design.

(b) - Use of Wireless Communication: Wireless radio-communication is used in IoT mainly for mobility needs. However, this mode of communication, by its very nature, is vulnerable and creates additional attack surfaces. Indeed, using the air as transmission media represents an inherent vulnerability of radio-communication, as the signal is physically accessible to anyone within range. This, consequently, makes easier some attacks such as jamming, eavesdropping, replay attacks and even cross-protocol attacks [21]. Moreover, these attacks have become too accessible thanks to Software-Defined Radio (SDR). SDR hardware such as the USRP - Universal Software Radio Peripheral [22] or the HackRF One [23] together with SDR software such as GNU Radio [24] or Universal Radio Hacker (URH) [25], [26] provide tremendous possibilities for the development of radio applications, including malicious ones, in record time and within affordable budgets. Another tool, even more affordable but very powerful, is the Flipper Zero [27], a portable multi-tool for pentesters and geeks in a toy-like body, but it can be used for malicious purposes such as deauthentication attacks, where malicious actors use spoofed deauthorization packets to disrupt legitimate user connections [28]. The deauthentication attack can be used to force devices to lose access to their current Wi-Fi (Wireless Fidelity) networks and then reconnect to a hacker-controlled network.

(c) - Manufacturers' interactions: From firmware updates to sensitive information gathering and sharing for economic purposes, the various interactions between the manufacturer and the device expose the latter to attacks, e.g., corrupted firmware attack.

3.2. Security Challenges

While, at the same time, securing IoT devices and networks remains a major concern due to multiple challenges, including:

(i) - Resources limitation: The resources (computing power, memory and energy) limitation of IoT devices make traditional security solutions not suitable to them. Indeed, while traditional computing devices - e.g., laptop - have high computing power, significant amount of memory, high-capacity storage drives and high-capacity easy-to-recharge energy storage components, the IoT devices, e.g., Smart Pacemaker, on the contrary, have limitations regarding at least one of these aforementioned resources (computing power, memory and energy). Consequently, traditional security solutions - e.g., Intrusion Detection/Prevention System (IDS/IPS), complex software-defined cryptography - are not adapted to IoT devices, as their operation requires significant resources in terms of computing power, memory and energy, beyond the capacity of most IoT devices.

(ii) - Use of multiple protocols: The industry of IoT is, so far, poorly governed by standards. IoT is still a highly multi-protocol world, where a large number of protocols are used, including BLE, Wi-Fi, LoRa, Sigfox, NB-IoT (Narrowband Internet of Things), ZigBee, Z-Wave, cellular

networks (e.g., 4G, 5G), etc. This multi-protocol aspect of the IoT is actually a disadvantage when it comes to security, as security solutions have to be protocol-independent. Furthermore, some protocols are proprietary (i.e., their specifications are not accessible) and it would be difficult to adapt one's security solution to these proprietary protocols. This multi-protocol nature of IoT remains one of the major challenges facing IoT security research. For example, a lightweight Intrusion Detection System (IDS) against IoT memory corruption attacks is presented in [29], where the authors achieved very interesting results with a detection accuracy of 99.98%, but their solution is not protocol-independent.

(iii) - Peer-to-Peer Communication: A key element in a traditional Local Area Network (LAN) is the gateway, through which incoming, outgoing and internal communications transit. This architecture has a major advantage when it comes to security, as the gateway represents an ideal control point of the traffic, which is done using firewalls. Whereas in IoT, direct machine-to-machine (M2M) communication is widely used, resulting in highly decentralized networks that make difficult the use of firewall-like solutions.

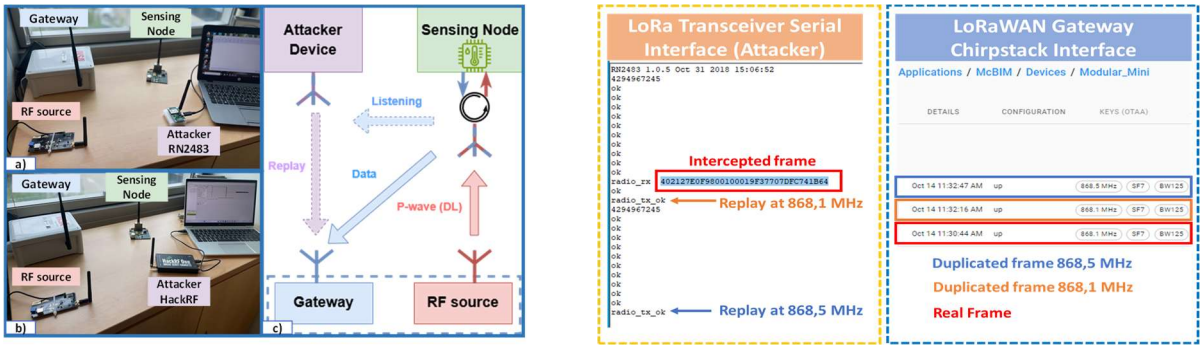
These constraints related to IoT require security solutions to be low computational complexity, lightweight in terms of memory footprint and protocol-independent. Therefore, traditional security solutions are not adapted to IoT devices; consequently, IoT security still remains a major concern. While the new age of Artificial Intelligence (AI) further accentuates this concern. Indeed, the use of AI in cybersecurity is as exciting as it is worrying. On the one hand, AI could be a powerful solution for improving IoT security [30] ; while on the other, its use for malicious purposes is proving to be a serious threat to the cybersecurity of IT systems in general [31] and IoT systems in particular [32]. Tools like ChatGPT can be powerful assistants in writing malware [33] [34].

With the ongoing rapid growth of IoT and considering the high vulnerability of IoT devices and networks, their security challenges and the diversity and quantity of threats targeting them, new security mechanisms are urgently needed for IoT devices and networks.

3.3. SHM device: A good example

Designed for the Civil Engineering Structural Health Monitoring, the cyber-physical device presented in [10] is an ideal example of a vulnerable IoT device. Due to the constraints associated with its deployment environment, this Sensing Node is powered by Wireless Power Transfer (WPT) [35], i.e, convert the received electromagnetic power into DC electrical energy. The device uses a simple super-capacitor as energy storage buffer to store just enough energy for transmitting, via the low-power LoRaWAN protocol, the few bytes from the embedded sensors' measurements. Data transmission by the Sensing Node requires between 21 mJ to 250 mJ depending on communication configurations (transmission power, payload length, bit rate). This represents a very limited quantity of available energy. Consequently, the Sensing Node uses the ABP (Activation by Personalization) activation mode of LoRaWAN. This mode bypasses the join procedure and uses hard-coded static security credentials, which makes this activation mode more energy-efficient than OTAA (Over the Air Activation) activation mode. But, ABP mode is more vulnerable to attacks than OTAA mode, as the latter uses the join procedure to join a LoRaWAN network [36]. Anyway, this Sensing Node is not compatible with OTAA mode, as it is a transmitter only and was not designed to support data reception.
Frame counters are important parameters in LoRaWAN security. They can, if not reset, help prevent replay attacks. However, due to energy shortage, this Sensing Node switches off

between two transmissions, causing reset of the frame counters. Consequently, this Sensing Node is vulnerable to attacks such as man-in-the-middle (MitM) attack and replay attack, the latter was successfully carried out on it, as shown on Figure 2.



* Power waveform (P-wave) as downlink (DL)

(A) **(B)**

Figure 2. (A) – Replay attack experimental setup: **(a)** setup with RN2483 LoRa transceiver as attacker device, **(b)** setup with HackRF One as attacker device, **(c)** diagram of replay attack process. **(B)** - LoRa transceiver serial interface displaying captured frame (left) and gateway ChirpStack network application with original and duplicated frames (right).

As part of the same project [10], a Sensing Node communicating via Bluetooth Low Energy (BLE), instead of LoRa, was developed. This Sensing Node uses connectionless Bluetooth advertising mode to broadcast the few bytes from the embedded sensors' measurements. This way of communicating data is also highly vulnerable. Bluetooth receivers within the range of the Sensing Node can capture the data. It is also easy to spoof this Sensing Node and communicate false information to the data collector [37].

4. Conclusion

In this chapter, through a few examples of applications covering various sectors, we have provided an overview of the enormous market represented by IoT. Market analysis providers predict considerable growth in this market in the coming years. This makes the issue of cybersecurity for IoT devices even more alarming, given the numerous types of cyberthreats targeting IoT devices. In addition to being used to build botnets, these devices are also often used as a gateway to larger infrastructures. Not to mention P2P attacks facilitated by their communication mode, that is radiocommunication; these attacks include MitM attacks, replay attacks, spoofing and impersonation attacks, cross-protocol attacks, etc.

Meanwhile, securing these vulnerable devices remains very challenging due to their resource-constrained nature, as perfectly illustrated by the example of the SHM device. This limitation in computing power, memory, and available energy makes IoT devices incompatible with traditional security solutions such as firewalls, complex cryptography, IDS, etc. There is a real need for security mechanisms that are tailored to the IoT.

Chapter 2: Context of the work and State-of-art

1. Context

Several papers in the literature [38], [39], [40], [41], [42], [43] show an overview of the incredible number of possible attacks on IoT devices. While the lack of security solutions persists, IoT continues to grow, with billions of devices already deployed worldwide. Sometimes deployed within critical network infrastructure in industrial and residential environments, these highly vulnerable devices make the networks they are a part of more vulnerable than ever because they are prime targets for attackers who use them as entry points to reach more important devices and large-scale networks. For instance, an attacker can use malicious tools such as Mirai [18] to get into a critical industrial network. The consequences of such attacks can be devastating. While Software-Defined Radio (SDR) and Artificial Intelligence (AI) promise to make attacks on IoT more and more simple and accessible with new tools. IoT security has become a major research topic in recent years. But research is facing major challenges in developing suitable security solutions for IoT. These challenges, which are inherent to the very nature of IoT (e.g., the resource-constrained nature of IoT devices), make traditional security solutions unsuitable for IoT. So, there is a need for new security mechanisms adapted to IoT devices and networks. This is the context in which the work presented in this manuscript was conducted.

2. Objective of this thesis

This thesis aims to propose original security solutions to allow IoT devices to discover their environment, to integrate into it confidently and to adapt to it regarding the number of IoT devices, the required data throughput, the required transmission range and power, as well as the available energy. This adaptability to the environment must take into account the mobility of these IoT devices, which must be able to belong to different IoT networks at different times, as well as how trustworthy are these different networks. In other words, it is all about finding new security mechanisms that take IoT-related constraints into account. This means security solutions with low memory footprint, low computational workload, energy-efficient, generic and protocol-independent, adapted to direct machine-to-machine (M2M) communication to ensure, for example, secure integration of a BYOD-type IoT device into a Peer-to-Peer (P2P) or Mesh topology IoT network. In addition, for successful attacks, countermeasures through hardware and/or software reconfiguration and re-adaptation techniques are also needed.

A common feature of IoT devices is their communication mode, i.e., wireless radio-communication. Indeed, IoT devices may be using different parameters and protocols (e.g., BLE, LoRa), but at the end of the stack, the transmission takes place over the air, *i.e.*, via electromagnetic waves. So, what IoT devices have in common is the type of the transmission media (electromagnetic waves), a parameter of the first layer in the OSI (Open Systems Interconnection) model, *i.e.*, the physical layer. This makes the physical layer (PHY-Layer) a very interesting one in the research for security solutions that meet IoT requirements, especially genericity and protocol independence. Thus, the work presented in this manuscript focuses on security at the physical layer of IoT devices and networks, specifically, through an innovative message authentication technique.

What if a unique authentication code could be associated with each transmitted message by an IoT device, an authentication code changing from one message to the next and transmitted in parallel with the message itself by the same IoT device? With such a system, the receiver is

certain that the received message does come from the expected transmitter. This makes it possible to counter attacks such as spoofing, replay, impersonation and Man-in-the-Middle (MitM), which are all very common attacks in IoT. Now imagine this system being completely independent of protocols (e.g., BLE, LoRa), resource-efficient and requiring no major modifications to the IoT device. It is exactly to an authentication system of this type that the majority of the work presented in this manuscript is dedicated. The proposed system is based on the RF wave polarization, and the results are very promising.

3. State-of-the-art

The literature regarding IoT security focuses on three main topics: intrusion detection / prevention, identification and authentication.

Intrusion detection / prevention in computer networks has always been of interest to IT security professionals, from the very beginning of the democratization of computer networks, long before the emergence of IoT. Over time, various types of intrusion detection / prevention system (IDS / IPS) have been proposed. For example, when intrusion detection is performed locally on a network node (computer), it is called a Host-based Intrusion Detection System (HIDS), and when intrusion detection is performed at network-scale, it is called a Network-based Intrusion Detection System (NIDS) [44]. HIDS examines, among other things, hardware usage by software, node's outgoing and incoming network traffic, in order to detect and counter intrusions before they affect the entire network, while NIDS examines the network traffic of the entire infrastructure in order to detect malicious traffic. Whereas HIDS is implemented on nodes, NIDS is implemented on traffic transit points such as firewalls and network switches.

IDSs are also classified, according to their operating principle, into signature-based IDSs, behavior-based IDSs and hybrid IDSs.

Signature-based IDS [45] compares network traffic to a database of known dangerous traffic patterns to identify malicious activities. Although, signature-based IDS accurately identifies known attacks by cross-referencing them with previously stored patterns in its knowledge base, it cannot detect newly discovered attacks. Indeed, signature-based IDS do not have signatures for newly created threats; therefore it cannot identify zero-day. Regular updates are necessary to the signature database to protect against newly discovered threats, i.e., to make it identify newly discovered attacks. However, signature-based IDS is computationally more effective than other approaches that make them perfect for systems with constrained resources hence IoT WSNs. In the literature, signature-based IDS can be referred to by other terms such as misuse-based IDS, rule-based IDS, knowledge-based IDS, etc.

Behavior-based IDS, also known as anomaly-based IDS, is mainly based on Machine Learning [46]. The work in [47], [48] is a good example of behavior-based IDS. The idea is firstly to use Machine Learning to establish a profile (a model) of legitimate radio-communications in a given environment (e.g., smart home, offices), and secondly to use this legitimate profile to detect any abnormal (unusual) behaviors (activities), i.e., any deviation from the pre-established legitimate profile. An example of an abnormal activity would be the occurrence of a communication on a frequency not usually used in this environment, or a communication at an unusual moment of the day. Behavior-based IDS is more costly and computationally more intensive than signature-based IDS. As shown by the work in [47], [48], Behavior-based IDS requires the generation and processing of large amounts of data, hundreds

of gigabytes. Also, monitoring radio-communications over a large frequency band requires high-performance, expensive tools.

Hybrid IDS aim to harness the strengths of both signature-based and behavior-based IDS to enhance intrusion detection. For example, [49] used a hybrid model combining a signature-based detection algorithm(Decision Tree, Naive Bayes, or its variants) with an unsupervised (clustered) anomaly-based (behavior-based) detection algorithm (DBSCAN or Isolation Forest) to detect both known as well as unknown attacks hence increasing the detection rate (DR) and reducing false-alarm rate (FAR).

Generally speaking, these intrusion detection techniques were initially designed for traditional computer networks using resource-abundant nodes such as standard laptops. These IDSs are poorly adapted to the IoT, where nodes are resource-constrained. Consequently, their adaptation to the IoT has become a major research topic, as witnessed by the literature.

The second topic on which IoT security research is focused is device identification, which is a classic feature of computer networks. For example, identification by addressing is systematically present in computer networks. Information such as host name, IP address and MAC address are some essential elements for the smooth running of exchanges within a computer network. However, these elements are not sufficient to guarantee the security of computer networks. For example, spoofing attacks remain possible. In recent years, a new identification technique has appeared in the literature: fingerprinting, which aims to identify a device through its inherent manufacturing imperfections. This technique, which IoT security researchers are very interested in, is the first security approach studied in this work.

The third topic on which IoT security research is focused is authentication, which is in fact a level of security above identification, since it means for the receiver to make sure that the received information actually comes from the expected sender, whose identity is already known to the receiver. The main subject of this work is a message authentication system based on Polarization Shift Keying (PoSK), aiming to associate an authentication code with each message emitted by a given radio-communicating device.

Let's start with the state-of-the-art regarding fingerprinting in IoT. Researchers have been interested in IoT devices fingerprinting for years now so much that it could be the most explored devices identification approach in the literature. Several papers about this identification approach have already appeared in the literature, highlighting its opportunities and challenges. In [50], Chatterjee et al. present "RF-PUF" for Radio Frequency - Physical Unclonable Functions. This expression "physical unclonable functions", introduced by Gassend et al. [51], [52], basically refers to any intentional add-on to a device hardware or any unintentional deviation (introduced by the manufacturing processes) from the theoretical data of a device hardware. These unintentional deviations are assumed to be unique for each device, even for two instances of the same product manufactured on the assembly lines at the same time. In analogy with human being, these inherent imperfections are for an electronic device what DNA, fingerprints, or voice are for a human being, that’s why they would be useful for electronic device identification. Indeed, these inherent imperfections, whose impact is not a concern when it comes to the normal operation of the device, could be used to build a signature for a given device. In the approach of devices authentication by fingerprinting, these inherent imperfections are known as "fingerprinting features". These imperfections in the transmitter components reflect on the RF signal properties such as frequency offset and I/Q imbalances. These are the

properties, together with the channel features for compensation (attenuation, distortion, and Doppler shift) that Chatterjee et al. use as fingerprinting features in [50]. Despite a certain efficiency of their solution, Chatterjee et al. illustrated, in some of their experiments, one of the major challenges that fingerprinting faces, which is the instability of the fingerprinting features regarding environment, i.e., with the environmental parameters. Indeed, changing the temperature of their experimental closed environment in discrete steps of 5 °C in the range of 0 °C–25 °C modified the transmitter RF signal properties, and the receiver ("the fingerprinter") considered every 5 °C change in the temperature as a different transmitter. This means that the used fingerprinting features are not resilient to environmental parameters, at least not to the temperature in this case. This is a major limitation of their solution, making it unusable as it is in a real-world IoT environment, as the same device signature ("fingerprint") can change from one moment to another depending on the environmental conditions.

In [53], Wang et al. present RF-DNA (Radio Frequency–Distinct Native Attribute) which can be seen as another name for RF-PUF. Wang et al. chose amplitude, phase, and instantaneous frequency as fingerprinting features. In their experiments, the accuracy of the devices' fingerprinting identification dropped drastically when the noise level increased in the experimental environment.

In [54], Vaidya et al. used clock oscillator and analog-to-digital converter (ADC) deviations as fingerprinting features. They experimented with COTS devices including Arduino boards. However, their approach is difficult to deploy in a real-world IoT environment because all the devices in the network have to run a local copy of their fingerprinting software, this is a major limitation of their approach since a real-world IoT environment is made of devices that are not usually designed to be rewritable by users. Moreover, an IoT environment can be very heterogeneous in terms of communication protocols and very dynamic with BYOD devices joining and leaving.

Electronic devices emit transient signals during the boot-up phase. Some authors are interested in using these transient signals as fingerprinting features [55], [56]. This method is again less suitable for a real-world IoT environment where most of the devices joining are already fully powered on. An interesting review of radio-frequency fingerprinting techniques is provided by [57].

Other ideas for fingerprinting can be found in the literature, including the addition of external circuitry dedicated to device-specific fingerprint generation, such as the addition of an extra layer of silicon by chip manufacturers or the addition of an external circuit by a specialist. This solution not only adds extra costs but also requires major modification in the hardware of a COTS IoT device, purchased on the market, which is definitely not convenient.

The first chapter highlights the challenges and possibilities of the devices fingerprinting approach through the use of Power Spectral Density (PSD) as a fingerprinting feature.

The state-of-the-art relating to the new message authentication system is going to be shorter because, to the best of our knowledge, we are the first to propose it, although the underlying principle, namely Polarization Shift Keying (PoSK), is not a new concept. Indeed, PoSK modulation has been used in optical communications for years now; it is a well-known modulation in this field [58], [59], [60]. Optical PoSK modulators capable of operating at tens of Gbps was developed [61]. On the other hand, the use of PoSK modulation in radio-

communications is more recent. In December 2004, a patent entitled "Polarization state techniques for wireless communications" was published [62]. More recent papers about PoSK can be found in the literature. In [63], the use of PoSK in satellite communications is discussed. In [64], fundamental properties of PoSK are investigated in wireless channels subject to fading, with a particular interest in Rayleigh and Rician fading channels. In [65], a polarization-based method for detecting and mitigating GNSS (Global Navigation Satellite System) signal spoofing attack is presented. For data transmission, there is also a lot of literature on polarization modulation using Reconfigurable Intelligent Surfaces (RIS) [66], [67], [68].

In the second chapter of this manuscript, we propose a, new and promising, PoSK-based message authentication system.

4. Threat Model

IoT devices have different functions or hardware capabilities and therefore, different threat models. Some devices are connected to the network and wait for instructions, while others are always on, sending and receiving data. Let's take the example of a critical network infrastructure made up of various IT systems including IoT devices. Let's assume that all network devices are physically reliable, i.e., their structure cannot be modified. The IoT devices wirelessly exchange confidential data (e.g., health data from a smartwatch) or critical data (e.g., a time set point for a chemical process or a temperature set point for a communicating thermostat). These data have to be authentic in order to counter potential malicious actions through data corruption that could have severe consequences. Thus, to guarantee this integrity, a solution could be the identification of the transmitter of the data in order to know whether or not it is a legitimate transmitter. Indeed, without such a solution and regarding the very nature of the wireless transmission medium, an attacker can transmit corrupted data or unwanted instructions to a device, for instance, the wireless transmission of an unwanted temperature set point to a communicating thermostat in order to increase the temperature to an unreasonable level.

With physical access to the network infrastructure and equipped with a malicious device that they fully control, with no need to even integrate that device into the network, the attacker can communicate with the IoT devices in the network through the wireless medium, which paves the way for malicious actions such as transmitting unwanted instructions and corrupted data. In such a position, the attacker can perform a plethora of attack scenarios against these devices, which can result in a breach of the three main aspects of security, namely: ***(i) – Availability:*** Prevent a device from performing its regular task, i.e., making a device unable to communicate with others through attacks such as denial of service (DoS), jamming, power off, and so on; ***(ii) – Confidentiality:*** Capture sensitive information through attacks such as eavesdropping; and, ***(iii) – Integrity:*** Modify the device, reduce its capabilities, or divert its use [21], which are breaches of its hardware / software integrity. Moreover, the integrity of transmitted data must be guaranteed in order to counter attacks through the transmission of corrupted data and false information.

Let's consider an IoT environment illustrates on Figure 3, with: ***Two wireless Sensing Nodes:*** SN1 is a transceiver and SN2 is an only transmitter. The Sensing Node presented in [10] is a good example of SN2 designed for the Civil Engineering Structural Health Monitoring. ***An Actuator Node (AN):*** a smart light bulb. ***A Gateway Device (GD):*** this can be a Bluetooth terminal or a LoRaWAN gateway. ***Two wireless Malicious Devices (MD):*** these can be

Software-Defined Radio (SDR) devices (e.g., USRP - Universal Software Radio Peripheral [22], HackRF One [23]). ***A Personal Computer (PC)***.

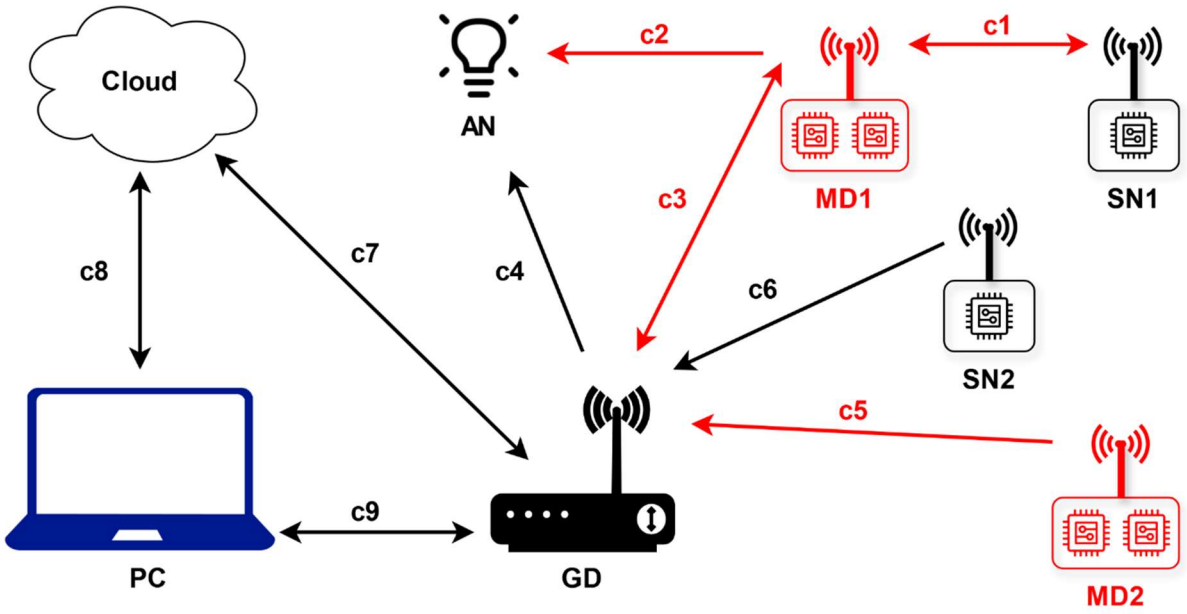


Figure 3. Threat model illustration in an IoT environment. SN = Sensing Node; MD = Malicious Device; GD = Gateway Device; AN = Actuator Node; PC = Personal Computer; cx = Channel x (x=1, 2, 3, 4, 5, 6, 7, 8, 9).

The main attacks targeted in this work are: ***Replay attacks:*** this consists of an attacker replaying, as received, a valid signal intercepted from the legitimate transmitter, with the aim of making the receiver (i.e., the targeted device) execute the expected action, for malicious purposes. As an example, on Figure 3, MD1 can intercept the light bulb (AN) turn-on/turn-off signal through channel c3 and replay it on channel c2 to turn the bulb on/off. As well as devices like this light bulb, electronically secured access systems (e.g., garage, parking and vehicle doors) are also prime targets for replay attacks; in these cases, the signal is intercepted from the wireless electronic access key. Although made more challenging by the rolling code mechanism, this attack remains possible. In the PoSK-based message authentication system, a message counter is added to the authentication to strengthen the system's ability to detect the following scenario: a message captured with its authentication key and replayed before the arrival of the next message is undetectable by the security gateway. Adding a message counter makes this scenario more detectable. ***Spoofing & mimicking attacks:*** The attacking device pretends to be its victim which is a legitimate device. In spoofing, the attacker needs to use the victim credentials to impersonate it. While mimicking targets a receiver that do not perform credentials verification, which is a common vulnerability in IoT machine-to-machine (M2M) communications; in this case the attacker just needs to reproduce the signals of the legitimate transmitter and transmit it to the receiver. As an example, on Figure 3, MD1 can mimic GD (or another remote control of AN) to control AN directly through the c2 channel. As an example of spoofing, MD2 can use SN2's credentials to communicate false information on SN2's behalf via the c5 channel. The emergence of Software-Defined Radio (SDR) has made spoofing and mimicking much easier. ***Man-in-the-Middle (MitM) attacks:*** MitM attacks mainly target traffic between two terminals. In this case, the attacker acts as a relay between the two terminals, which believe they are

communicating directly without any intermediary. In this way, the attacker captures messages coming from one terminal and eventually modifies them before communicating them to the other terminal. This attack is illustrated, on Figure 3, by the channels c1 and c3, where MD1 is the attacker and SN1 and GD are the victims.

The work presented in this manuscript, device identification using fingerprinting and message authentication using PoSK, aims to counter these attacks that take advantage of the propagating nature of radio waves in order to establish countermeasures and to ensure the integrity of data exchanged between devices. This is achieved through received message authentication and transmitter identification, even if the received message is meaningful for the receiver (the targeted device). Thus, the latter checks that the received message indeed comes from a legitimate transmitter. The targeted environments of this initial work are critical, difficult-to-access and risky static environments such as industrial plants, civil engineering (e.g., nuclear power plants) Structural Health Monitoring (SHM), etc.

5. Conclusion

This thesis offers two major contributions on the security at the physical layer in IoT. The first contribution relates to the analysis of devices fingerprinting, which aims to associate a unique and constant signature with a given device, based on inherent manufacturing-introduced imperfections in that device hardware. This contribution analyses the possibilities of the fingerprinting approach using Power Spectral Density (PSD) as device signature. The main question answered is whether or not it is possible to identify a transmitter from the PSD of its signal. The first chapter of the manuscript is dedicated to this contribution.

The second contribution proposes a new innovative message authentication system using the radio wave polarization. Basically, this consists in a cryptographically controlled modification of the polarization of the transmitter outgoing wave, in order to transmit, in parallel with the main message, authentication data for this main message. The experimental results of this new message authentication system show that this technique is a promising way of achieving secure communications in IoT where devices are resource-constrained.

Chapter 3: Robustness of the PSD-based Devices Fingerprinting

1. Introduction

As previously mentioned, fingerprinting aims to associate a unique and constant signature with a given device, for identification purposes. The signature is built from the inherent manufacturing-introduced imperfections in the device hardware. For a wireless communicating device, the fingerprinting features (i.e., the inherent imperfections) can be extracted at two different levels and be used for the fingerprinting of this device. ***(1) - In-situ fingerprinting***: focus on the deviations in some specific components of the hardware such as clocks, ADC/DAC, amplifiers, and so on, in order to perform fingerprinting in-situ, i.e., locally on the device. As an example, in [54], the fingerprinting is performed in-situ. ***(2) - Ex-situ fingerprinting***: uses an external radio receiver ("the fingerprinter") to perform the fingerprinting of the transmitter. In this case, the fingerprinting is carried out by the "fingerprinter" through the impact the transmitter components deviations would have had on the RF signal properties such as frequency offset, phase noise, I/Q imbalances, and so on. The deviations in the transmitter RF signal properties are, in indeed, the echo of the deviations in the components of this transmitter hardware. As an example, in [50], the fingerprinting is performed ex-situ.

The ex-situ fingerprinting is centralized fingerprinting where one "fingerprinter" can be used to perform the fingerprinting of several transmitters, whereas in in-situ fingerprinting, each node (device) is a "fingerprinter". In ex-situ fingerprinting, the computation workload is shifted to a central device (i.e., the "fingerprinter"), a gateway, for example, can be used as "fingerprinter". Figure 4 presents the in-situ and ex-situ fingerprintings.

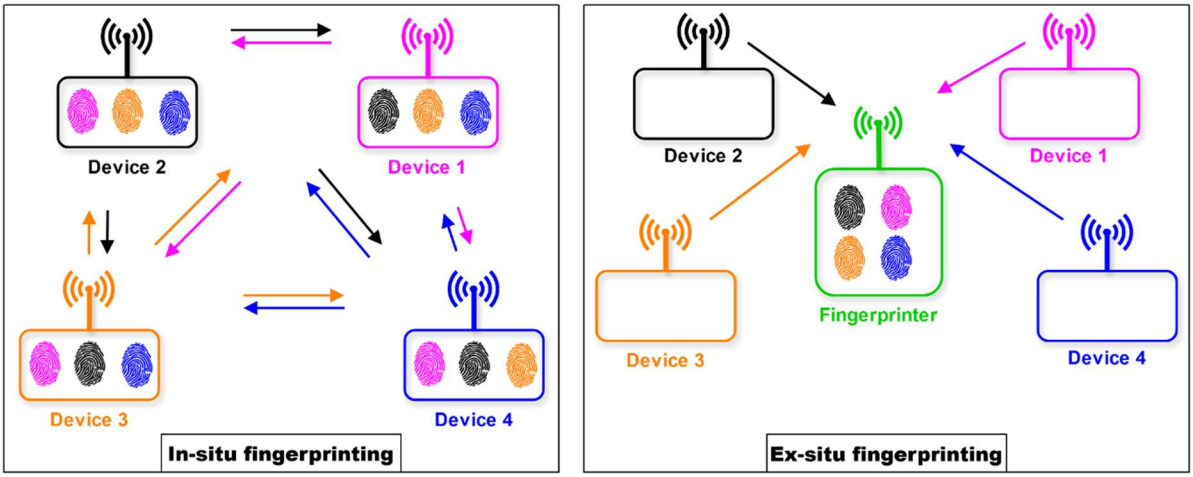


Figure 4. Illustration of in-situ and ex-situ fingerprintings.

In this chapter, an ex-situ fingerprinting is performed, where the "fingerprinter" uses the Power Spectral Density (PSD) of the transmitter signal as a fingerprinting feature. Devices identification using fingerprinting, especially ex-situ fingerprinting, is one of the security approaches adapted to IoT resources-constrained devices. Although fingerprinting is one of the promising possibilities for improving IoT security, as this manuscript was written, this approach still faces major challenges limiting its efficiency. In this chapter, we are going to highlight the possibilities and the challenges of the devices fingerprinting.

2. Power Spectral Density (PSD)-based Fingerprinting

2.1. Related Work

In the literature, PSD is used for a variety of applications. For example, in [69], biometric identification based on the PSD of the electroencephalograph (EEG) signal is investigated. When it comes to device identification, [70] uses PSD to perform a classification based on the transient unintentional modulation effects. Another paper, namely [71], proposes a device fingerprinting based on the frequency profile of the communications, extracted from the PSD of the signals emitted by the device. This latter approach is the closest to the work presented in this chapter; the major difference is that the authors did not take into account the possibility that the devices could have very close properties, nor that the environment could change a lot. This means a partial projection of the attacker model and deployment environment. Their solution was designed for static environments where the devices do not move and the ambient conditions remain more or less constant, which is not true for most real-world IoT environments.

2.2. Analytic Development

2.2.1. PSD and Correlation Function

Consider a transmitted signal $x(t)$, proposed for simplicity, a periodic power signal. Then it can be written as

$$x(t) = \sum_{n=-\infty}^{\infty} C_n e^{i2\pi f_n t} \quad (2)$$

where $f_n = nf_0$ and $f_0 = 1/T$ with T the period of x. The autocorrelation function $R_x(\tau)$ is defined as

$$\frac{1}{T}\int_{(T)} x(t)\,\overline{x(t+\tau)}\,dt \quad (3)$$

where $\overline{x}$ denotes the complex conjugate of x. A simple calculation leads to

$$R_x(\tau) = \sum_{n=-\infty}^{\infty} |C_n|^2\, e^{i2\pi f_n \tau} \quad (4)$$

The power density function of x is the Fourier transform of $R_x(\tau)$ and is given by

$$PSD_x(f) = TF[R_x(\tau)] = \sum_{n=-\infty}^{\infty} |C_n|^2 \delta(f - f_n) \quad (5)$$

2.2.2. PSD through a channel

Consider a multipath fading channel of N paths. The channel output which is the input signal for the receiver, can be written as in [72].

$$y(t) = \sum_{k=1}^{N} a_k(t)\, x(t - \tau_k(t)) \quad (6)$$

where $a_k(t)$ and $\tau_k(t)$ are, respectively, the attenuation and the propagation delay associated with the *n*th multipath component. If the transmitter and receiver are at rest and if there is no change in the transmission medium, the transmission delay associated with the *n*th propagation path and the path attenuation are constant, that is $\tau_k(t) = \tau_k$ and $a_k(t) = a_k$. In that case, we have

$$y(t) = \sum_{k=1}^{N} a_k \, x(t - \tau_k) \tag{7}$$

The channel is fixed and can be represented by its impulse, i.e.,

$$h(t) = \sum_{k=1}^{N} a_k \delta(t - \tau_k) \tag{8}$$

or its transfer function

$$H(f) = \sum_{k=1}^{N} a_k e^{-i2\pi f \tau_k} \tag{9}$$

The output signal $y(t)$ is then obtained through the following convolution product

$$y(t) = \int_{-\infty}^{\infty} h(u) \, x(t - u) \, du \tag{10}$$

and the PSD being the Fourier transform of the autocorrelation of $y(t)$, we have

$$PSD_y(f) \; = \; |H(f)|^2 PSD_x(f) = \sum_{n=-\infty}^{\infty} |C_n|^2 |H(f_n)|^2 \delta(f - f_n) \tag{11}$$

But

$$|H(f)|^2 = H(f)\overline{H(f)} = \sum_{k=1}^{N} a_k e^{-i2\pi f \tau_k} \sum_{l=1}^{N} a_l e^{i2\pi f \tau_l}$$

$$= \sum_{k=1}^{N} {a_k}^2 + \sum_{k=1}^{N} \sum_{l=k+1}^{N} a_k a_l (e^{-i2\pi f(\tau_k - \tau_l)} + e^{i2\pi f(\tau_k - \tau_l)})$$

$$= \sum_{k=1}^{N} {a_k}^2 + 2 \sum_{k=1}^{N} \sum_{l=k+1}^{N} a_k a_l \cos(2\pi f(\tau_k - \tau_l)) \tag{12}$$

$$= [a_1 \quad \cdots \quad a_N] \underbrace{\begin{bmatrix} 1 & \cos(2\pi f(\tau_1 - \tau_2)) & \cdots & \cos(2\pi f(\tau_1 - \tau_N)) \\ \cos(2\pi f(\tau_1 - \tau_2)) & 1 & \cdots & \cos(2\pi f(\tau_1 - \tau_N)) \\ \vdots & \vdots & \vdots & \vdots \\ \cos(2\pi f(\tau_1 - \tau_N)) & \cos(2\pi f(\tau_2 - \tau_N)) & \cdots & 1 \end{bmatrix}}_{M(f)} \begin{bmatrix} a_1 \\ \vdots \\ a_N \end{bmatrix}$$

Note that because $|H(f)|^2 \geq 0$, the symmetric matrix $M(f)$ is positive semidefinite for all f.

Analyzing the off-diagonal terms of $|H(f_n)|^2$, their absolute values $|\cos(2\pi n f_0(\tau_k - \tau_l))|$ are sensitive to the values of the delays τ_k and are close to one if

$$\tau_k - \tau_l \approx \frac{j}{2f_0} = j\frac{T}{2}, j \in \mathbb{Z} \tag{13}$$

As expected, the PSD of signal $y(t)$ can be significantly affected by the channel and its use as a fingerprinting feature is really questionable, even more so when the channel characteristics are changing due to the changes in the transmission medium.

2.3. Methodology

2.3.1. Aims of This Work

An ideal fingerprinting solution would be a one that is able to build a unique and time-constant signature for a given device from its inherent imperfections introduced by the manufacturing processes. This signature should remain the same for the entire lifetime of the device, it should not change in any circumstances. In such a case, the fingerprinting features (i.e., inherent imperfections) used to build this signature are considered to be relevant.
Since these inherent imperfections reflect on the RF signal that the device emits, this work aims to figure out whether or not the PSD of this RF signal would be a relevant fingerprinting feature, meaning unique for a given device and stay constant under any circumstances, and over the device entire lifetime. We will, hence, examine the behavior of the PSD regarding two influential elements, namely:

- Ambient conditions (temperature, relative humidity, etc.);
- The data carried by the emitted RF signal, i.e., the data modulating the RF carrier.

These elements are two among others (e.g., device aging) but are enough to address the question of whether or not the PSD would be a relevant fingerprinting feature for device identification.

2.3.2. Experiments Overview

To assess the influences of the environment and the data on the PSD of the RF signal, we carried out a certain number of experiments. These experiments were performed on Bluetooth Low Energy (BLE) devices presented on Figure 6. These devices were designed at the LAAS-CNRS [73], [74] laboratory as part of a project that aims to monitor the structural health of reinforced concrete throughout its lifetime. To do so, these devices measure some physical parameters (e.g., temperature, relative humidity, and resistivity) of reinforced concrete and then wirelessly transmit these data via the BLE technology to a data collector for processing.

In this work, we are interested in the PSD of the BLE signal emitted by these devices. Their hardware and software architectures have evolved over time and, when this work is carried out, we have three different versions.

To address the influence of environmental conditions on the PSD, we performed experiments in two different environments, namely: ***(1) - An anechoic chamber*** which is an external-wave-isolated chamber; the electronic noise is, therefore, very low in this chamber. ***(2) - In open space***, where the environmental parameters are less favorable to the propagation of the electromagnetic waves.

To address the influence of the data modulating the BLE RF carrier on the PSD, we performed experiments with different transmitted data by performing a spoofing attack on the BLE device, since the identifier is part of the transmitted data. We configured all the BLE devices with the same identifier in order to have them transmit exactly the same data, and we compared the PSDs. We also looked at the case of a same BLE device when it is experimented with two different identifiers.

Table 1 below summarizes the experiments in this study. "Experiment 1" aimed to evaluate the influence of environmental parameters on the PSD. For this purpose, we compared the PSDs of an experiment when carried out in the anechoic chamber with the PSDs of the same experiment when carried out in open space. In this experiment, all the devices kept their original identifier, in contrast to the two other experiments. Indeed, "Experiment 2" and "Experiment 3" aimed to evaluate the influence of the data on the PSD; they involved data modulating the RF carrier of the BLE, i.e., the transmitted data. For this purpose, we changed the identifier of the same BLE device in "Experiment 2" and we assigned the same identifier to four BLE devices in "Experiment 3" knowing that the identifier is part of the data transmitted by the device.

Table 1. Summary of the PSD-based fingerprinting experiments.

Experiment Parameters / Experiment Goal	Experiment Name	Experiment Environment	BLE Device Name	BLE Device Identifier	Distance between BLE Device and Receiver	Number of PSDs (i.e., Number of Transmission)
Assess the influence of environmental parameters on the PSD.	Experiment 1	Anechoic chamber	devC2	E0512496A1C2	2 m	20
			dev48	E05124969648	2 m	20
			devCA	E0512496C3CA	2 m	20
			dev02	B87C6FCD6102	2 m	20
		Open space	devC2	E0512496A1C2	2 m	20
			dev48	E05124969648	2 m	20
			devCA	E0512496C3CA	2 m	20
			dev02	B87C6FCD6102	2 m	20
Assess the influence of the data modulating BLE RF carrier on the PSD.	Experiment 2	Anechoic chamber	devCA	E0512496C3CA	2 m	20
			devCA	E05124969648	2 m	20
	Experiment 3	Anechoic chamber	dev48	E05124969648	2 m	20
			devCA	E05124969648	2 m	20
			dev02	E05124969648	2 m	20
			dev09	E05124969648	2 m	20

In "Experiment 1", our objective was to compare the PSDs in anechoic chamber with those in open space. In "Experiment 2", we had the same device, devCA, which transmitted in first case with its own identifier; then, in second case, with the identifier of dev48, we compared the PSDs of these two cases. In "Experiment 3", the three devices devCA, dev02, and dev09 all used dev48's identifier, so these four devices transmitted the same data. Could we distinguish their PSDs? This question is answered later on, in section "2.4.2. Results".

In "Experiment 3", the devC2 device was replaced by the dev09 device because the transmission power of devC2 (the first ever version) was very low compared to the other devices. This difference would make our PSD comparison more difficult and less reliable because, as it will be demonstrated, the transmission power has an effect on the PSD.

2.3.3. PSDs Normalization and Comparison

To compare two PSDs, we needed a method that can give the difference between them in order to know how close or different they are. Let's consider two signals, s_1 and s_2, whose PSD spectrums are PSD_1 and PSD_2, respectively. The spectrums PSD_1 and PSD_2 are in the same frequency band named B. To compare these two PSDs, we used a correlation method that first normalizes PSDs amplitudes and then compares amplitude differences at certain frequencies, as outlined on Figure 5.

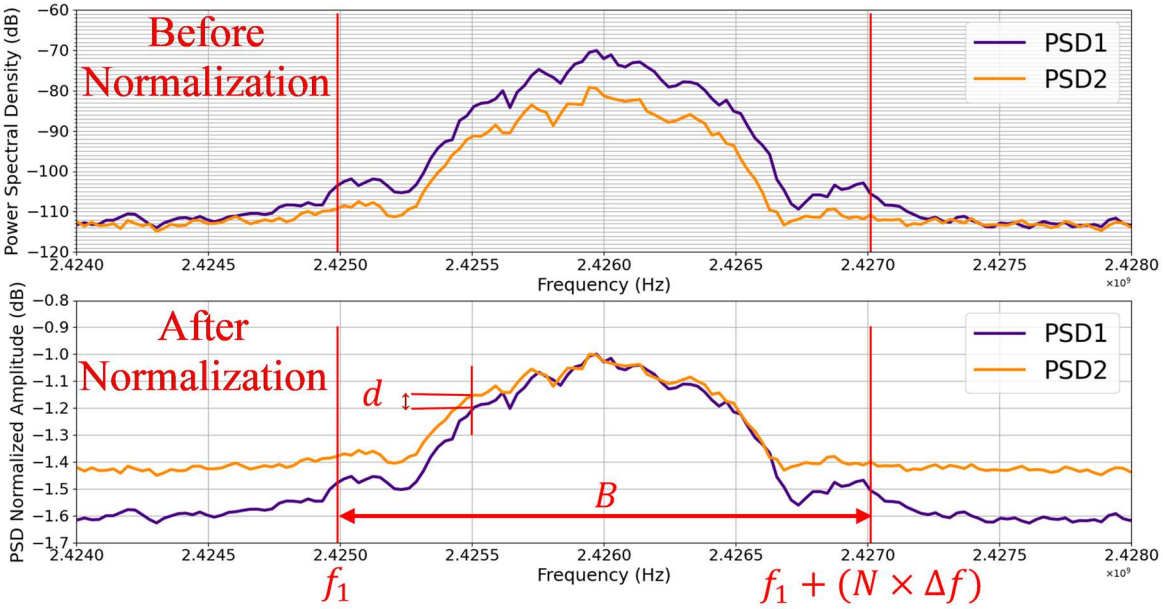


Figure 5. Comparison of two different PSDs before and after amplitude normalization. Parameter d is frequency-dependent and is given by $d = |PSD_1^*(f) - PSD_2^*(f)|$, where * denotes the normalized amplitude. In this figure, d is shown at $f = 2.4255$ GHz as an example.

Within band B (Figure 5), the below conditions (16), (17), (18), and (19) are checked at N frequencies spaced from one to the next by Δf. The parameter $d = |PSD_1^*(f) - PSD_2^*(f)|$ is compared to ε in condition (16). For band B, we chose 2 MHz, which is the channel bandwidth of Bluetooth Low Energy (BLE). This comparison method, which is actually a correlation function, can be presented in two steps as follows.

☞ ***Amplitude Normalization***

As it will be demonstrated, the amplitude of the PSD varies with several parameters (e.g., the distance between the transmitter and the receiver); hence, a comparison approach focused only on amplitude will be less relevant for comparing two PSDs for fingerprinting purposes, especially in a dynamic IoT environment. For this reason, with the formula (14), we will first normalize each of our two PSDs by dividing them with their respective maximum amplitudes.

$$\forall f \in B, PSD^*(f) = PSD(f) \,/\, |PSD_{max}| \qquad (14)$$

Where $PSD(f)$ denotes the amplitude of the spectrum at frequency f, PSD_{max} denotes the maximum amplitude of the spectrum, and $PSD^*(f)$ denotes the normalized amplitude at frequency f.

☞ ***Comparison***

So now the two PSDs have been normalized, their initial amplitude difference is no longer a concern. We are going to use a comparison method which is based on the following idea: provided that they are in the same frequency band, that is B, without any frequency offset between them, the two spectrums PSD_1 and PSD_2 have the same profile if and only if:

$$\forall f \in B, PSD_1^*(f) = PSD_2^*(f) \quad (15)$$

Where PSD_1^* and PSD_2^* are the normalized amplitudes of the PSD spectrums of signals s_1 and s_2, respectively.
However, equation (15) represents an ideal scenario where the two PSDs are perfectly identical, so it is actually a theoretical equation and overoptimistic. In practice, we will allow an acceptable tolerance ε as follows:

$$\forall f \in B, d = |PSD_1^*(f) - PSD_2^*(f)| \leq \varepsilon \quad (16)$$

Note that the comparison does not include all frequencies $f \in \mathbb{R}$ in band B but rather addresses some specific frequencies defined by the PSD estimation function, i.e., the Welch periodogram method. Nevertheless, these considered frequencies increase from one to the next by a fixed step that we call Δf. This Δf is the frequency resolution of the comparison; the lower Δf, the more accurate the comparison because the number of considered frequencies is higher. This frequency resolution Δf is defined by the PSD estimation function as well.

However, at a given frequency f, the fulfillment of condition (16) is necessary but not enough to claim that the two PSDs have the same profile between f and $f + \Delta f$. Indeed, condition (16) could be satisfied in f while the two PSDs evolve in different directions between f and $f + \Delta f$, e.g., PSD_1 rises while PSD_2 falls or vice versa, which is a strong proof of the two PSDs being different. To take into account this important possibility, one of the three conditions below must be satisfied in addition to condition (16).

$$PSD_1^*(f) > PSD_1^*(f + \Delta f) \;\&\&\; PSD_2^*(f) > PSD_2^*(f + \Delta f) \quad (17)$$

$$PSD_1^*(f) < PSD_1^*(f + \Delta f) \;\&\&\; PSD_2^*(f) < PSD_2^*(f + \Delta f) \quad (18)$$

$$PSD_1^*(f) = PSD_1^*(f + \Delta f) \;\&\&\; PSD_2^*(f) = PSD_2^*(f + \Delta f) \quad (19)$$

Where && denotes the logical "AND" condition.

These three conditions are actually a simplified form of the comparison of the first derivatives between f and $f + \Delta f$ of the two PSDs. These conditions are sufficient in this study since the frequency resolution of the comparison is quite good.
Hence, to summarize, we consider that the two PSDs have the same profile between f and $f + \Delta f$ when condition (16) is satisfied and one of the conditions (17), (18), or (19) is also satisfied, i.e., 16 AND (17 OR 18 OR 19). Thus, after going through the entire band B, we compute a similarity percentage S_p between the two PSDs with the following formula:

$$S_p = \frac{N_F}{\frac{B}{\Delta f} + 1} \times 100$$

$$S_p = \frac{N_F \times \Delta f}{B + \Delta f} \times 100 \quad (20)$$

Where N_F is the number of frequencies, out of a total of N, for which condition (16) and one of the conditions (17), (18), or (19) are satisfied.
The higher S_p , the closer the profiles of the two PSDs, provided that they are both in the band B, there is no frequency offset between them, and, finally, the value of ε in condition (16) is acceptable. Indeed, S_p depends on the value of ε, whose choice is empirical in this work. The lower ε, the more relevant the comparison result, whereas the higher ε, the higher the similarity percentage could be, but it will not represent the reality. For example, if we ignore conditions (17), (18), and (19) and assume $\varepsilon = \infty$, the similarity percentage S_p will be always equal to 100% but will have no meaning.

This approach allows us to compare the profiles of the PSDs regardless of the difference between their amplitudes, which is more relevant in a dynamic IoT environment where the devices move a lot, knowing that the amplitude of the PSD is dependent to the distance between the transmitter and the receiver. However, a disadvantage of this approach could be the high number of operations to perform. Indeed, the greater the number N of considered frequencies, that is, the smaller Δf, the higher the computational workload, although the comparison is more accurate. Another advantage of this approach is that it also takes into account the inherent frequency offset devices can have, which is a very important parameter. Indeed, two PSDs with perfectly identical profiles but offset in frequency are not considered identical.

2.4. Experiments and Results

2.4.1. Experimental Setup

The key elements of our experiments are the BLE devices preseneted on Figure 6. The different blocks making up the architecture of these devices are shown on Figure 6(a). The aim was to develop a battery-free wireless module, remotely powered by electromagnetic wireless power transfer in far-field. Thus, for its power supply, the module uses an antenna to capture electromagnetic waves, which are then rectified into DC using a rectifier. For wireless energy transfer, the module's designers chose the 868 MHz and 2.45 GHz ISM band frequencies. In the module shown on Figure 6(a), the designers used a single-band rectifier for each of the two frequencies for RF-to-DC conversion.

The rectified voltage is then used to recharge a storage capacitor via a Power Management Unit (PMU). Two PMUs are present in the module shown on Figure 6(a), a PMU based on the AEM30940 integrated circuit (IC) and a PMU based on the BQ25570 IC. One or other PMU is used to recharge the storage capacitor.

Then, this storage capacitor is used as a power supply for the embedded sensors and the IC that measures and transmits the data. For this IC, the designers chose the ultra-low-power high-performance BLE SoC QN9080 transceiver from NXP Semiconductors. A 2.45 GHz PIFA antenna is used for BLE data transmission. The measured and transmitted data are the temperature and humidity obtained from the HDC2080 sensor and the resistivity (of the

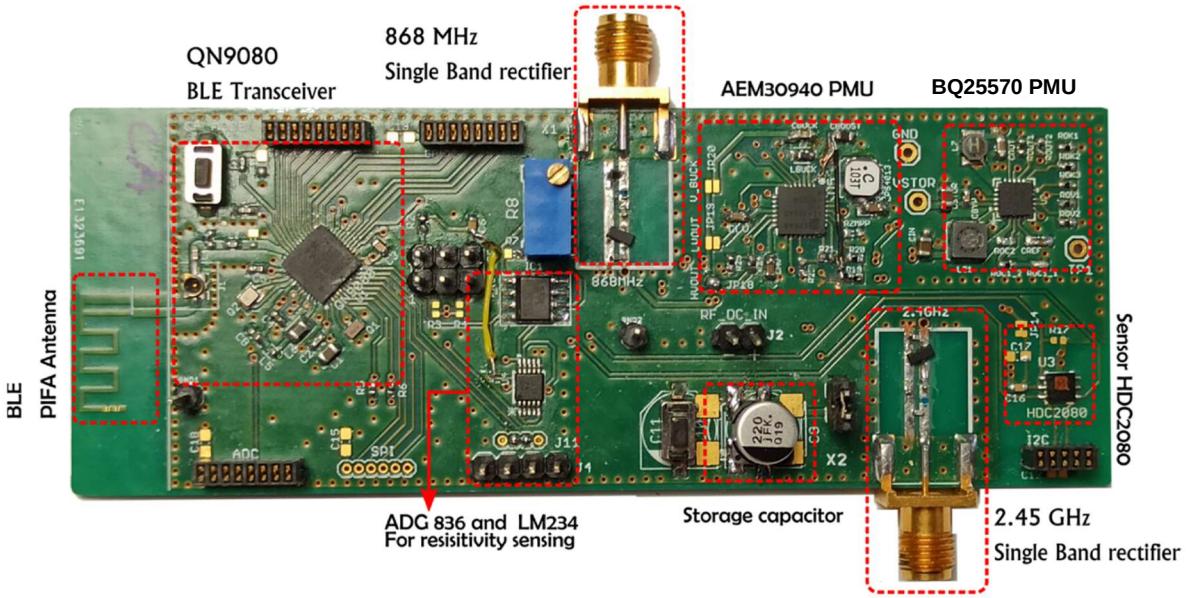
reinforced concrete) obtained from a circuit made up of an ADG836 (an IC containing two independently selectable SPDT switches) and an LM234 (an adjustable current source). Remember, this module was designed at the LAAS-CNRS [73], [74] laboratory as part of a project that aims to monitor the structural health of reinforced concrete throughout its lifetime. Reference [73] provides more information about this BLE module.

At the time this manuscript is written, these BLE modules exist in three versions. Figure 6(b) shows one module from the very first version (v1), two modules from the second version (v2) and two modules from the third version (v3). The module shown on Figure 6(a) is a second-version module. There are no major differences between versions in terms of operating principle. Version three implements a single PMU based on the AEM30940 IC and uses a flexible substrate antenna for wireless energy harvesting. The QN9080 BLE transceiver is used in all three versions. Modules from each version were used in this work. All five modules shown on Figure 6(b) were used in the experiments conducted in this chapter.

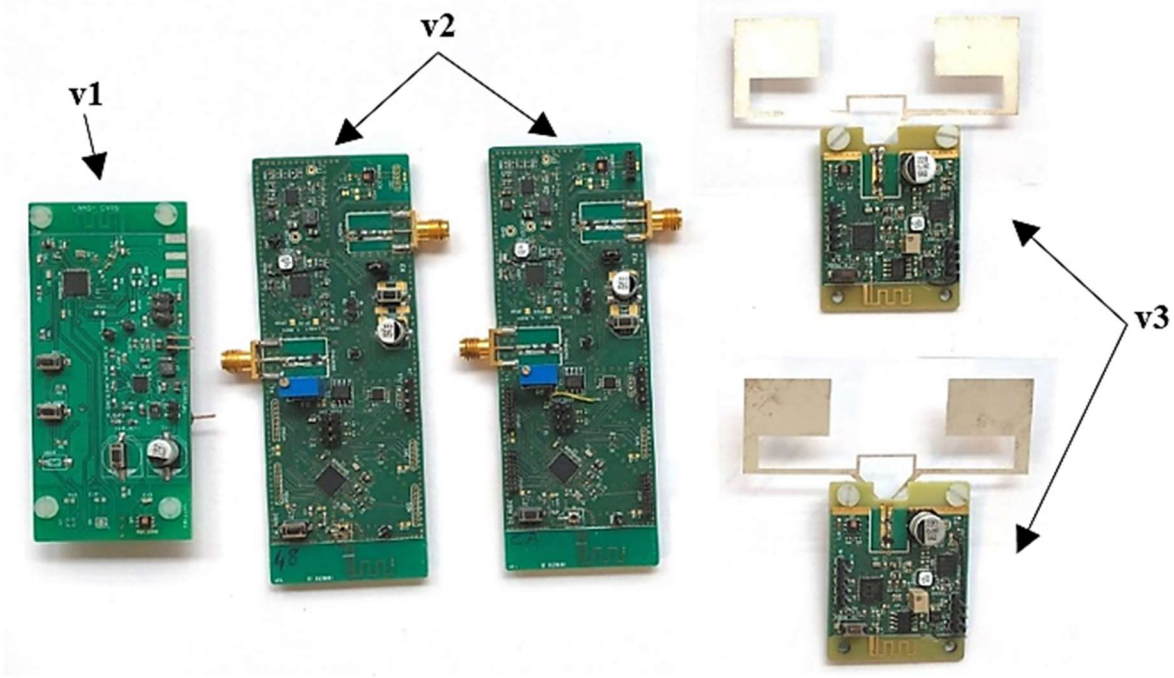
The aim of this work was to find out whether it would be possible or not for a receiver to identify a BLE device that emitted the received signal thanks to the PSD of this signal, and whether this identification remains possible in any environmental conditions and independently of the transmitted data, i.e., data modulating the BLE carrier.

In our experiments, we used the RSA306B real-time spectrum analyzer with SignalVu-PC software for receiving and recording the BLE signal as I/Q samples. Then, we used these data in a python implementation of the Welch's average periodogram method [75] to estimate the power spectral density.

Throughout our experiments, we kept the distance between the transmitter (i.e., the BLE device) and the receiver (i.e., RSA306B) at 2 m for two reasons, namely: ***(1) -*** The dimensions of our anechoic chamber made it difficult to exceed this distance; ***(2) -*** Fortunately, regarding what we were trying to prove about the use of a PSD as a fingerprinting feature, it was not necessary to experiment with different distances between the transmitter and the receiver in this work. Indeed, based on the results of the 2 m experiments, we could already explain the behavior of the PSD with respect to the distance between the transmitter and the receiver. More details on this assertion are provided in section "2.5. Discussion".



(a)



(b)

Figure 6. (a) The different blocks making up the BLE module (source: [72]). **(b)** The BLE devices used in the PSD-based fingerprinting experiments. These devices were designed in the context of the same project; they are different versions of the same product whose hardware and software architectures have evolved over time. Thus, v1 (the board at the left end) refers to the first ever version; v2 (the two boards in the middle) and v3 (the two boards at the right end) refer to the second and third versions, respectively.

Figure 7 shows the experimental setup in the anechoic chamber, which was the same setup used in open-space experiments, and Figure 8 schematizes the experimentation in a more readable way.

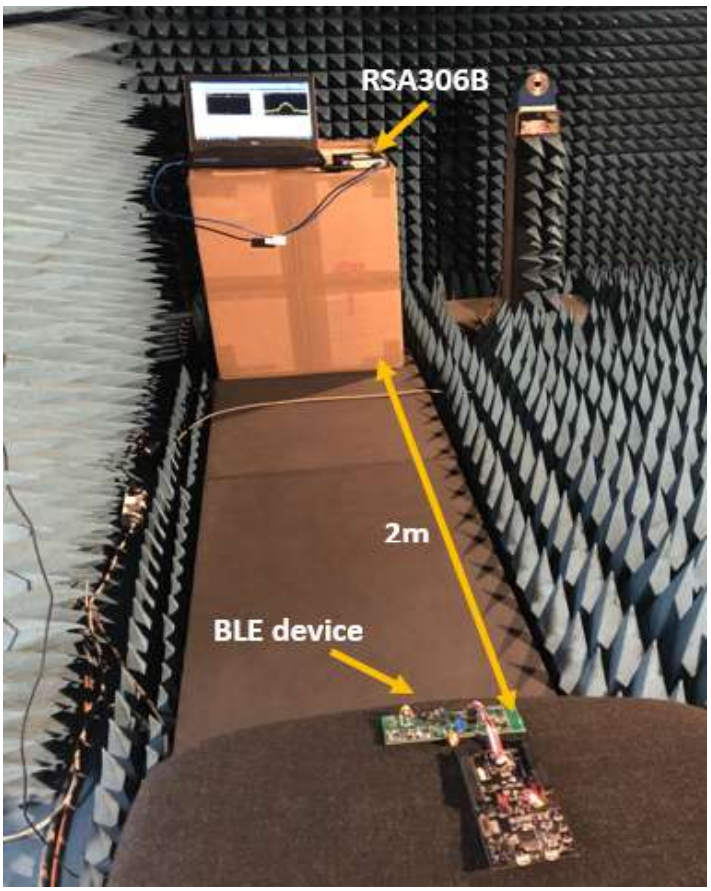


Figure 7. PSD-based fingerprinting experimental setup in the anechoic chamber. The BLE device emits the signal at 2 m from the RSA306B. The latter is equipped with a BLE antenna and driven by the Tektronix SignalVu-PC software. This way, we can capture the BLE signal in real-time and record it as IQ samples on the PC in order to be used later in a python script for estimating the power spectral density using Welch's average periodogram method.

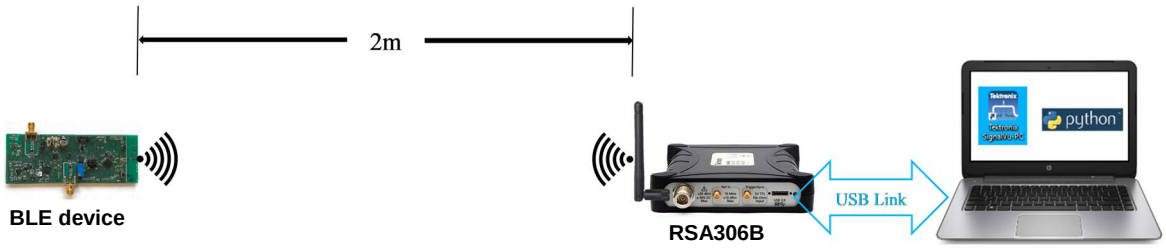


Figure 8. Schematic of the PSD-based fingerprinting experimental setup.

On Figure 7, we can see that the experimentation is quite simple and do not require a lot of equipment. It is, therefore, an easily reproducible experiment and, for this reason, we are going

to give some configuration information about certain components in order to guide those who will try to reproduce this experiment.

We configured the BLE devices in non-connectable advertising mode [76], where they advertise information to any listening device. It is important to know that any transmission of information by a BLE terminal (i.e., via the BLE protocol) starts with a step called the advertising phase, during which the terminal broadcasts information that contains its intentions such as a request to establish a connection with another terminal or a message broadcasted to any nearby listening terminal, as it was the case for the BLE devices in this work.

Among the 40 channels used by the BLE in the 2.402 GHz to 2.480 GHz band, 3 channels are exclusively dedicated to advertising and are called primary advertising channels. At every advertising event, the advertising information is transmitted on these 3 channels one after another. So, we configured our receiver (the RSA306B) to be listened to one of these channels to capture and record the BLE signal each time we triggered the advertising on the BLE device. These devices were similar to BLE beacons where we controlled the triggering of the advertising as well as the advertising data. Thus, we were able to experiment PSD as a function of the data modulating the BLE carrier.

The advertising mode was more suited to our experimentation because it allowed us to avoid the frequency-hopping mechanism used by the BLE protocol when the communication involved the 37 other channels. Indeed, with the frequency selection pattern used by the hopping mechanism being random, the PSD of the signal of the same message changed with the change of the frequency selection pattern, which could have misled us in our analysis of the PSD. More information about this BLE protocol, widely used in IoT, can be found in [77], [78].

The second component to consider in this experiment was the RSA306B real-time spectrum analyzer. This device allowed us to capture and record the BLE signal as I/Q samples in a CSV format file. This device allowed to listen in real-time mode up to 40 MHz bandwidth, which was more than enough for our application. Beyond 40 MHz bandwidth, the RSA306B works in sweep mode. Some of the most important configuration parameters of this device are: the center frequency to be listened to, the span which must be higher than the BLE channel bandwidth (i.e., higher than 2 MHz), the acquisition triggering setup, the acquisition duration, and the sampling frequency. For our experiment, the values of these parameters are given in Table 2.

To guarantee the integrity of our results, we had to record the entire advertising signal. To do so, we had to choose a convenient acquisition duration, that is, a one longer than the duration of the advertising on a channel. To find the duration of the advertising on a channel, we observed the power consumption of the BLE device during advertising. Figure 9 shows the profile of the power consumption of the BLE device during an advertising period; the three peaks represent the three channels. We can see that each peak corresponding to the advertising on a channel lasted only about 0.5 ms. This is why we chose to perform the acquisition over 1 ms from the occurrence of the trigger event; hence, 1 ms is the value of the parameter "Acquisition Length". This was more than enough time to acquire the entire advertising signal on a channel, as witnessed by all the PSD plots in this work.

Table 2. Configuration parameters of the RSA306B real-time spectrum analyzer.

Parameter	Value	Comments
Frequency	2426 GHz	Second advertising channel.
Span	4 MHz	This is two times the BLE channel bandwidth.
Sampling frequency	56 MS/s	
Acquisition Length	1 ms	This is 56k samples.
Acquisition start-up	Triggered on RF input	Trigger level = −55 dBm. Slope = Rise.
Save acq data on Trigger	Checked	File format selected = CSV
Ref Lev	−33 dBm	Internal preamplifier on.

The "Trigger Level" value depends on the power of the received signal; in our case, this power was very low. Fortunately, the RSA306B had an internal preamplifier automatically activated when the value of the parameter "Ref Level" was lower than or equal to −30 dBm.

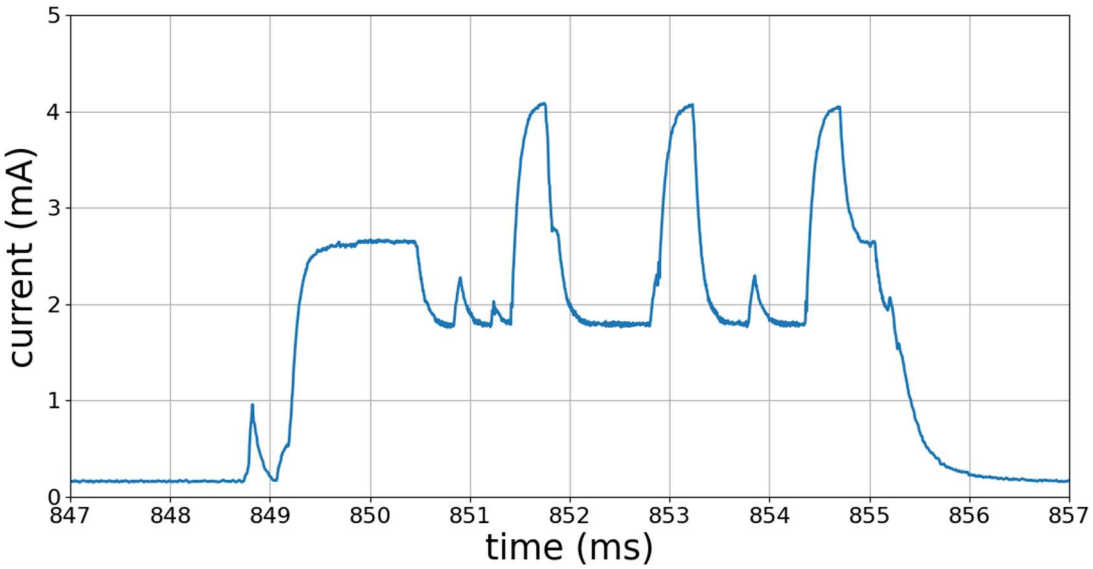


Figure 9. Profile of the power consumption of one of our BLE devices during advertising, i.e., the broadcast of the advertising packet on each of the three primary advertising channels, one after another. Each peak corresponds to the advertising on a channel.

It is worth mentioning that the order of advertising on the three channels was random, although this did not affect our acquisition which was only triggered when the advertising arrived on the channel our receiver (the RSA306B) was listening to, i.e., the second channel.

2.4.2. Results

In this section, we will present the results of our experiments.
First of all, we will present an example of 20 PSDs (20 transmissions) from one of the BLE devices at 2 m from the RSA306B receiver in order to check for possible dynamics on the PSD from one transmission to another while all the parameters (environment, data, distance, etc.) have remained unchanged.
This section will also present the practical application of our comparison approach.
Next, we will move on to the results of our experiments starting with "Experiment 1", the comparison of the PSDs from anechoic chamber experiments with those from open-space experiments.

We will close this section with the results of the experiments "Experiment 2" and "Experiment 3" by examining the PSDs for different data, that is, data modulating the BLE carrier, i.e., the data transmitted by the BLE device.

☞ ***Experiment 0: Example of 20 PSDs' Acquisition***

Figure 10 shows 20 PSDs from the same device in a static experimental setting, i.e., no parameter changed during the 20 transmissions.

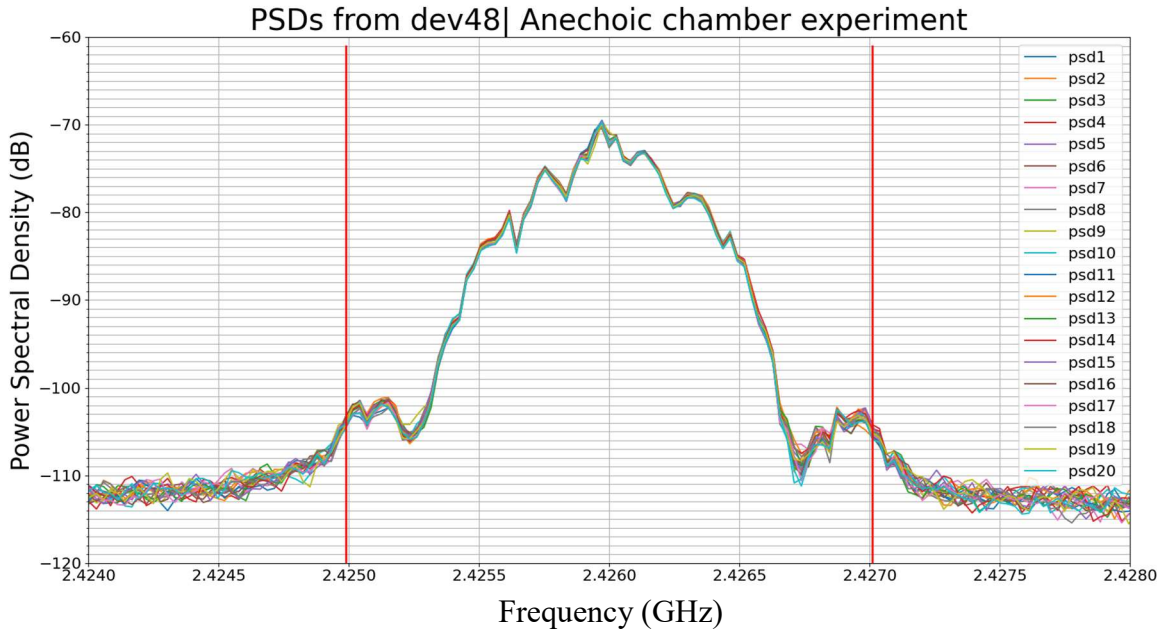


Figure 10. Twenty PSDs of a same BLE device in a static experimental setting. The region between the red lines is the band B where the PSDs are compared; we chose B = 2 MHz, which is the channel bandwidth of the BLE. So, we ignored the ambient noise to have a more relevant analysis.

On Figure 10, we can see that the plot formed by the 20 PSDs was quite thin especially in the band of interest, B, the region between the two vertical red lines. We chose signals captured in an anechoic chamber because if their PSDs showed differences, then this dynamic would be worse in open space where there is more noise than in an anechoic chamber.
At first glance, one could conclude that these 20 PSDs had exactly the same profile. But to be sure, we are going to compare each one with another one, that is, 10 pairs in total. Table 3 shows the results of this comparison knowing that the region between the two red lines (Figure 10) is the one considered by the comparison.

The width of this region, which we call the band B, is: B(Hz) = 2,427,011,718.75 − 2,424,988,281.25 = 2,023,437.5. The frequency resolution Δf computed by the PSD estimation algorithm is Δf(Hz) = 27,343.75. Thus, our comparison algorithm checks whether condition (16) and one of conditions (17), (18), or (19) are satisfied or not at N = (B/Δf) + 1 = 75 frequencies within interval [2,424,988,281.25, 2,427,011,718.75] (Hz) for two values of ε, namely: ε(dB) = 0.025 and ε(dB) = 0.05. Remember that these values are completely empirical. The smaller ε, the more severe the comparison, but the more it is representative of the reality.

The parameter N_F is the number of frequencies, out of a total of N = 75, at which condition (16) and one of conditions (17), (18), or (19) are satisfied.

Table 3. Results of the comparison of the 20 PSDs.

Pair of PSDs	ε(dB) = 0.025		ε(dB) = 0.05	
	N_F	S_p (%)	N_F	S_p (%)
psd1 vs. psd2	69	92.00	69	92.00
psd3 vs. psd4	67	89.33	70	93.33
psd5 vs. psd6	70	93.33	70	93.33
psd7 vs. psd8	71	94.66	71	94.66
psd9 vs. psd10	67	89.33	70	93.33
psd11 vs. psd12	73	97.33	74	98.66
psd13 vs. psd14	71	94.66	71	94.66
psd15 vs. psd16	70	93.33	70	93.33
psd17 vs. psd18	70	93.33	73	97.33
psd19 vs. psd20	67	89.33	68	90.66

Ideally, the percentage of similarity S_p should have the value of 100% in all the cases in Table 3 but we can see that there is no case of a 100% similarity percentage according to our algorithm. This is the proof that even in a static experimental setting, the PSD may have some dynamics, i.e., the PSD may have minor deviations, from one transmission to the next.
However, one can notice that the value of ε had only a slight influence on the value of S_p in this case, which makes sense since the 20 transmissions were being performed at the same distance from the receiver and in the same environmental conditions, so the amplitude variation was quite small. In this case, the changes in the value of S_p came mainly from the minor variations in the profile of the PSDs.
Note that the values of band B, frequency resolution Δf, and, therefore, N=75 remained the same for all PSDs comparisons in this work.

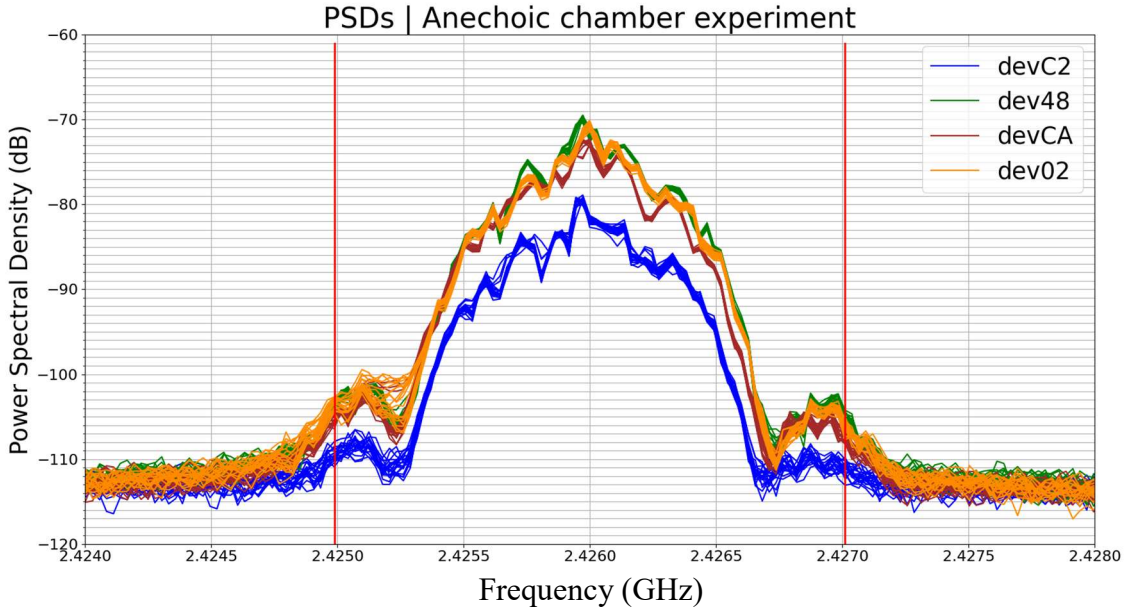
☞ ***Experiment 1: PSD Measurement in Various Environment***

The two graphs (a) and (b) on Figure 11 show the PSDs of the BLE signal for different BLE devices.

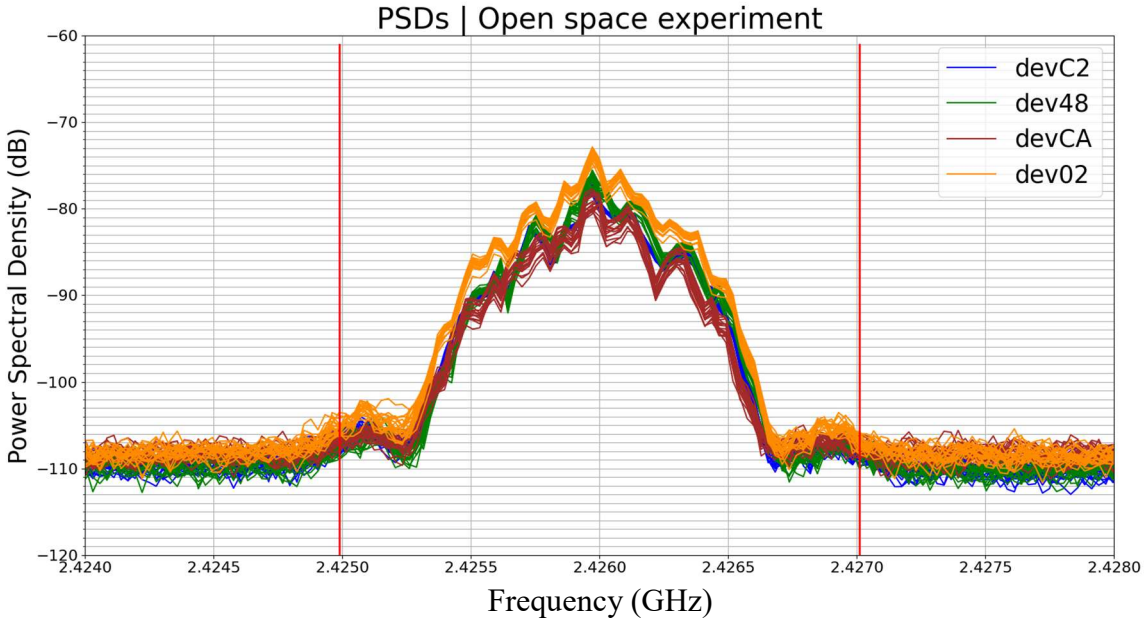
In each graph, each color actually represents a set of 20 plots (i.e., 20 PSDs) corresponding to 20 transmissions of the same data by the same BLE device at the same position and distance from the receiver (the RSA306B). This is true for each color in a graph in this work.
From one graph to the other, plots of the same color represent the same BLE device, e.g., the green plots in graph (a) and the green plots in graph (b) are all PSDs of the device dev48, while plots of graph (a) correspond to the transmission in the anechoic chamber and the ones of graph (b) correspond to the transmission in open space.

It is also worth remembering that, in this experiment, the transmitted data are different from one device to another since the BLE device identifier is part of these data. All the devices had their original unique identifier without any change. Graph (a) shows the results of the experiment in the anechoic chamber and graph (b) shows the results of the same experiment in open space.

We can notice that the amplitudes of the plots on graph (a) are higher than the amplitudes of the plots on graph (b); for example, the green plots have an amplitude of around −69.5 dB on graph (a) and −75.5 dB on graph (b). Actually, the amplitude of the PSD reflects the amount of power received by the receiver; the higher the received power the higher the amplitude of the PSD. The lower the received power, the lower the amplitude of the PSD, and the more the PSD's profile changes.



(a)



(b)

Figure 11. (a) PSDs of anechoic chamber experiment **(b)** PSDs of open-space experiment.

This difference of amplitudes between the graph (a) plots and graph (b) plots means that under the same conditions except environmental, the amount of power arriving at the receiver in the anechoic chamber is higher than the amount of power arriving at the receiver in open space. This is already noticeable at only 2 m between the transmitter and the receiver.
It can also be noticed that on Figure 11, in the case of the same device (plots of the same color from one graph to the other), the graph (a) PSD plots are thinner than the graph (b) PSD plots. This means that the amount of power arriving at the receiving antenna from one transmission to another is more stable in the anechoic chamber than in open space. The channel has a significant influence on the signal and does not allow for stable PSDs.

Despite this evidence, we carried out comparisons using our comparison algorithm. The goal of this experiment was to analyze the influence of environmental parameters on the PSD of a given device; hence, comparing the PSDs of two different devices was of no interest. Instead, what was interesting was to compare the PSDs from the anechoic chamber with the ones from open space for the same device, this is what we carried out. So, we compared the plots of the same color from one graph to the other. As previously mentioned, each color on a graph is actually a set of 20 plots, so we compared the averages rather than performing a 20 versus 20 comparison. Therefore, the process consisted of calculating the average of one of the groups of 20 PSDs of a graph, normalizing this average and, finally, comparing it with the normalized average of the same group of 20 PSDs of the other graph, with the compared PSDs being the same color on the two graphs. For instance, for dev48, we compared the normalized average of the green plots in graph (a) with the normalized average of the green plots in graph (b). Table 4 shows the results of this comparison.

Table 4. Results of the comparison of the average PSDs of anechoic chamber experiments with the ones of open space experiments for each BLE device.

BLE Device Name	ε(dB) = 0.025		ε(dB) = 0.05	
	N_F	S_p (%)	N_F	S_p (%)
devC2	65	86.66	65	86.66
dev48	39	52.00	46	61.33
devCA	32	42.66	40	53.33
dev02	31	41.33	56	74.66

The parameters B and Δf keep the same values as in Experiment 0.

These results (Table 4) show that the higher the difference of amplitudes, the more influential ε is and the more the profiles of the PSDs are different according to our comparison algorithm. Indeed, on Figure 11(b), one can notice that the lower the amplitude, the more abrupt the slopes of the plots, and the greater the change of the profile of the PSD. This fact is related to the hardware and computing methods of the PSD and not related to the BLE device or to the signal. When the amplitude decreases, a sort of compression occurs which causes the details of the plot to disappear, i.e., the PSD's profile is changing.

These results highlight the effects of environmental parameters on the PSD of the received signal. Increasing the distance between the transmitter and the receiver would amplify even more these effects of environmental parameters on the PSD. Indeed, in the case of RF wireless communications, any parameter influencing the amount of power arriving at the receiving antenna also influences the PSD. The transmitted power, the distance between the transmitter antenna and the receiver antenna, the positioning of the transmitting antenna in relation to the

receiving antenna regarding polarizations, multipath propagation and fading effects, the temperature, the relative humidity, and interferences with other transmitters are some of these parameters.
However, observing graphs (a) and (b), one can notice that the profile of the PSD is not as sensitive to the change of environmental parameters as its amplitude is.

☞ ***Experiment 2: PSD vs. Transmitted Data, a Case of the Same Device Transmitting Two Different Identifiers***

As previously mentioned, these BLE devices, designed at the LAAS-CNRS [73], [74] for structural health monitoring of the reinforced concrete, measure some physical parameters (e.g., temperature, relative humidity, and resistivity) and wirelessly transmit these data via the BLE technology. These physical parameters, including temperature, measured by the BLE devices are obviously not stable and could change on-the-fly according to the weather. So, in order to always transmit exactly the same data over time, we assigned constant values to these parameters by hard-coding them in the firmware and, thus, ignored the values measured by the different sensors of the BLE device. In this way, we performed data-related experiments, that is, this experiment (Experiment 2) and the next one (Experiment 3), by changing only the identifier of the BLE device as the identifier was part of the data transmitted by the device.

In this experiment (Experiment 2) carried out inside the anechoic chamber, we used one and only one BLE device. Firstly, we performed the experiment using the device with its original identifier without any change. Secondly, we changed only the device identifier and we repeated the same experiment. The results are shown on Figure 12.

The two groups of 20 plots each on Figure 12 are from the same BLE device, devCA. The blue plots are PSDs of this device, devCA, when using its original identifier (E0512496C3CA) and the dark-orange plots are PSDs of this same device, devCA, when using the device dev48's identifier (E05124969648). We have spoofed the identity of the device dev48.

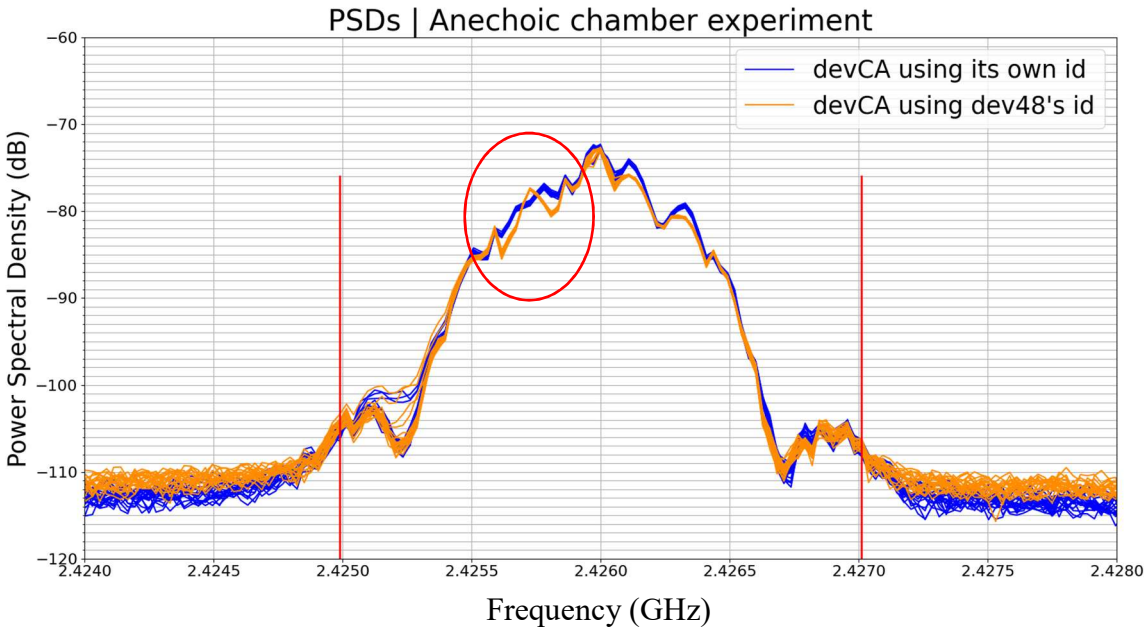


Figure 12. PSDs of the same BLE device but measured with two different identifiers.

We compared the PSDs (the average of the 20 PSDs) of the device devCA when it had its real identifier with the PSDs when it had the identifier of device dev48. This means, we compared the normalized average of the blue plots with the normalized average of the dark-orange plots. Table 5 shows the results of this comparison.

Table 5. Results of the comparison of the PSDs of the same device transmitting two different identifiers.

BLE Device Name	ε(dB) = 0.025		ε(dB) = 0.05	
	N_F	S_p (%)	N_F	S_p (%)
devCA	64	85.33	68	90.66

The parameters B and Δf keep the same values as in Experiment 0.

In this comparison, ε, again, has less influence on the similarity percentage S_p. This makes sense as we can see on Figure 12 that the PSDs have almost the same amplitude, so the value of S_p is more impacted by the difference between the profiles than the difference of amplitudes. So, although being the same device, the PSDs of the device when using its original identifier are different from the PSDs of the same device when using another identifier. The part circled in red on Figure 12 highlights some apparent differences. Remember that the identifier is part of the transmitted data.

One can already notice that this time, the profile of the PSD is more sensitive to the changes of transmitted data compared to its amplitude. While, in the previous experiment (Experiment 1), we found that the amplitude of the PSD was more sensitive to environmental conditions compared to its profile.

The next experiment (Experiment 3) is an extended version of this experiment (Experiment 2). Indeed, in "Experiment 3", we assigned the same identifier to four different BLE devices, in order to make them always transmit exactly the same data.

☞ ***Experiment 3: PSD vs. Transmitted Data, a Case of Four Different Devices Transmitting with the Same Identifier***

In this experiment, we carried out identity spoofing by giving the identifier of one of the 4 BLE devices to the 3 others. We ended up with 4 BLE devices always transmitting exactly the same data. The results are shown on Figure 13.

The plots on the graph on Figure 13(a) are an overlapping of the 4 groups of plots on the graphs on Figure 13(b). Previously, on Figure 11(a), where the transmitted data are different from one device to another, we can see that the PSDs are much more distinguishable than they are on Figure 13(a), where devices are transmitting the same data. As usual, each color on a graph is actually a set of 20 plots corresponding to 20 PSDs, i.e., 20 transmissions, from the same device.

Here, we compared the PSDs, that is, again, the normalized average of the 20 PSDs of each device with those of the three others, that is, a combination of 2 among 4; hence, 6 pairs of PSDs, as shown in Table 6.

These results in Table 6 are very interesting. One can notice that the comparisons in which devCA is involved have more often low similarity percentages and these cases are also where the value of ε is the more influential; this is due to the inherent frequency offset between devCA

and the other devices. For instance, Figure 14 shows the PSDs of dev48 (green plots) and devCA (brown plots), and we can see that the PSDs of devCA are shifted from dev48's PSDs. This is the case where the similarity percentage is the lowest.

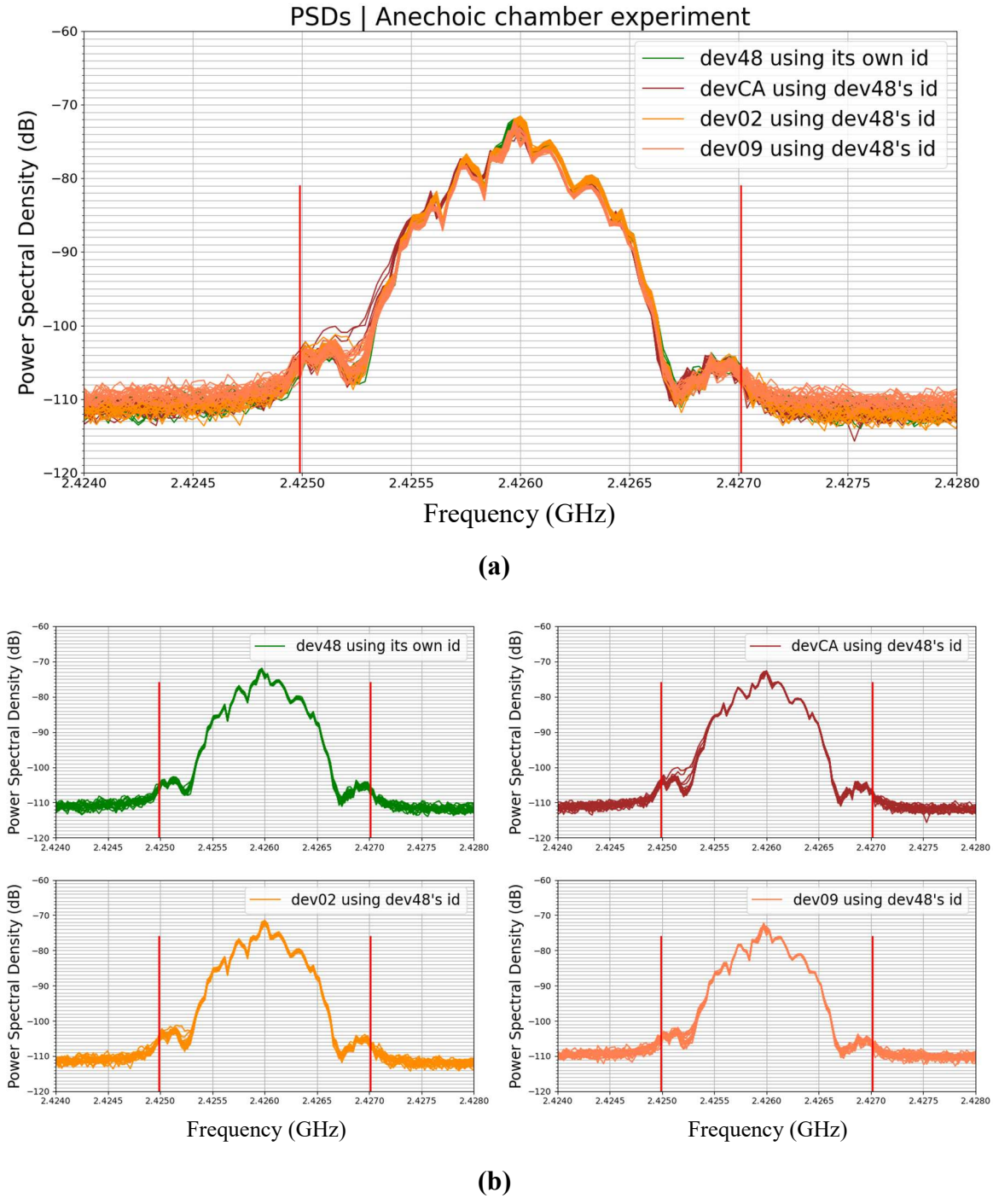


Figure 13. PSDs of four different BLE devices using the same identifier (the one of dev48) and, thus, always transmitting exactly the same data. Graph (**a**) plots are an overlapping of graph (**b**) four groups of plots.

We could have corrected this frequency offset before comparing the PSDs but we chose to show it because it may be an interesting point for the readers of this work. What is interesting about

the frequency offset is that it is inherent to the device; this is why it has even been used as a fingerprinting feature in some work, for instance, in [50].

Table 6. Results of the comparison of the PSDs of four BLE devices transmitting the same data.

Pair of PSDs	ε(dB) = 0.025		ε(dB) = 0.05	
	N_F	S_p (%)	N_F	S_p (%)
dev48 vs. devCA	38	50.66	51	68.00
dev48 vs. dev02	67	89.33	68	90.66
dev48 vs. dev09	58	77.33	62	82.66
devCA vs. dev02	41	54.66	54	72.00
devCA vs. dev09	57	76.00	64	85.33
dev02 vs. dev09	57	76.00	63	84.00

The parameters B and Δf keep the same values as in Experiment 0.

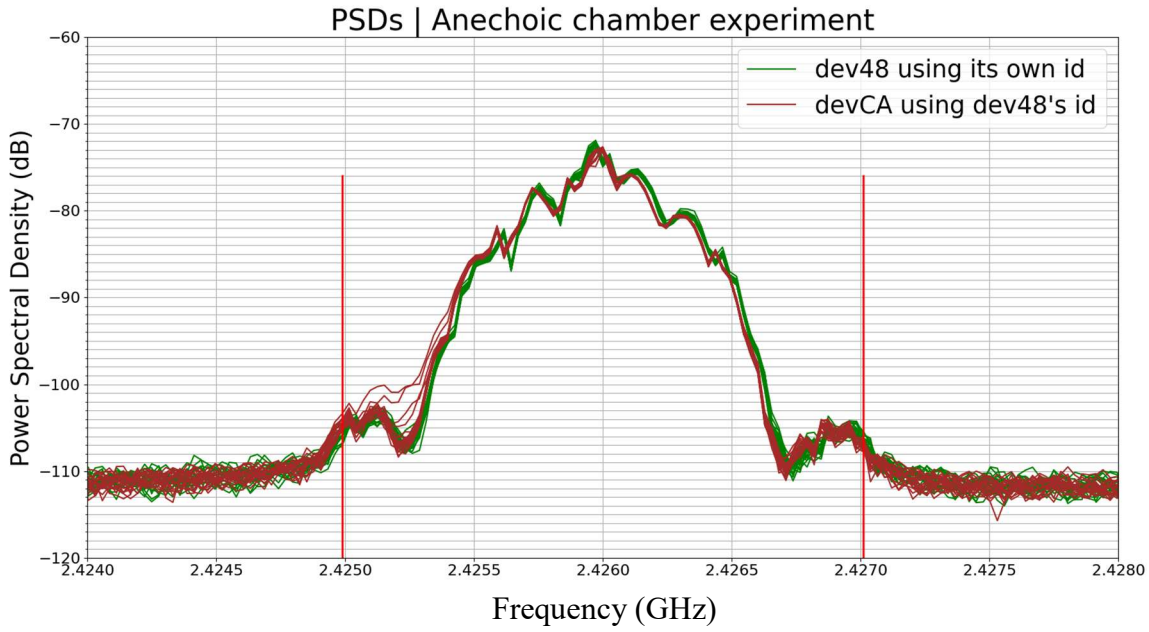


Figure 14. PSDs of dev48 (green plots) and devCA (brown plots) BLE devices using the same identifier (the one of dev48) and, thus, always transmitting exactly the same data.

The other parameter that we could consider is amplitude, but the latter had almost no influence on the results of this comparison since the 4 average PSDs had almost the same amplitude. Indeed, the average PSD amplitudes were −72.23 dB, −72.88 dB, −72.07 dB, and −72.90 dB for dev48, devCA, dev02, and dev09, respectively.

So, the main factor that makes the results in Table 6 not an exact representation of the reality is the frequency offset. However, the profiles of the PSDs were not exactly the same either; it can be seen, for example, that there is a hollow at the center frequency (2.426 GHz) of the green plot (dev48 using its own id). These slight differences may actually come from hardware imperfections of the devices (local oscillators, filters, power amplifiers, etc.). However, these slight differences will be difficult to exploit in the context of fingerprinting in order to identify

these devices. Indeed, a fingerprinting algorithm, that does not take frequency offset into account, could conclude that these four groups of plots are from the same device and not from four different devices.

As in the previous experiment (Experiment 2), the profile of the PSD is again more sensitive to the changes of transmitted data than its amplitude. Indeed, the profile of the PSD changed a lot with the change of the data modulating the RF carrier; this is the most important information in this experiment.
We also note that the similarity percentage between 2 PSDs from 2 different devices, for the same data, in the anechoic chamber (Table 6) can be greater than the similarity percentage between 2 PSDs from the same device, for the same data, but from two different environments (Table 4).

2.5. Discussion

This work highlighted the weaknesses of PSD as a fingerprinting feature. The stability under any circumstance of the PSD's two most important parameters, amplitude and profile, for a given device, is essential for the use of the PSD as a fingerprinting feature. However, our experiments pointed out the limits of the robustness of these two parameters to the environmental conditions and to the transmitted data, i.e., the data modulating the RF carrier. We have shown that the PSD is not resilient to the changes of either environmental parameters or transmitted data.

As expected from a theoretical point of view, we found that the amplitude of the PSD was a quantification of the amount of RF power arriving at the receiving antenna. The higher (lower) this power is, the higher (lower) the amplitude of the PSD. Therefore, all parameters affecting this power also affect the amplitude of the PSD, and among these influencing parameters are the environmental ones such as free-space-loss attenuation, polarization mismatch, multipath and fading effects, interferences, and so on. In addition, the operating temperature can influence the hardware stability of both the transmitter and the receiver, which can impact RF signal properties.

That is why, to answer our question of whether or not the PSD would be a relevant fingerprinting feature, it is not necessary to carry out the experiments for different distances between the transmitter (the BLE device) and the receiver (the RSA306B). The amount of received power will just be decreasing as the distance increases or the other way around; the amount of received power will be increasing as the distance decreases. Therefore, the amplitude of the PSD will simply evolve in the same way.
However, the profile of the PSD was less sensitive than its amplitude to the variations of the amount of received power, but below a certain level of this power, even the profile of the PSD started to distort.

When it comes to data modulating the RF carrier, their influence on the PSD was more hindering than the influence of the environmental parameters to the use of the PSD as a fingerprinting feature. Indeed, while the environment had more influence on the amplitude of the PSD, the data had more influence on the profile of the PSD, which is more difficult to handle in a fingerprinting context.

Actually, the PSD is signal-specific, the signal changed with the transmitted data, and the PSD is specific to the signal; in other words, the PSD varied according to the data, which makes it

completely useless as a fingerprinting feature in a dynamic IoT environment where the data transmitted by a device can be different from one transmission to another.

Therefore, an environment wherein the PSD could be considered as a relevant feature for fingerprinting would be a static IoT environment with static devices, with each device transmitting always the same message (data) or a limited number of predefined messages, like a home automation facility.

Even for such an environment, fingerprinting by PSD is not that efficient. Indeed, if it is performed in a centralized way (ex-situ fingerprinting) as it was in this work, a simple way to confuse the fingerprinting system would be to get close enough to a device and transmit the same data as this device with the same transmission power.

In addition, these experiments, conducted on several BLE technology-based devices, showed that a collision is highly possible. Considering the large number of devices that can make up a Wireless Sensor Network (WSN), this probability is really not negligible. So, the device authentication cannot rely on PSD alone, while today's IoT environments can include thousands of devices and there are billions of them already deployed around the world.

However, the PSD can be part of the features of an Intrusion Detection System (IDS). These PSD analysis are particularly interesting, since, as stated in [79], the efficiency of an IDS relies heavily not only on its implementation techniques but also on the relevance of its features and their extraction method. Indeed, relevant features allow increasing the accuracy and reducing the detection time of the IDS. This work shows that the PSD, if part of an IDS's features, even in a static IoT environment, may introduce a large number of false positives if the comparison is too severe or a large number of false negatives if the comparison is less severe. Remember that features refer to properties that fingerprinting uses to establish a signature for a given device or that an IDS uses to determine whether a communication is legitimate or not.

Fingerprinting remains an attractive approach for IoT security, due in part to its genericity, which is one of the most needed characteristics for IoT security solutions. Indeed, fingerprinting is a protocol-independent approach, which makes it suitable for IoT device networks that are sometimes highly protocol-heterogeneous, with devices using different protocols including proprietary ones. Different carrier frequencies and modulation patterns can also be present in an IoT network.

However, there are still several challenges to overcome, such as finding relevant fingerprinting features, unwanted parameter effects management, as well as fingerprinting solutions cost optimization. Indeed, one of the weaknesses of fingerprinting is that the more the devices in a network are similar, the more likely it is to have some signatures so close that it is difficult to disassociate their corresponding devices. For example, the probability of successfully distinguishing several instances of the same device, manufactured in a chain on the same day, will be low compared to cases with completely different devices. This is why fingerprinting requires the use of very precise algorithms and, therefore, very powerful and expensive hardware to increase the efficiency of this approach.

3. Conclusion

In this chapter, we have highlighted the possibilities and challenges of fingerprinting-based device identification.
The fingerprinting is limited by several challenges, such as:

- The electronic devices instability regarding physical environmental phenomena (temperature, electromagnetic interference, etc.) as well as the impact of aging on the devices.
- The dynamic nature of the wireless transmission channel, i.e., Channel properties are not constant and can change over time.

These phenomena make it very difficult to associate a time-independent (i.e., constant over time) signature (fingerprint) with a given device. Our hands-on work on the PSD as fingerprinting feature illustrates this reality by showing that different devices can produce the same PSD and a same device can produce different PSDs, depending on the transmitted message and the experimental environment. Consequently, the use of PSD as a fingerprinting feature is only conceivable in a static IoT environment where devices do not move, the transmitted messages do not change and the ambient conditions remain sufficiently stable. In conclusion, PSD alone as a fingerprinting feature is not enough for fingerprinting-based devices identification in a dynamic IoT environment.

Another limitation of fingerprinting is, paradoxically, the advances in manufacturing processes, which reduce more and more device imperfections, making it very difficult to associate a unique signature (fingerprint) with a given device. For example, it would be difficult to build two different signatures for two copies of the same IoT product manufactured on the same production lines, unless very powerful and extremely accurate, therefore very expensive and energy-intensive, computation machines were used for the fingerprinting.

The challenge for fingerprinting remains finding relevant fingerprinting features that are free from the aforementioned limitations, thus allowing to associate with a given device a unique and lifetime-constant signature (fingerprint) built from its inherent imperfections.

As an example, we analyzed, through analytic development and experimental results, the behavior and reliability of PSD under the influence of environmental parameters and the changes of the transmitted data. Our results prove that the PSD cannot be used alone and without any precaution as a reliable fingerprinting feature in any IoT environment for several reasons.

Firstly, the amplitude of the PSD is sensitive to the environmental parameters related to the communication channel; secondly, the profile of the PSD is sensitive to the transmitted data. There are also other parameters (not considered in this study), e.g., device aging, that can impact the PSD over the time. Therefore, the PSD cannot be considered as a relevant fingerprinting feature. We have come to the following conclusion: using PSD alone as a fingerprinting feature in a real-world IoT environment is not a reliable solution for identifying devices that can continuously change their position and transmit different data from one time instant to the next.

Nevertheless, the variation of the amplitude of the PSD will be easy to handle in a fingerprinting context by using a normalized amplitude approach instead of absolute amplitude, for example. Actually, the amplitude will not be as important as the profile in identifying a transmitter using the PSD of its emitted signal. Two PSDs with the same profile but different amplitudes are likely from the same device that will have transmitted at two different distances from the

fingerprinting receiver. On the other hand, it will be difficult to identify a device using the PSD of its emitted signal if the profile of this PSD changes from one transmission to the next.

The ideal IoT environment for using the PSD as a fingerprinting feature will be an IoT environment where the environmental parameters are not changing, the multipath and fading effects related to propagation channel are very limited, the IoT devices are not moving, and each device is always transmitting the same data on the same frequency with the same modulation. However, these conditions, especially the last one, are not realistic. The IoT environments meeting these conditions are not numerous. So, PSD fingerprinting is not a robust device identification solution and would not be suitable for most real-world IoT environments.

Finding solutions to manage the limitation of the fingerprinting approach based on PSD is among the new directions for device fingerprinting research. Some dedicated hardware can be added for fingerprinting purposes on the transmitter and/or the receiver and a dedicated/standardized bit frame (for PSD fingerprinting analysis/identification) can be used in the beginning of a secure (with the fingerprinting approach) wireless communication.

Additionally, the results presented in this work can be helpful in any study involving PSD, even for applications other than fingerprinting.

In the next chapter, we propose a new security technique for IoT devices that consists of authenticating these devices (or messages) using radio wave polarization.

Chapter 4: Polarization Shift Keying (PoSK)-based Message Authentication System

1. Introduction

An innovative security approach based on Polarization Shift Keying is presented in this chapter. One of the fundamental properties of a propagating electromagnetic wave is its polarization, which is antenna-dependent and does not usually change during the transmission, in common environments. The Polarization Shift Keying consists of a dynamic change of the used polarization during transmission. The proposed security approach uses this dynamic control of the polarization (i.e., Polarization Shift Keying) to simultaneously transmit the message and authentication data associated with this message. This is a kind of second modulation of the main message where the different used polarizations denote the symbols of this meta-modulation. Then, a security gateway retrieves these symbols and recovers the authentication code. This creates a secure communication link between the two terminals at the physical layer. Basically, this is a cryptographically controlled modification of the polarization of the emitter outgoing wave. By analogy with the frequency hopping mechanism – used, for instance, in Bluetooth communication –, this can be called a polarization hopping mechanism. The Polarization Shift Keying mechanism do not alter the original waveform and its primary modulation enough to increase the bit error rate. This work shows that this technique is a promising way of achieving secure communications in IoT where devices are resource-constrained in terms of processing, memory and energy. Moreover, this technique is independent of the communication protocol used.

The proposed PoSK-based message authentication system is designed to prevent the attacks previously mentioned in the section "6. Threat Model", by associating with each message an authentication code that is transmitted and received simultaneously with the message. For instance, by replaying even a valid command to a targeted receiver (e.g., unlock car doors), the latter will ignore the command if the authentication code associated with it, is missing or false. This system allows to validate the authenticity of a message and to ensure that it is coming from the expected transmitter.

To the best of our knowledge, this work uses for the first time PoSK as secondary modulation to simultaneously transmit a message and its authentication code. The mechanism consists in dynamically controlling polarization of the RF wave during its emission. The polarization switching pattern is based on a cryptographic algorithm. Then, the receiver needs to retrieve this cryptographic code to authenticate the received message.

2. PoSK-based Message Authentication System

2.1. Overview

Figure 15 shows an overview of the PoSK-based message authentication system architecture, where the standard antennas are replaced by Active Antenna Systems (AAS). An AAS refers to an antenna that provides multiple polarizations, together with a Cryptographic Polarization Controller (CPC). The latter dynamically controls the polarization of the outgoing RF signal following a cryptographic algorithm (e.g., AES) that has been initialized with a secret key shared between the two terminals. The Security Gateway is in charge of authenticating the message from the wireless sensor by retrieving the authentication code (i.e., the polarization switching pattern) used by the Wireless Sensor when transmitting the message.

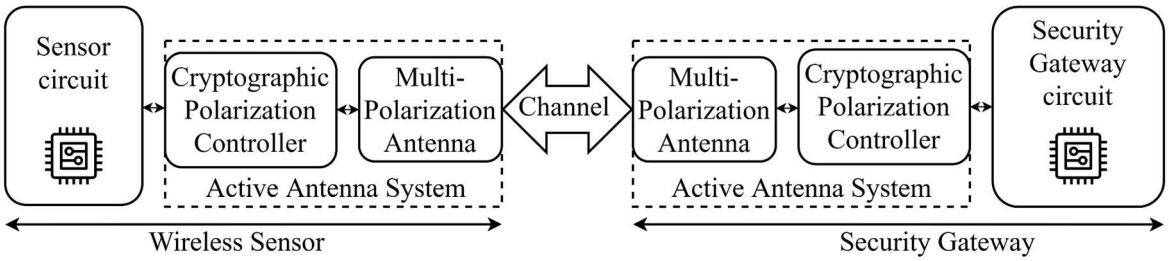


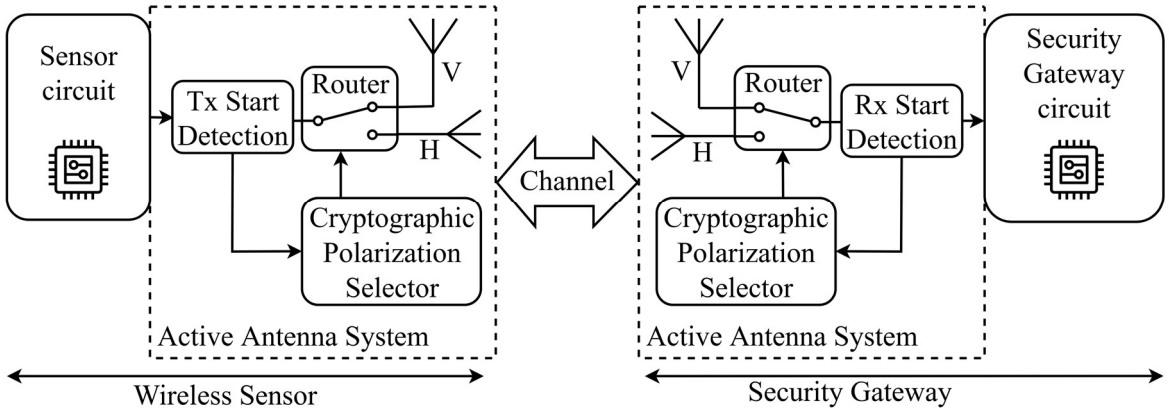
Figure 15. Overview of the PoSK-based message authentication system.

This is the way PoSK-based message authentication is performed. It is a kind of second modulation of the radio wave, where the used polarizations represent the symbols of this second modulation. It is a non-invasive security approach since the only part of the device affected is the antenna part; no other hardware or software modifications are required; all initial parameters, including primary modulation, are kept.

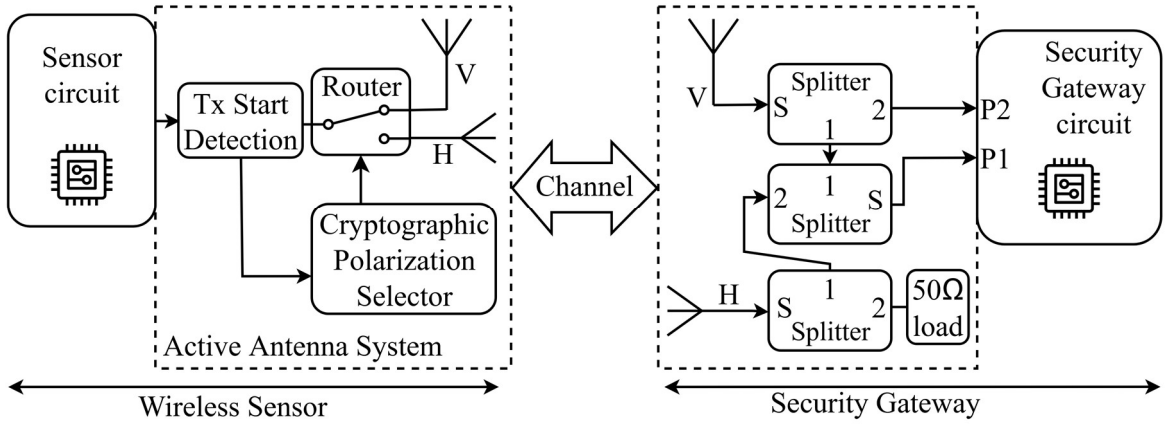
As the primary objective of this initial work is to validate the approach, it is dedicated to the 2-state PoSK (Binary-PoSK or 2-PoSK), i.e., the use of two polarizations. With the right cryptographic polarization controller, it is possible to use either linear or circular polarizations.

2.2. Architecture design of Linear Polarizations Shift Keying

For linear Polarizations Shift Keying, the key component of the cryptographic polarization controller is the switch, used as a polarization router. A Single Pole Double Throw (SPDT) switch is used to perform binary-PoSK. The antenna is either a single-piece dual-polarized antenna, or two identical antennas of the same (linear) polarization perpendicularly positioned to each other to create a vertical polarization (V) and a horizontal polarization (H). Figure 16 shows the two architectures, with linear polarizations, investigated in this work.



(a)



(b)

Figure 16. Architectures of the linear PoSK-based message authentication system: **(a)** - Synchronization-required architecture; **(b)** - Synchronization-free architecture.

The Cryptographic Polarization Selector block is the one that runs the cryptographic algorithm. In the first architecture shown on Figure 16(a), the normal transmission should always be done in co-polarization V to V or H to H, that is why a precise time synchronization is required between the Wireless Sensor Active Antenna System and the Security Gateway Active Antenna System via their Cryptographic Polarization Selector modules; while the second architecture shown on Figure 16(b) is the easiest to implement since it is free from this time synchronization. However, in the latter architecture, the Security Gateway requires two ports, as can be seen on Figure 16(b): the main message is received on port P1 and the authentication signal on port P2. Indeed, in synchronization-free architecture, the receiver (i.e., the Security Gateway) uses both H and V polarizations simultaneously, and manages to retrieve the main message and its authentication code from two different ports, by using three "Splitter" blocks, two of which perform power division and the one in the middle performs power addition. In fact, the Splitter block, depending on its configuration, performs a division by two (1 = 2 = S/2) or a combination (S = 1+2).

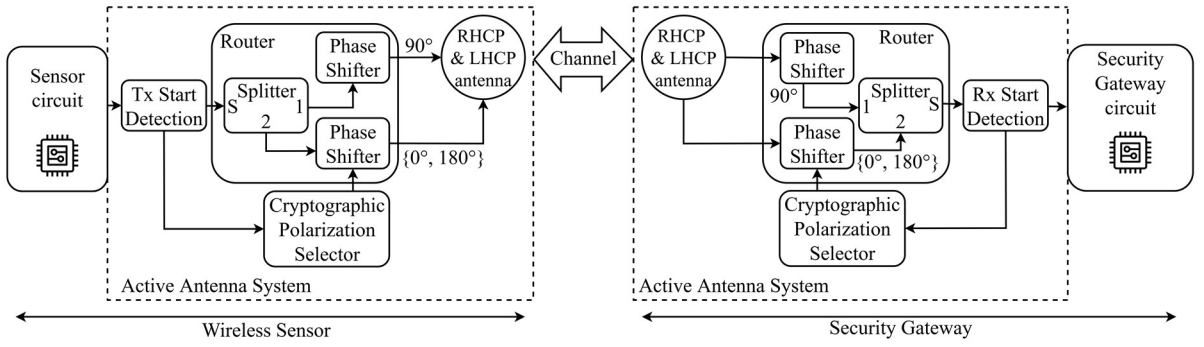
In simple terms, the message is obtained by summing the two components V and H, and the authentication code is obtained by listening on just one of the two polarizations, in this case, V.

In the Active Antenna System, the "Tx/Rx Start Detection", the "Cryptographic Polarization Selector" and the "Router" blocks together constitute the Cryptographic Polarization Controller. On the transmission side, the "Tx/Rx Start Detection" block allows the detection of the beginning of the transmission. This block is basically a power detector. If the used power detector has no output to forward the signal to the Router, a power divider can be used to divide the signal so that a fraction of the power can be used for the detection and the majority of the power can be forwarded to the Router for emission. The amount of power required for the detection depends on the sensitivity of the power detector used; the more sensitive the detector, the lower the power required for the detection. The "Tx/Rx Start Detection" block then communicates the transmission start information to the Cryptographic Polarization Selector, which then triggers the dynamic control of the polarization through the Router.

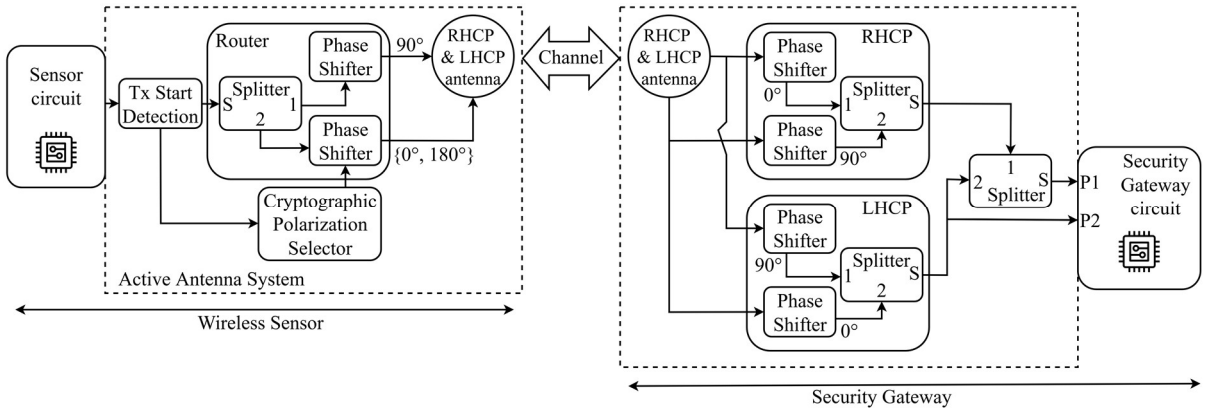
It is important to note that, on a terminal, if cables are needed to connect the two antennas (V and H) to the "Router" (or to the "Splitters", depending on the architecture), the two cables must have the same length, to avoid introducing an undesired phase shift that could affect the signals received by the Security Gateway, especially in the synchronization-free architecture (Figure 16(b)), since the result of the addition of two signals depends on the phase shift between them.

2.3. Architecture design of Circular Polarizations Shift Keying

What are different from architectures with linear polarizations and those with circular polarizations are the routers and the antennas. In this work, to perform 2-state circular PoSK, a single-piece 2-port circular polarization antenna is used. The antenna radiates either in Right-Hand Circular Polarization (RHCP) mode or in Left-Hand Circular Polarization (LHCP) mode, depending on the value of the phase shift between its two excitation ports. When this phase shift is +90°, the antenna radiates in RHCP, when it is -90°, the antenna radiates in LHCP. Figure 17 shows the two PoSK architectures with circular polarizations, investigated in this work. The operating principle is the same as for linear polarizations.



(a)



(b)

Figure 17. Architectures of the circular PoSK-based message authentication system: **(a)** - Synchronization-required architecture; **(b)** - Synchronization-free architecture.

On each of the two terminals in the synchronization-required architecture (Figure 17(a)) and on the Wireless Sensor in the synchronization-free architecture (Figure 17(b)), two phase shifters ("Phase Shifter" blocks) are used to dynamically control the antenna polarization. One of the two phase shifters is set to a fixed phase shift value, +90° in this case; and the phase shift value of the other phase shifter is dynamically controlled by the "Cryptographic Polarization Selector" block. The two possible values for this latter phase shifter being 0° and 180°, at the end of the day the two possible phase shift values between the antenna ports are indeed +90° (+90° - 0°) (RHCP mode) and -90° (+90° - 180°) (LHCP mode).

One might ask, why not remove the first phase shifter, the one configured to a fixed phase shift value of +90°, and simply do -90° / +90° (instead of 0° / 180°) with the phase shifter controlled by the "Cryptographic Polarization Selector" block? In fact, the primary objective of this +90° configured phase shifter is to compensate for the time delay and insertion losses introduced by the other phase shifter (the one controlled by "Cryptographic Polarization Selector"), so that the signals on both paths arrive together at the block where they meet, with the same power level and without any time offset between them. On the Wireless Sensor, the block where the two paths meet is the emitting antenna, and on the Security Gateway it is a "Splitter" block that performs an addition of the two paths (i.e., signals).

Since time delay actually means additional undesired phase shift, this delay compensation is necessary for the antenna to function properly, especially in emission, where it must have a phase shift of -90° (or +90°) to radiate in LHCP (or RHCP).

So, the choice of using two phase shifters is not an insignificant one. Indeed, one of the two phase shifters, in addition of introducing phase shift when necessary, also serves to compensate insertion losses and undesired phase shift introduced by the signal's travel time along the path of the other phase shifter. This also applies to the Security Gateway of the synchronization-free architecture (Figure 17(b).

For this compensation to be effective, the two phase shifters must be identical. For the same reasons, it is also very important that, on a terminal, the two cables used to connect the two ports of the antenna to the two phase shifters have the same length, to avoid introducing an undesired phase shift that would not only impair the proper operation of the antenna, but also affect the signals received by the Security Gateway, since the result of the addition of two signals depends on the phase shift between them.

In the synchronization-required architecture (Figure 17(a)), normal transmission must always be done in co-polarization (RHCP to RHCP or LHCP to LHCP). This is why the "Cryptographic Polarization Selector" block is present on both terminals, in order to allow both antennas to be in the same mode (RHCP or LHCP) at any given time t during the transmission. This is not necessary in the synchronization-free architecture, as in this case the Security Gateway uses both polarizations (RHCP and LHCP) at the same time, and manages, through a special circuit made up of phase shifters and splitters, to retrieve the message on port P1 and the message authentication code on port P2. In simple terms, the message is obtained by summing the two components, RHCP and LHCP, and the authentication code is obtained by listening on just one of the two polarizations, in this case, LHCP.

3. Authentication Approach

This is a dynamic authentication system for messages. Basically, the authentication is based on the received instantaneous power. But, the authentication approach used is different depending on whether the architecture requires time synchronization or not. In all the architectures, the Security Gateway is in charge of authenticating messages coming from the Wireless Sensor.

3.1. Authentication Approach in Synchronization-required architectures

For architectures where time synchronization is required (Figure 16(a) and Figure 17(a)), the normal transmission is always done in co-polarization. Therefore, the power received – via a static channel *c* defined by a given Wireless Sensor-to-Security Gateway distance *d* (about 3 meters in the presented experimentations) – is theoretically constant over time. But, since a real-world channel is dynamic, the Security Gateway will be rather checking that the received instantaneous power remains within some acceptable range. Relation *(23)* expresses this approach.

$$P_d(c,t) \in [P_L, P_H], with\ P_L > P_{crossPol}(t) \quad (23)$$

Where *P* is the instantaneous power received by the Security Gateway located at a distance *d* from the Wireless Sensor and $P_{crossPol}$ is the received power in a cross-polarization transmission (e.g., V to H, or RHCP to LHCP). Variables *c* and *t* represent channel and time, respectively. P_L and P_H represent, respectively, the lower and upper limits of the acceptable range for the power *P*. If, at a time *t*, power *P* falls outside this interval, the Security Gateway deduces that the received message does not come from the expected Wireless Sensor (i.e., the one it is synchronized with), and the received message is classified as non-authentic. This approach is quite channel-dependent. For example, the values of P_L and P_H must be readjusted with the distance *d* between the Wireless Sensor and the Security Gateway.

3.2. Time Synchronization Technique between PoSK and Modulation symbols in order to mitigate PoSK impact on the Bit Error Rate (BER)

As previously mentioned, in architectures where time synchronization is required (Figure 16(a) and Figure 17(a)), the normal transmission is always done in co-polarization, i.e., V to V or H to H for linear polarizations (Figure 16(a)) and RHCP to RHCP or LHCP to LHCP for circular polarizations (Figure 17(a)). This requires a precise time synchronization between the two Active Antenna System blocks via their Cryptographic Polarization Selector modules.

Wireless nodes time synchronization is a research topic of its own. It is a widely explored topic by the scientific community [80], [81], [82], [83], and well-established synchronization techniques are available nowadays. This is why the synchronization is not a major point in this work, as it is taken for granted. The time synchronization technique to be used is the choice of those implementing the PoSK.

However, in this work, a time offset to be considered in the synchronization is proposed in order to mitigate the impact of PoSK on the communication quality. This is because the time taken to switch from one polarization to the other (i.e., the switching time for the switches or the phase shifting time for the phase shifters) is non-zero and therefore causes a brief interruption in the transmission at each polarization switching. For example, in architectures with linear polarizations (Figure 16), during the switching time, the Sensor circuit output signal is

connected to neither of the two antennas, i.e., polarizations. The symbol change of the main modulation (e.g., Phase Shift Keying (PSK)), i.e., the main message, will be missed if it occurs within this switching time, which could significantly degrade communication quality by increasing the Bit Error Rate (BER). Time synchronization has to respect relations (21) and (22) to prevent the symbol change of the main modulation from occurring during the polarization switching time.

$$F_{PoSK}(n) = 2^n F_S \, , n \in \mathbb{Z} \tag{21}$$

$$\varphi(k) = \frac{1}{4 * F_{PoSK}} + \frac{k}{2 * F_{PoSK}} = \left(\frac{1 + 2k}{4 * F_{PoSK}}\right), k \in \mathbb{N} \tag{22}$$

Where F_{PoSK} is the Cryptographic Polarization Selector output rate, F_S is the main modulation symbol rate, and φ is the time offset between F_{PoSK} and F_S, i.e., the beginning of the PoSK relative to the beginning of the transmission, as illustrated on Figure 18 with $n = \{-1,0,1\}$ and $k = \{0,1\}$. It can be seen on Figure 18 that the symbol change of the main modulation (i.e., main message) never occurs during the polarization switching, regardless of the F_{PoSK} value. Indeed, the phase shifts of the main message, which uses Phase Shift Keying in this example, always occur far from the edge transitions of the $F_{PoSK}(-1)$, $F_{PoSK}(0)$ and $F_{PoSK}(1)$ signals.

Note that relations (21) and (22) are just empirical examples, simplified on purpose for the experimentation. In fact, in this case, F_{PoSK} is either a multiple or a sub-multiple of F_S, depending on the value of n.

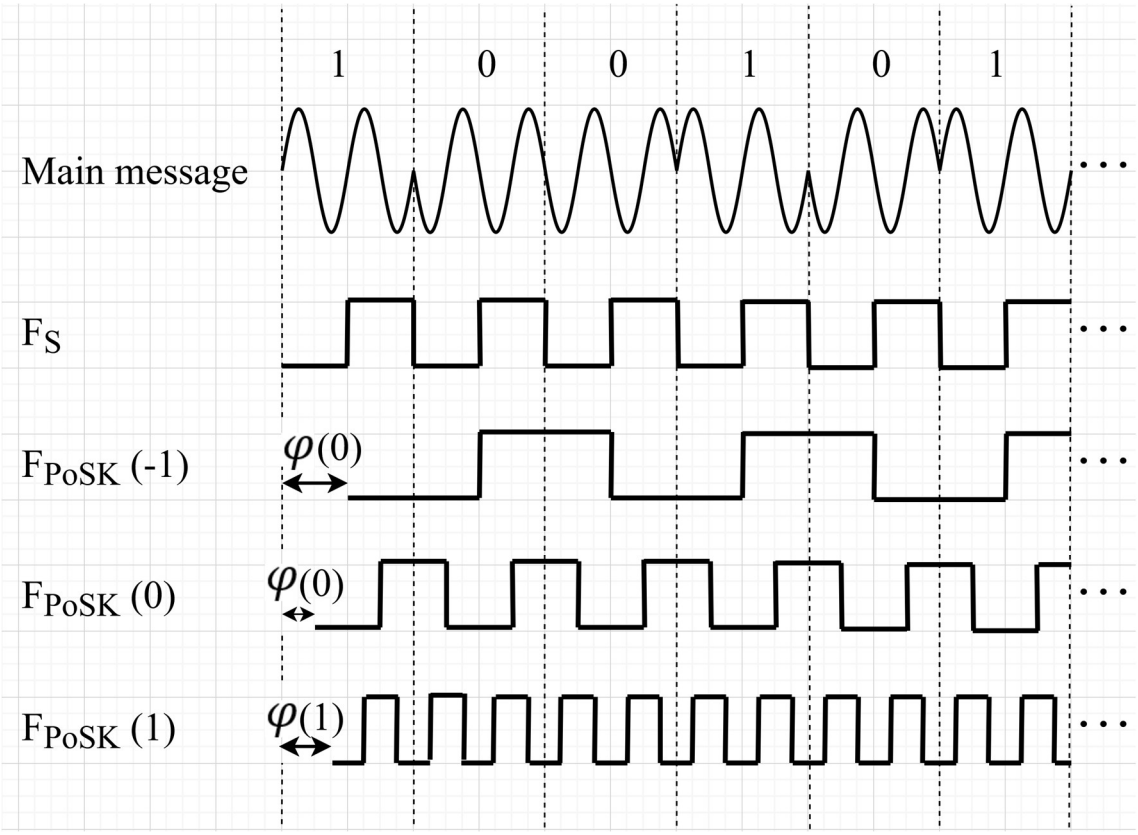


Figure 18. Illustration of the time offset between the beginning of transmission and the beginning of PoSK.

3.3. Authentication Approach in Synchronization-free architectures

For architectures where time synchronization is not required (Figure 16(b) and Figure 17(b)), authentication is based on the assumption that the received power in a co-polarization transmission is higher than the received power in a cross-polarization transmission. Relation (24) expresses this hypothesis.

$$P_{coPol}(t) \gg P_{crossPol}(t) \quad (24)$$

Where P_{coPol} is the received power in a co-polarization transmission (e.g., V to V, or RHCP to RHCP) and $P_{crossPol}$ is the received power in a cross-polarization transmission (e.g., H to V, or RHCP to LHCP). Variable *t* represents the time.
Actually, relation *(24)* means that the instantaneous power of the signal received on only one of the two polarizations will be the authentication code. Port *P2* on the Security Gateway on Figure 16(b) and Figure 17(b) is dedicated to receiving this authentication signal. Indeed, port *P2* is listening only vertical polarization (V) on Figure 16(b), and only LHCP polarization on Figure 17(b). While, the main message is received on port *P1*, at the same time with the authentication signal.
The Security Gateway knows the expected code because it is running the same cryptographic algorithm as the Wireless Sensor, with the same initial shared secret key. The Security Gateway compares the received code with the expected one, and the message is validated as authentic if the two codes are the same.

4. Experimentation

4.1. Software-Defined Radio (SDR)

As SDR tools are key elements in the experiments carried out in this work, it is worth devoting a paragraph to this topic. As its name suggests, SDR is an approach of the radio-communication that is more software-oriented.

A traditional radio-communication device (e.g., the former FM receivers) consists mainly of a printed circuit board (PCB) containing a set of not very flexible analog electronic components, generally dedicated to the implementation and processing of a specific waveform (a single modulation) in a predetermined frequency band. This traditional radio architecture is not very flexible, and is difficult to evolve once it has been manufactured.

SDR proposes a two-component architecture: a hardware component and a software component. The hardware component consists mainly of a radio front-end, which is usually very wide-band and very flexible, with the possibility of configuring parameters such as transmission (reception) RF carrier frequency, sample rate, gain, etc. The software component focuses mainly on digital baseband signal processing, using DSP and FPGA chips. Figure 19 shows an overview of the SDR system architecture.

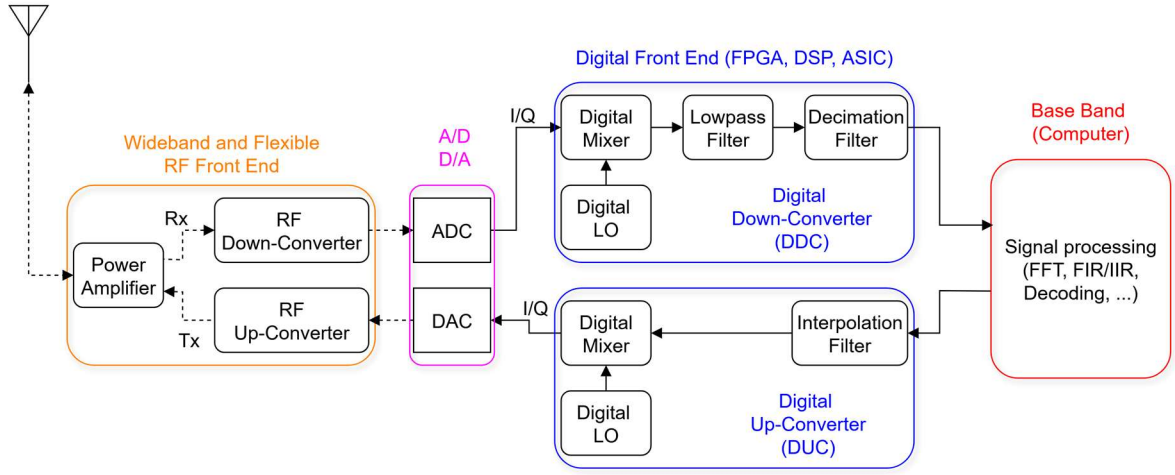


Figure 19. Software radio architecture.

In brief, a software radio is a radio system which performs the required signal processing in software instead of using dedicated integrated circuits in hardware. The benefit is that since software can be easily replaced in the radio system, the same hardware can be used to create many kinds of radios for many different communications standards; thus, one software radio can be used for a variety of applications. This results in highly flexible and scalable radio systems, covering a wide range of communications standards (e.g., WiFi, Bluetooth, ZigBee, GSM, etc.). These properties make SDR tools ideal for fast prototyping and laboratory experimentation.

The main SDR devices used in this work are:

The USRP B210 which, according to its documentation [84], provides a fully integrated, single-board, Universal Software Radio Peripheral (USRP) platform with continuous frequency coverage from 70 MHz – 6 GHz. Designed for low-cost experimentation, it combines the AD9361 RFIC direct-conversion transceiver providing up to 56 MHz of real-time bandwidth, an open and reprogrammable Spartan6 FPGA, and fast SuperSpeed USB 3.0 connectivity with convenient bus-power. Full support for the USRP Hardware Driver (UHD) software allows you to immediately begin developing with GNU Radio, prototype your own GSM base station with OpenBTS (an open-source software-based GSM base station), and seamless transition code from the USRP B210 to higher performance, industry-ready USRP platforms. Figure 20 shows the USP B210 and a schematic of its internal architecture as presented in its documentation.

The HackRF One, from Great Scott Gadgets, is a Software Defined Radio peripheral capable of transmission or reception of radio signals from 1 MHz to 6 GHz. Designed to enable test and development of modern and next generation radio technologies, HackRF One is an open source hardware platform that can be used as a USB peripheral or programmed for stand-alone operation. HackRF One's main features are: 1 MHz to 6 GHz operating frequency, half-duplex transceiver, up to 20 million samples per second, 8-bit quadrature samples (8-bit I and 8-bit Q), compatible with GNU Radio, Universal Radio Hacker, and more, software-configurable R_X and T_X gain and baseband filter, software-controlled antenna port power (50 mA at 3.3 V), SMA female antenna connector, SMA female clock input and output for synchronization, Hi-Speed

USB 2.0, USB-powered, open source hardware, [23]. Figure 21 shows the HackRF One and its block diagram as provided by Great Scott Gadgets.

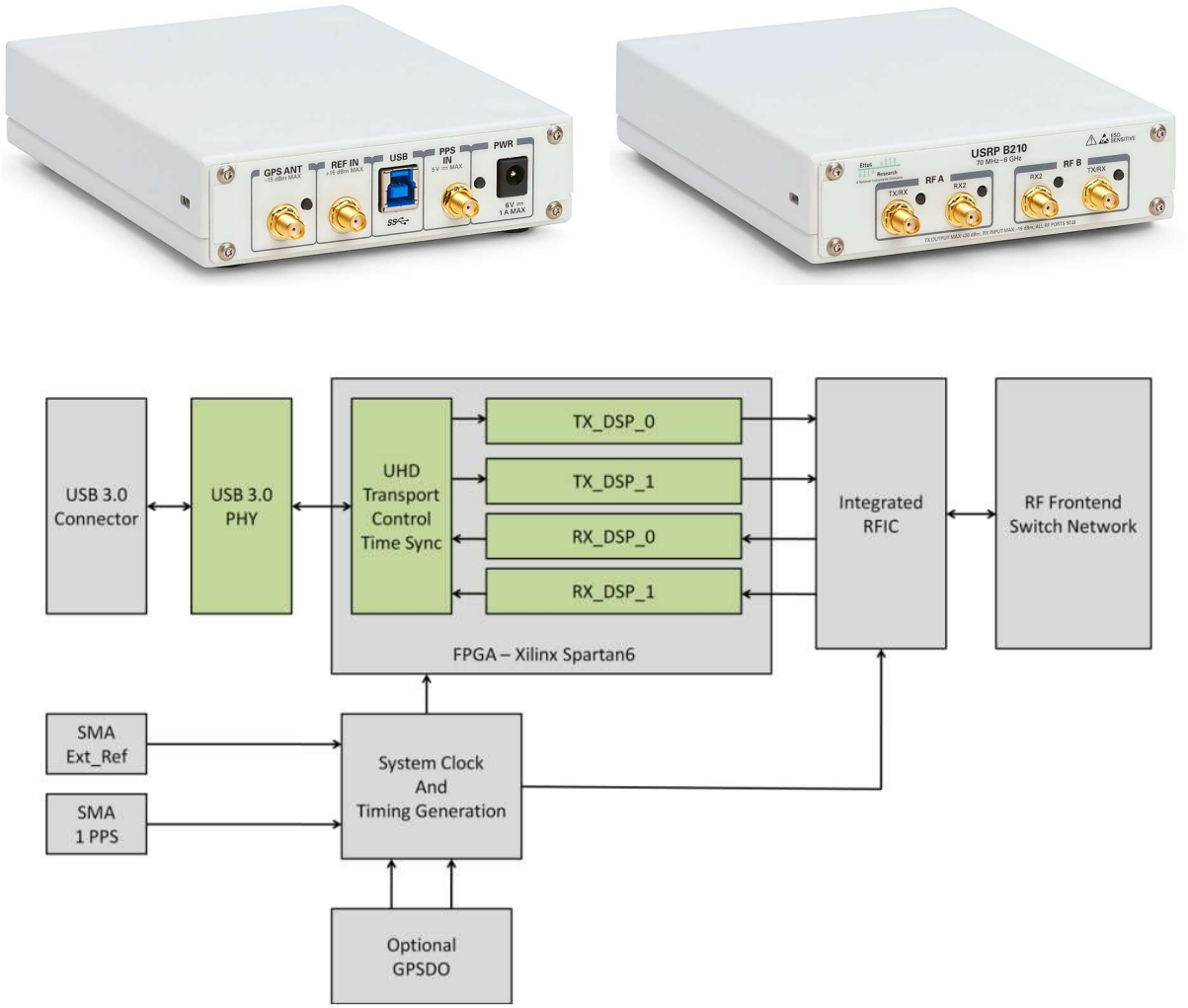


Figure 20. USRP B210.

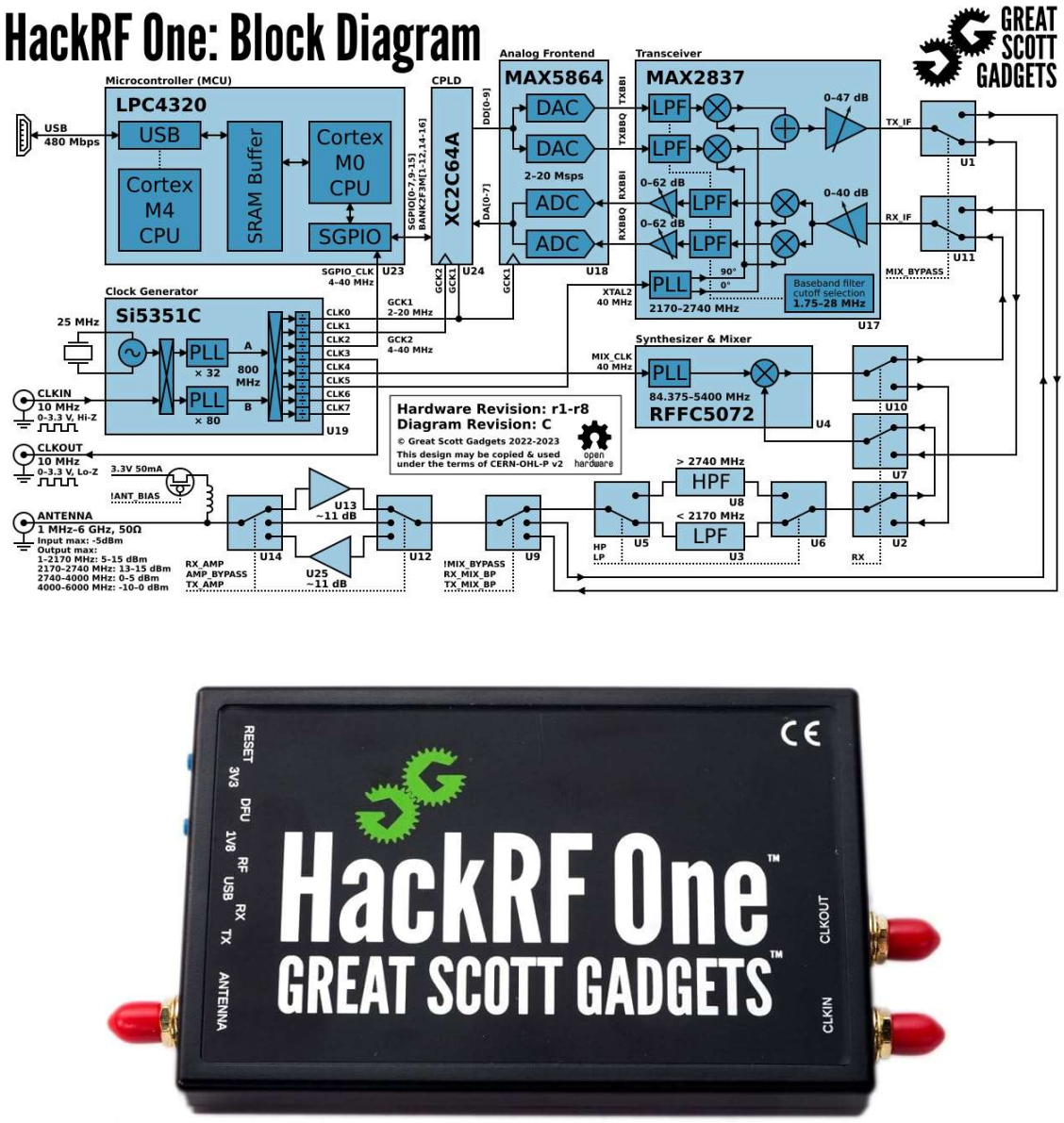


Figure 21. HackRF One.

When it comes to software, the main SDR software tools used in this work are:

GNU Radio: a free and open-source software development toolkit that provides signal processing blocks to implement software radios. It can be used with readily-available low-cost external RF hardware to create software-defined radios, or without hardware in a simulation-like environment. It is widely used in research, industry, academia, government, and hobbyist environments to support both wireless communications research and real-world radio systems [24]. GNU Radio supports most USRP devices, including the USRP B210 used in this work. It is also compatible with HackRF One.

Universal Radio Hacker (URH): a complete suite for wireless protocol investigation with native support for many common Software-Defined Radios. URH allows easy demodulation of signals combined with an automatic detection of modulation parameters making it a breeze to identify the bits and bytes that fly over the air. As data often gets encoded before transmission, URH offers customizable decodings to crack even sophisticated encodings like CC1101 data whitening. In addition to its capabilities for receiving, demodulating and decoding, URH also allows for generating, modulating and transmitting data [25], [26]. URH supports also USRP B210 and HackRF One.

4.2. Experimentation tools

The experimental tools corresponding to the blocks of the architectures on Figure 16 and Figure 17 are presented with their structure and their functionality in the next sub-chapters:

4.2.1. Wireless Sensor and Security Gateway circuits

For each of these two blocks, a SDR device HackRF One [23] or USRP B210 [84] is used. Each of these two devices offers more than enough performance for the experiments in this work. Indeed, both of these devices more than meet our requirements, namely: a transmission carrier frequency of 2.45 GHz, a bandwidth of just 1 MHz, and the gain required for transmission over a distance of just 3 meters.

4.2.2. Power Detection

A Linear Technology's LTC5536 -based power detector was designed in our research team. It is used for the detection of the beginning of the transmission. Figure 22 shows the electrical schematic of this power detector and its PCB.

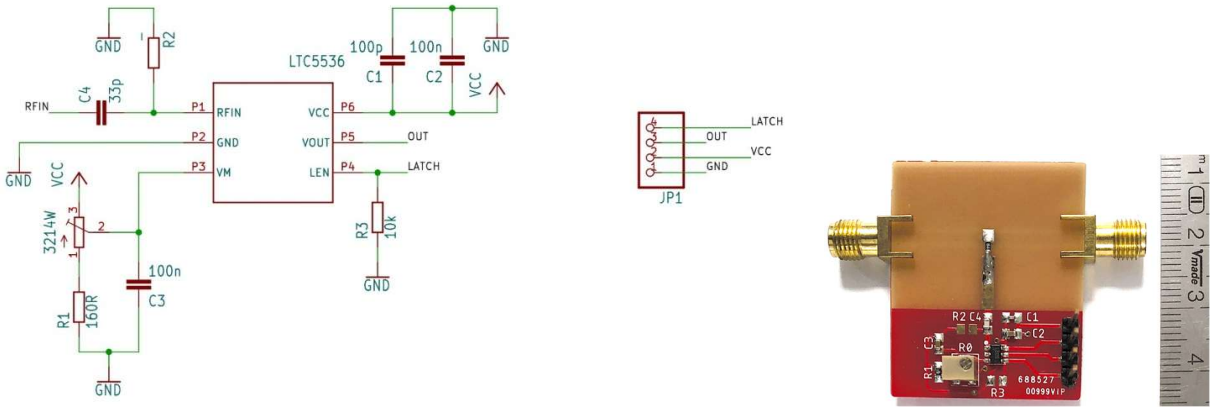


Figure 22. 2.45 GHz RF Power detector.

The LTC5536 is an RF power detector for RF applications operating in the 600 MHz to 7 GHz range. In addition to this Wide Input Frequency Range, the LTC5536 also has a Wide Input Power Range, ranging from –26 dBm to 12 dBm. It is also a fast and accurate detector. According to its datasheet, the RF input voltage is peak detected using an on-chip Schottky diode. The detected voltage is compared against a reference voltage at V_M (see Figure 22).

The response time from the RF input (RFIN) to V_{OUT} (detection pin) can be as little as 20 ns. The comparator output is latched when LEN is high or is transparent when LEN is low. In the final PCB, a second RF port has been added to output the RF signal, as shown on Figure 22.

4.2.3. Single Pole Double Throw (SPDT) switch

The Router block, in architectures with linear polarization (Figure 16), is a SPDT switch. Designed in our research team, a custom-made switch - based on the Texas Instruments THS4505 Wideband, Low-distortion, Fully Differential Amplifiers (Analog Device's AD8137 as equivalent) and LT1013 dual precision operational amplifier - was used in this work. Figure 23 shows the electrical schematic of this SPDT switch and its PCB.

This double shunt, PIN Diode, SPDT switch transmits the signal from its RF_in port to one of its two outputs, RF_out_1 and RF_out_2, depending on the voltage, -5V or +5V, applied to the CTRL pin (see Figure 23). It is important to note that -5V can be replaced by 0V, making it easy for a microcontroller to control the switch directly, without the need for a DC-DC converter.

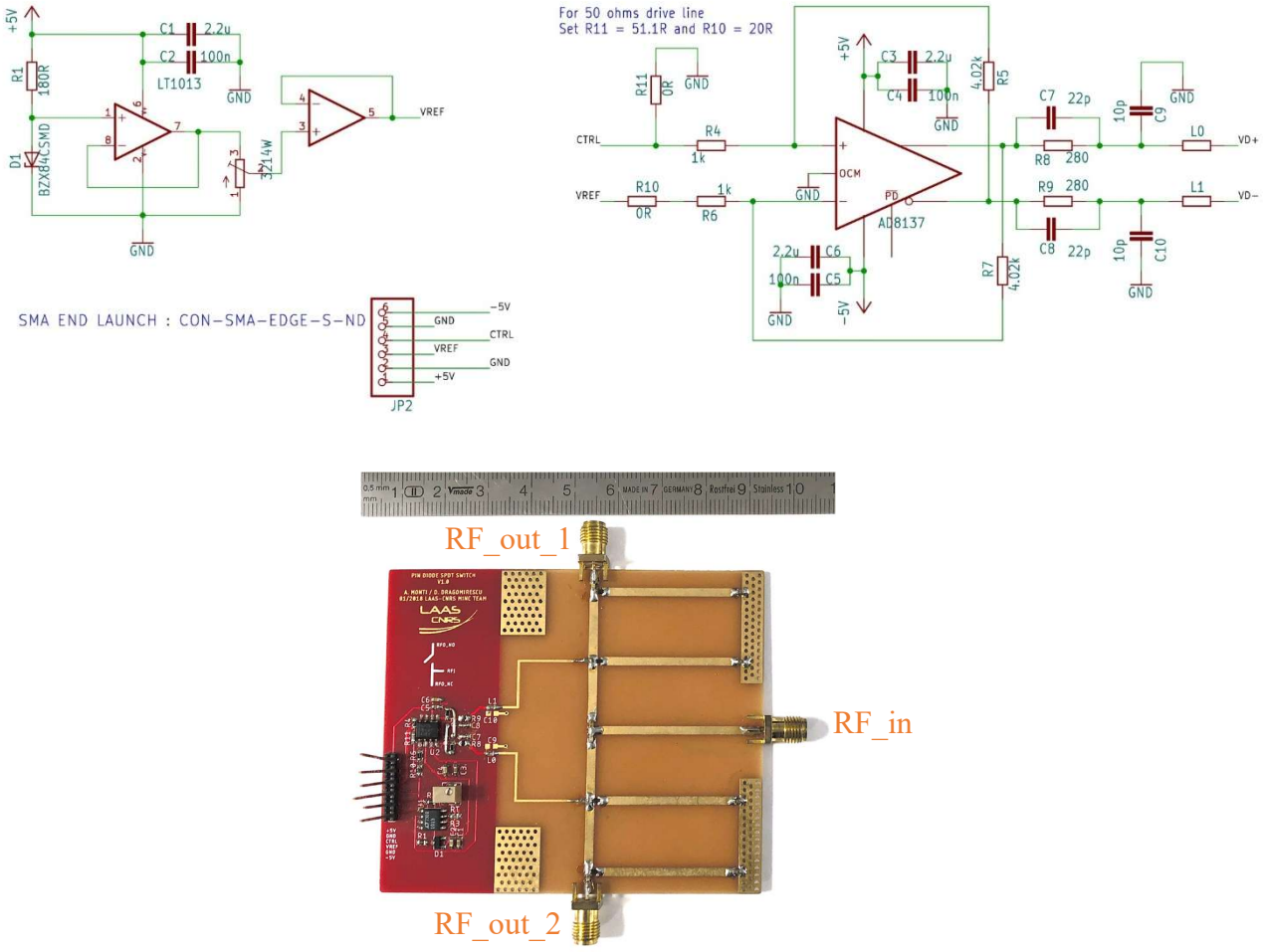
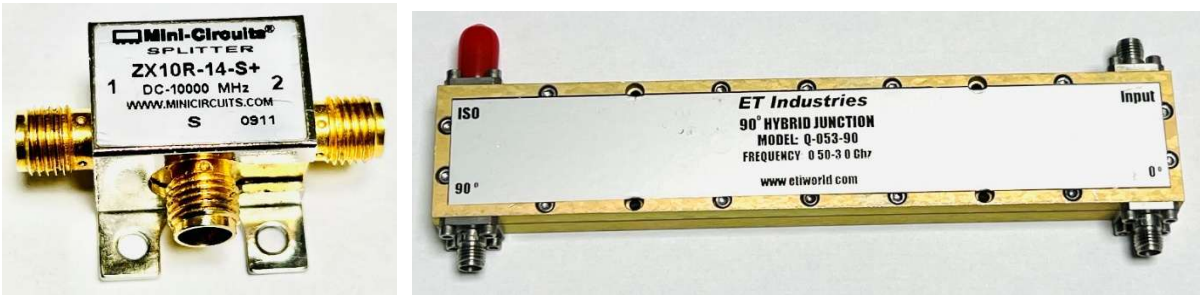


Figure 23. Single Pole Double Throw (SPDT) Switch.

4.2.4. Splitter

The Mini-Circuits ZX10R-14-S+ [85] was used as power splitter/combiner. This is a 2 Way-0° Resistive 50Ω Power Splitter/Combiner operating in the very wide band from DC to 10 GHz, which obviously includes our 2.45 GHz frequency of interest. An image of this component is shown on Figure 24(a). Another splitter/combiner was also used in this work, namely, the Q-053-90, a 90 Degree Hybrid Coupler from ET Industries. Figure 24(b) shows an image of this component. The advantage of this other symmetrical splitter is that it introduces a 90° phase shift between its two outputs, so it is both a splitter/combiner and a 90° phase shifter. This makes it a useful component in the experimentation of circular PoSK, especially for the operation of our 2-port LHCP- RHCP- enabled antenna, since this antenna requires its two excitation ports to be in positive (negative) phase quadrature to radiate in LHCP (RHCP). The Q-053-90 replaces a set of an ZX10R-14-S+ splitter/combiner, an EV1HMC647ALP6 [86] phase shifter (presented in the next section) and its controller (Arduino board).



(a) **(b)**

Figure 24. (a): Power Splitter/Combiner ZX10R-14-S+; **(b)**: Q-053-90, a 90 Degree Hybrid Coupler from ET Industries.

4.2.5. Phase Shifter

The evaluation board EV1HMC647ALP6 [86] of the Analog Devices HMC647ALP6E phase shifter was used in this work. Figure 25 shows an image of this component and some characteristic diagrams of the HMC647ALP6E as provided by its datasheet.

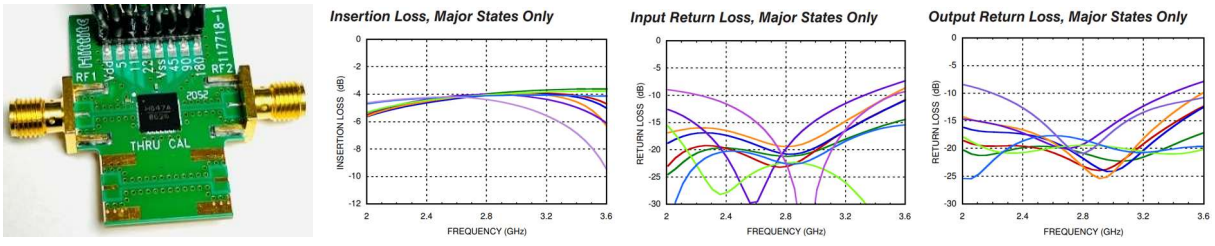


Figure 25. The evaluation board EV1HMC647ALP6 of the Analog Devices HMC647ALP6E phase shifter. The diagrams are from the HMC647ALP6E datasheet.

The HMC647ALP6E is a GaAs MMIC 6-Bit digital phase shifter, operating in the 2.5 - 3.1 GHz frequency band. As we could not find a phase shifter operating at 2.45 GHz, we chose this component, whose characteristics seemed acceptable for our application. Indeed, according to its datasheet, at 2.45 GHz, this component has an Insertion Loss of around -4 dB, an Input Return Loss of less than -10 dB, a Phase Error of around 0° and an Output Return Loss also less than -10 dB. But, despite these values, we found that this component worked poorly at 2.45 GHz, which led us to experiment with a software-defined phase shifter as well, an approach we were already interested in.

4.2.6. Cryptographic Polarization Selector

An Arduino UNO board was used to run the cryptographic polarization selection algorithm. This choice is mainly motivated by the short time it takes to develop an application in the Arduino environment.
Generally speaking, the most important criteria to be taken into account in the hardware (FPGA or ASIC) or software (e.g., Python, C/C++) implementation of this block are: power consumption, which must be as low as possible, memory footprint, which must be as small as possible, and execution time, which must be as short as possible. In the IoT context, particular emphasis is placed on the power consumption criterion for this component, as for all other components of the PoSK-based authentication system.
Nevertheless, for this experimental work, any other microcontroller, instead of the Arduino, would do the job.

4.2.7. Linear Polarization Antenna

For linear Polarizations Shift Keying experimentation in this work, each terminal uses two identical antennas of the same (linear) polarization perpendicularly positioned to each other, as shown on Figure 26.
Actually, this antenna, on Figure 26, is designed to provide both H and V polarizations, but its architecture results in significant gain difference between the two polarizations. This is why we use only the vertical component of the antenna. Consequently, two antennas, perpendicularly positioned to each other, are used for the two polarizations, as shown in Figure 26, right-hand image.

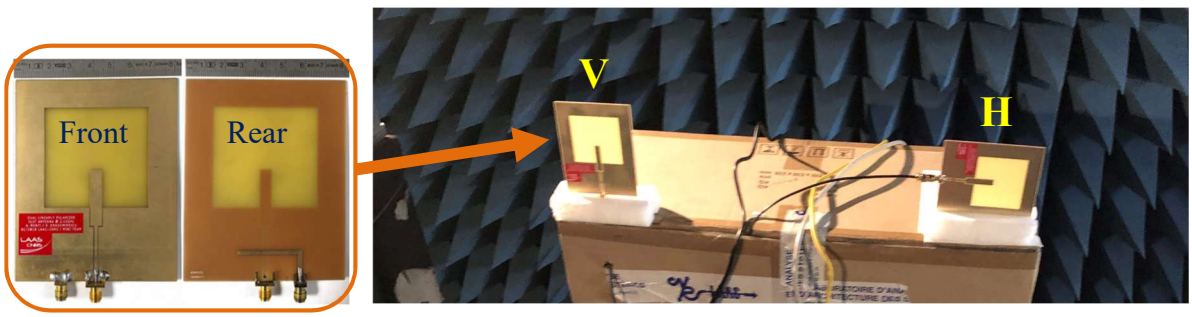


Figure 26. 2.45 GHz dual-linearly polarized antenna.

In this case, it is important that the two antennas used for the V and H polarizations to be identical in terms of gain and radiation pattern, to avoid them contributing to the phenomenon of Undesired Polarization Sensitivity, which we will see later in this manuscript.

4.2.8. Circular Polarization Antenna

For circular Polarizations Shift Keying experimentation in this work, a single-piece two-port antenna [87], shown on Figure 27, was used. We designed this antenna dedicated to this application.

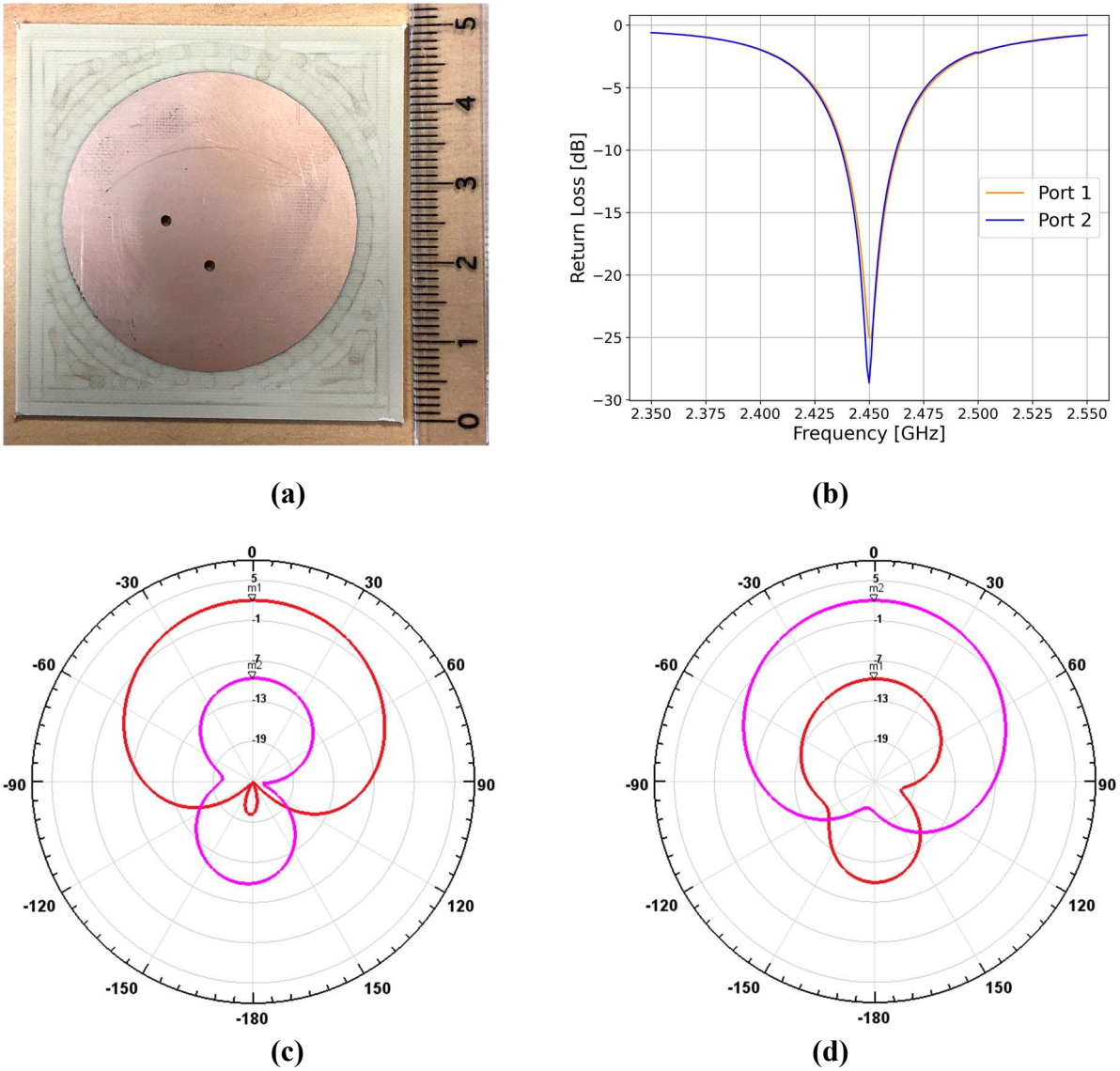


Figure 27. 2-port 2.45 GHz RHCP and LHCP -enable antenna. **(a)**: Image of one of the fabricated antennas. **(b)**: Return Loss of the 2 ports of the antenna, measured using a VNA. **(c)**: Realized Gain in LHCP mode, from Ansys HFSS simulation of the antenna. **(d)**: Realized Gain in RHCP mode, from Ansys HFSS simulation of the antenna. NOTE: The gap between the m1 and m2 markers is actually almost identical in both cases (c) and (d), at just over 11 dB.

On Figure 27(b), one can see that the Return Loss of each port is less than -10 dB between 2.4380 GHz and 2.4626 GHz, with an excellent match (less than -20 dB Return Loss) at the frequency of interest, i.e., 2.45 GHz.

Figure 27(c) shows the Realized Gain at 2.45 GHz in LHCP mode. The pattern in red represents the LHCP component and the pattern in pink represents the RHCP component. The two markers m1 and m2 are both at angle 0°. Marker m1 is at 2.0219 dB and marker m2 at -9.5939 dB, giving an Axial Ratio (AR), the gap between LHCP and RHCP, of just over 11 dB. Realized Gain is the standard gain reduced by its impedance mismatch factor.

Figure 27(d) shows the Realized Gain at 2.45 GHz in RHCP mode. The pattern in pink represents the RHCP component and the pattern in red represents the LHCP component. The two markers m1 and m2 are both at angle 0°. Marker m1 is at -9.6868 dB and marker m2 at 2.0378 dB, giving an AR of just over 11 dB, as in LHCP mode on Figure 27(c).

This antenna radiates either in RHCP or LHCP, depending on whether its two excitation ports are in positive or negative phase quadrature, i.e., +90° or -90° phase shift between the two ports. The carrier frequency used in this work is the 2.45 GHz of the ISM (Industrial, Scientific, and Medical) band.

4.2.9. Universal Radio Hacker (URH)

On the software side, GNU Radio [24] and Universal Radio Hacker (URH) [25], [26] were used for transmissions. Tailor-made GNU Radio flowgraphs were developed for demodulation and other signal processing operations, as the version, to this day, of URH has limited demodulation performance.

In this work, URH is used mainly for emission and reception. It is not used for signal processing and message decoding. In emission, URH is used to create, modulate and emit the message. At the receiving end, URH is used to receive and store the message in I/Q format in a complex file. Then, this complex file will be processed by the appropriate custom-made GNU Radio demodulator.

Figure 28 shows the transmission and reception processes with URH. In the transmission process (Tx), the block ***Generator*** corresponds to the Generator tab in URH, where one can save the message with ***Generate file*** for later use, or go directly to the emission tab with ***Send data***.

URH provides other very useful functions that are not presented on Figure 28. For example, the Generator's ***Fuzzing*** tab is extremely useful for quickly generating messages based on a given mathematical law. For example, a message containing numbers from 0x000 to 0xFFF in steps of 1 can be created in 10 seconds by following a few simple steps.

Another very useful Generator's tab is the ***Pauses*** tab, which lets you add and remove pauses in the message. URH also offers the possibility of duplicating messages.

On the receiving side (Rx), URH enables real-time observation of radio-communications, as well as recording radio-communications in complex I/Q format for later use.

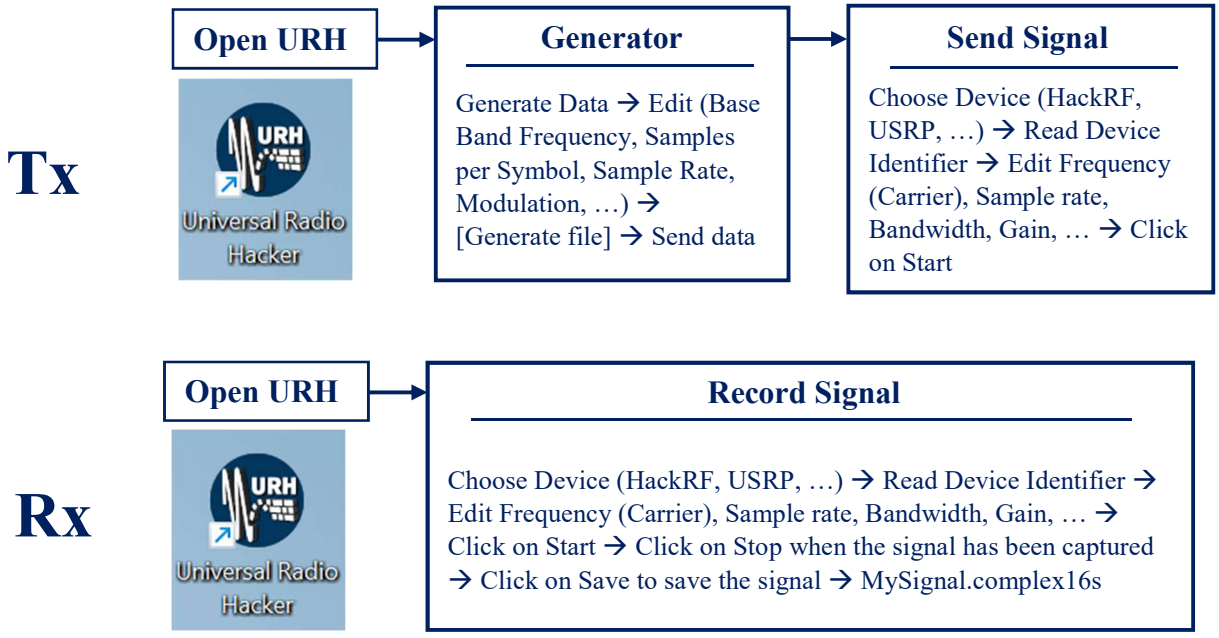


Figure 28. URH emission (Tx) and reception (Rx) processes.

4.2.10. GNU Radio

In this work, GNU Radio is mainly used for processing and decoding the received message and, sometimes, its authentication code as well. The main signal processing operations are filtering and demodulation. We have developed three demodulators for the three modulation patterns (2-ASK, 2-FSK, BPSK) explored in this work.
In some experiments, particularly in the experimentation of the software-oriented architectures, discussed later in the manuscript, GNU Radio was also used for both emission and reception, in place of URH. This is because, in software-oriented architectures, the emitter is a bit particular with the use of two ports for example, which was difficult to handle with URH, but easy to handle with GNU Radio.
Also, in this work, power calculation and display were often done directly in the same python script, without going through GNU Radio, although this is also simple with GNU Radio.

4.2.10.1. 2-ASK demodulation

2-ASK is the simplest of our three modulation patterns. Its demodulation is just as simple. Figure 29 shows our 2-ASK demodulator GNU Radio flowgraph. Here, we use the envelope detection technique through the ***Complex to Mag*** block, which calculates the magnitude of the signal provided at its input.

In this example, the baseband carrier is 1 MHz and the bit rate is 100 kbps. The ***File Source*** block reads the complex file containing the 8-bit signed I/Q samples (IQIQIQ...). The ***IChar To Complex*** block reconstructs the complex signal from the I/Q samples. The ***Multiply Const*** block then normalizes the signal between 0 and 1 by dividing it by 128. The ***Throttle*** block is used to control the flowgraph execution speed when no SDR device is used in the flowgraph. Without the ***Throttle*** block, GNU Radio will try to execute the flowgraph as faster as possible.

The ***Throttle*** block is not required when an SDR device (e.g., USRP B210, HackRF One) is used.

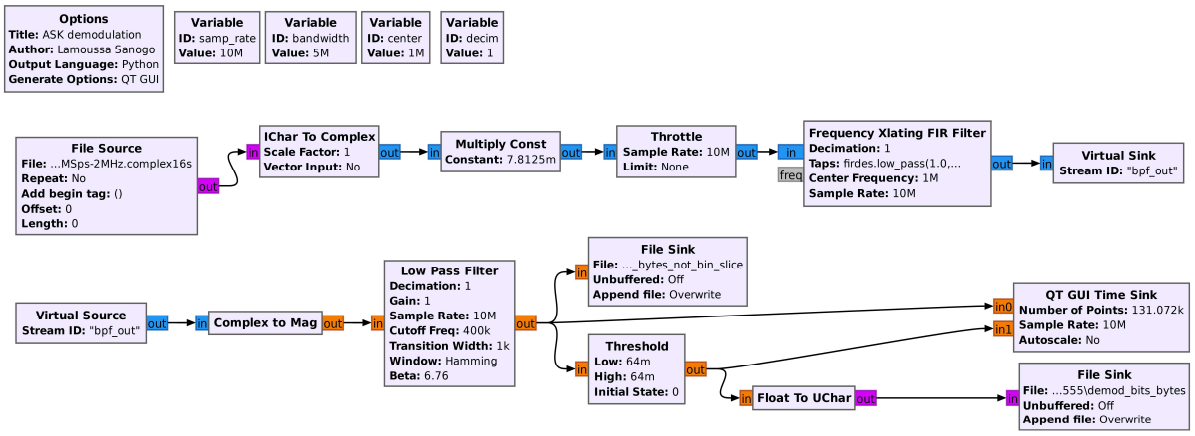


Figure 29. 2-ASK demodulator GNU Radio flowgraph.

The ***Frequency Xlating FIR Filter*** block not only filters the signal to remove possible noise, but also performs frequency translation to bring the signal back to around 0 Hz (DC); this block's low-pass filter arguments, in this flowgraph, are: 1.0, 10×10^6 Hz, 5×10^6 Hz, 40×10^3 Hz for gain, sample rate, bandwidth and transition width, respectively.

The ***Virtual Sink*** and ***Virtual Source*** blocks connect the ***Frequency Xlating FIR Filter*** and ***Complex to Mag*** blocks. These two blocks are just used to create virtual connections for flowgraph good readability purposes.

After obtaining the signal magnitude with the ***Complex to Mag*** block, the ***Low Pass Filter*** block is used to clean the signal before binary decoding. The cutoff frequency of this low-pass filter must be at least equal to the modulation symbols rate, which is equal to the bit rate in this case of 2-state modulation. So, the low-pass filter cutoff frequency must be at least 200 kHz. As the signal has been well filtered before arriving at this filter, the 400 kHz cutoff frequency does the job nicely. About the transition width of this filter, one can still increase it to reduce the computational load if necessary.

After this low-pass filter, the ***Threshold*** block is used to convert the signal into binary signal, which is then stored in a file using the ***File Sink*** block. This file is then read by a Python script for display in a figure and for decoding the content of the received message. The ***Binary Slicer*** block can be used in place of the ***Threshold*** block, but to do so, the signal must first be centered on zero, as the ***Binary Slicer*** block outputs a binary 1 for a positive float input and outputs a binary 0 for a negative float input.

4.2.10.2. 2-FSK demodulation

The flowgraph of the 2-FSK demodulator is architecturally very similar to that of the 2-ASK, as shown on Figure 30. But, unlike in the 2-ASK, this time the center frequency of the ***Frequency Xlating FIR Filter*** block is half the sum of the two frequencies used for the 2-FSK, that is, (1 MHz + 500 kHz) / 2, giving 750 kHz.

The two other most important new additions are the ***Simple Squelch*** and ***Quadrature Demod*** blocks.

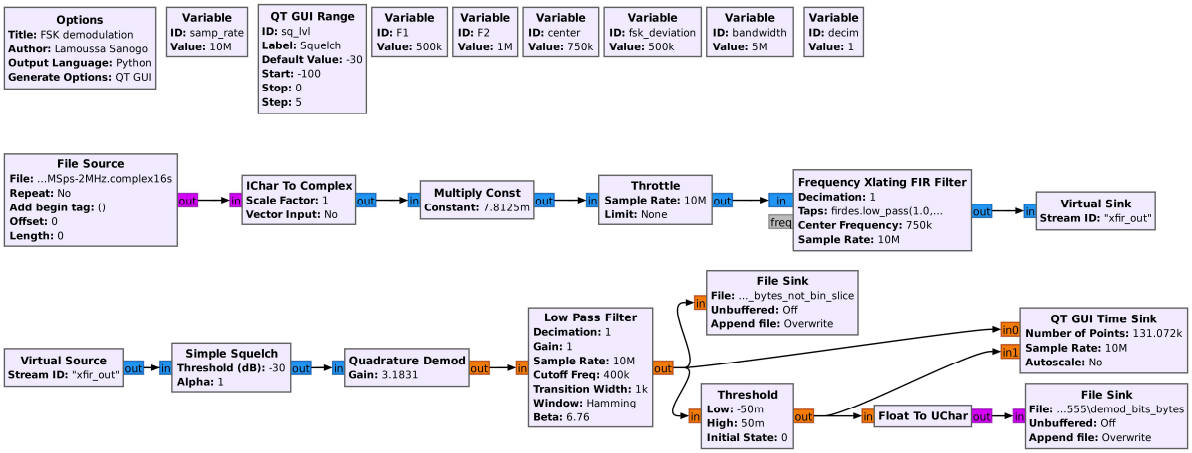


Figure 30. 2-FSK demodulator GNU Radio flowgraph.

Basically, the ***Simple Squelch*** block is used to lower the noise floor of the signal. The output equals the input if the average input is higher or equal the threshold, and zero otherwise. The average is computed using a Single Pole IIR filter. It uses the magnitude squared for both averaging and for comparing to the threshold.

The ***Quadrature Demod*** block is most useful in demodulating FM, FSK, GMSK, and similar modulations that carry information through changes in frequency. Since frequency is a change of phase over time (in math terms, frequency is the differential of phase), in order to frequency demodulate a carrier, this block calculates the phase change between consecutive samples by multiplying each complex sample with a conjugated version of the sample that came before it. The argument of the multiplier output is the phase difference, in radians, between these two consecutive samples. To change this from radians to cycles, this value must be divided by 2π. Then, to convert this to frequency, this value must be divided by the sample period, i.e., this value must be multiplied by the sample rate. That is why the value of the parameter ***Gain*** of the ***Quadrature Demod*** block is given by ***samp_rate/(2*math.pi*fsk_deviation)***, the division by *fsk_deviation* is used to normalize the gain and make it unitless.

The other blocks in this flowgraph play the same roles as in the 2-ASK demodulator, described previously.

4.2.10.3. BPSK demodulation

The flowgraph of the BPSK demodulator also remains very similar to the two previous ones, 2-ASK and 2-FSK, from an architectural point of view, as it can be seen in Figure 31.

The main new addition for the BPSK demodulator is the ***Costas Loop*** block, which in fact performs BPSK demodulation. As explained in its GNU Radio documentation, the ***Costas Loop*** is a carrier recovery module, great for synchronizing to BPSK, QPSK, and 8PSK. The ***Costas Loop*** locks to the center frequency of a signal and down-converts it to baseband. It uses a second order loop and is therefore defined with a loop bandwidth parameter whose value is about ***2π / 100***.

Depending on the noise level and the quality of the signal at the input of the ***Costas Loop*** block, the value of the Loop Bandwidth parameter can be adjusted in order to obtain a more or less clean signal at the ***Costas Loop*** block output. In the flowgraph shown on Figure 31, the Loop Bandwidth value is ***2π / 150***.

The other parameter ***Costas Loop*** needs to know is the order of the PSK modulation, that is, 2 for BPSK, 4 for QPSK, and 8 for 8PSK.

For the BPSK, the real part of the output signal of the ***Costas Loop*** block is the demodulation of the baseband BPSK signal and the imaginary part is the error signal. This is why the ***Complex To Float*** block is used just after the ***Costas Loop*** block, in order to recover the useful signal. In the case of QPSK and 8PSK, the ***Constellation Decoder*** block is required at the output of the ***Costas Loop*** block, instead of the ***Complex To Float*** block.

The other blocks in the flowgraph play the same roles as in the 2-ASK and 2-FSK demodulators, described previously.

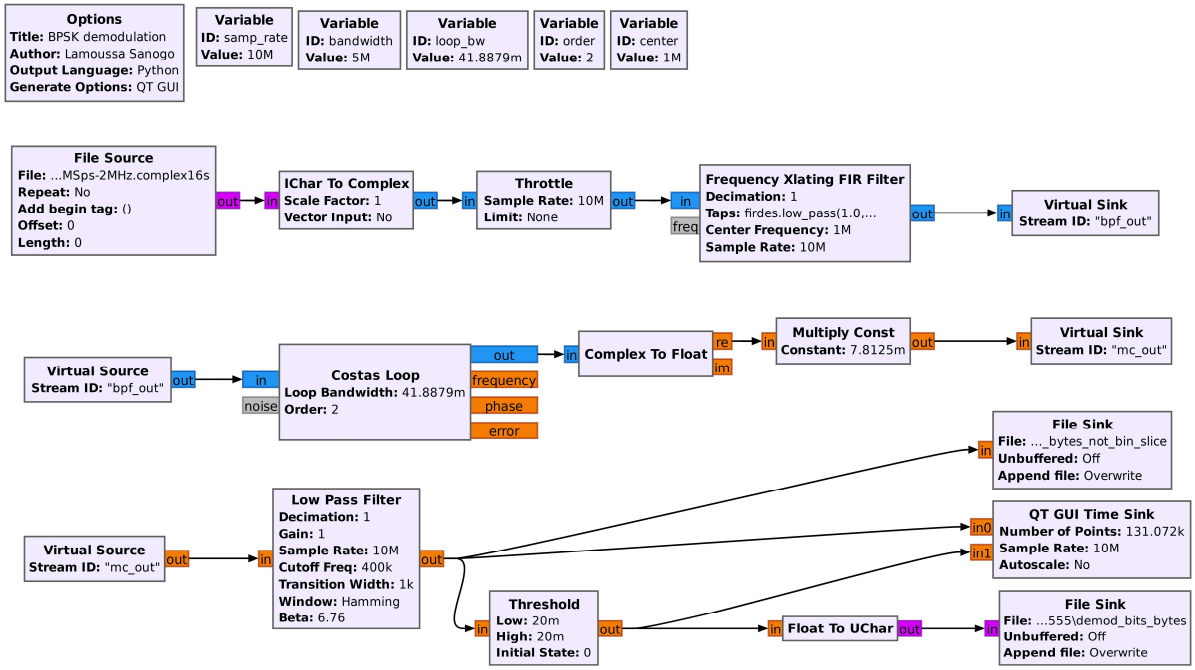


Figure 31. BPSK demodulator GNU Radio flowgraph.

4.3. Experimental setup

The PoSK-based message authentication system was experimented with different values of the PoSK frequency (F_{PoSK}); in different environments (anechoic chamber and regular environment); with different modulations for the main message, i.e., 0 % / 100 %, 25 % / 100 % and 50 % / 100 % ***Amplitude Shift Keying (ASK)*** modulations; 64 kHz / 128 kHz and 500 kHz / 1 MHz ***Frequency Shift Keying (FSK)*** modulations; -90° / +90° and 180° / 0° ***Phase Shift Keying (PSK)*** modulations.

A generic overview of the experimental setup is shown on Figure 32. Indeed, Figure 32 summarizes our experimental process of the PoSK-based message authentication system.

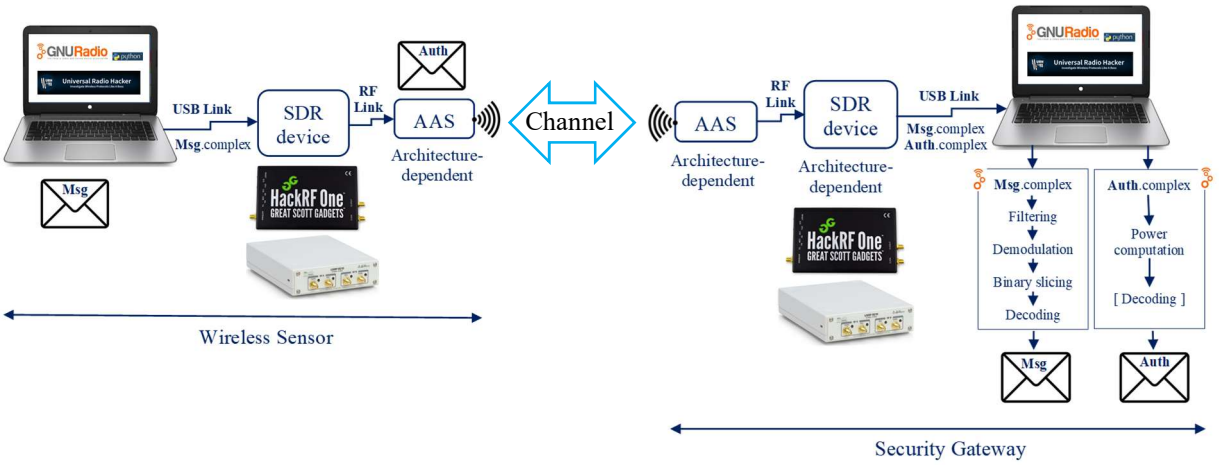


Figure 32. PoSK-based message authentication system experimental setup overview.

On Figure 32, the Wireless Sensor is sending to the Security Gateway the message ***Msg*** and its authentication code ***Auth.*** First of all, the message ***Msg*** is created with the desired parameters, including information, main modulation, symbols rate and sample rate. This task is performed using one of the two software tools, GNU Radio or URH, resulting in the ***Msg.complex*** file containing the message ***Msg*** modulated with the desired parameters. This file is then read by an SDR device (HackRF One or USRP B210) communicating with the PC (laptop) via USB. While reading the message, the SDR device emits it via the Active Antenna System (AAS), which is PoSK architecture-dependent (linear or circular polarizations, synchronization-free or -required).

Indeed, for architectures with linear polarizations, the AAS is made up of a power detector for the "Tx/Rx Start Detection" block, an Arduino UNO board for the "Cryptographic Polarization Selector" block, an SPDT switch for the "Router" block and two identical linear polarization antennas for the H and V corresponding radiative elements (see Figure 16). While, for architectures with circular polarizations, the "Router" block is not an SPDT switch but made up of a Splitter and two identical Phase Shifters, and the radiative element is the single-piece 2-port circular polarization antenna which radiates either in RHCP mode or in LHCP mode, depending on whether its two excitation ports are in positive or negative phase quadrature (see Figure 17).

During the transmission, the AAS performs Polarization Shift Keying following the authentication code ***Auth*** associated with the message ***Msg.complex***. This authentication code is based on the successive outputs of the used cryptographic polarization selection algorithm, that is, the lightweight Linear Feedback Shift Register (LFSR) shown on Figure 33.

As previously mentioned, the AAS of the Wireless Sensor and the one of the Security Gateway need to be time synchronized or not, depending on the architecture. For architectures where time synchronization is required (Figure 16(a) and Figure 17(a)), the two AAS are identical, while for synchronization-free architectures (Figure 16(b) and Figure 17(b)), circuits allowing the SDR device to receive the message and its authentication code simultaneously on two different ports are used. Unlike the AAS, these circuits do not require polarization control

(routing). These architectural differences mean that the Security Gateway SDR device is architecture-dependent, since for synchronization-free architectures, it must have at least two ports and be able to receive simultaneously on these two ports.

Then, when the message ***Msg.complex*** and its authentication code ***Auth.complex*** have been received, the next step is to retrieve the message content (the information) and the authentication code. This is carried out using signal processing techniques and the GNU Radio (Python) software. The message content recovery process involves filtering, demodulation, binary slicing and decoding (return to section "5.3.2. GNU Radio" for more information on the demodulators developed in this work). The decoded message is then written to a file or sent to a Python script for graphical display.

The authentication code recovery involves calculating and decoding the instantaneous power of the ***Auth.complex*** signal. Note that, for architectures where time synchronization is required, authentication uses the power of the signal of the message itself, i.e., the ***Auth.complex*** file is actually the ***Msg.complex*** file.

RMS power in Watts is given by the relation (25). It is then expressed in dBm using relation (26).

$$P_{RMS}(W) = (I^2 + Q^2)/2R \tag{25}$$

$$P(dBm) = 10 * log_{10}[P_{RMS}(W)] + 30 \tag{26}$$

Where I and Q are the In-Phase and Quadrature components of the complex signal, and *R* represents the adaptation, *50Ω* systems are used in this work, so *R=50Ω*. More details on the relations (25) and (26) are provided in [88].

4.4. Cryptographic polarization selection algorithm

In this experimental context, a lightweight Linear Feedback Shift Register (LFSR) is used as a preliminary cryptographic polarization selection algorithm for its ease of implementation. Figure 33 shows the architecture of the used LFSR algorithm.

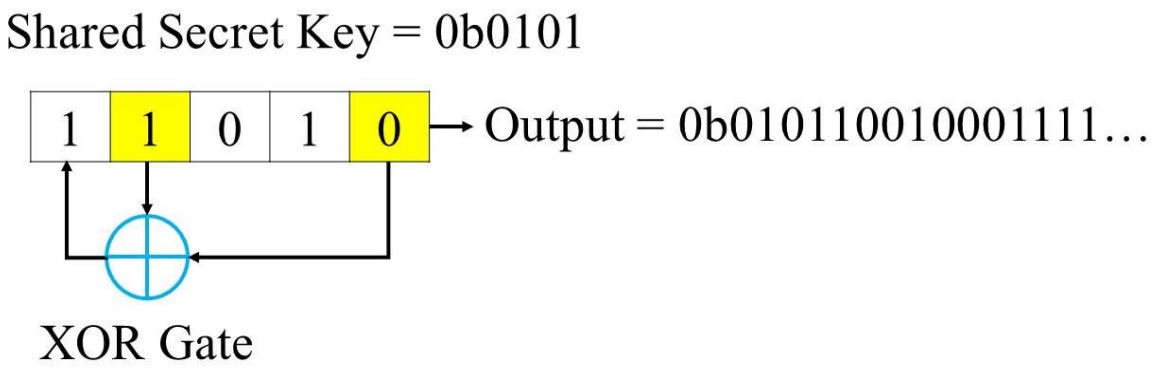


Figure 33. Architecture of the LFSR used in the PoSK experimentation in this work.

Obviously, it is possible to use either a more complex LFSR configuration or any other cryptographic algorithm such as AES (Advanced Encryption Standard). However, it is important to consider as far as possible the important requirements of low computational complexity, low memory footprint and, if possible, an algorithm capable of generating a large number of unique more or less long binary sequences.

4.5. Linear Polarizations Shift Keying Experimentation

4.5.1. Synchronization-required architecture

Here, it is about the experimentation of the PoSK architecture with linear polarizations and with time synchronization between the Wireless Sensor and the Security Gateway (Figure 16(a)). In order to make accurate measurements and do precise analysis and interpretations, the PoSK has been experimented at a very low frequency first. . So, $F_S = 4\ baud$ was used as the main message (i.e., main modulation) symbol rate, that is, 4 bits per second for the 2-symbol modulations; $F_{PoSK} = 16\ Hz$ is the Cryptographic Polarization Selector output rate, i.e., the bit rate of the authentication code. With the experimental hardware used in this work, the synchronization would be even more difficult to implement at high PoSK frequencies.

Figures 34-36 show the results of an experimental example where the main modulation is FSK 64 kHz / 128 kHz. Since it is amplitude-independent, FSK is an appropriate modulation for understanding the impact of PoSK on the transmitted RF signal. Figure 34[a] shows the signal received by the Security Gateway and Figure 34[b] shows its demodulation where one can read the main message, that is, the hexadecimal sequence 0xa12345. Figure 34[c] shows the instantaneous power of this signal. Figure 34(b) shows a transmission of this same message, but in cross-polarization (without PoSK) from vertical (V) to horizontal (H) polarizations. From Figure 34(a) and Figure 34(b), one can already claim that authentication, based on relation (23), is possible as expected. Indeed, taking the power profile after the moving average filtering on Figure 34[c] as $P_d(c,t)$, gives $P_L \approx -10\ dBm$ and $P_H \approx -7\ dBm$, while the power profile after the moving average filtering on Figure 35[b] represents $P_{crossPol}(t) \approx -21\ dBm$ which is at least 10 dBm lower than P_L. Relation (23) is therefore satisfied throughout the transmission of this message, but only if the instantaneous power is filtered (with a moving average in this example). Indeed, two interesting phenomena can be observed on Figures 34-36.

The first phenomenon, to be called Undesired Polarization Sensitivity, is the dependency of the received signal to the polarization. In other words, the Security Gateway does not receive the exact same amount of power depending on whether the co-polarization transmission is vertical (V) to vertical (V) or horizontal (H) to horizontal (H). This phenomenon, which can be seen on Figure 34[a], has the consequence of adding an undesired amplitude modulation (2-ASK in this case of 2-PoSK). The main message shown on Figure 34[a] is actually FSK-modulated and not ASK-modulated. Another consequence of this phenomenon is that it reveals the authentication code via the message signal. Indeed, the power profile shown on Figure 34[c] corresponds to the output (bit-by-bit inverted because of the initial polarization) of the LFSR algorithm shown on Figure 33. Fortunately, the used cryptography is dynamic, i.e., the transmitted authentication code becomes invalid.

This Undesired Polarization Sensitivity is caused by the transmission channel (propagation is better in V to V than in H to H, or vice versa) and the switch losses, which are lower on one of its outputs than on the other.

Although the signal on Figure 34[a] was error-free demodulated using an FSK demodulator, the Undesired Polarization Sensitivity remains particularly disturbing when the main message uses an amplitude-dependent modulation such as ASK (Amplitude Shift Keying) and QAM (Quadrature Amplitude Modulation). So, it is important to avoid it, and to do so, relation (27) must be satisfied.

$$|P_{XX}(t) - P_{YY}(t)| = 0 \tag{27}$$

Where X and Y are the polarizations used to perform the PoSK, and $P_{XX}(t)$ and $P_{YY}(t)$ are received powers in co-polarization X to X and Y to Y transmissions, respectively. In this example, X and Y correspond to V (Vertical) and H (Horizontal) polarizations, respectively. In this case, it is important that the two antennas used for the V and H polarizations to be identical in terms of gain and radiation pattern, to avoid them contributing to this phenomenon.

There are ways of correcting the Undesired Polarization Sensitivity.

A first possible solution would be to use a dedicated port, equipped with a circular polarization (linear polarizations insensitive) antenna, for the reception of the main message only, and use the PoSK port (equipped with the Active Antenna System) for the reception of the authentication signal only. Figure 34(c) shows the signal received by the Security Gateway when equipped with a circular polarization antenna in replacement of the Active Antenna System. This circular polarization antenna is shown on Figure 37, it is a simple 1-port LHCP antenna, not to be confused with the 2-port circular polarization antenna used for the circular PoSK experimentation.
On Figure 36[a], one can see that the Undesired Polarization Sensitivity has disappeared, the amplitude remains constant whether the emission is in vertical (V) or horizontal (H) polarization, as in the case without PoSK.

Another possible way of correcting Undesired Polarization Sensitivity is by using signal processing techniques in the reception software. For example, a hardware or software-defined equalizer or AGC (Automatic Gain Control) system can be used to correct the effect of the undesired polarization sensitivity. However, correcting Undesired Polarization Sensitivity at the emission side is the most advantageous, as this avoids the authentication code being revealed via the message signal; fortunately, this authentication code is dynamic, i.e., it changes continuously.

In addition, the switch having the same loss on each of its outputs contributes to the avoidance of the Undesired Polarization Sensitivity. Also, as previously mentioned, it is important that the two antennas used for the V and H polarizations to be identical in terms of gain and radiation pattern, to avoid them contributing to this phenomenon.

The second interesting phenomenon one can observed on Figures 34-36 is the impact of the polarization switching time (i.e., the time taken by the switch to go from one output to the other) on the main message signal. As mentioned in the section "3. Time synchronization", during the switching time, neither the Sensor circuit (Tx) nor the Security Gateway circuit (Rx) are connected to either of their two antennas, i.e., polarizations, which results in a brief interruption in the transmission at each polarization switching. This phenomenon appears as spikes (sharp and brief power drops) in the instantaneous power, as shown on Figure 34[c]. Therefore, to avoid a significant number of false positives in the authentication, it is important to remove

these spikes using, for example, a filter (a moving average was used in the case shown on Figure 34[c]).

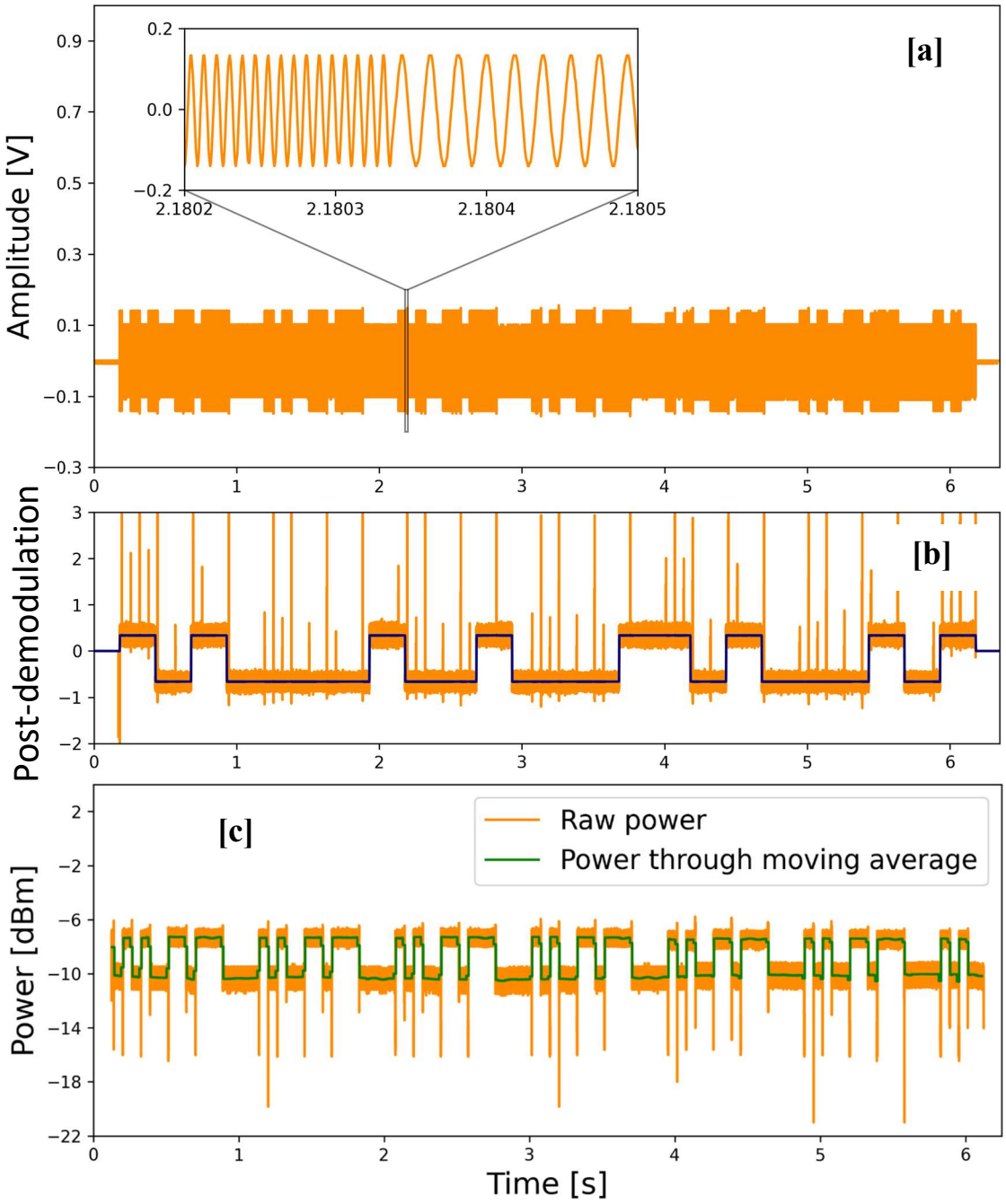


Figure 34. Linear PoSK; Synchronization-required hardware-oriented architecture; Transmission with PoSK; Signals received by the Security Gateway: **[a]** – Received 64 kHz / 128 kHz FSK-modulated analog signal; **[b]** – The demodulation of the 64 kHz / 128 kHz FSK-modulated analog signal shown in [a]; **[c]** – The instantaneous power of the 64 kHz / 128 kHz FSK-modulated analog signal shown in [a].

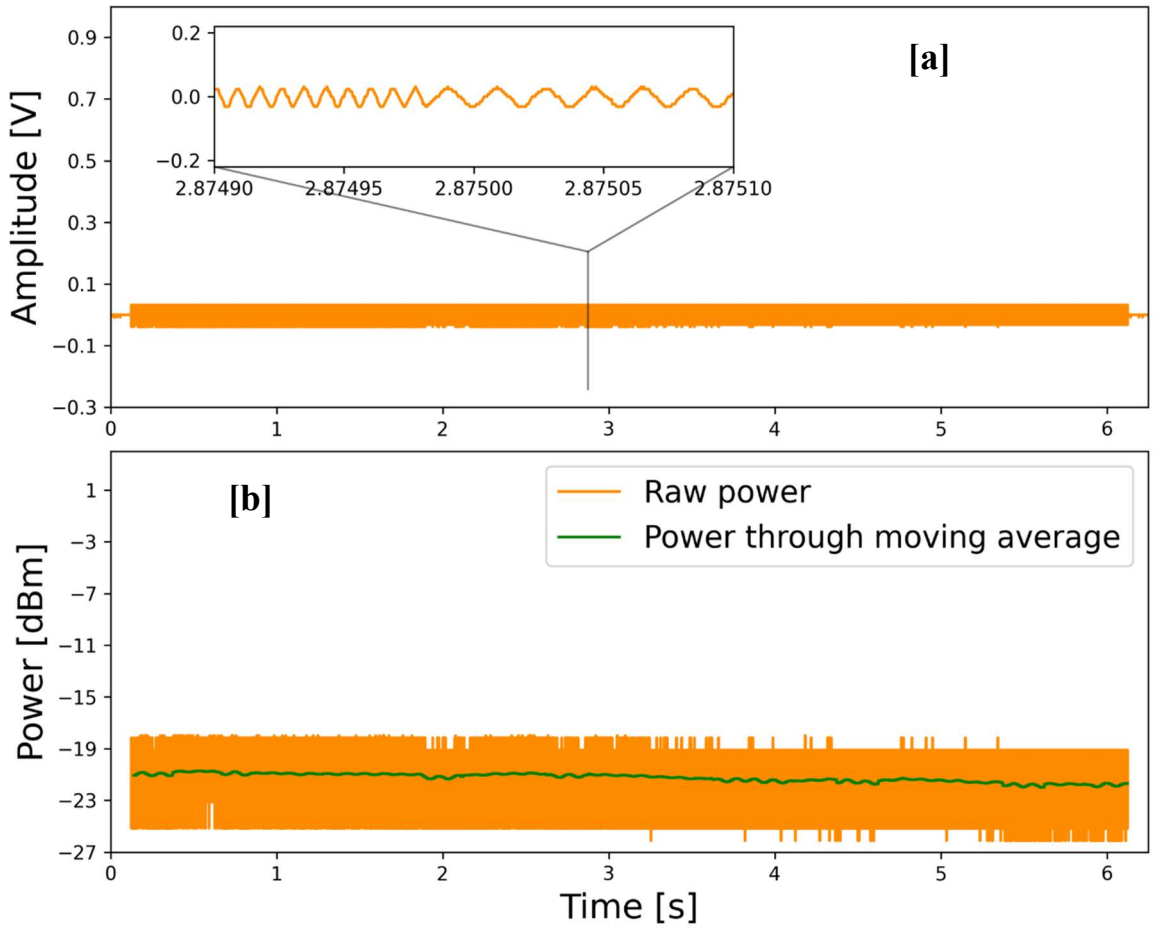


Figure 35. Linear PoSK; Synchronization-required hardware-oriented architecture; Transmission in cross-polarization V to H; Signals received by the Security Gateway: **[a]** - Received 64 kHz / 128 kHz FSK-modulated analog signal; **[b]** - The instantaneous power of the 64 kHz / 128 kHz FSK-modulated analog signal shown in [a].

This phenomenon also appears as noise in the demodulated signal, as shown on Figure 34[b] and Figure 36[b]. In this work, a simple low-pass filter with a cut-off frequency at least equal to the main message symbols rate (F_S) was used to remove this noise before binary slicing, in order to reduce the risk of bit errors due to PoSK. The results of the filtering are presented on Figure 36[c]; the same result is obtained whether the transmission was done with or without PoSK.

These first experiments were carried out in anechoic chamber and the results are promising. The Wireless Sensor authentication by the Security Gateway, based on relation (23), is indeed possible and the PoSK did not alter the signal enough to cause bit errors in the main message, in these standard modulations (2-ASK, 2-FSK and 2-PSK). Throughout a massive series of experiments over several months, the main message was always error-free recovered after an easy and successful demodulation using GNU Radio. Several papers were presented at various international conferences [89], [90], [91].

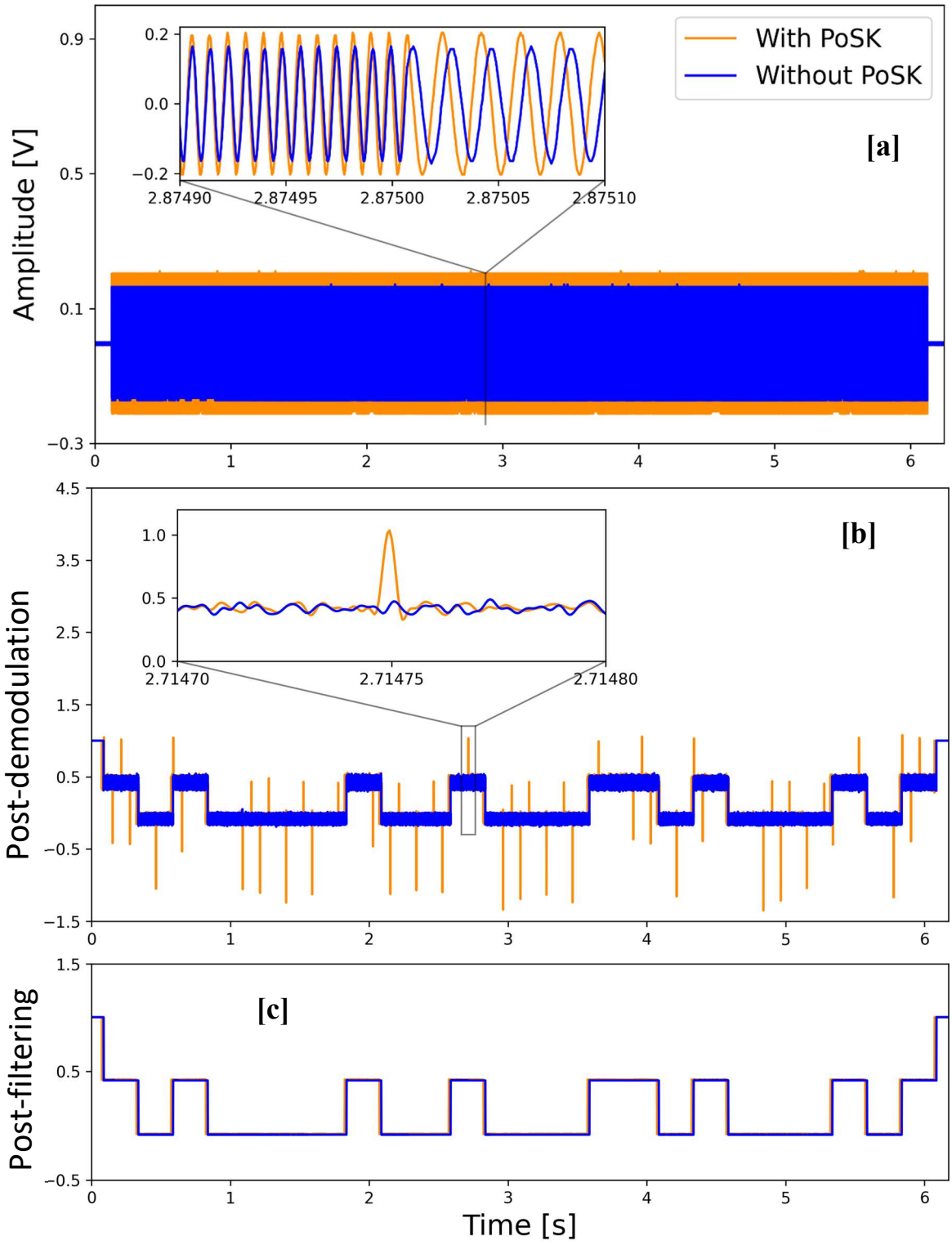


Figure 36. Linear PoSK; Synchronization-required hardware-oriented architecture; Reception with a circular polarization antenna (shown on Figure 37) of the signal emitted with and without PoSK; Signals received by the Security Gateway: **[a]** - Received 64 kHz / 128 kHz FSK-modulated analog signals; **[b]** - The demodulations of the 64 kHz / 128 kHz FSK-modulated analog signals shown in [a]; **[c]** - Post-filtering version of the signals shown on [b]: to reduce the risk of bit errors due to PoSK, a low-pass filter with a cut-off frequency at least equal to the main message symbols rate (F_S) can be used to clean the signal before binary slicing.

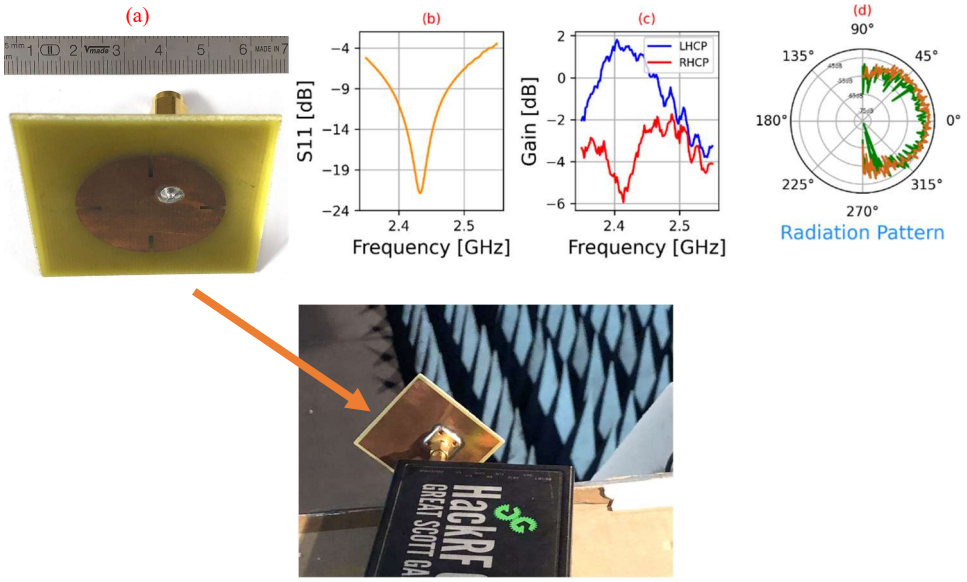


Figure 37. Linear PoSK experimentation: the circular (LHCP) polarization antenna used by the Security Gateway, which led to the results shown on Figure 34(c). For the green plot of the radiation pattern, the reference emitting antenna is at 0° angle relative to the circularly-polarized antenna, and 90° angle for the brown plot.

The experimentation of this PoSK architecture with time synchronization (Figure 16(a)) made us realize how difficult the implementation of time synchronization can be. This is why, we proposed an architecture that does not require time synchronization between the Wireless Sensor Active Antenna System and the Security Gateway Active Antenna System (Figure 16(b)). The next section is dedicated to the experimentation of this synchronization-free architecture.

4.5.2. Synchronization-free architecture

Here, it is about the experimentation of the PoSK architecture with linear polarizations and without time synchronization between the Wireless Sensor and the Security Gateway (Figure 16(b)). The goal is to retrieve the main message received on port P1 of the Security Gateway and the authentication code on its port P2 (see Figure 16(b)). Again, in order to make accurate measurements and do precise analysis and interpretations, the PoSK was experimented at a very low frequency first with $F_S = 4$ *baud* as main message modulation symbols rate and $F_{PoSK} = 16$ *Hz* as Cryptographic Polarization Selector output rate, i.e., the bit rate of the authentication code.

Figures 39-40 show the results of an experimental example of this synchronization-free architecture. Figure 39[a] shows the main message modulated in FSK 64 kHz / 128 kHz, received on port P1 and Figure 39[b] shows its demodulation where one can read the main message, that is, the hexadecimal sequence 0xa12345.

These results are also very interesting. For example, as can be seen on Figure 39[b], the impact of the polarization switching time, - which was strongly present in the synchronization-required architecture -, has more or less disappeared in this experimental example of an architecture that does not use a switch on the Security Gateway. The removal of the switch from the Security Gateway could be the reason for this enhancement. The impact of the polarization switching time is so minor in this example that filtering is barely needed for the error-free decoding of both the main message and the authentication code.

However, the Undesired Polarization Sensitivity is also present in this architecture, as it can be seen in the instantaneous power of the main message signal shown on Figure 39[c]. Relation (27) remains the condition to be satisfied to avoid this phenomenon; in this case *X* and *Y*, again, correspond to V and H polarizations, respectively.

Figure 40[a] shows the authentication signal received on the port P2 of the Security Gateway. Figure 40[b] shows the instantaneous power of this authentication signal. One can see that the profile of this instantaneous power is actually the output (bit-by-bit inverted because of the initial polarization) of the LFSR algorithm shown on Figure 33. One can also see that this profile is indeed a bit-by-bit inversion of the profile of the instantaneous power of the main message, shown on Figure 39[c], which, as previously discussed, is an undesired revelation of the authentication code.

This architecture was experimented in a regular environment and not in anechoic chamber. Therefore, these results are also a demonstration of the resilience of the PoSK to noise, although more experiments are needed to assess the level of this resilience. The experimental environment is presented on Figure 38.

The two other main message modulations experimented with are ASK and PSK. Figure 41 shows an example of a main message, modulated in ASK 50% / 100%, received on port P1 of the Security Gateway. Figure 42 shows an example of a main message, modulated in PSK -90° / +90°, received on port P1 of the Security Gateway.

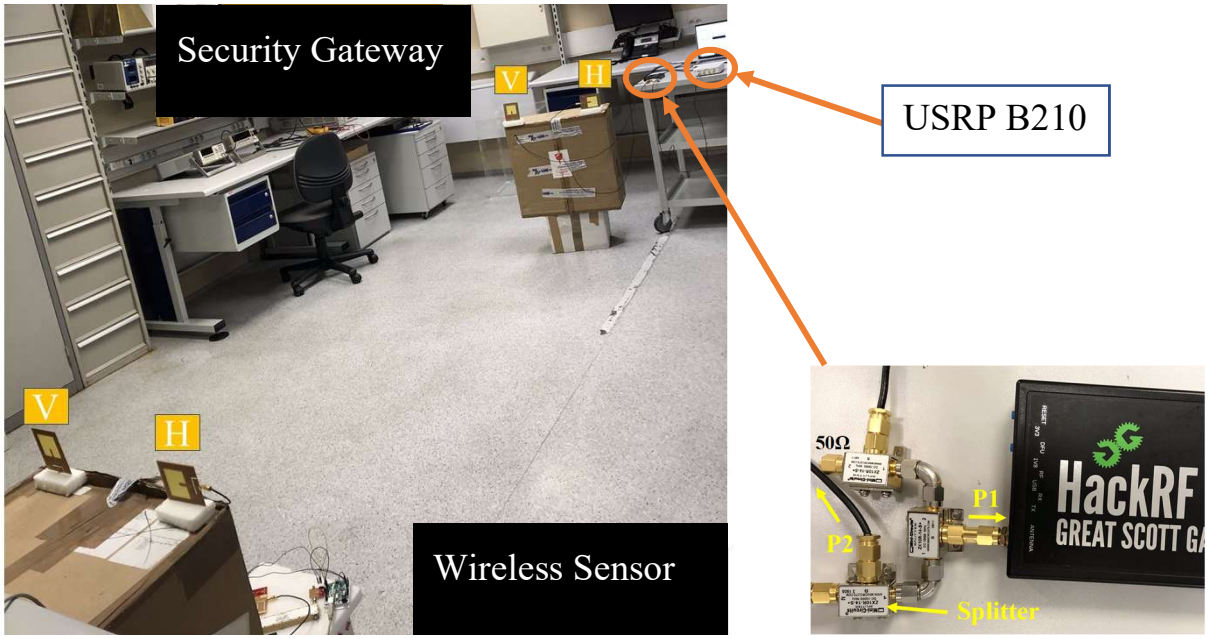


Figure 38. Linear PoSK experimentation: Synchronization-free hardware-oriented architecture (Figure 16(b)) Security Gateway. The port P2 was connected to another SDR device, an USRP B210, as the HackRF One has only one port.

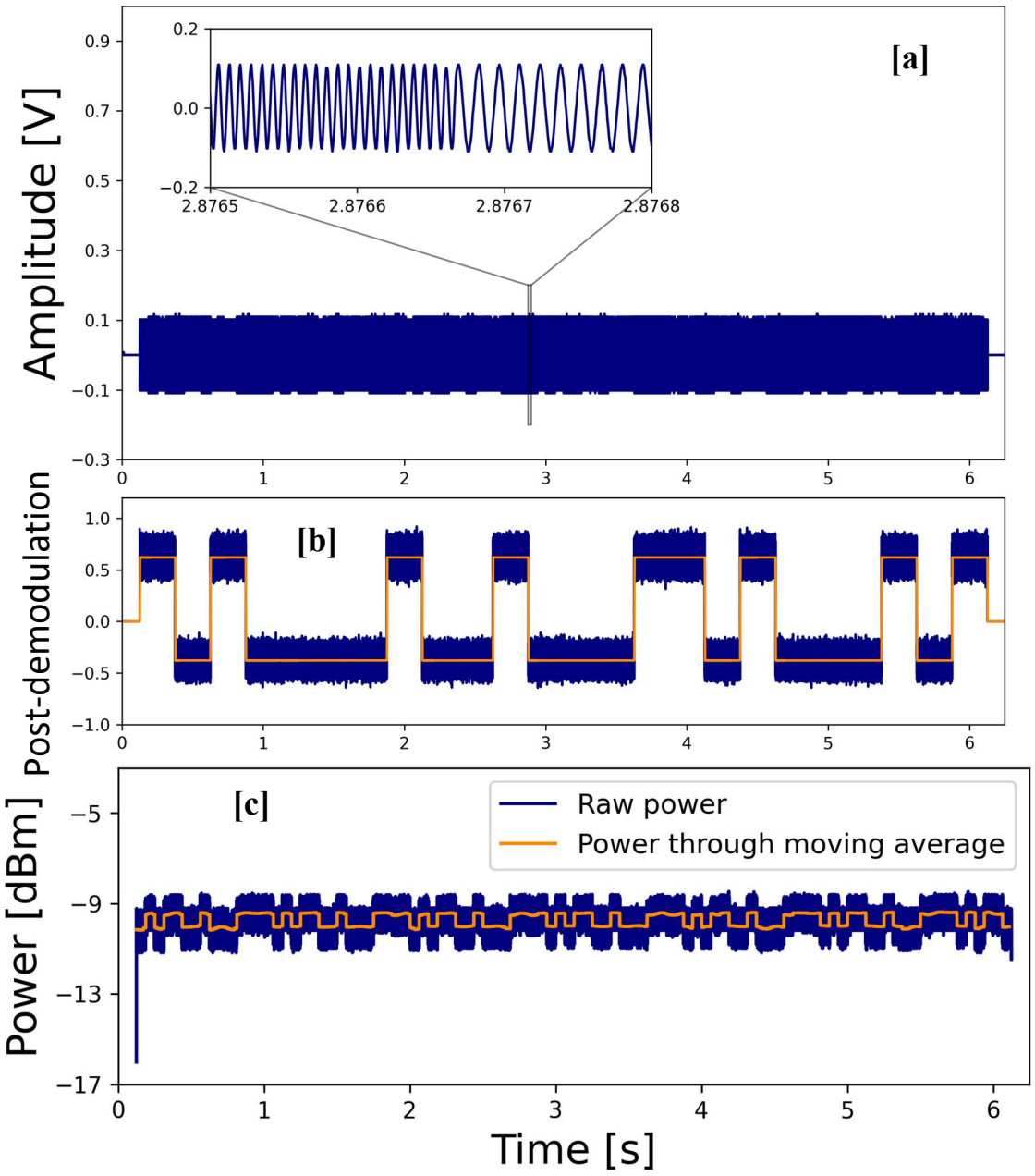


Figure 39. Linear PoSK, Synchronization-free hardware-oriented architecture, Signals received by the Security Gateway, Transmission with PoSK, Main message received on port P1: **[a]** - Received 64 kHz / 128 kHz FSK - modulated analog signal; **[b]** - The demodulation of the 64 kHz / 128 kHz FSK-modulated analog signal shown in [a]; **[c]** - The instantaneous power of the 64 kHz / 128 kHz FSK-modulated analog signal shown in [a].

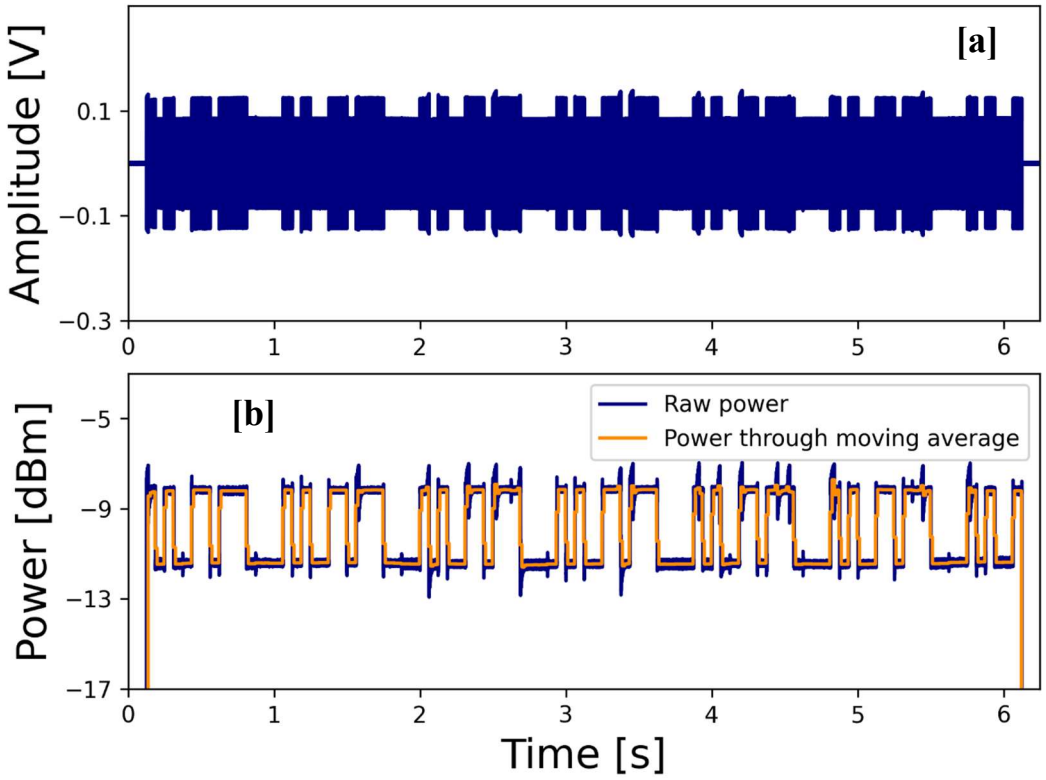


Figure 40. Linear PoSK, Synchronization-free hardware-oriented architecture, Signals received by the Security Gateway, Transmission with PoSK, Authentication code received on port P2 (see Figure 16(b)): **[a]** - Analog raw authentication signal; **[b]** - The instantaneous power of the analog authentication signal shown in [a].

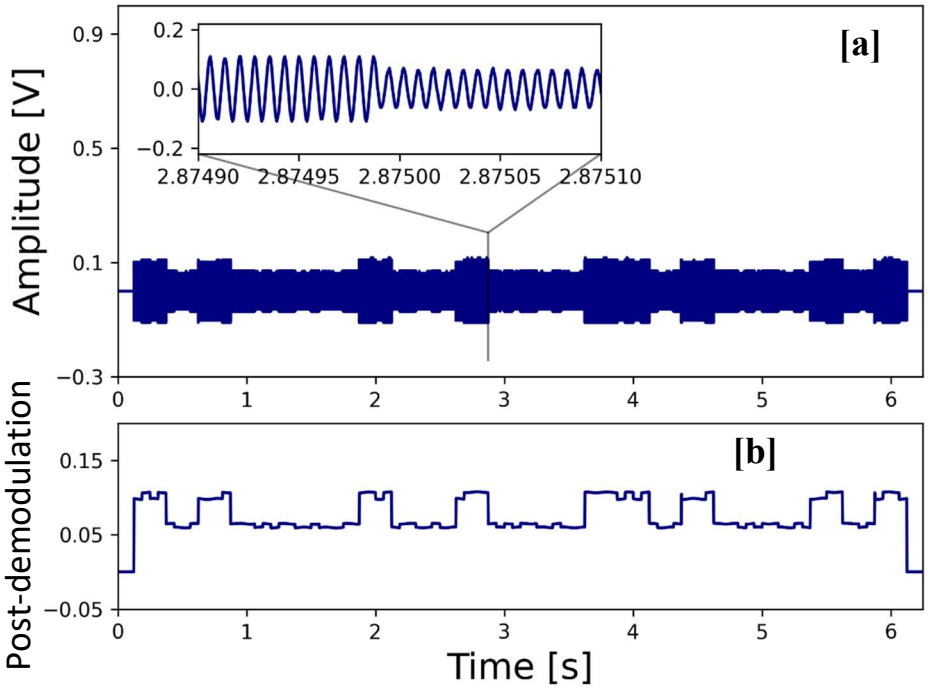


Figure 41. Linear PoSK, Synchronization-free hardware-oriented architecture, Signals received by the Security Gateway, Transmission with PoSK, Main message received on port P1: **[a]** - Received 50% / 100% ASK - modulated analog signal; **[b]** - The demodulation of the 50% / 100% ASK-modulated analog signal shown in [a].

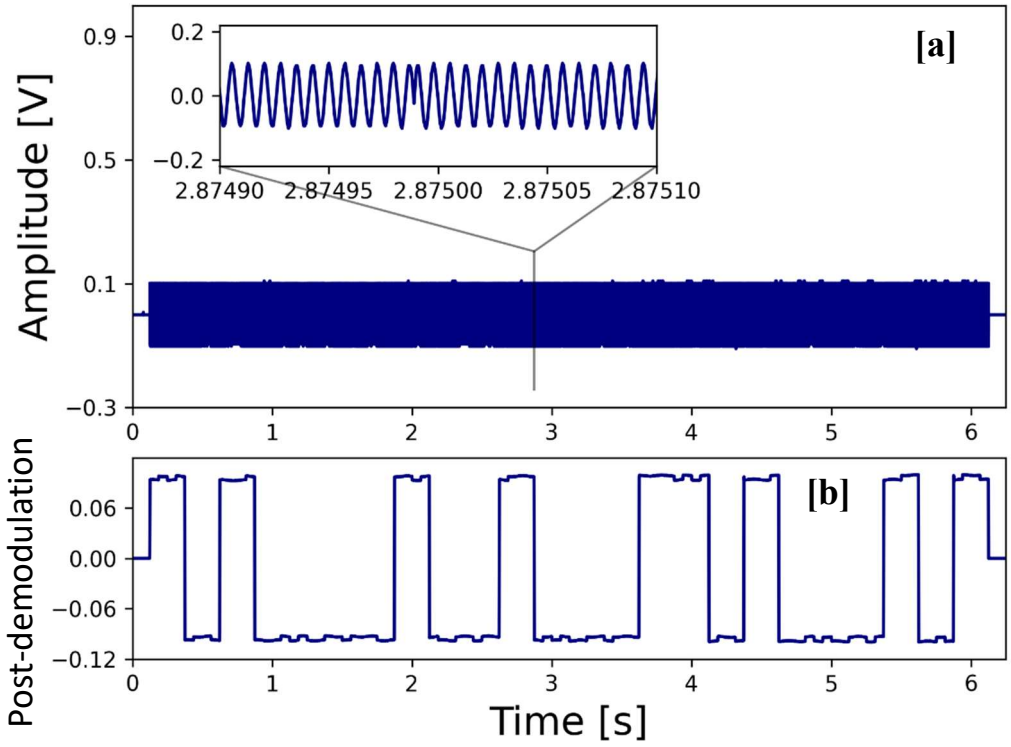


Figure 42. Linear PoSK, Synchronization-free hardware-oriented architecture, Signals received by the Security Gateway, Transmission with PoSK, Main message received on port P1: **[a]** - Received -90° / +90° PSK - modulated analog signal; **[b]** - The demodulation of the -90° / +90° PSK-modulated analog signal shown in [a].

4.5.3. Software-Defined Cryptographic Polarization Controller (CPC)

One of the advantages of software-defined radio (SDR) is its use for rapid prototyping of radio applications, which can be very helpful in research work such as this. In the original idea of the PoSK-based message authentication system, the CPC - Cryptographic Polarization Controller (see Figure 15) is intended for hardware implementation on an FPGA or ASIC platform. However, the use of SDR as an experimental tool has raised our interest in its software implementation, an approach we call software-oriented approach. It turns out that this approach provides great flexibility. It could be integrated during the design phase of the IoT device; or after its deployment by adding the necessary hardware and then updating the firmware. However, this approach could suffer from a limitation in polarization keying frequency, F_{PoSK}, due to longer execution times of the cryptographic algorithm. This approach could also be more power-hungry than the hardware-oriented approach.

In the case of linear polarizations, Figure 43 shows this software-oriented architecture of the PoSK-based message authentication system. In the software-oriented architecture, a second port is required on the Wireless Sensor.

Note that, on a terminal, if cables are needed to connect the two antennas (V and H) to the two ports (P1 and P2), the two cables must have the same length, to avoid introducing an undesired phase shift that could affect the signals received by the Security Gateway, since the result of the addition of two signals depends on the phase shift between them. On Figure 43, one can see that in this experimental case there is no need for cables, the antennas are directly connected to the ports of the USRP B210.

On Figure 43, blocks surrounded in orange are software-defined. For the software implementation of the Cryptographic Polarization Controller block on the Wireless Sensor side, we used Python on GNU Radio. Indeed, custom functions can be implemented in GNU Radio using the Embedded Python Block [92], [93]. This Embedded Python Block can only be used in the flowgraph it was created in, in contrast to an out-of-tree (OOT) module which allows to create a custom Python block as an OOT module which can be installed and used in any flowgraph.

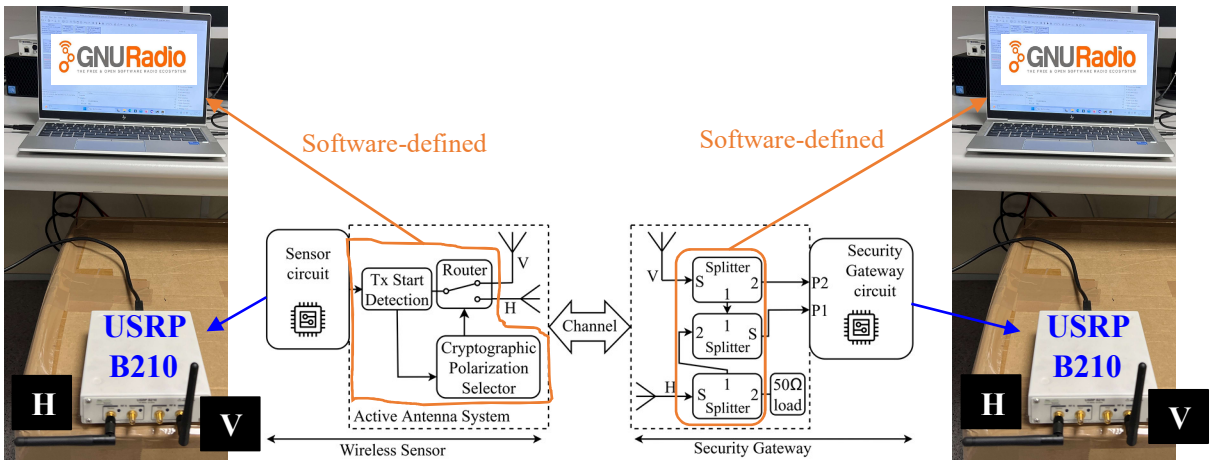


Figure 43. Linear PoSK-based message authentication system, Synchronization-free architecture, Experimentation with Software-Defined Cryptographic Polarization Controller.

Figure 44 presents the GNU Radio flowgraphs for the Wireless Sensor and the Security Gateway, corresponding to the software-oriented architecture shown on Figure 43. Figure 44(a) shows the Wireless Sensor flowgrah and Figure 44(b) shows the Security Gateway flowgraph.

The most important section of the Cryptographic Polarization Controller's Python code is shown on Figure 44(a) in the blue rectangle. In this section, the variable ***new_bit*** represents the current value of the output of the cryptographic polarization selection algorithm. The parameter ***PoSK_Period***, shown on the Cryptographic Polarization Controller block, represents the period at which the variable ***new_bit*** is updated. The maximum possible value of the parameter ***PoSK_Period*** is limited by the execution time of the cryptographic polarization selection algorithm. In these experiments, we have an average execution time of 0.00098 seconds, that is, a maximum PoSK frequency of about 1 kHz. Basically, this value represents the execution time of the LFSR algorithm shown on Figure 33 (a XOR gate, a right shift and the update of the variable ***new_bit***) plus the code shown on Figure 44(a) in the blue rectangle. The execution machine is an HP Elitebook 11th Generation with an Intel(R) Core(TM) i5-1135G7 processor clocked at 2.4 GHz, with 16 GHz RAM memory and 64-bit Windows 11 Professional as OS.

This result shows us that one of the limitations of the software-oriented architecture is the limitation (by the execution time) of the maximum value of the PoSK frequency (F_{PoSK}), i.e., the maximum frequency at which the execution of the cryptographic polarization selection algorithm can be repeated and the polarization eventually changed.

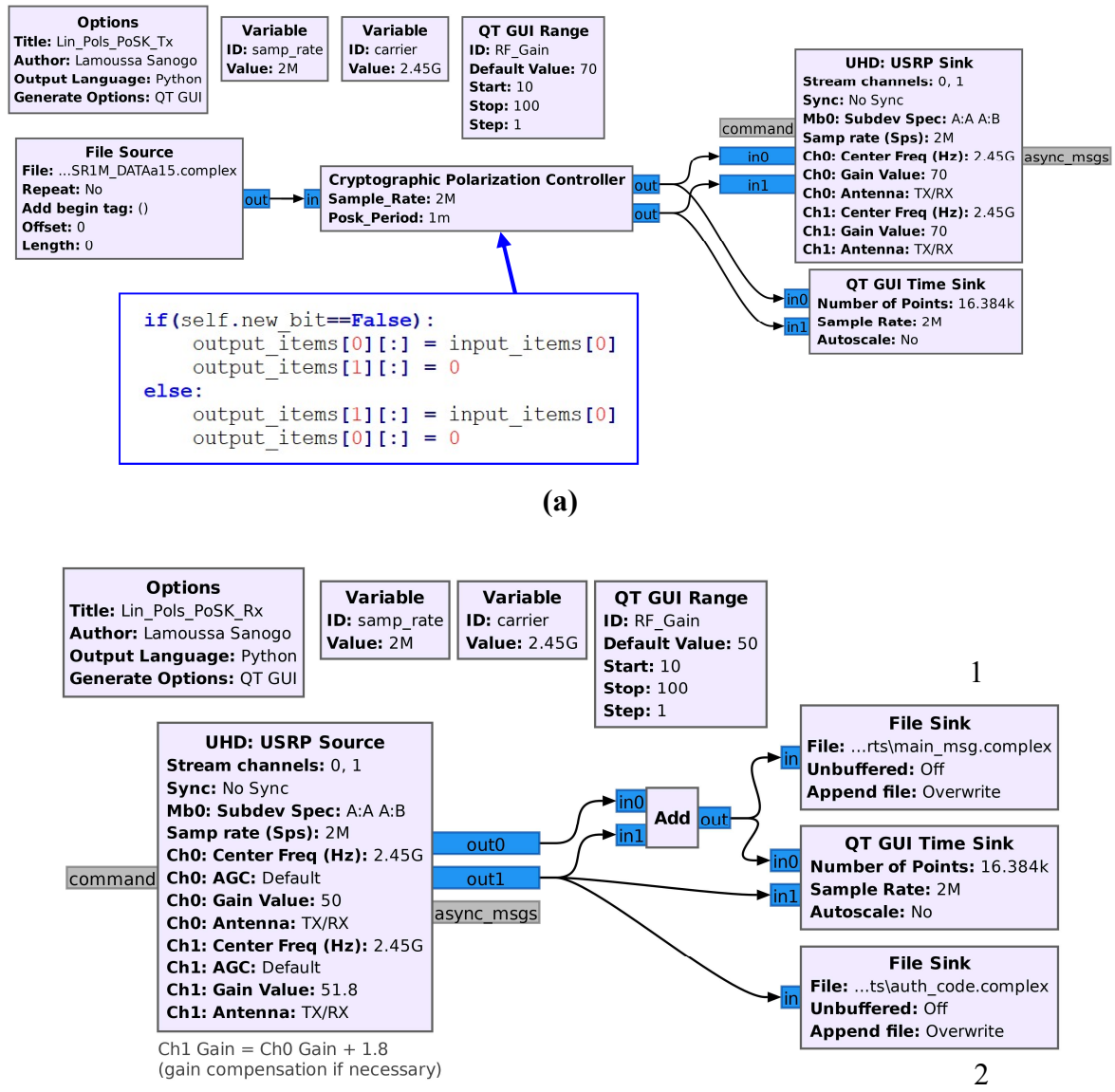


Figure 44. GNU Radio flowgraphs for linear PoSK-based message authentication system, Synchronization-free architecture, Experimentation with Software-Defined Cryptographic Polarization Controller. **(a)**: Wireless Sensor flowgraph. **(b)**: Security Gateway flowgraph.

This also means a limitation of the maximum possible bit rate of the authentication code. On this point, the hardware-oriented architecture, using the hardware switch, wins out over the software-oriented architecture. Indeed, with the hardware-oriented architecture, we can go up to 1 MHz for F_{PoSK}, which is 1000 times higher than with the software-oriented architecture. Therefore, the software-oriented architecture is not suitable if a high bit rate is required for the authentication code, which may be necessary when the main message to be authenticated is very short, for example.

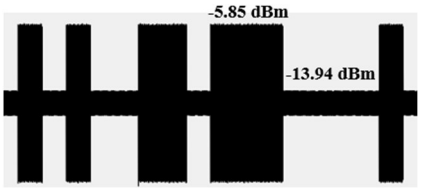
However, this difference in performance between the two architectures should be put into perspective, as the 1 MHz PoSK frequency was experimented with a square-wave signal generated by a waveform generator (see section "5.8. Assessment of the robustness of the

PoSK-based message authentication system"), and therefore without the cryptographic polarization selection algorithm, in other words, without the LFSR program which takes more time to execute. Especially, as Python language, which is a very high-level language, is not the best choice for such an implementation. A better result (an optimized and faster program) could be achieved with a low-level language such as C or even assembly. A fairer comparison of the two architectures, from the point of view of maximum PoSK frequency, will be possible when the hardware implementation of the cryptographic polarization selection algorithm will be done on FPGA or on ASIC.

Let's go back to how this software-oriented architecture works. If the value of the variable ***new_bit*** is 0, the signal at the ***Cryptographic Polarization Controller*** block's input is sent to its first output (emission in polarization V), and if the value of the variable ***new_bit*** is 1, the signal at this block's input is sent to its second output (emission in polarization H). The ***File Source*** block is used to read the file containing the message signal to be transmitted, and the ***UHD: USRP Sink*** block is used to transmit with the USRP B210. The two ports of the USRP B210 (in0 and in1 of the ***UHD: USRP Sink block***) are required in this case, as the Wireless Sensor uses two ports in the software-oriented architecture. On the receiving side, ***UHD: USRP Source*** allows reception with another USRP B210, whose two ports (out0 and out1 of the ***UHD: USRP Source*** block) are also used in this case (synchronization-free architecture). The two ***File Sink*** blocks are used to save the main message (File Sink 1) and the authentication signal (File Sink 2) in two different files, representing Security Gateway ports P1 and P2, respectively. The ***QT GUI Time Sink*** block offers a time-domain display.

At this point, one can say that the software-oriented architecture is less cumbersome and more flexible, at least in an experimental context, than the hardware-oriented architecture. As a proof of this flexibility, we can see on Figure 44(b), at the receiving end, a gain compensation is performed on one of the two ports of the ***UHD: USRP Source*** block, in order to bring the two signals' amplitudes to the same level. The integration of this software-oriented architecture would also be easy during the device design. This integration would consist in adding a second antenna port and adding to the firmware the implementation of the Cryptographic Polarization Controller. However, compared to the hardware-oriented architecture, it would be more difficult to integrate the software-oriented architecture on already deployed IoT devices that did not originally anticipate the use of PoSK-based message authentication system.

When it comes to results, unsurprisingly, we obtained the same results as in hardware-oriented architecure (see previous section "5.6.2. Synchronization-free architecture"), where the Wireless Sensor and the Security Gateway use full-hardware front-ends for the PoSK-based message authentication system, i.e., blocks surrounded in orange on Figure 43 are all hardware. Even better, with the software-oriented architecture we obtained a higher gap (more than 8 dBm) between the power received on the authentication port P2 (see Figure 43) when the transmission is done in polarization V than when it is done in polarization H, as shown on Figure 45. This means a more robust authentication code against noise, as the gap between the logic levels 0 and 1 is larger, more than 8 dBm in the example shown on Figure 45. In addition to this authentication signal recorded by the ***File Sink 2*** block (Figure 44(b)), Figure 45 also shows the main message recorded by the ***File Sink 1*** block (Figure 44(b)).



Authentication code signal => Figure 44(b) => ***File Sink 2*** block



Main message signal => Figure 44(b) => ***File Sink 1*** block

Figure 45. Linear PoSK, Synchronization-free architecture experimentation with Software-Defined Cryptographic Polarization Controller, Authentication code and Main message signals.

Despite these good results and an architecture that appears more flexible, more lightweight and easier to implement, this software-oriented architecture is nevertheless very likely the one that consumes more energy than the hardware-oriented architecture. A study of the power consumption of this software-oriented architecture was not carried out in this initial work, and is therefore part of future work. Another, but less critical, limitation of this software-oriented architecture will be the execution time of the cryptographic polarization selection algorithm, which could be longer than in the hardware-oriented architecture. This means a lower maximum frequency at which the execution of the cryptographic polarization selection algorithm can be repeated.

4.6. Circular Polarizations Shift Keying Experimentation

Since the experimentation of the time synchronization between the Wireless and the Security Gateway, remains the same as in the case of the architecture with linear polarizations, and considering its difficult implementation, synchronization-required architecture with circular polarizations was not experimented in this work. Only the synchronization-free architecture (Figure 17(b)) was experimented.

As with linear polarizations, in the synchronization-free architecture with circular polarizations, the objective remains the recovery of the main message received on port P1 of the Security Gateway and the authentication code on its port P2 (see Figure 17(b)). Again, in order to make accurate measurements and do precise analysis and interpretations, the PoSK was experimented at a very low frequency first with $F_S = 4\ baud$ as main message modulation symbols rate and $F_{PoSK} = 16\ Hz$ as Cryptographic Polarization Selector output rate, i.e., the bit rate of the authentication code.

In this case too, we investigated both hardware- and software- oriented approaches of the Cryptographic Polarization Controller (CPC).

4.6.1. Experimentation with Hardware CPC

For the experimentation of the hardware-oriented synchronization-free architecture, the evaluation boards of the HMC647ALP6E [86] phase shifter (EV1HMC647ALP6) was used. This component represents the "Phase Shifter" block of the PoSK architectures with circular polarizations, shown on Figure 17. Figure 25 shows an image of the EV1HMC647ALP6 and some characteristic diagrams of the HMC647ALP6E as provided by its datasheet. For the "Splitter" block, the Q-053-90, a 90 Degree Hybrid Coupler from ET Industries, can be used instead of the Mini-Circuits ZX10R-14-S+ [85], as shown on Figure 46. A better picture of this component is provided by Figure 24(b). The advantage of this other symmetrical splitter is that it introduces a 90° phase shift between its two outputs, which makes it an ideal splitter for our

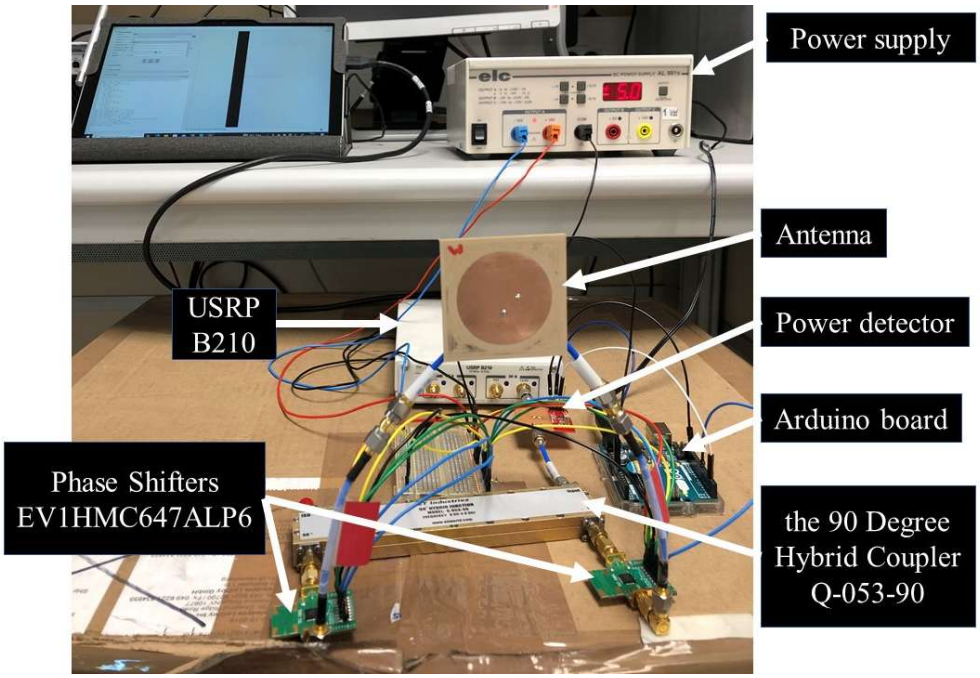
2-port LHCP- RHCP- enabled antenna, since this antenna requires its two excitation ports to be in positive (negative) phase quadrature to radiate in LHCP (RHCP). The use of the Q-053-90 can be particularly interesting on the Security Gateway side in the synchronization-free architecture, as this avoids the use of EV1HMC647ALP6, which require at least one controller and are more expensive.

For the "Tx/Rx Start Detection" block, the custom-made Linear Technology's LTC5536 -based power detector, described in section "5.2.2. Power Detection", was used for the detection of the beginning of the transmission throughout this work. Figure 22 shows the electrical schematic of this power detector and its PCB. The Arduino board remains the cryptographic polarization selector.

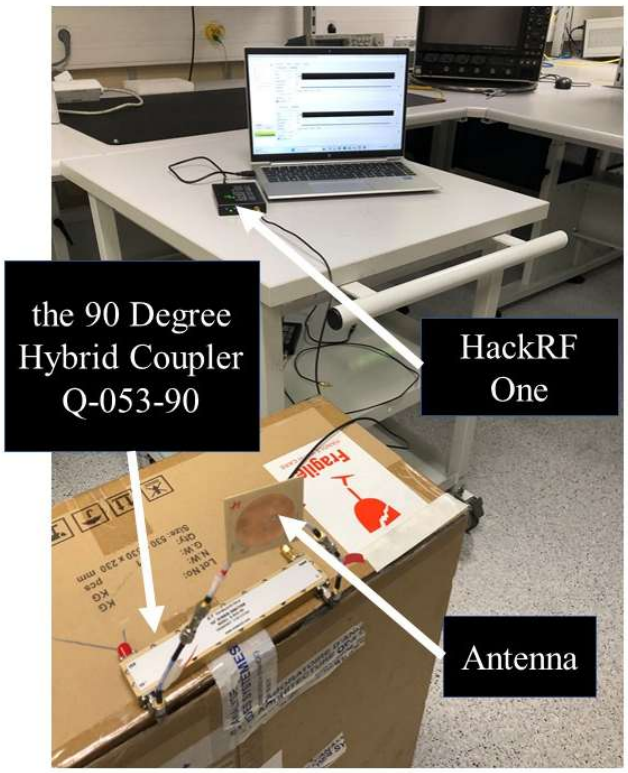
Figure 46 shows an experimental setup in which the Security Gateway (Figure 46(b)) was configured to receive on only one of the two polarizations, the LHCP - i.e., the Security Gateway of Figure 17(b) without the RHCP part, and without the port P1 -, while the Wireless Sensor (Figure 46(a)), when transmitting the message, was performing PoSK using EV1HMC647ALP6 [86] phase shifters (see Figure 17(b)). In other words, Figure 46 shows an experimental setup in which the signal received by the Security Gateway is what will be received on its port P2 in a full implementation (see Figure 17(b)), while the Wireless Sensor, when transmitting the message, was performing PoSK using EV1HMC647ALP6 [86] phase shifters.

On the Wireless Sensor experimental setup shown on Figure 46(a), remember that the two cables used to connect the two ports of the antenna to the two EV1HMC647ALP6 phase shifters have the same length, to avoid introducing an undesirable phase shift. On the Security Gateway side (Figure 46(b)), the same applies to the two cables used to connect the two ports of the antenna to the two ports of the 90 Degree Hybrid Coupler Q-053-90, the latter is a combiner in this configuration.

Figure 47[a] shows a signal received by the Security Gateway in this configuration (Figure 46). This is actually the authentication signal that would be received on the port P2 of the Security Gateway in a full implementation of the circular PoSK synchronization-free architecture (Figure 17(b)). So, Figure 47[b] shows the authentication code, that is, the instantaneous power of the authentication signal shown on Figure 47[a].



(a)



(b)

Figure 46. Circular PoSK experimentation, Synchronization-free hardware-oriented architecture: **(a)**: The Wireless Sensor; **(b)**: The Security Gateway configured to receive on only one of the two polarizations, the LHCP (port P2 on Figure 17(b)), i.e., the Security Gateway of Figure 17(b) without the RHCP part, and without the port P1.

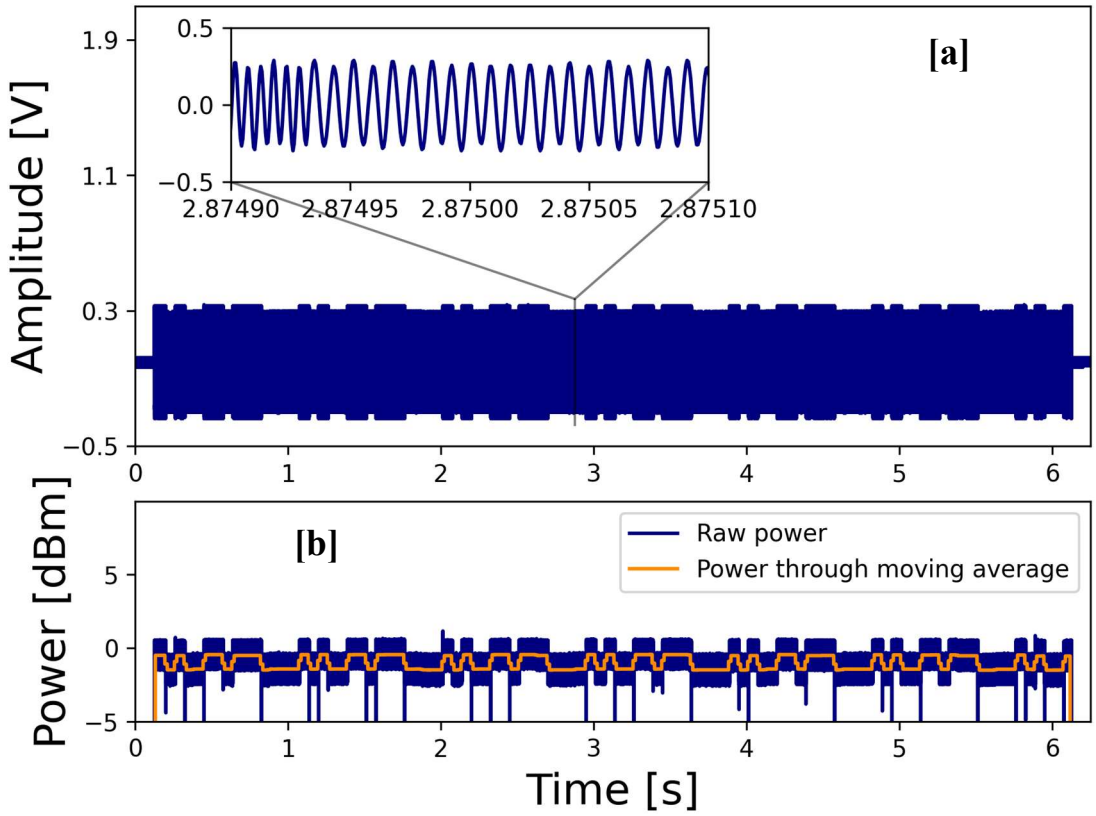


Figure 47. Circular PoSK, Synchronization-free hardware-oriented architecture, Signal received by the Security Gateway on its port P2, while the Wireless Sensor, when transmitting the message, was performing PoSK using the evaluation boards of the HMC647ALP6E phase shifter (EV1HMC647ALP6). **[a]** - Received 64 kHz / 128 kHz FSK-modulated analog signal; **[b]** - The instantaneous power of the 64 kHz / 128 kHz FSK-modulated analog signal shown in [a]. The experimental setup of this experiment is shown on Figure 46.

But, this result on Figure 47 is not very good. As it can be seen on Figure 47[b], the difference between the received powers on the authentication port P2 of a RHCP transmission and a LHCP transmission is very small (about 1 dBm), which means that the gap between logic levels 0 (LHCP transmission) and 1 (RHCP transmission) of the authentication code is very small; therefore, the authentication code will be very sensitive to noise in such a case.

In other words, the Security Gateway is receiving on its port P2, almost the same amount of power in a co-polarization LHCP-to-LHCP transmission as in a cross-polarization RHCP-to-LHCP transmission. This is due to the fact that the HMC647ALP6E [86] phase shifter is not designed to operate at 2.45 GHz, although in its datasheet, its characteristics at this frequency seem to be acceptable for this application. We realized that the problem was, indeed, coming from the EV1HMC647ALP6 [86] phase shifters by experimenting with a software-defined Cryptographic Polarization Controller, hence a software-defined phase shifter.; and the results obtained with this software-oriented architecture are very good, as we will see in the next section.

When it comes to PoSK-related phenomena, it can be seen on Figure 47[b] that the impact of the polarization switching time is present in the authentication signal in this experimental example of an architecture that does not use any switches. Indeed, in this case, the polarization switching time phenomenon is rather caused by the time it takes the phase shifter to stabilize

on its new phase, i.e., the time required for the phase shift to be effective. The presence of the Undesired Polarization Sensitivity phenomenon, especially in the main message signal, in this hardware-oriented architecture, was not investigated in this work and should therefore be part of future work.

4.6.2. Experimentation with Software-Defined CPC

From an architectural point of view, the software-oriented architecture with circular polarizations remains similar to that with linear polarizations, as shown on Figure 48. In the software-oriented architecture, a second port is required on the Wireless Sensor.

Note that, on a terminal, the two cables used to connect the two ports of the antenna to the two ports of the USRP B210 have the same length, to avoid introducing an undesired phase shift that would not only impair the proper operation of the antenna, but also affect the signals received by the Security Gateway, since the result of the addition of two signals depends on the phase shift between them.

Figure 49 shows the circular polarization antenna we designed for this application, as well as its connection to the USRP B210 serving as Security Gateway.

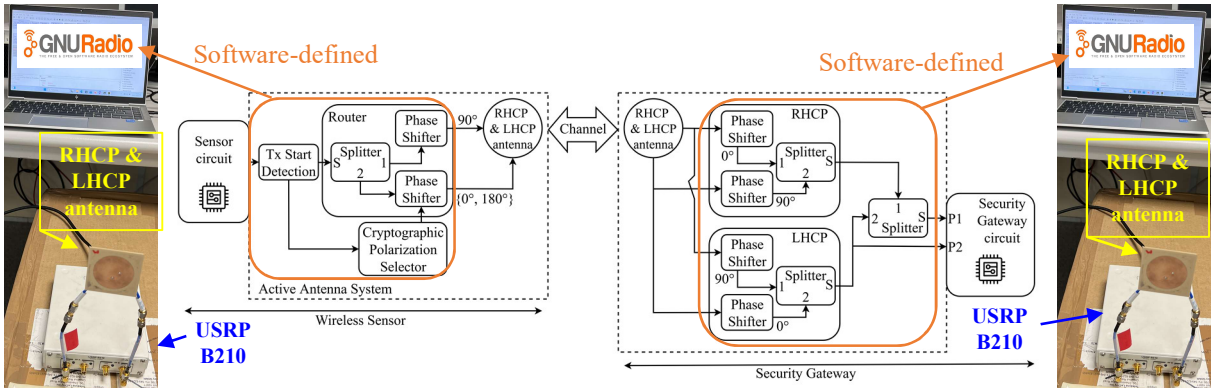
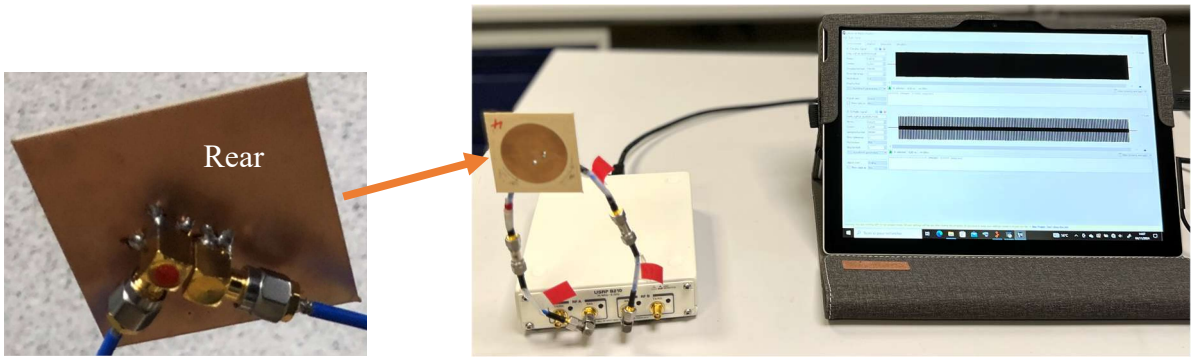


Figure 48. Circular PoSK-based message authentication system, Synchronization-free architecture, Experimentation with Software-Defined Cryptographic Polarization Controller.

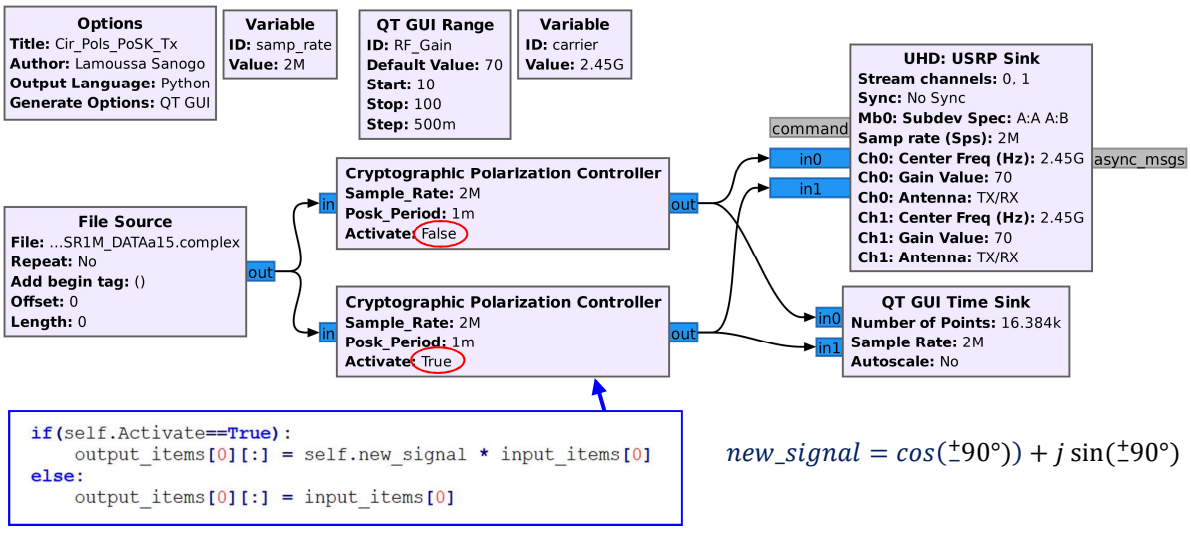
On Figure 48, blocks surrounded in orange are software-defined. For the implementation of the Cryptographic Polarization Controller block (Wireless Sensor side), we again used Python on GNU Radio. Indeed, custom functions can be implemented in GNU Radio using the Embedded Python Block [92], [93]. Figure 50 shows the GNU Radio flowgraphs of the Wireless Sensor and the Security Gateway, corresponding to the software-oriented architecture shown on Figure 48.



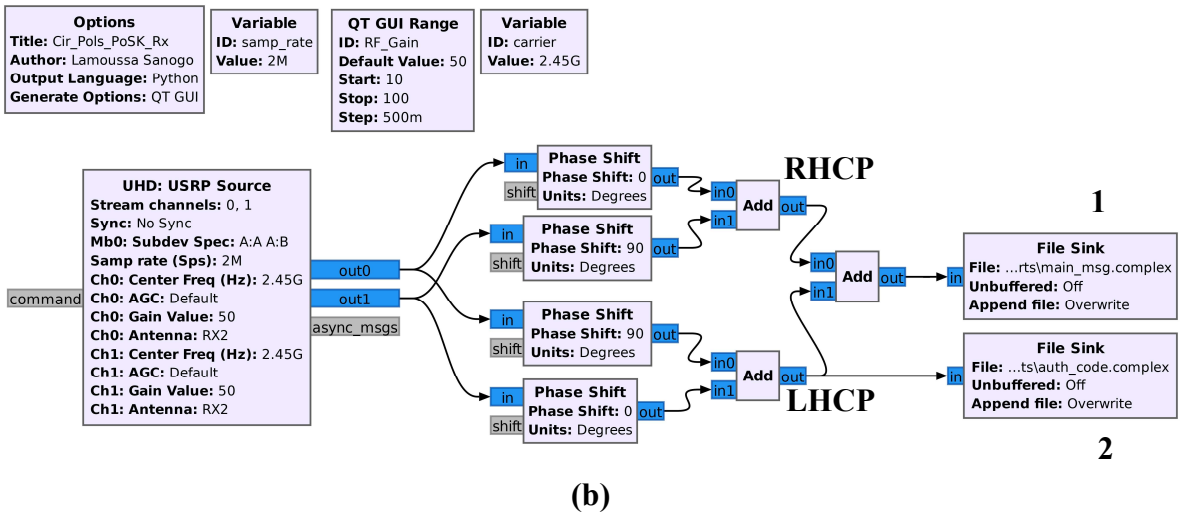
(a) **(b)**

Figure 49. Circular PoSK experimentation, Synchronization-free software-oriented architecture: **(a)** - the single-piece 2-port circular polarization antenna; **(b)** - Security Gateway experimental setup in the synchronization-free software-oriented architecture (the Cryptographic Polarization Controller is software-defined). The two cables connecting the antenna to the USRP B210 must be the same length to avoid an undesired offset in the phase shift between the two ports.

On the Wireless Sensor flowgraph (Figure 50(a)), the ***File Source*** block is used to read the file containing the message signal to be transmitted. The ***UHD: USRP Sink*** block is used to transmit with the USRP B210, and the ***QT GUI Time Sink*** block provides a time-domain display. Before moving on to the most important block in this flowgraph, that is, the Cryptographic Polarization Controller block, let's remember how circular PoSK works. As shown on Figure 48, a single-piece 2-port antenna is used to transmit in either RHCP or LHCP polarization, depending on the value of the phase shift between its two ports, hence the use of the two ports (in0 and in1 of the ***UHD: USRP Sink*** block) of the USRP B210.



(a)



(b)

Figure 50. GNU Radio flowgraphs for circular PoSK-based message authentication system, Synchronization-free architecture, Experimentation with Software-Defined Cryptographic Polarization Controller. **(a)**: Wireless Sensor flowgraph. **(b)**: Security Gateway flowgraph.

Now, let's discuss the ***Cryptographic Polarization Controller*** block, which is used twice in the flowgraph of the Wireless Sensor (Figure 50(a)).

This ***Cryptographic Polarization Controller*** block, when activated (parameter ***Activate*** = ***True***), introduces +/-90° phase shifts to the signal at its input before sending it to its output. This results in the Python code shown on Figure 50(a) in the blue rectangle, which is the most important section of this block's Python code. In this section of the code, the argument of the two functions ***cos*** (cosinus) and ***sin*** (sinus) of the variable ***new_signal*** changes according to the current value of the output of the cryptographic polarization selection algorithm. This argument is -90° when the output of the cryptographic polarization selection algorithm is 0, i.e., ***new_signal*** = ***-j***, which allows to perform the -90° phase shift. And the argument of the functions ***cos*** and ***sin*** is +90° when the output of the cryptographic polarization selection algorithm is 1, i.e., ***new_signal*** = ***+j***, which allows to perform the +90° phase shift. The parameter ***PoSK_Period***, shown on the Cryptographic Polarization Controller block, represents the period at which the argument of the functions ***cos*** and ***sin***, and therefore the variable ***new_signal***, is updated. The maximum possible frequency of this update is limited by the execution time of the cryptographic polarization selection algorithm. In these experiments, we have an average execution time of 0.001 seconds, that is, a maximum PoSK frequency of about 1 kHz. Basically, this value represents the execution time of the LFSR algorithm shown on Figure 33 (a XOR gate, a right shift and the update of the variable ***new_signal***) plus the code shown on Figure 50(a) in the blue rectangle. The execution machine is an HP Elitebook 11th Generation with an Intel(R) Core(TM) i5-1135G7 processor clocked at 2.4 GHz, with 16 GHz RAM memory and 64-bit Windows 11 Professional as OS.

Once again, this result shows us that one of the limitations of the software-oriented architecture is the limitation (by the execution time) of the maximum value of the PoSK frequency (F_{PoSK}), i.e., the maximum frequency at which the execution of the cryptographic polarization selection algorithm can be repeated and the polarization eventually changed. This also means a limitation of the maximum possible bit rate of the authentication code. Therefore, as with linear polarizations, software-oriented architecture with circular polarizations may not be suitable if a

high bit rate is required for the authentication code, which may be necessary when the main message to be authenticated is very short, for example.

However, for a fairer comparison of the hardware- and software- oriented architectures, it would be preferable to use another language for the implementation instead of Python. Indeed, Python is a very high-level language and is not ideal in such an application. A better result (an optimized and faster program) could be achieved with a low-level language like C or even assembly.

Let's go back to how this software-oriented architecture works. When the ***Cryptographic Polarization Controller*** block is not activated (parameter ***Activate = False***), the block simply forwards the signal as it received it, i.e., it sends its input signal to its output without any modification. One might ask what the advantage of this parameter ***Activate*** is then? The answer is that this parameter allows the use of two instances of the ***Cryptographic Polarization Controller*** block, as shown in the Wireless Sensor GNU Radio flowgraph on Figure 50(a). However, this answer leads to another question: why are we using two instances of the ***Cryptographic Polarization Controller*** block, one of which is activated (parameter ***Activate = True***) and the other is deactivated (parameter ***Activate = False***)? The aim of this maneuver is to compensate, as best as possible, for the time delay introduced by the ***Cryptographic Polarization Controller*** block, so that the signals on both paths arrive together at the ***UHD: USRP Sink*** block without no additional undesired phase shift. So, the only purpose of the deactivated instance (parameter ***Activate = False***) is to compensate for the time delay introduced by the activated instance (parameter ***Activate = True***) located on the other path. This delay compensation is necessary because the delay actually means adding an unwanted phase shift, which could affect the proper operation of the 2-port antenna, which must have a phase shift of -90° (or +90°) between its two excitation ports to radiate in LHCP (or RHCP). With this delay compensation, we have a system that works very well despite the fact that the compensation is not perfect, since the deactivated instance does not perform the multiplication by the variable ***new_signal***, this instance is therefore very likely faster than the activated one.

To further improve this time delay compensation, one could make the deactivated instance of the ***Cryptographic Polarization Controller*** block always perform a multiplication by 1, in order to make the execution time of the two instances to be as close as possible. Another possibility would be as follows: the deactivated instance could always multiply by +j, that is, a constant phase shift of +90°, and the activated instance would multiply by 1 (a phase shift of 0°) or -1 (a phase shift of 180°) following the output of the cryptographic polarization selection algorithm. In fact, this latter possibility of configuration is the one that matches the architectures of the circular PoSK-based message authentication system presented in Figure 17.
Remember that in hardware-oriented architecture, two identical phase shifters are used to compensate for time delay and phase shifter insertion losses.

On the Security Gateway GNU Radio flowgraph (Figure 50(b)), the reception is made with the ***UHD: USRP Source*** block, which is materialized by another USRP B210 whose two ports (out0 and out1 of the ***UHD: USRP Source*** block) are also used to connect the two ports of the receiving antenna. As shown on Figure 50(b), the circuit for receiving simultaneously the main message signal and the authentication signal is made up of a set of phase shifters and adders, at the ends of which two ***File Sink*** blocks are connected. The ***File Sink*** 1 block (on the top) represents the port for the main message; and the ***File Sink*** 2 block (on the bottom) represents

the port for the authentication code. This flowgraph corresponds to the configuration shown on Figure 48, the ***File Sink*** 1 block represents port P1, the ***File Sink*** 2 block represents port P2, and the ***Add*** (the adders) blocks represent the splitters.

When it comes to results, unsurprisingly, once again we obtained the same results as in hardware-oriented architecture (see previous section "5.7.1. Experimentation with Hardware CPC"), where the Wireless Sensor and the Security Gateway use full-hardware front-ends for the PoSK-based message authentication system, i.e., blocks surrounded in orange on Figure 48 are all hardware. The quality of the results is even better with the software-oriented architecture. Indeed, this software-oriented approach turned out to be, very effective for proving the feasibility of circular Polarizations Shift Keying, since excellent results have been achieved with this software-defined approach. Figures 51-52 show an example of experimental results of this software-oriented architecture. Figure 51[a] shows the signal of the main message, modulated in FSK 64 kHz / 128 kHz, received by the Security Gateway on its port P1 (see Figure 17(b)) and Figure 51[b] shows the demodulation of this signal where one can read the main message, that is, the hexadecimal sequence 0xa12345. It can be seen on Figure 51[b] that the impact of the polarization switching time on the main message is significant in this experimental example of an architecture that does not use any switches. Indeed, in this case, the polarization switching time phenomenon is rather caused by the time it takes the phase shifter (hardware or software-defined) to stabilize on its new phase, i.e., the time required for the phase shift to be effective. As seen in the previous section ("5.7.1. Experimentation with Hardware CPC"), this phenomenon is also present, at least in the authentication signal, in the hardware-oriented version of the same architecture (see Figure 47[b]).

However, there is a positive point in this example, that is, the Undesired Polarization Sensitivity phenomenon is not present, as it can be seen on Figure 51[c], which shows the instantaneous power of the main message signal. This is proof that this phenomenon can be avoided in this circular PoSK, synchronization-free, software-oriented architecture. Nevertheless, the architectures using circular polarizations are also vulnerable to the Undesired Polarization Sensitivity, a phenomenon to which the transmission channel remains a major contributor (propagation is better in RHCP to RHCP than in LHCP to LHCP, or vice versa). In addition, in architectures with circular polarizations, another factor that can strongly contribute to the Undesired Polarization Sensitivity phenomenon is phase noise, since the result of the addition of two signals depends on the phase shift between them, the phase noises of the transmitter and the receiver become potential sources of fluctuations in the power levels of the received signals.

Figure 52[a] is the authentication signal received on the port P2 of the Security Gateway (see Figure 17(b)) and Figure 52[b] shows the instantaneous power of this authentication signal, which is actually the output (bit-by-bit inverted because of the initial polarization) of the LFSR algorithm shown on Figure 33. In this example, the impact of the polarization switching time on the instantaneous power of the authentication signal is so minor that filtering is not necessary, in contrast to the main message signal. However, this is not enough to confirm that in the case of the circular PoSK, synchronization-free, software-oriented architecture, the authentication signal is not vulnerable to the phenomenon of polarization switching time. Especially since this phenomenon is present in the hardware-oriented version of the same architecture, (see previous section "5.7.1. Experimentation with Hardware CPC") as it can be seen on Figure 47[b]. Therefore, further investigation is required before claiming that the phenomenon of polarization switching time will never appear in the software-oriented, synchronization-free, circular PoSK architecture.

The other excellent result in this experimental example is the very large gap, about 16 dBm (see Figure 52[b]), between the received power in a RHCP transmission and the received power in a LHCP transmission. This means the gap between logic levels 0 and 1 of the authentication code is very large. By way of comparison, remember we have a mere 1 dBm gap (see Figure 47[b]) in the hardware-oriented version of the same architecture, due to the poor performance at 2.45 GHz of the HMC647ALP6E [86] phase shifter. The higher the gap, the more resilient to noise is the authentication code.

Considering the good quality of the results obtained in this initial work, this software-defined approach for the Cryptographic Polarization Controller deserves further investigation in future work.

Despite these good results and an architecture that appears more flexible, more lightweight and easier to implement, this software-oriented architecture is nevertheless very likely the one that consumes more energy than the hardware-oriented architecture. A study of the power consumption of this software-oriented architecture was not carried out in this initial work, and is therefore part of future work. Another, but less critical, limitation of this software-oriented architecture will be the execution time of the cryptographic polarization selection algorithm, which could be longer than in the hardware-oriented architecture. This means a lower maximum frequency at which the execution of the cryptographic polarization selection algorithm can be repeated.

However, the integration of the software-oriented architecture would be easy during the design of the device, by adding a second antenna port and by adding the implementation of the Cryptographic Polarization Controller in the firmware. On the other hand, compared to the hardware-oriented architecture, it would be more difficult to integrate software-oriented architecture into already deployed IoT devices that do not originally foresee the use of PoSK-based message authentication system.

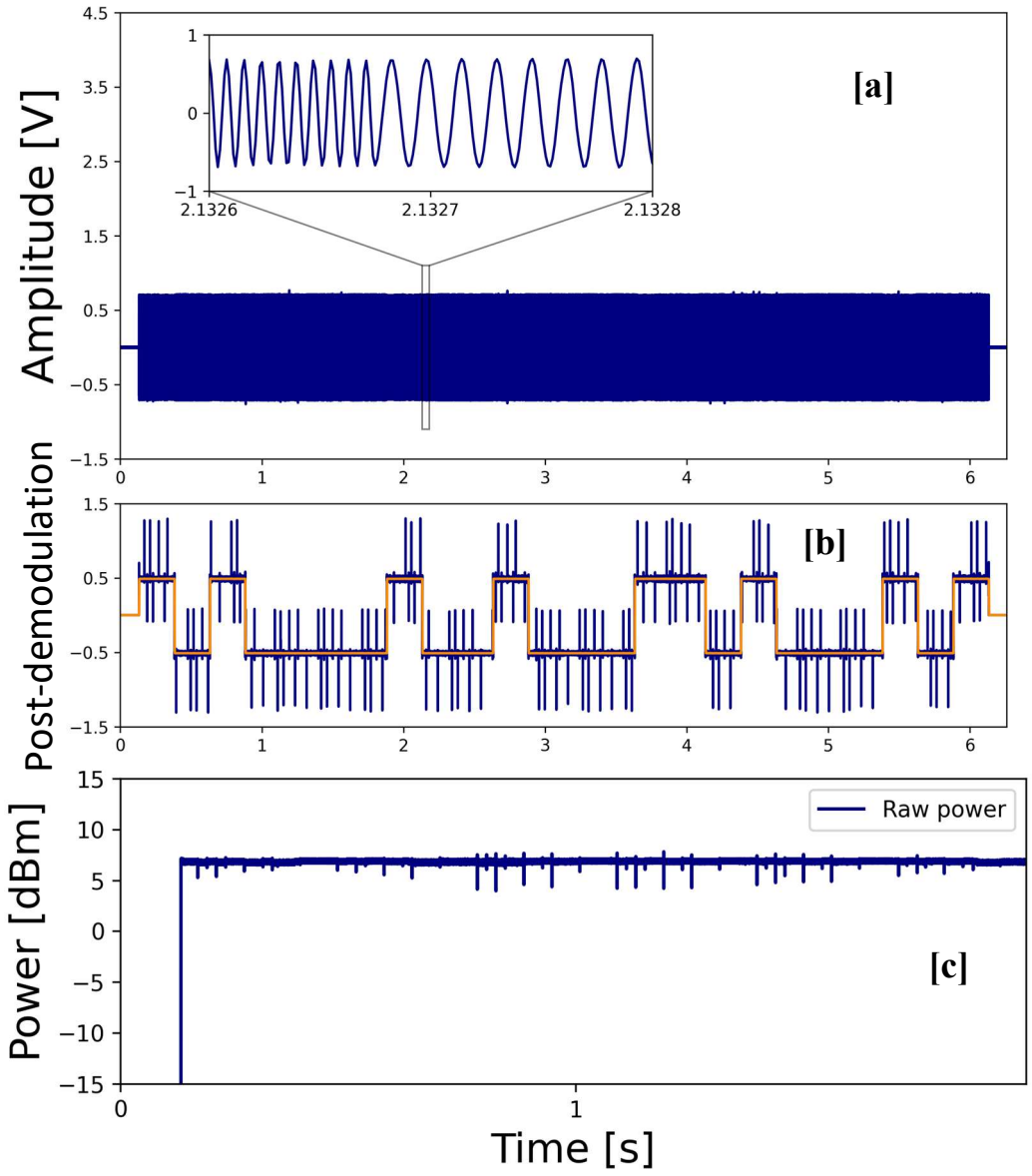


Figure 51. Circular PoSK, Synchronization-free software-oriented architecture, Signals received by the Security Gateway, Transmission with PoSK, Main message signal received on port P1: **[a]** - Received 64 kHz / 128 kHz FSK -modulated analog signal; **[b]** - The demodulation of the 64 kHz / 128 kHz FSK-modulated analog signal shown in [a]; **[c]** - The instantaneous power of the 64 kHz / 128 kHz FSK-modulated analog signal shown in [a].

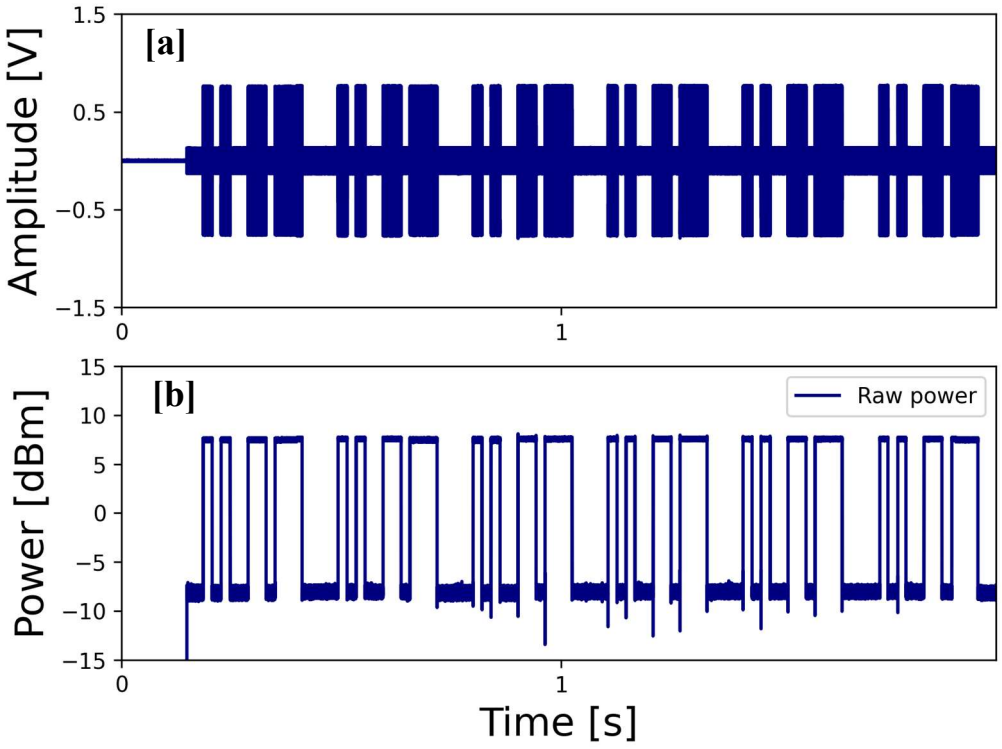


Figure 52. Circular PoSK, Synchronization-free software-oriented architecture, Signals received by the Security Gateway, Transmission with PoSK, Authentication code received on port P2 (see Figure 17(b)): **[a]** - Analog raw authentication signal; **[b]** - The instantaneous power of the analog authentication signal shown in [a].

4.7. Assessment of the robustness of the PoSK-based message authentication system

To assess the robustness of the PoSK-based message authentication system and highlight its impact on the main message signal, a square-wave signal generated by a waveform generator was used as PoSK signal on the Wireless Sensor, instead of Arduino UNO board running LFSR script. Figure 53 shows Wireless Sensor side experimental setup.

The number of polarization switches is at its maximum when a square-wave signal is used as PoSK signal; which makes this experiment the one in which PoSK should have the most significant impact on the main message signal. A realistic symbol rate, $F_S = 100\ kbaud$, was used for the main message, and $F_{PoSK}(kHz) = \{1, 10, 100, 500, 1000\}$ were used as different values of the Cryptographic Polarization Selector (the waveform generator) output rate.

This experiment was carried out on the two architectures with linear polarizations (Figure 16(a) and Figure 16(b)). To bypass the need for synchronization, which would be difficult at the F_{PoSK} frequencies mentioned above, in the case of the synchronization-required architecture (Figure 16(a)), the experiment was carried out with a circular polarization antenna on the Security Gateway in replacement of the Active Antenna System. This circular polarization antenna is shown on Figure 37, it is a simple 1-port LHCP antenna, not to be confused with the 2-port circular polarization antenna used for the circular PoSK experimentation.

The main message was the 49164-bits hexadecimal sequence 0xaaa001…fff555 and the main modulation was the FSK 500 kHz / 1 MHz.

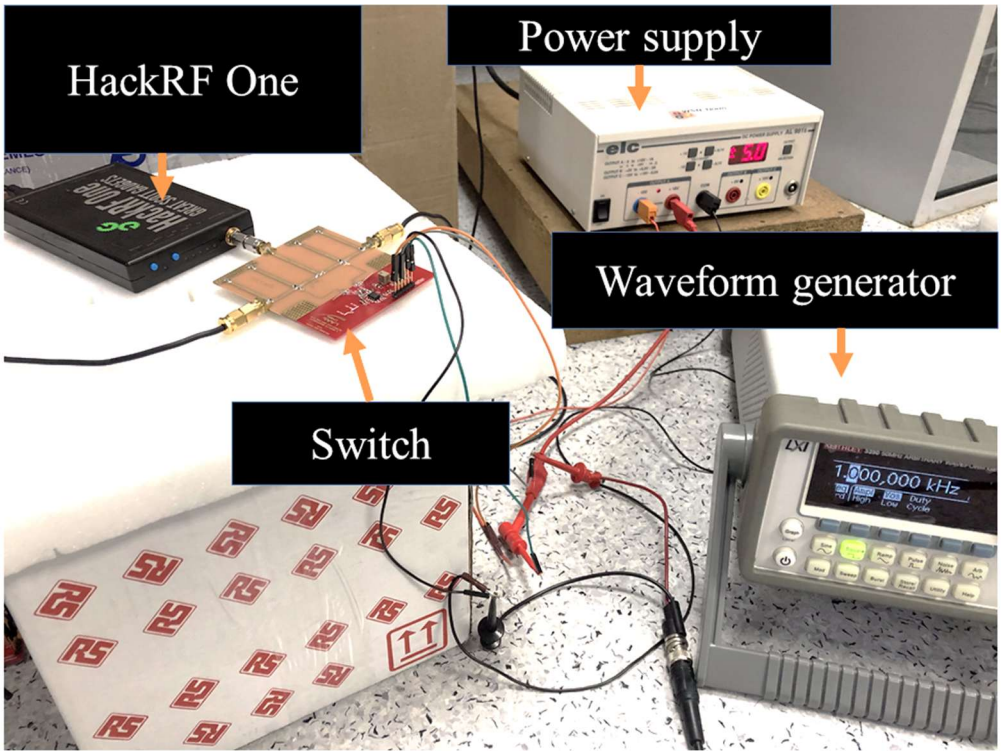


Figure 53. Linear PoSK experimentation; Assessment of the robustness of the PoSK-based message authentication system, Wireless Sensor experimental setup.

The two architectures produced the same experimental result. Figures 54-55 show the result from the synchronization-free architecture (Figure 16(b)) experimentation; this figure shows only the first 10 bits of the demodulation of the main message. On Figure 54, the impact of the PoSK can be seen as fluctuations in the demodulated signals; the higher the polarization switching rate, the higher its impact on the main message signal. For example, at $F_{PoSK} = 1$ *MHz* square-wave signal, with *100 kbps* of bit rate, there is 10 polarization switches in every bit. This phenomenon is the impact of polarization switching time.

However, compared with the previous experiments where an Arduino UNO board was used as Cryptographic Polarization Selector, we can see that in this experiment with a waveform generator as Cryptographic Polarization Selector, the level of the noise introduced by the PoSK is low. For example, we can see that, in Figure 34[b] and Figure 36[b], the absolute values of the amplitudes of the fluctuations introduced by the PoSK are much larger than those on Figure 54. This could be due to the waveform generator's switching time being shorter than the Arduino UNO board's GPIO switching time. This also means that the communication interruption caused by polarization switching is shorter with the waveform generator than with the Arduino UNO board. It also means that the switching time of our SPDT switch depends on its controller. It is important to note that with the Arduino UNO this switch is controlled using 0V/+5V, whereas with the waveform generator it is controlled using -5V/+5V. The latter levels are those recommended by its designer.

Increasing F_{PoSK} amplifies the impact of polarization switching time on the main message signal. Nevertheless, in this experiment, the main message can be recovered from the raw demodulated signals, shown on Figure 54, by a simple binary slicing through a thresholding, without even need for filtering. But, to reduce the risk of bit errors due to PoSK, a low-pass filter with a cut-off frequency at least equal to the main message symbols rate (F_S) can be used

to clean the signals before the thresholding and binary slicing. Figure 55 shows the signals on Figure 54 after a low-pass filter with 400 kHz of cut-off frequency was applied to them.

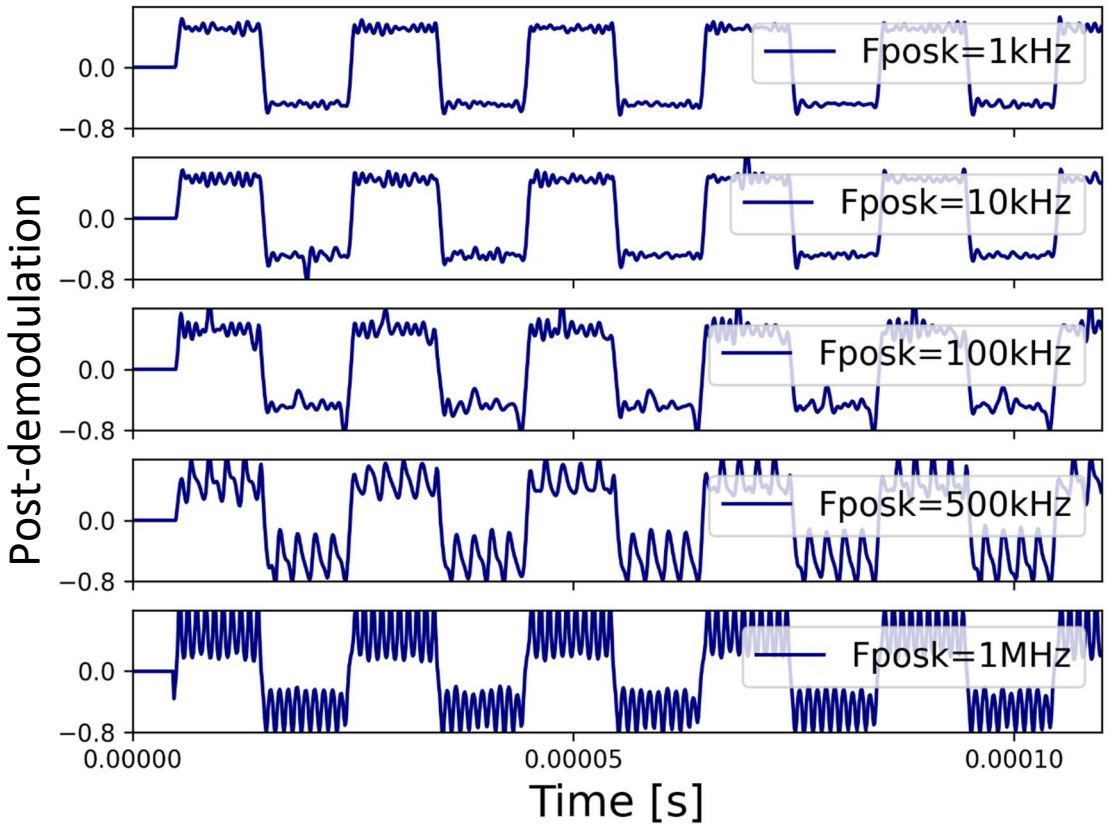


Figure 54. First 10 bits of the demodulation of the main message for different values of the polarization keying frequency (F_{PoSK}), The demodulations of the 500 kHz / 1 MHz FSK-modulated analog signals. The symbol rate is 100 kbaud. The bit rate is 100 kbps.

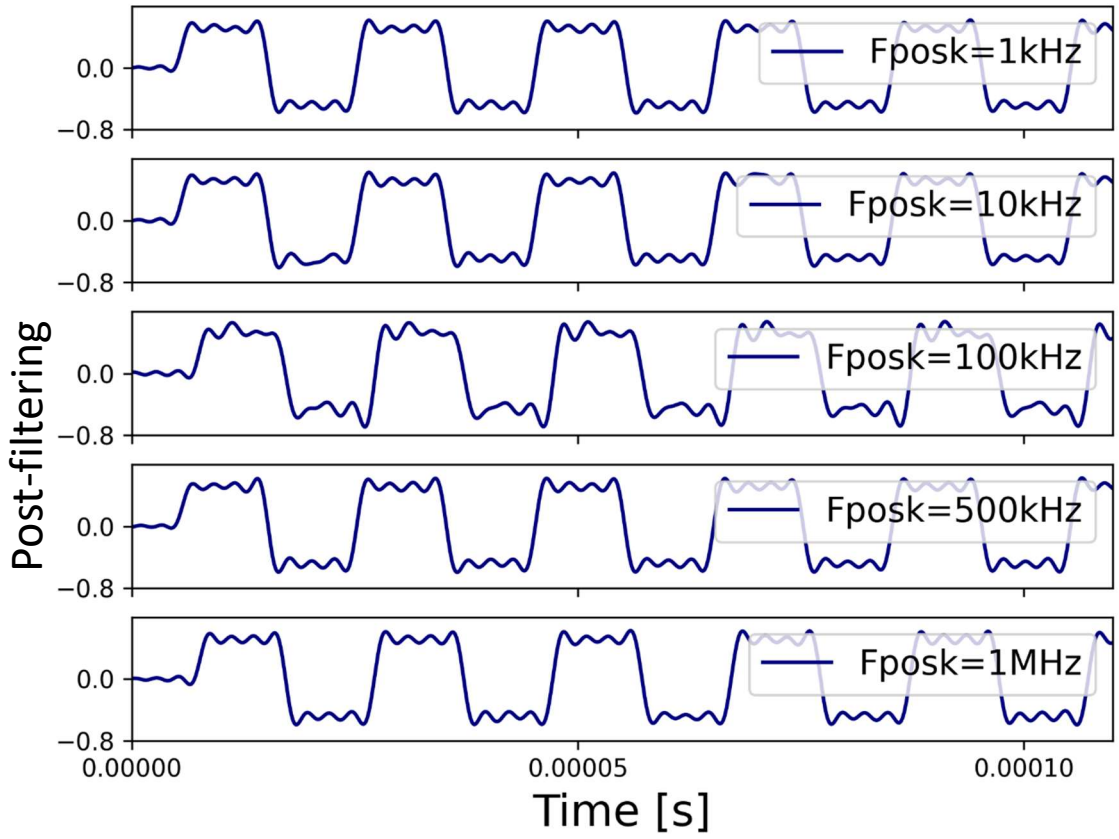


Figure 55. First 10 bits of the demodulation of the main message for different values of the polarization keying frequency (F_{PoSK}), Post-filtering version of the signals shown on Figure 54: to reduce the risk of bit errors due to PoSK, a low-pass filter with a cut-off frequency at least equal to the main message symbols rate (F_S) can be used to clean the signal before binary slicing, here a low-pass filter with 400 kHz of cut-off frequency was applied to them. The symbol rate is 100 kbaud. The bit rate is 100 kbps.

However, we will see later in the section "Impact of PoSK on communication quality: Bit Error Rate (BER)/Packet Error Rate (PER)" that this experiment is not sufficient to claim that the impact of PoSK on the main message signal would be so easy to overcome. Indeed, in this experiment we have a separation of 500 kHz between the two frequencies used in the modulation; the resulting FSK-modulated signal is robust against noise, including noise introduced by the PoSK. In reality, FSK deviations can be much weaker and more vulnerable to noise.

5. PoSK-based Message Authentication System Experimentation in real data transmission: Image, Video and File transmission

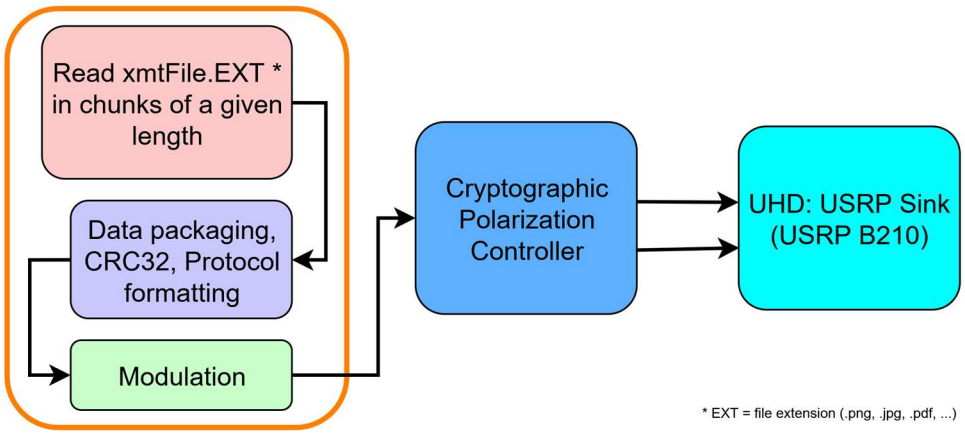
So far, the main message sent by the Wireless Sensor to the Security Gateway was a synthetic and noise-free signal. This message, generated by the URH software, contained a sequence of numbers, modulated with the desired parameters (modulation type, baseband frequency, number of samples per symbol, etc.). This message was then transmitted with PoSK by the Wireless Sensor using either URH or GNU Radio. One of these two software was also used by the Security Gateway to receive the main message and the authentication code signals. Then, signal processing chains on GNU Radio allowed to retrieve the main message and the authentication code from the received signals.

This section discusses the transmission with PoSK of real and meaningful data such as images, PDF (Portable Document Format) files, videos, and so on. As this manuscript was being finalized, we were able to carry out an initial experiment on the transmission with PoSK of such data. The experimental architecture is linear PoSK; synchronization-free, software-oriented architecture. Figure 56 shows the structure of the corresponding GNU Radio flowgraphs. On the Wireless Sensor side (Figure 56(a)), the blocks surrounded by the orange rectangle enable the file ***xmtFile.EXT*** to be read and prepared for transmission. On the Security Gateway side (Figure 56(b)), the blocks surrounded by the orange rectangle enable the processing and reconstruction of the received file named ***rcvFile.EXT***. If the transmission completes successfully from start to end, the contents of the files ***xmtFile.EXT*** and ***rcvFile.EXT*** will be identical. Note that the names of these files are just examples, and ***EXT*** should be replaced with the file extension (e.g., png, jpg, pdf). The parts surrounded by orange rectangles are adaptations to our requirements of the work in [95]. In previous experiments, the File Sink and File Source blocks were almost sufficient for these parts.

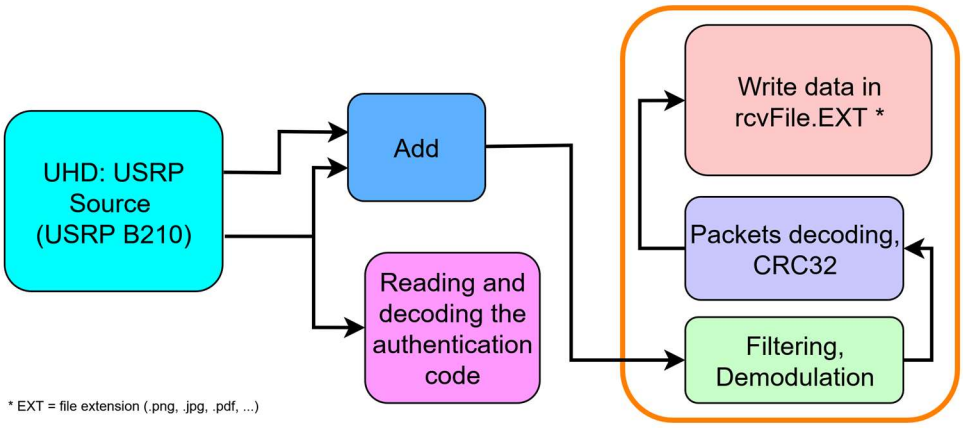
The Cryptographic Polarization Controller block remains the same as previously. It is a GNU Radio Embedded Python Block whose task, in this architecture, is to output the signal it receives on its input to one of its two outputs following the cryptographic polarization selection algorithm.

The technique used in reception to simultaneously receive the main message (the file ***xmtFile.EXT***) and the authentication code also remains the same. In this architecture, the main message is obtained by adding the powers received on the two ports (corresponding to the two polarizations H and V) of the USRP B210. The authentication code is obtained by listening on only one of the two ports.

In this initial experiment, only BPSK and 2-FSK have been experimented as the main message modulation for transmitting real data.



(a)



(b)

Figure 56. Linear PoSK; Synchronization-free software-oriented architecture; GNU Radio flowgraphs structure for real data transmission. **(a)** – Wireless Sensor; **(b)** – Security Gateway.

This experiment produced good results even for BPSK, despite the fact that BPSK could be one of the most sensitive modulations to the effects of PoSK, namely polarization switching time and undesired polarization sensitivity effects. However, it is important to wait until the receiver is synchronized with the transmitter before sending data. This may require the use of a preamble data which length depends on the convergence time of the receiver.

Several times, we successfully transmitted with PoSK images and PDF files without any loss. File sizes exceeding 100 kilobytes (KB) were successfully transmitted; which is far from negligible data size in the context of IoT. These results further reinforced our claims regarding the relevance of the PoSK-based message authentication system; this is a relevant authentication system.

In the next section, we will carry out further experiments with the aim of finding optimal values for certain parameters, at which the PoSK-based massage authentication system has the lowest impact on communication quality by affecting the bit error rate as least as possible. Among these parameters, one can mention the type of modulation used for the main message. For example, does the modulation of the main message being FSK or PSK change anything in terms of the impact of PoSK on communication quality? Another parameter for which optimal values must be found is the polarization keying frequency, F_{PoSK}. To do this experimentally, bit error rates at different F_{PoSK} values must be compared.

6. Impact of PoSK on the communication quality: Bit Error Rate (BER) / Packet Error Rate (PER)

So far, PoSK-based message authentication system has been the subject of massive experimentation over several months, in both anechoic chamber and regular environment, with a considerable volume of transmissions and various modulations for the main message (ASK 0 % / 100 %, ASK 25 % / 100 %, ASK 50 % / 100 %, FSK 64 kHz / 128 kHz, FSK 500 kHz / 1 MHz, PSK -90° / +90° and PSK 180° / 0°). More than 2,000,000 bits were transmitted, including more than 1,000,000 bits in the experiments of the section "Assessment of the robustness of the PoSK-

based message authentication system". Using GNU Radio, the transmission has been always successfully demodulated and the main message error-free recovered. Nevertheless, more in-depth investigation on the impact of the PoSK-based message authentication system on the quality of the communication was needed through experiments closer to real-world everyday transmissions.

6.1. Experimental Setup

To study the impact of PoSK on the communication quality, we conducted experiments that were more representative of everyday radiocommunications. Indeed, using the programs from the previous section, we conducted a massive campaign of real data transmissions with the aim of measuring the impact of PoSK on the PER.
To do this, we performed these transmissions for different modulations for the main message and for different values of the polarization keying frequency F_{PoSK}.
The choice of 2-FSK and BPSK allows to experiment with two modulations with quite different spectral profiles. F_{PoSK} values are chosen on an ongoing basis according to changes in communication quality, the experimental range being from 1 Hz to 5 MHz.

The experimental setup is shown on Figure 57.

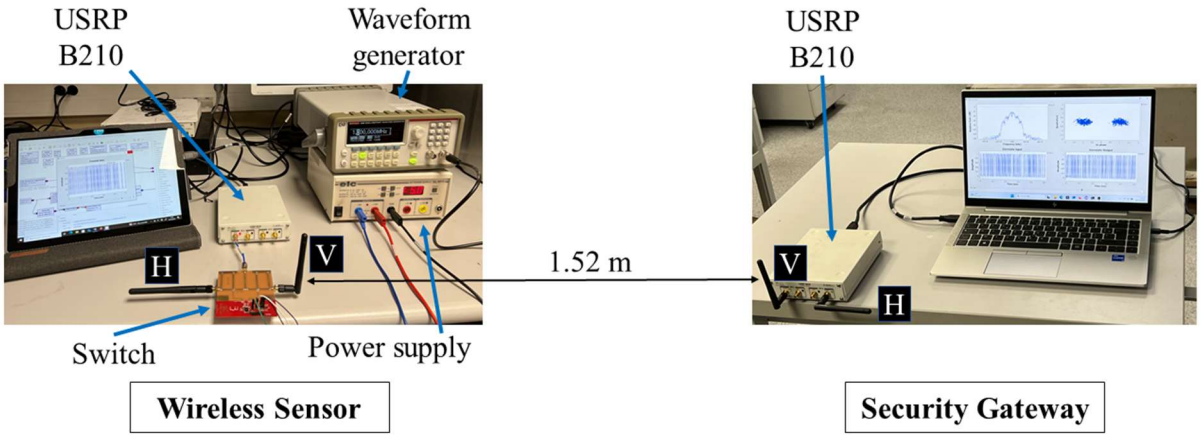


Figure 57. Experimental setup for studies of the impact of PoSK on the PER.

For these experiments, a synchronization-free hybrid architecture is used with a hardware-oriented wireless sensor and a security gateway that replaces the network of splitter devices with GNU Radio addition blocks. A hardware-oriented approach on the wireless sensor side allows higher PoSK frequencies of up to 5 MHz to be achieved using the switch and waveform generator. For the security gateway, a software-oriented approach was chosen for simplicity's sake. Indeed, the type of security gateway architecture is irrelevant in the experimental setup shown on Figure 57.

To measure PER, the wireless sensor transmits an 86 KB image to the security gateway using a given modulation (2-FSK or BPSK), at a carrier frequency of 2.45 GHz and a given polarization keying frequency F_{PoSK} (between 1 Hz and 5 MHz). Depending on the percentage of bits lost, we readjust the F_{PoSK} frequency and retransmit the image. We repeat this process over and over again to see if at some point the image will be received in its entirety.

And indeed, we found that above a certain F_{PoSK} value, the image is always received in its entirety, which corresponds to a PER of 0%; this also means that the BER is so low that all packets are recovered correctly. In other words, above a certain value for the polarization keying frequency, PoSK no longer has any impact on the transmission. The search for this frequency is carried out in a kind of dichotomous manner: starting from DC, we gradually adjust the polarization keying frequency until no more packets are received. We then search for the frequency at which packets proper reception gradually resumes again and we continue adjusting the polarization keying frequency until the PER definitively drops to 0%.

The payload size is more than 50 bytes per packet, which means that the 86 KB file is transmitted in over 1.000 packets. In case of error, the packet is rejected and is not retransmitted because retransmission of the packet would bias the PER measurement. A 32-bit Cyclic Redundancy Check (CRC32) is used in this experiment.

The very interesting results of these experiments are presented in more detail in the following section, "Results".

6.2. Results

6.2.1. BPSK

In BPSK, two phases are used to modulate the 86 KB image transmitted by the wireless sensor. The complex signal resulting from baseband modulation is then up-converted to the 2.45 GHz carrier frequency for transmission.
The experiment was repeated for three different transmission speeds, i.e., three different symbol rates Rs or baud rates. Remember that the symbol rate Rs is also the bit rate Rb in a two-state modulation such as BPSK. The three experimental values of Rs are 31.25 kbaud, 62.5 kbaud, and 125 kbaud, that is, 31.25 kbps, 62.5 kbps, and 125 kbps.

The evolution of the PER as a function of the polarization keying frequency F_{PoSK} is shown on Figure 58 for the three bit rates.

As can be seen on Figure 58, these results are very interesting. In the three cases of symbol rates, which are also bit rates, the PER definitely drops to 0% (the image is received in its entirety) as soon as the polarization keying frequency slightly exceeds the bit rate. Indeed, the three plots drop to zero at frequencies 35 kHz for symbol rate Rs=31.25 kbaud, 65 kHz for symbol rate Rs=62.5 kbaud, and 130k for symbol rate Rs=125 kbaud, respectively.

Above these specific frequencies, PoSK no longer has any impact on the transmission, all packets are received and decoded correctly; we will call this band the "authorized F_{PoSK} band". Whereas, for frequencies below these specific frequencies, a band we will refer to as the "prohibited F_{PoSK} band", PoSK significantly degrades the quality of the communication, reaching a PER of 100% for certain F_{PoSK} frequencies such as 5 kHz for example (see Figure 58). This means that the receiver was able to recover 0 packets out of more than 1.000 transmitted packets; this also means that the BER is so high that no packet is recovered correctly. This gives an idea of the degradation in communication quality caused by PoSK when the polarization keying frequency is chosen in this "prohibited F_{PoSK} band".

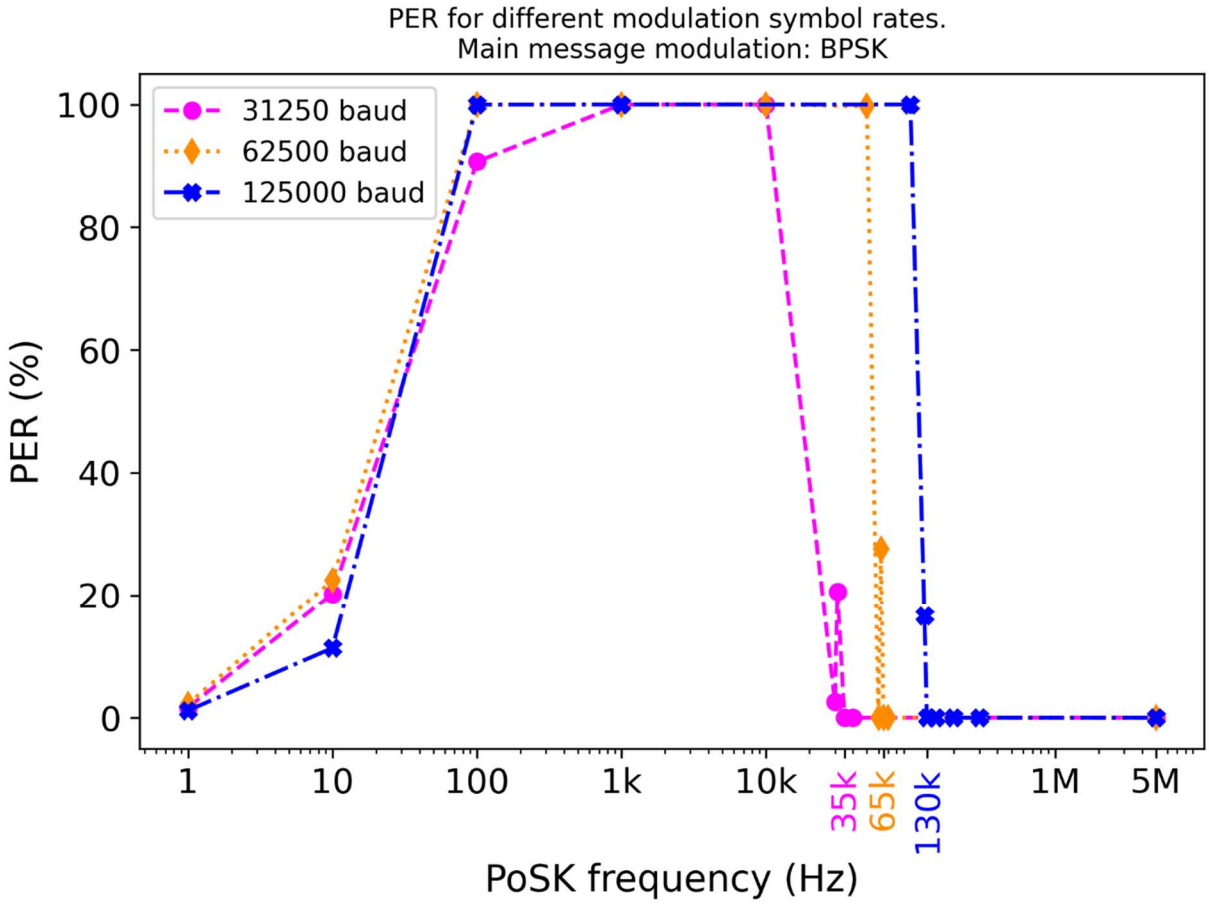
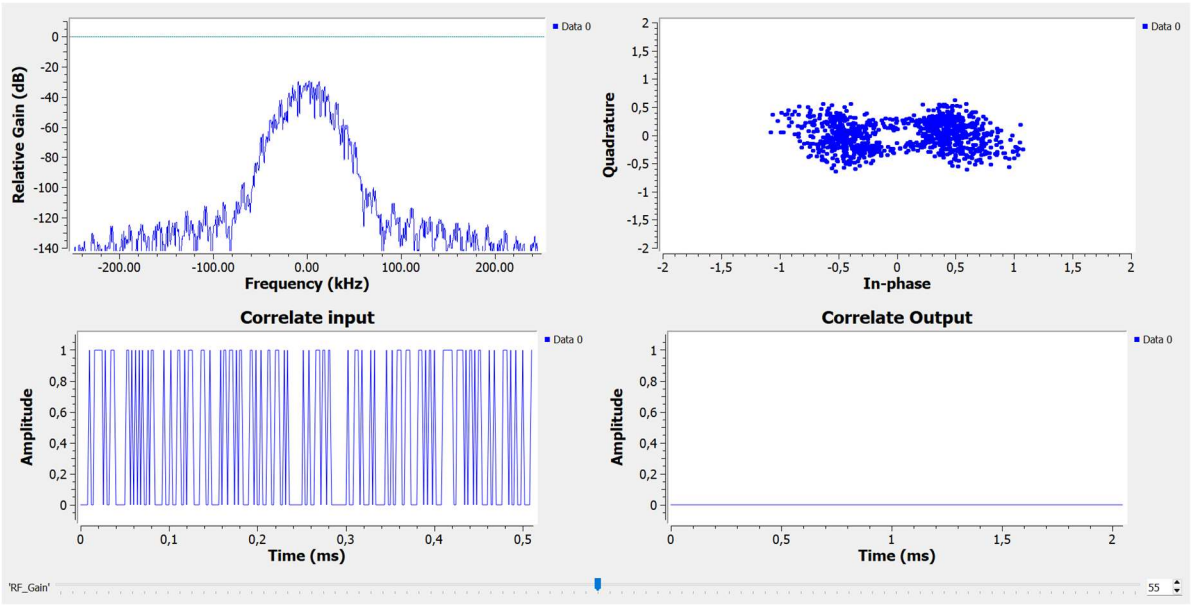


Figure 58. The PER as a function of polarization keying frequency, in BPSK-modulated message transmission, experimented for three different BPSK modulation symbol rates. The PER falls to 0% definitely at the frequencies highlighted in color.

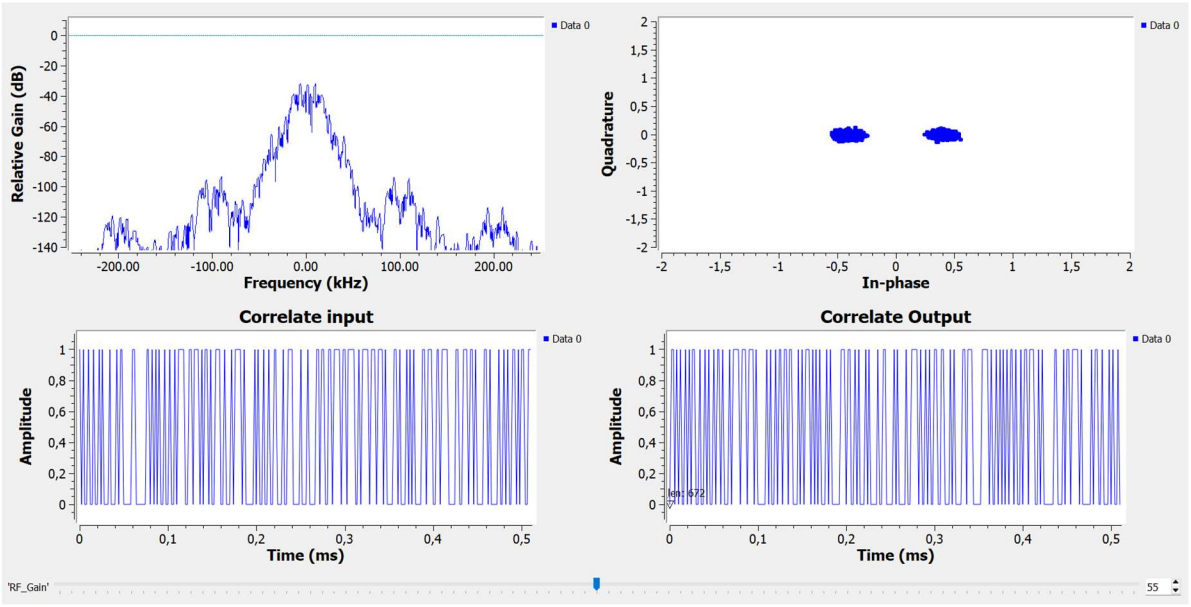
The explanation for this phenomenon is as follows: when the polarization keying frequency is higher than the spectral band occupied by the main message, the noise introduced by the PoSK no longer constitutes an obstacle to proper signal reception, as this noise is rejected by the filter used by the receiver.

The spectral band of the BPSK modulated signal ranges from -Rs to Rs, that is, a bandwidth of 2Rs. A low-pass filter at the receiver with a cutoff frequency equal to or slightly higher than Rs thus rejects the noise introduced by the PoSK for polarization keying frequencies, F_{PoSK}, higher than Rs. The narrower the filter transition width, the closer F_{PoSK} value can be to Rs without affecting the PER, but this comes at the cost of increased computational load. In an IoT application, it would be preferable to avoid filter transition widths that are narrower than needed. In this experiment, we used a transition width of 40 kHz. The sampling frequency, 2 MHz, is decimated by 4, after filtering, to bring it down to 500 kHz.

Through constellation diagram, it turns out that BPSK is exactly one of the best modulations for visualizing the impact of PoSK on the quality of the transmission. As an example, Figure 59 shows the BPSK constellations for two different values of F_{PoSK}. In the case of Figure 59(a), F_{PoSK} is chosen from the "prohibited F_{PoSK} band", whereas in the case of Figure 59(b), it is chosen from the "authorized F_{PoSK} band". In this example, the symbol rate is 31.25 kbaud.



(a)



(b)

Figure 59. BPSK-modulated message reception at 31.25 kbaud symbol rate, when it was transmitted with **(a)**: a prohibited polarization keying frequency (10 kHz) related to the symbol rate and **(b)**: an authorized polarization keying frequency (100 kHz) related to the symbol rate.

With a symbol rate of 31.25 kbaud, the 10 kHz frequency for the F_{PoSK} is not appropriate, as this frequency lies within the spectral band of the message, the "prohibited F_{PoSK} band". The low-pass filter used by the receiver therefore allows the noise introduced by the PoSK to pass through, preventing the receiver from synchronizing on the two phases of the BPSK, as shown in the constellation on Figure 59(a). As a result, the receiver is unable to recover the packets. The packet rejection rate is 100%, as also shown on Figure 58. Of the 86 KB transmitted, 0 KB is recovered; which also means that the BER is so high that no packet is recovered correctly. Whereas at a frequency of 100 kHz for F_{PoSK}, which is a value from the "authorized F_{PoSK} band",

i.e., outside the spectral band of the message, the noise introduced by PoSK no longer prevents proper signal reception, as this noise is rejected by the low-pass filter used by the receiver. The latter synchronizes very quickly on the two phases of BPSK, as shown by the constellation on Figure 59(b). The 86 KB image is received correctly in its entirety. The packet rejection rate is 0%, as also shown on Figure 58. The PoSK no longer has any impact on the PER, i.e., the PoSK no longer has any impact on the transmission.

6.2.2. Improved PER using Power Squelch

Power Squelch is a common technique in radiocommunications. Basically, it means setting a minimum power level for the receiver to consider and then ignoring any power levels below that minimum threshold. It is used, for example, to mute a radio receiver while waiting for a transmission. But its advantage that interests us in these experiments is the fact that it reduces the noise floor. It turns out that this reduction of the noise floor reduces the impact of PoSK on communication quality and improves PER.

We performed exactly the same experiment as in the previous section, but this time we added power squelch just after the low-pass filter on the receiver. The evolution of the PER as a function of the polarization keying frequency F_{PoSK} is shown on Figure 60 for the three baud rates.

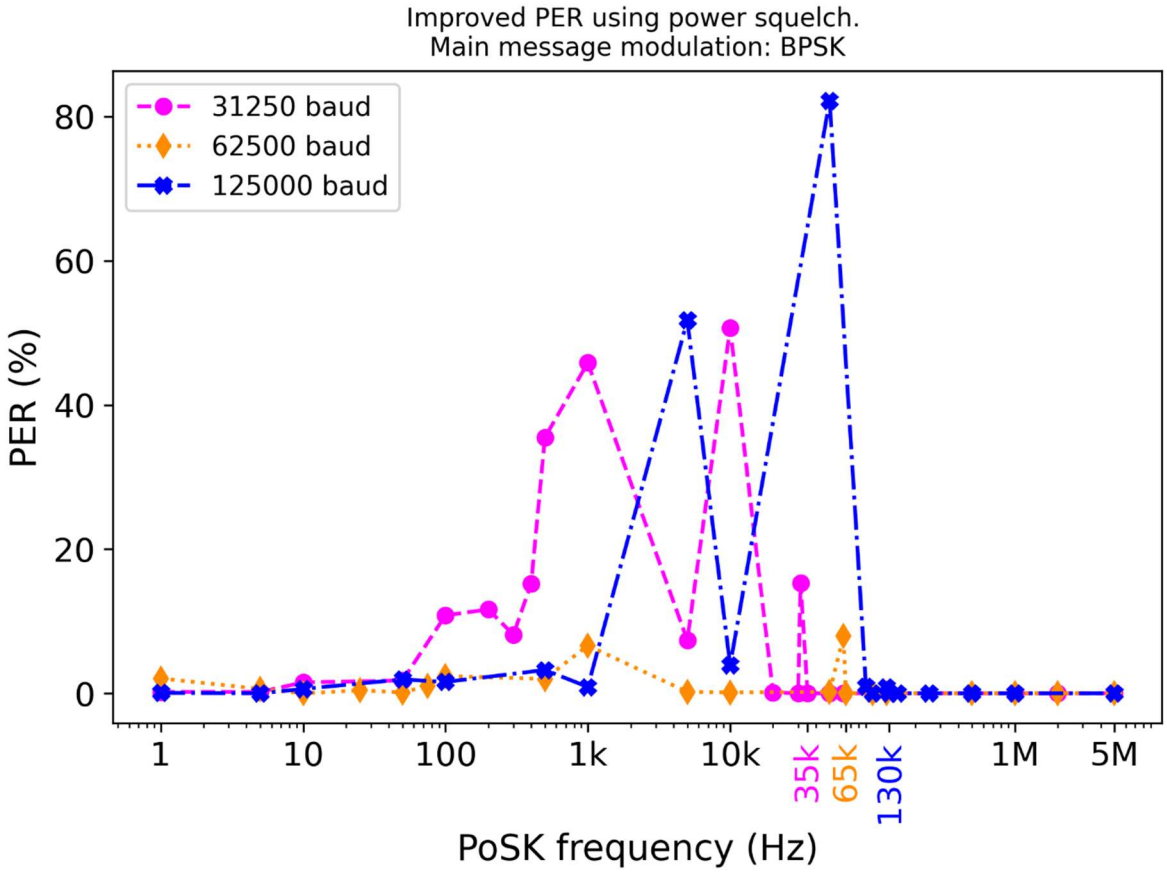


Figure 60. The PER as a function of polarization keying frequency, in BPSK-modulated message transmission, experimented for three different BPSK modulation symbol rates. Power Squelch was added to improve the PER. The PER falls to 0% definitely at the frequencies highlighted in color.

Figure 60 shows the results of the same experiment as Figure 58, but performed with power squelch added after the low-pass filter of the receiver.

This improvement in PER can be particularly beneficial when F_{PoSK} is chosen from the "Prohibited F_{PoSK} band"; whereas, it is not of much interest when F_{PoSK} is chosen from the "Authorized F_{PoSK} band". However, although better than the results without power squelch (see Figure 58), the evolution of the PER remains very unstable and random in the "Prohibited F_{PoSK} band" when power squelch is added. In any case, it is preferred not to choose F_{PoSK} in the "Prohibited F_{PoSK} band".

In this experiment, GNU Radio's "Power Squelch" block was used to implement power squelch with a threshold (Threshold parameter) of -40 dB. The choice of Threshold value matters: too high values will alter the useful spectral band and increase the PER, while too low values will be of no importance and will have no impact on the quality of the communication.

6.2.3. 2-ASK

Since the spectral profile of 2-ASK modulation is similar to that of BPSK, one can expect the same result in terms of the impact of PoSK on the PER, in the case of 2-ASK modulation. Indeed, the spectral bandwidth of the 2-ASK modulated signal is also 2Rs; the difference with BPSK is that the 2-ASK spectrum is symmetrical around the baseband carrier (two symmetric sidebands around the carrier), due to the fact that the signal is real, not complex as in BPSK. So, for 2-ASK, in addition to filtering, a frequency translation is necessary to bring the signal back around 0 Hz.

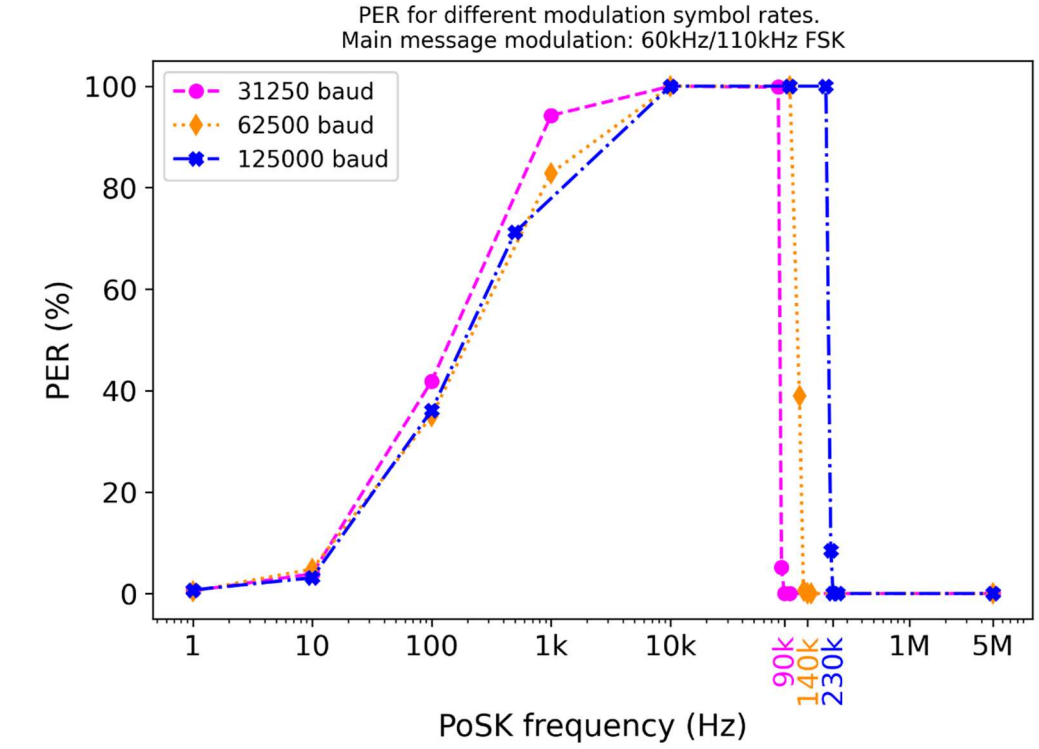
6.2.4. 2-FSK

When it comes to 2-FSK, its spectral profile is quite different from those of BPSK and 2-ASK. In 2-FSK, two frequencies, which we will call f_1 and f_2, are used to modulate the signal before being up-converted to the 2.45 GHz carrier for transmission. The FSK deviation Δf refers to half the absolute value of the frequency separation (the difference between f_1 and f_2). Two experiments were carried out for the FSK.

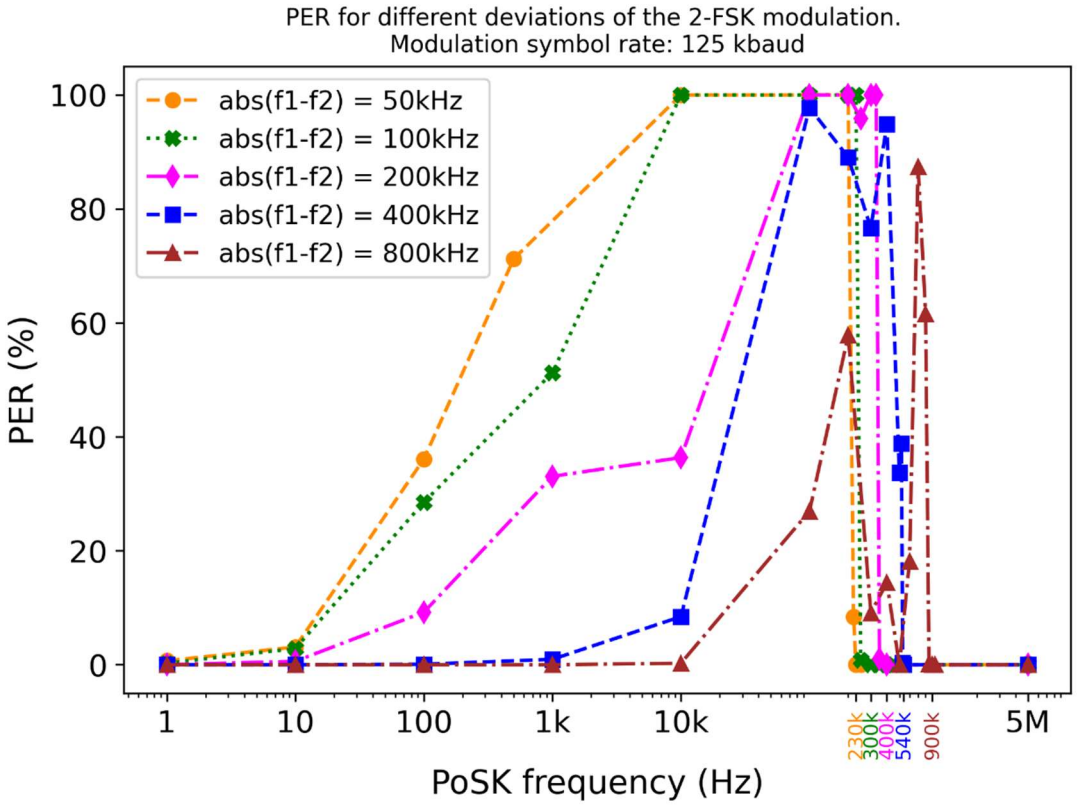
- With a unique frequency deviation $\Delta f = 25\ kHz$, i.e., a frequency separation of 50 kHz, the first experiment measures the PER at three different values of the symbol rate Rs, as in the case of BPSK. In fact, the same values 31.25 kbaud, 62.5 kbaud, and 125 kbaud were used. Remember that the symbol rate Rs is also the bit rate Rb in a two-state modulation such as 2-FSK.
- The second experiment uses a unique symbol rate Rs = 125 kbaud, and measures the PER at five different values of the frequency deviation Δf.

Figure 61 shows the evolution of the PER as a function of the polarization keying frequency F_{PoSK} for these two experiments.

In this experiment, GNU Radio's "Simple Squelch", block was used to implement power squelch with a threshold (Threshold parameter) of -50 dB. This block is similar to the "Power Squelch" block with a few differences. However, the threshold of -50 dB is a little too low, making this block less helpful and without significant impact on communication quality. The PER evolution on Figure 61(a) is therefore similar to that on Figure 58, the case of BPSK without power squelch.



(a)



(b)

Figure 61. The PER as a function of polarization keying frequency, in 2-FSK-modulated message transmission, experimented **(a)**: for three different FSK modulation symbol rates, with a unique FSK deviation (50 kHz); and **(b)**: for five different FSK deviations, with a unique symbol rate (125 kbaud). The PER falls to 0% definitely at the frequencies highlighted in color.

As might be expected after seeing what happened in the case of BPSK, in this case too one can observe two frequency bands in the PER evolution. The first band ranges from DC to the frequency at which the PER definitely falls to 0%: this is the band we have called the "Prohibited F_{PoSK} band"; and the second band is the band of frequencies above the frequency at which the PER definitely drops to 0%: this is the band we have called the "Authorized F_{PoSK} band"; as can be seen on Figure 61(a).

The explanation is also more or less the same as in the case of BPSK. Indeed, it will be again a question of the spectral band of the message, and the underlying logic remains the same. When the polarization keying frequency, F_{PoSK}, is higher than the spectral band occupied by the main message, the noise introduced by the PoSK is no longer constitutes an obstacle to proper signal reception, as this noise is rejected by the filter used by the receiver.

So, we need to know the spectral profile of 2-FSK modulation. In these experiments, we estimate the spectral bandwidth of the 2-FSK modulation using Carson's rule, given by the following relation:

$$BW = 2(\Delta f + f_m) = 2(\frac{h}{2} + 1)f_m$$

Where:

BW is the spectral bandwidth containing 98% of the power of the frequency-modulated signal, according to Carson.

Δf is the peak frequency deviation, that is, half the absolute value of f_1-f_2 for 2-FSK modulation.

f_m is the highest frequency in the modulating signal, that is, Rb=Rs for 2-FSK modulation.

$h = \frac{2\Delta f}{f_m}$ is the modulation index.

$h < 1$ is referred to as Narrowband Frequency Shift Keying (NBFSK). $h > 1$ is referred to as Wideband Frequency Shift Keying (WBFSK). $h = 0.5$ is referred to as Minimum Shift Keying (MSK). The larger Δf is, the larger h is, the larger the spectral bandwidth of the FSK-modulated signal is, but the easier the detection is for the receiver. Conversely, the smaller Δf is, the smaller h is, the smaller the spectral bandwidth of the FSK-modulated signal is, but the more difficult the detection is for the receiver. In 2-FSK, Δf corresponding to $h = 0.5$ is the smallest frequency deviation for which f_1 and f_2 can be orthogonal, i.e., no interference between symbols.

Thus, for our first experiment using a single deviation of 25 kHz, i.e., *abs(f_1-f_2)=50 kHz*, the spectral bandwidths of the message are estimated to be:

For Rs=31250 baud: $BW = 2(25\ kHz + 31.25\ kHz) = \ 112.5\ kHz$. That is, $h = 1.6$

For Rs=62500 baud: $BW = 2(25\ kHz + 62.5\ kHz) = \ 175\ kHz$. That is, $h = 0.8$

For Rs=125000 baud: $BW = 2(25\ kHz + 125\ kHz) = \ 300\ kHz$. That is, $h = 0.4$

In this case, we can see on Figure 61(a) that the PER drops definitely to 0% even before F_{PoSK} becomes greater than the spectral bandwidth of the message estimated using Carson's rule. For example, for Rs=31250 baud, BW=112.5 kHz, the PER drops definitely to 0% as soon as at 90 kHz F_{PoSK}. This could be due to the fact that Carson's rule is more flexible regarding the width

of the signal's spectral band; the width estimated by Carson's rule may be greater than the minimum band actually need for the message to be demodulated correctly.

But we can clearly see that the logic remains the same as in the case of BPSK: we have a "Prohibited F_{PoSK} band" in which PoSK can significantly degrade communication quality if F_{PoSK} is taken in this band. Then we have a second band, which we have called the "Authorized F_{PoSK} band" in which the PoSK no longer affects the quality of communication if F_{PoSK} is taken in this second band. This second band begins at a frequency at which the PER definitively falls to 0%; these are frequencies highlighted in color on Figure 58, Figure 60, and Figure 61. The explanation is that when the polarization keying frequency, F_{PoSK}, is outside the actual spectral band of the message, the noise introduced by the PoSK can be rejected at the reception using a filter.

In the case of 2-FSK, before filtering, a frequency translation is necessary to bring the signal back to around 0 Hz. Fortunately, a single GNU Radio block can perform these two operations and even more: the "Frequency Xlating FIR Filter" block. This block performs the following operations: frequency translation, filtering, and decimation. In this case, its "Center Frequency" parameter is set to the center of the band, i.e., $(f_1+f_2)/2$. It is fed with a low-pass filter ("Taps" parameter) with a cutoff frequency of $BW/2$ and a transition width of 20 kHz, i.e., 1% of the sampling frequency, 2 MHz. The "Decimation" parameter takes an integer whose value depends on the sampling frequencies, which in turn depend on the spectral bandwidth of the message, which in turn depends on the FSK deviation and the symbol rate. For example, "Decimation" takes the value 4 to go from 2 MHz, the sampling frequency of the receiver (USRP B210), to 500 kHz for the rest of the processing chain.

For Rs=125000 baud, we have a modulation index of 0.4, which is lower than the minimum 0.5 required to guarantee orthogonality between the two frequencies and avoid interference between symbols. Despite this unfavorable condition, PoSK had no particular impact on the communication quality related to this situation.

To confirm our findings on the choice of polarization keying frequency with regard to the spectral band of the message, we conducted the second experiment whose results are shown on Figure 61(b). This figure shows the evolution of the PER as a function of the polarization keying frequency for different values of FSK deviation when a single symbol rate, 125 kbaud, is used.

Thus, for this second experiment, the spectral bandwidths of the message are estimated to be:

For *abs(f1-f2)=50 kHz*: $BW = 2(25\ kHz + 125\ kHz) = 300\ kHz$. That is, $h = 0.4$

For *abs(f1-f2)=100 kHz*: $BW = 2(50\ kHz + 125\ kHz) = 350\ kHz$. That is, $h = 0.8$

For *abs(f1-f2)=200 kHz*: $BW = 2(100\ kHz + 125\ kHz) = 450\ kHz$. That is, $h = 1.6$

For *abs(f1-f2)=400 kHz*: $BW = 2(200\ kHz + 125\ kHz) = 650\ kHz$. That is, $h = 3.2$

For *abs(f1-f2)=800 kHz*: $BW = 2(400\ kHz + 125\ kHz) = 1050\ kHz$. That is, $h = 6.4$

As can be seen on Figure 61(b), the same trend is confirmed once again. The PER falls definitively to 0% when the polarization keying frequency is greater than the actual spectral band of the message. As mentioned previously, the spectral width estimated by Carson's rule may be greater than the minimum band actually needed for the message to be demodulated

correctly. This is why the PER definitively falls to 0% earlier, while F_{PoSK} is still even in the "Prohibited F_{PoSK} band".

In the first case, where *abs(f1-f2) = 50 kHz*, once again the modulation index is lower than the minimum 0.5 required to guarantee orthogonality between the two frequencies and avoid interference between symbols. Despite this unfavorable condition, once again PoSK had no particular impact on the communication quality related to this situation.

Another phenomenon that can be observed on Figure 61(b) is that the wider the message bandwidth, the more robust the transmission is against noise introduced by the PoSK. For example, in the last case where *abs(f1-f2) = 800 kHz*, one can see that the average PER is much better than in the other cases. This is not a phenomenon specific to PoSK, but rather to FSK modulation in general: the higher the FSK deviation relative to the symbol rate, the more robust the transmission is against noise. However, such FSK deviation is not reasonable in practice, as spectrum is a precious resource.

This case, where *abs(f1-f2) = 800 kHz*, also shows why we were able to demodulate, decode, and error-free recover the message in the experiments conducted to assess the robustness of PoSK, presented in the section "Assessment of the robustness of the PoSK-based message authentication system". In the experiments carried out in the said section, we used FSK 500 kHz / 1 MHz, i.e., a frequency separation, *abs* (f_1-f_2), of 500 kHz. The result is a transmission that is robust to noise. Therefore, these experiments would not be sufficient to understand the impact of PoSK on communication quality as a function of polarization keying frequency. This new study on the impact of PoSK on communication quality was therefore necessary.

6.3. Criteria for choosing polarization keying frequency

The results of this study on the impact of PoSK on the quality of the communication lead us to some very interesting conclusions.
It could be that the impact of PoSK on communication quality depends entirely on the choice of polarization keying frequency, F_{PoSK}, in relation to the spectral occupation of the signal of the message to be transmitted.

PoSK degrades communication quality when the polarization keying frequency is chosen within the actual spectral band of the message, because the filter used by the receiver then allows the noise introduced by PoSK into the message signal to pass through along with this signal. So, to choose F_{PoSK} in this band, which we have named the "Prohibited F_{PoSK} band", it is necessary to find solutions to remove the noise introduced by PoSK without affecting the integrity of the message signal or altering it so much that error-free recovery of the message becomes impossible. The sources of noise related to PoSK that we have identified in this work being the impacts of polarization switching time and Undesired Polarization Sensitivity, the challenge is to find solutions to prevent or correct the effects of these two undesired phenomena, in order to be able to use an F_{PoSK} belonging to the message's spectral band, the "Prohibited F_{PoSK} band".
On the other hand, the spectral band in which one would naturally think of choosing the polarization keying frequency is the band outside the actual spectral band of the message; this is the band we call the "Authorized F_{PoSK} band". Indeed, when F_{PoSK} is chosen outside the actual spectral band of the message, PoSK no longer has any impact on communication quality, as the noise introduced by PoSK can be easily rejected by the filter used by the receiver. In addition

to the cutoff frequency, another important parameter of this filter is its transition width. The lower the value of this parameter, the closer the minimum acceptable F_{PoSK} (without affecting communication quality) can be to the upper limit of the message's actual spectral band. However, a low transition width results in an increase in the filter's computational load.

Another very exciting conclusion from these results is that the impact of PoSK on communication quality therefore depends only indirectly on the type of modulation of the main message and on the bit rate.

In these experiments, the F_{PoSK} signal is a square wave generated by the waveform generator. In a cryptographic context, the dynamics of the cryptographic output should be high enough to remain outside the actual spectral band of the main message.

7. Energy Consumption

Energy consumption is more of a concern on the Wireless Sensor than on the Security Gateway, since the latter can be a custom-made non-energy-constrained device. The energy consumption added by the PoSK mechanism is the consumption of the Active Antenna System's Cryptographic Polarization Controller block (see Figure 15). Ultimately, this block will be implemented on an ASIC (Application-Specific Integrated Circuit) or on a FPGA (Field Programmable Gate Arrays).

In the Cryptographic Polarization Controller block, the two sub-blocks most in demand, and therefore most consumptive, are:

(i) – ***Cryptographic Polarization Selector block (see Figure 16 and Figure 17):*** the consumption of this block depends mainly on the computational workload of the cryptographic polarization selection algorithm. The higher the computational workload of the used algorithm, the higher the power consumption will be. Therefore, it is important to implement this block using low-energy ASIC or FPGA technologies. In this experimental work, this block was experimented with an Arduino UNO board running the cryptographic polarization selection algorithm, a lightweight LFSR whose architecture is shown on Figure 33. The main reason for this choice is how quickly one can develop in the Arduino environment.

(ii) – ***Router block:*** In the 2-PoSK architectures with linear polarizations, the switching from one polarization to the other is carried out by the "Router" block (see Figure 16). As previously mentioned in section "5.2. Experimentation tools and general parameters", this block was experimented using a custom-made SPDT switch, an image of which can be seen on Figure 23. In architectures with circular polarizations, the "Router" block is not an SPDT switch, but made up of a Splitter and two identical Phase Shifters (see Figure 17). The latter case was experimented using the evaluation board EV1HMC647ALP6 of the Analog Devices HMC647ALP6E [86] phase shifter. The power consumptions of these two components, the SPDT switch and the EV1HMC647ALP6, are shown on Figure 62.

These measurements on Figure 62 were carried out with the help of the DTJ1524D05 [94], an isolated chassis mount DC/DC converter. This component is used between the main supply and the load, which is the device whose consumption is to be measured, that is, the SPDT switch or the phase shifter. The main supply device powers the DTJ1524D05 with 0V / 24V, i.e., the DTJ1524D05 input voltage. Then, the DTJ1524D05 outputs -5V / +5V to power the load (the SPDT switch or the EV1HMC647ALP6). The device used as main supply is the N6705B DC Power Analyzer, a device that also provides measures recording functions.

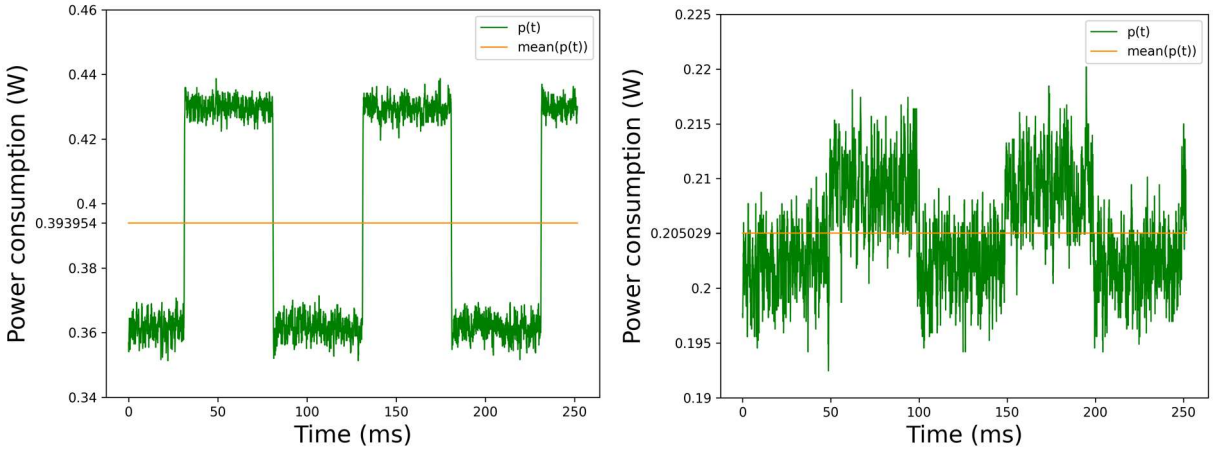


Figure 62. (a) – Power consumption of the custom-made SPDT switch shown on Figure 23; **(b)** – Power consumption of the evaluation board EV1HMC647ALP6 shown on Figure 25.

Thus, the power consumption, in Watts (W), of the load is given by the relation (28).

$$P(t) = U \,.\, I(t) \,.\, E_{ff} \qquad (28)$$

Where U is the supply voltage (24V in this case), $I(t)$ is the instantaneous supply current and E_{ff} is the efficiency of the DC/DC converter (84% according to its datasheet).

Figure 62(a) shows the SPDT switch's power consumption, which is around 0.36 W when its input is connected to one of its outputs and around 0.43 W when its input is connected to its other output. Its average power consumption is about 0.39 W, or 390 mW.

When it comes to the power consumption, of the evaluation board EV1HMC647ALP6, shown on Figure 62(b), we can see that this component consumes almost half less power than the SPDT switch. In these measurements, the component was making phase shifts of 0° and 180°, this is why Figure 62(b) shows a two-level pattern. The average power consumption of this component is about 0.205 W, or 205 mW.

These measurements may not be very accurate, as the power calculation (see relation (28)) used the typical (non-experimental) efficiency of the DTJ1524D05, 84%, as provided by its manufacturer. Nevertheless, these power consumptions are relatively high for energy-constrained IoT devices. Therefore, energy consumption optimization is part of future work. These power consumptions can be further reduced through the design, the choice of components (e.g., low-energy SPDT switch), and the miniaturization and integration of the Active Antenna System's Cryptographic Polarization Controller block (see Figure 15), in a real-world application. An implementation on FPGA or on ASIC of this block will allow more accurate power consumption measurements.

8. Discussion

8.1. Linear vs Circular Polarizations Shift Keying

A propagating electromagnetic wave can change polarization on its way due to an obstacle. Linear polarizations are more sensitive to this phenomenon than circular polarizations. Indeed, circular polarizations are more robust to propagation constraints (obstacles, mobility), making them more suitable to dynamic environments than linear polarizations. For example, a simple 90° rotation of a linear polarization antenna is enough to switch from vertical (V) to horizontal (H) polarization, or vice versa. To handle this in PoSK-based message authentication system, both antennas must rotate together, and both terminals (Wireless Sensor and Security Gateway) must rotate together. This can be difficult to manage, that's why linear polarizations are suitable for static setup and LOS (Line Of Sight) channels only.

The impact of polarization switching time is present in both architectures with linear polarizations because of the switching time of the SPDT switch and in both architectures with circular polarizations because of the phase shifting time of the phase shifter.

The phase shifts related to the PoSK, in architectures with circular polarizations, are actually unwanted phase shifts in the main message, and can be particularly disturbing when a phase-dependent modulation (e.g., PSK) is used as the main message modulation. These phase shifts must be corrected, at the reception, using signal processing techniques.

Since the Undesired Polarization Sensitivity is channel-dependent and the authentication is based on the received power, the mobility management is necessary to use the PoSK in dynamic environments whether linear or circular polarizations are used. The targeted environments in this initial work are critical static environments such as industrial plants, civil engineering (e.g., nuclear power plants) Structural Health Monitoring (SHM), etc.

8.2. Synchronization-required vs Synchronization-free Architectures

The absence of synchronization in the Synchronization-free architectures (Figure 16(b) and Figure 17(b)) is the major advantage of these architectures over Synchronization-required ones (Figure 16(a) and Figure 17(a)). Indeed, synchronization is difficult to implement, especially at high PoSK frequencies (F_{PoSK}). Another advantage of Synchronization-free architectures is the absence of dynamic polarization switching on the Security Gateway, which may contribute to the reduction of the impact of polarization switching time on the main message. In addition, Synchronization-free architectures are likely more energy-efficient than Synchronization-required architectures, as the Cryptographic Polarization Selector and the "Tx/Rx Start Detection" blocks are not required in the former. However, Synchronization-free Architectures require a second port and more Splitters and Phase Shifters on the Security Gateway.

8.3. Software-Defined Cryptographic Polarization Controller

The Active Antenna System's Cryptographic Polarization Controller block (see Figure 15) has to be ultimately implemented on an ASIC or on a FPGA, an architecture we have called hardware-oriented architecture. However, a software-defined approach for this block seems also interesting, in this case, we call it software-oriented architecture. The first advantage of this latter approach is that it simplifies PoSK implementation, as the only remaining hardware block is the Multi-Polarization Antenna, i.e., the radiative elements. This significantly reduces

the complexity of PoSK implementation compared to hardware-oriented architecture, at least from an experimental point of view.

Another point of comparison between the two architectures is flexibility, which also leans in favor of the software-oriented architecture. Indeed, it is easier to make changes to software than to an ASIC after it has been manufactured.

Another point of comparison between the two architectures is their integration into IoT devices. Indeed, integrating the hardware-oriented architecture into an already deployed IoT device, i.e., COTS-like devices, is a matter of replacing the standard antenna with a PoSK-enabled Active Antenna System (AAS), whereas integrating the software-oriented architecture will be more difficult. Indeed, requiring two ports on each terminal (i.e., Wireless Sensor and Security Gateway) to connect the antennas, this software-defined approach is not easy to integrate into already deployed IoT devices that did not expect to use PoSK. In addition of adding a second antenna port, it will also be necessary to add the PoSK (the cryptographic polarization selection algorithm) implementation to the firmware. Consequently, the integration of the software-oriented architecture should be considered early at the design stage of the device.

Another point of comparison that leans in favor of the hardware-oriented architecture is the maximum possible PoSK frequency (F_{PoSK}), i.e., the maximum achievable value for F_{PoSK}, or the maximum speed of the polarization switching. Indeed, with the hardware switch in linear PoSK experimentation, we can go up to 1 MHz for F_{PoSK}, whereas with the Python implementation on GNU Radio, we can barely reach 1 KHz for F_{PoSK}, that is, 1000 times lower than with the hardware-oriented architecture. However, this comparison is a little unfair because in the case of 1 MHz F_{PoSK}, a waveform generator was used to control the switch, so it is a square-wave signal instead of the cryptographic polarization selection algorithm output signal.

In the software-oriented architecture, the maximum possible value of F_{PoSK} is defined by the execution time of the used cryptographic polarization selection algorithm, which depends on the used programming language, optimization techniques and the performance of the computing device used. Figure 44 and Figure 50 show implementation examples of the software-oriented architecture (using Python programming language via GNU Radio) of the Synchronization-free architectures with linear polarizations (Figure 16(b)) and circular polarizations (Figure 17(b)), respectively. However, this limitation in the maximum possible F_{PoSK} value is less critical. It means a limitation in the maximum achievable bit rate for the authentication code. Therefore, the software-oriented architecture would not be the best option if a high bit rate is required for the authentication code. This may be necessary if the main message to be authenticated is relatively very short in terms of transmission time. However, high values for F_{PoSK}, in the megahertz (MHz) range, should only be used when necessary (i.e., depending on the spectral bandwidth of the message, the length of the message to be authenticated and the desired length of the authentication codes), as this obviously increases energy consumption, which is the next point of comparison between the two approaches, hardware-oriented and software-oriented architectures.

An interesting idea that makes the limitation in the maximum achievable value for F_{PoSK} even less constraining would be to do a whole communication authentication instead of message authentication. In other words, the Wireless Sensor transmits a whole message on one polarization or on the other, following the cryptographic polarization selection algorithm. However, this idea, which is actually the use of a 1-bit authentication code to authenticate a

message, will be less robust from a security point of view, compared to a multi-bit authentication code.

Another very important point of comparison between the two architectures is the energy consumption. A preliminary study of the energy consumption of the hardware-oriented architecture was carried out and presented in section "7. Energy Consumption". We obtained relatively high power consumptions, that is, an average consumptions of about 390 mW for the SPDT switch and 205 mW for the EV1HMC647ALP6 phase shifter. These values are indeed high in IoT. Therefore, further and more precise study of energy consumption and its optimization should be part of future work. For energy consumption optimization, one could be interested, among other things, in the design of the switch and the phase shifter, the choice of the components (e.g., choose low-energy SPDT switch), with an emphasis on miniaturization and integration. A hardware implementation on FPGA or on ASIC of the Wireless Sensor and the Security Gateway will allow more accurate power consumption measurements. A study of the energy consumption of the software-oriented architecture has not been carried out in this work, and will therefore be part of future work as well. The most important thing to note is that, whether it is hardware (e.g., ASIC, FPGA) or software-defined (e.g., C, C++, Python) implementation, the objectives remain operational and energy efficiency.

Table 7. Comparative table of hardware-oriented and software-oriented PoSK-based message authentication system architectures.

	Implementation	*Flexibity*	***Integration into an already deployed IoT device***	***Integration early at the IoT device design stage***	***Maximum PoSK frequency (F_{PoSK} max)***	*Consumption*
Hardware-oriented architecture	High complexity	Low flexibility	Easy integration	Easy integration	High	Can be improved
Software-oriented architecture	Low complexity	High flexibility	Difficult integration	Easy integration	Limited by the execution time of the cryptographic polarization selection algorithm	Not investigated in this work

8.4. Undesired Polarization Sensitivity

The Undesired Polarization Sensitivity is due to the fact that, on a given channel, the amount of power received by the Security Gateway in an X to X transmission is different from the amount of power received in a Y to Y transmission, where X and Y are the used polarizations, i.e., linear polarizations V and H or circular polarizations RHCP and LHCP. This phenomenon is channel-dependent, therefore mobility-dependent. In addition, the switch losses lower (or higher) on one of its outputs than on the other also contributes to this phenomenon in architectures with linear polarizations, and phase noise contributes to it in architectures with circular polarizations.

As a consequence, the Undesired Polarization Sensitivity reveals the authentication code through an undesired amplitude modulation of the message signal. Fortunately, the authentication code is dynamic, i.e., the transmitted authentication code becomes invalid. So, to guess the next code and get ahead of the legitimate device, the attacker needs to know both the used cryptographic algorithm and the initial secret key that was shared between the Wireless Sensor and the Security Gateway. Therefore, this phenomenon has no significant impact on the efficiency of this PoSK-based authentication technique.

Nevertheless, to prevent this phenomenon from degrading communication quality, it is important to correct it as best as possible, especially when an amplitude-dependent modulation (e.g., ASK, QAM) is used for the main message, since this phenomenon introduces undesired amplitude modulation. Some possible solutions and techniques (e.g., AGC – Automatic Gain Control) for preventing / correcting Undesired Polarization Sensitivity are proposed in section "5.6.1. Synchronization-required architecture".

If Figure 51[c] presents a proof that the Undesired Polarization Sensitivity can be avoided in the circular PoSK, synchronization-free, software-oriented architecture, yet, it would not be prudent to claim, through this initial work, that some of the architectures of the PoSK-based authentication system would not be vulnerable to the Undesired Polarization Sensitivity phenomenon. Indeed, given the very nature of this phenomenon, all the architectures investigated in this work, whether they use linear or circular polarizations, whether they are synchronization-required or synchronization-free, whether they are hardware-oriented or software-oriented, could be vulnerable to the Undesired Polarization Sensitivity.

8.5. Polarization Switching Time

Switching from one polarization to the other is not instantaneous; indeed, this takes a certain lapse of time to be effective. This time corresponds to the switching time of the SPDT switch in architectures with linear polarizations, and to the phase shifting time of the phase shifter in architectures with circular polarizations. Consequently, each polarization switching causes a brief interruption in the transmission, resulting in sharp and brief power drops in the instantaneous power (e.g., Figure 34[c]) and spikes in the demodulation of the main message (e.g., Figure 34[b], Figure 36[b]). To reduce the risk of bit errors in the main message, due to PoSK, a low-pass filter with a cut-off frequency at least equal to the symbols rate (F_S) was used to remove the spikes before binary slicing. It is also important not forget to remove the sharp and brief power drops in the instantaneous power before decoding the authentication code; otherwise, there will be several false positives in the authentication. These power drops can be removed using also a low-pass filter, but with a cut-off frequency at least equal to the symbols rate of the authentication code.

The duration of the polarization switching time is very important in this PoSK-based authentication system. The more the duration of the modulation symbol ($1/F_S$) is higher than the duration of the polarization switching time, the more efficient the filtering, the lower the risk of bit errors due to PoSK. Consequently, for this application, one of the most important criteria when choosing a switch (or a phase shifter, depending on the architecture) is switching time, that is, how fast the switch can go from one output to another. This switching time must be as short as possible.

On the other hand, very high values in the MHz range for F_{PoSK} should only be used when necessary (i.e., depending on the spectral bandwidth of the message, the length of the message

to be authenticated and the desired length of the authentication codes) and this is a good thing because, the higher F_{PoSK} the higher the energy consumption in an environment where energy is already scarce. However, the higher F_{PoSK}, the more it is possible to associate long authentication codes with very short messages. High F_{PoSK} values should only be used if really necessary.

If Figures 39-40 present a proof that the impact of the polarization switching time can almost disappear in the linear PoSK, synchronization-free, hardware-oriented architecture, again, it would not be prudent to claim, through this initial work, that some of the architectures of the PoSK-based authentication system would not be vulnerable to the impact of the polarization switching time. More investigations will be needed to identify with greater certainty which architectures are less affected by this phenomenon.

8.6. Inter-Polarization Cross-talk

Here, inter-polarization cross-talk refers to the phenomenon wherein some power is received on the polarization X when the actual transmission is in co-polarization Y to Y, or vice versa (some power is received on the polarization Y when the actual transmission is in co-polarization X to X), where X and Y are used polarizations. Actually, this phenomenon is not very disturbing in the PoSK-based message authentication system architectures experimented in this work.

In the Synchronization-required architecture with linear polarizations (Figure 16(a)), the inter-polarization cross-talk is not an issue because when one antenna (i.e., polarization) is in use, the other antenna (i.e., polarization) is disconnected to the Security Gateway. In the Synchronization-free architecture with linear polarizations (Figure 16(b)), the inter-polarization cross-talk is not very disturbing either, as the two polarizations are always required simultaneously by the Security Gateway.

Architectures with circular polarizations (Figure 17(a), Figure 17(b)) are the most advantageous regarding inter-polarization cross-talk because these architectures use a single-piece 2-port antenna instead of two separated antennas.

9. Conclusion

This work represents a proof of concept of the use of radio signal Polarization Shift Keying (PoSK) to secure IoT communications regardless of protocols. Transmitter using the PoSK-based message authentication system was successfully experimented. It was demonstrated that this transmitter (i.e., Wireless Sensor) authentication via the instantaneous power and the main message error-free recovering are possible, by the Security Gateway. The PoSK-based message authentication system was experimented with different shift keying rate (F_{PoSK}) of the polarization; in different environments (anechoic chamber and regular environment); with different standard modulations (ASK, FSK, PSK) for the main message.
The impact of the PoSK-based message authentication system on the main message signal was outlined and some possible solutions for overcoming the undesired effects introduced by this authentication system was proposed. For example, filtering was proposed to remove the impact of polarization switching time. Also, the automatic gain control (AGC) technique was proposed to correct the impact of undesirable polarization sensitivity.

For a robust implementation of the PoSK-based message authentication system and to mitigate its impact on communication quality, one should not forget to consider the following points when necessary:

- ***Clearly identify the sources of the Undisered Polarization Sensitivity and Polarization Switching Time phenomena in order to better address their impacts***: Although the causes of these two phenomena have been well exposed in this initial work, it would still be useful to study these phenomena more in-depth, in order to identify possible PoSK-based message authentication system architectures that could be more resistant to these phenomena. Because, in this initial work, we found indeed that not all the architectures are equal regarding these phenomena. For examples, Figures 39-40 present a proof that the impact of the polarization switching time can almost disappear in the linear PoSK, synchronization-free, hardware-oriented architecture and Figure 51[c] presents a proof that the Undesired Polarization Sensitivity can be avoided in the circular PoSK, synchronization-free, software-oriented architecture.

- ***Management of the received power variation over the time:*** Since the authentication of the transmitter (the Wireless Sensor) by the receiver (the Security Gateway) is based on the received power, it is necessary to consider the variation of this power over the time.
This point was not a priority in this initial work, as the management of the received power variation is not a new topic in radio-communications, it is an already widely covered topic; therefore, it is taken for granted in this work. However, the management of the received power variation is necessary for the use of the PoSK-based message authentication system in a dynamic IoT environment, since parameters influencing the received power include the distance between the two terminals and the channel noise.
Among the techniques used to correct variations in received power is Automatic Gain Control (AGC), which a receiver uses to automatically and dynamically adjust its gain to maintain a stable signal level despite fluctuations or fading. In a hardware-oriented architecture of the PoSK-based message authentication system, an AGC can be realized using a Variable Gain Amplifier (VGA) combined with a circuit for measuring the received power level. In a software-oriented architecture of the PoSK-based message authentication system, a software-defined AGC can be implemented with a function for measuring the received power level associated with a digital equalizer function, or with the use of tools like Adaptive Algorithm.
However, in the case of the PoSK-based message authentication system, the AGC should take into account the different power levels related to the system itself. Indeed, in the authentication signal received by the Security Gateway, there are as many useful power levels as there are polarizations used to perform PoSK-based message authentication. These power levels are inherent to the PoSK-based message authentication system, none of them should not be corrected by the AGC. For example, in 2-PoSK, as in this work, the authentication signal contains two useful power levels, one for logic 0 and the other for logic 1, as previously seen. If one of these two power levels is corrected by the AGC, there will be no authentication code, as the result will be a single-amplitude authentication signal, in other words, a signal with a single power level.

- ***Correction of the PoSK-related phase shifts:*** In circular PoSK, to switch from one polarization to another (e.g., from LHCP to RHCP or vice versa), -90°/+90° phase shifts are performed. These phase shifts cause unwanted phase shifts in the main message signal. This can be particularly disturbing if the modulation used for the main message is phase-dependent, such as PSK. In this case, it would be useful to correct these unwanted phase shifts to avoid bits errors in the main message. Nevertheless, the Security Gateway, having knowledge of the expected polarization change pattern (authentication code), can recognize unwanted PoSK-related phase shifts and ignore them in the demodulation of the main message. For the synchronization-required architectures, taking into account relations (21) and (22) in the time synchronization can make it easier to recognize unwanted phase shifts, as these will never occur at the same time as the symbol change of the main message.

The polarization switching time, which has been well explained throughout this manuscript, can also be a source of unwanted phase shifts in the main message signal, even in linear PoSK. Indeed, as this polarization switching time is non-zero, at the end of the polarization change, transmission resumes on the new polarization at a sample (of the signal being transmitted) subsequent (i.e., a new phase) to the sample at which the polarization change began, with the potential loss of several samples, or even periods of the carrier, between the two samples.

- ***Management of the effects of phase noise:*** In the PoSK-based message authentication system architectures, presented on Figure 16 and Figure 17, one can see that splitters are used to add or split signals; it is the addition that interests us here. Indeed, since the result of the addition of two signals depends on the phase shift between them, the phase noises of the transmitter and the receiver become potential sources of trouble. In reception, these noises can cause slight fluctuations in the amplitude of the main message signal, this can even go so far as to cause fading. The circular PoSK architectures are the most vulnerable to phase noise because of the phase shifts (of the PoSK) during the transmission. To avoid phase noise-related issues as much as possible, phase shifters with the lowest possible phase noise should be used. An equalizer or an AGC system can also be used in this case to equalize the amplitude of the main message signal, while taking into account cases where the modulation of the main message involves amplitude, i.e., the modulation of the main message is amplitude-dependent (e.g., ASK, QAM).

- ***Use hardware-oriented or software-oriented Cryptographic Polarization Controller (see Figure 15)***: Although the original idea of the PoSK-based message authentication system favors a hardware implementation on FPGA (or on ASIC), we were nevertheless interested in a software implementation, using GNU Radio (Python), of this "Cryptographic Polarization Controller" block. As previously seen, the experimental results obtained with the software-defined Cryptographic Polarization Controller are of such good quality that this software-defined approach deserves further attention. For example, it would be interesting to implement with a programming language that allows better optimization than Python, such as C or assembly language. Remember, whether it is a hardware implementation on FPGA (or on ASIC) or a software implementation using programming languages such as Python, C/C++ or others, the objectives remain operational and energy efficiency. As a reminder, compared to the software-oriented approach, the hardware-oriented approach is easier to integrate into an IoT device that has already been deployed.

- ***Energy consumption:*** Energy consumption is one of the most important points to consider, especially when it comes to security solutions for IoT where devices are energy-constrained. In the PoSK-based message authentication system, energy consumption depends on the components used in the Cryptographic Polarization Controller (see Figure 15) and the PoSK frequency (F_{PoSK}). The higher the F_{PoSK}, the higher the consumption. This is why high values for F_{PoSK} should only be used when necessary, i.e., depending on the spectral bandwidth of the message, the length of the message to be authenticated and the desired length of the authentication codes. It is a matter of finding the optimum trade-offs. In an implementation of the PoSK-based message authentication system, a quantification of the energy consumption must be carried out and must be part of the criteria for choosing the architecture to be used: hardware-oriented or software-oriented architecture. Energy consumption must be optimized as much as possible.

- ***Choose an appropriate cryptographic polarization selection algorithm (e.g., AES, ASCON)***: In this initial work, a lightweight LFSR (shown on Figure 33) was used as a cryptographic polarization selection algorithm for its ease of implementation. Actually, the

choice of this algorithm is a trade-off between desired level of security, power consumption, computing power, and available memory. A cryptographically complex algorithm may provide a higher level of security but can be difficult to implement on an IoT device due to an insufficient amount of computing power, memory, or available energy. The suitable choice is an algorithm that is complex enough to offer a certain level of security but lightweight in terms of computation workload, memory usage, and energy consumption. Ideally, a lightweight version of one the powerful and commonly used algorithms, such as AES or ASCON.

- ***When the main message is using an advanced complex modulation pattern***: In this initial work, 2-state elementary modulations (2-ASK, 2-FSK, 2-PSK) are used as the main message modulations. However, the use of a complex modulation pattern, such as QAM, should not be more problematic than the use of elementary modulation because, as the results of our studies have shown, the impact of PoSK on communication quality does not depend on the modulation but rather on the choice of polarization keying frequency in relation to the spectral band of the message. If the polarization keying frequency is chosen outside the actual spectral band of the message, the type of modulation could be of no importance. Nevertheless, if necessary, one have to use more advanced filtering techniques and error correction algorithms specifically tailored to mitigate any degradation introduced by the PoSK-based message authentication system in the main message signal. In this AI era, we can even imagine using AI tools for signal processing. It would be great to have a Machine Learning model that has learned the effects of PoSK on the main message signal and can therefore correct them efficiently.

- ***Validate an implementation of the PoSK-based message authentication system before deployment*:** Although possible, automatic detection of a Wireless Sensor, by the Security Gateway, has not been implemented in this initial work. Indeed, the reception and decoding of the main message and the authentication code are done manually, using URH software or GNU Radio software for reception. GNU Radio software was also used for everything related to signal processing. The whole process has to be implemented in such a way that everything is done automatically, as well as automatic raising of alarms when an anomaly occurs (e.g., detection of an illegitimate Wireless Sensor trying to communicate with the Security Gateway). In this way, we can experiment the PoSK-based message authentication system with a Wireless Sensor Network, using several Wireless Sensors, some of which considered legitimate and others illegitimate. This will also make it easier to study the performance of the PoSK-based message authentication system through parameters such as false-positive, false-negative, true-positive and true-negative.
  It is also good to know that an implementation on FPGA or on ASIC of the Cryptographic Polarization Controller block (see Figure 15) will allow more precise studies on the hardware footprint of the PoSK-based message authentication system, its computational workload, power consumption, maximum achievable PoSK frequency (F_{PoSK} max), etc…
  An implementation of the PoSK-based message authentication system must also be validated in terms of its impact on the Bit Error Rate, through experiments in very noisy environment and in the presence of multipath propagation.

General Conclusion and Perspectives

The vulnerability of IoT has made its security a major research topic. This thesis entitled "Security for Internet of Things: hardware authentication mechanisms for communicating objects" is part of this context and focuses on security at the PHY layer, which could be a very interesting layer in the research for generic and protocol-independent security solutions.

At the beginning, we studied devices fingerprinting, a well-known approach in the literature. We have highlighted the possibilities and challenges of this approach through a hands-on study in which we explored the possibility of identifying a transmitter via the Power Spectral Density (PSD) of its signal. We have come to the conclusion that PSD alone as a fingerprinting feature is not sufficient for authentication, as PSD depends on the transmitted data and is also environment-sensitive. Therefore, it is impossible to associate a unique and time-constant PSD-based signature with a given device. This makes PSD an irrelevant fingerprinting feature. The major challenge facing fingerprinting remains to find relevant fingerprinting features.

In this thesis, we proposed a new and very promising security approach, which is a messages authentication system based on Polarization Shift Keying (PoSK) technique. The results of our proof-of-concept work on this approach prove its feasibility and its relevance. We have demonstrated that it is indeed possible to transmit message authentication data in parallel with the message itself, using the polarization of the radio wave as a carrier for the authentication data.

Our experimental results have shown that PoSK has no impact on the bit error rate (BER) if the polarization keying frequency (F_{PoSK}) is outside the actual spectral band of the main message. On the other hand, if the polarization switching frequency (F_{PoSK}) lies within the actual spectral band of the main message, the polarization switching time affects the message signal by introducing noise that can impede the proper demodulation of the signal. We propose filtering as a means of removing the noise introduce by the polarization switching time before and/or after demodulation, but this is not always effective depending on the noise level. It is therefore preferable to choose a polarization keying frequency (F_{PoSK}) outside the actual spectral band of the main message to be authenticated.

On the transmission side (Wireless Sensor), it is important to avoid as much as possible the phenomenon of Undesired Polarization Sensitivity in order to prevent the authentication code from being leaked through the message signal. Fortunately, this code is dynamic, meaning that once transmitted, it becomes invalid. Among the ways of avoiding this undesired polarization sensitivity are: ensuring that the antennas used for the different polarizations have the same gain, and ensuring that the polarization router has the same loss on each of its outputs.

On the receiver side (Security Gateway), the effect of Undesired Polarization Sensitivity can be corrected using a hardware or software-defined equalizer or AGC (Automatic Gain Control) system, to avoid misleading the receiver regarding the modulation of the main message. This is particularly important when the modulation of the message is amplitude-dependent, as undesired polarization sensitivity results in the introduction of an undesired amplitude modulation.

Just as the frequency hopping mechanism has enhanced the security of some protocols (e.g. Bluetooth), PoSK-based message authentication system, which is basically a kind of polarization hopping mechanism, will enhance the security of radiocommunications regardless

of protocols. Moreover, PoSK is a low-energy mechanism, making it very suitable for IoT resource-constrained devices. The approach is also non-invasive, since the only part of the device affected is the antenna part. It is a dynamic message authentication system, low computational complexity and lightweight memory footprint.

This PoSK-based authentication approach is also a robust security system. Indeed, since the authentication is dynamic (the authentication code changes continuously), for an attacker to succeed in thwarting this system, he/she must be able to guess the next authentication code. To do this, the attacker needs to know the algorithm used, the shared initialization secret key and the current state (output) of the algorithm. In addition, considering the fact that a transmitted authentication code becomes automatically invalid, one can say that this PoSK-based security system is hard to thwart.

In the future, we will extend this work by studding two major axes:

i) ***Integration and experimentation on IoT devices*****:** experimentation of the PoSK-based message authentication system in a network of real-world wireless sensors, with real-world IoT devices communicating via Bluetooth or LoRa, for example. The robustness of PoSK message authentication to large scale IoT network will be tested.

ii) ***High-order PoSK:*** This proof-of-concept work uses 2-PoSK, i.e., two polarizations (Horizontal and Vertical or RHCP and LHCP) are used to perform PoSK in order to transmit the authentication code in parallel with the main message. To make this authentication technique even more robust, it would be interesting to use more than two polarizations, referred to as high-order PoSK. This can be performed by using a Meta-surface, i.e., a Reconfigurable Intelligent Surface (RIS), i.e., a Software-Defined Antenna (SDA). Long-term perspectives include the experimentation of high-order PoSK.

List of contributions

❖ International journals

[1] Lamoussa Sanogo, Eric Alata, Alexandru Takacs, Daniela Dragomirescu. "Intrusion Detection System for IoT: Analysis of PSD Robustness". Sensors, 2023, 23 (4), pp.2353. ⟨10.3390/s23042353⟩. ⟨hal-04154781⟩.

[2] Sanogo et al.. "Polarization Shift Keying for IoT Communications Security: An Innovative Solution for Message Authentication in Wireless Sensor Networks". IEEE IoT Journal. 2025 (Submittted).

❖ International conferences with proceedings

[3] Lamoussa Sanogo, Eric Alata, Taki E. Djidjekh, Gaël Loubet, Alexandru Takacs, Daniela Dragomirescu, "Polarization Shift Keying for Device Authentication in Wireless Sensor Network", European Microwave Week, 22-27 september 2024, Paris, France.

[4] Lamoussa Sanogo, Eric Alata, Gaël Loubet, Taki E. Djidjekh, Alexandru Takacs, Daniela Dragomirescu, "A Hardware Add-On for IoT Secure Communications: Polarization Shift Keying Smart Antenna System", The 10th IEEE World Forum on the Internet of Things (IoT), 10-13 November 2024, Ottawa, Canada.

[5] Lamoussa Sanogo, Eric Alata, Gaël Loubet, Taki E. Djidjekh, Alexandru Takacs, Daniela Dragomirescu, "Circular Polarization Shift Keying for IoT Communications Security: A proof of concept", The 31st IEEE International Conference on Electronics Circuits and Systems, 18-20 November 2024, Nancy, France.

[6] Taki E. Djidjekh, Lamoussa Sanogo, Gaël Loubet, Alassane Sidibé, Daniela Dragomirescu, Alexandru Takacs, "A New Security and Identification Concept for SWIPT Systems in IoT Applications", International Microwave Symposium, 16-21 June 2024, Washington, DC.

[7] Taki E. Djidjekh, Gaël Loubet, Lamoussa Sanogo, Alassane Sidibé, Guillaume Delai, Daniela Dragomirescu, Alexandru Takacs, "Backscattering Rectifier for Security and Identification in the context of Simultaneous Wireless Information and Power Transfer", European Microwave Week, 22-27 september 2024, Paris, France.

[8] Alassane Sidibe, Gaël Loubet, Alexandru Takacs, Lamoussa Sanogo, Daniela Dragomirescu. "Battery-free Bluetooth Low Energy Wireless Sensor Powered by Radiative Wireless Power Transfer", NewCAS 2023, IEEE, Jun 2023, Edinburg,, United Kingdom. ⟨10.1109/NEWCAS57931.2023.10198088⟩. ⟨hal-04157047⟩.

[9] Alexandru Takacs, Gaël Loubet, Taki E. Djidjekh, Lamoussa Sanogo, Daniela Dragomirescu, "Battery-free Wireless Sensors for IoT applications", The 15th International Conference on Communications (COMM), October 03-04, 2024, Bucharest, Romania.

[10] Gaël LOUBET, Taki E. DJIDJEKH, Lamoussa SANOGO, Alexandru TAKACS, and Daniela DRAGOMIRESCU, "Backscattering and Wireless Power Transfer in Battery-free Sensors for Structural Health Monitoring of Reinforced Concrete Structures", The IEEE RFID Technology and Applications, 27-30 October 2025, Valence, France.

❖ International conferences without proceedings

[11] Lamoussa Sanogo, Eric Alata, Alexandru Takacs, Daniela Dragomirescu, "Intrusion Detection System for IoT : Analysis of PSD robustness", École d'hiver Francophone sur les Technologies de Conception des Systèmes Embarqués Hétérogènes (FETCH 2023), 01-03 February 2023, Lavey-les-Bains, Vaud, Switzerland.

❖ French national conferences

[12] Gaël Loubet, Alassane Sidibe, Taki E. Djidjekh, Lamoussa Sanogo, Alexandre Rumeau, Alexandru Takacs, Daniela Dragomirescu, “Réseau de capteurs Bluetooth Low Energy, sans fil et sans batterie, télé-alimentés par transfert de puissance électromagnétique rayonnée”, 23ème Journées Nationales Microondes, 4-7 Juin 2024, Antibes Juan-Les-Pins.

❖ GDR - French research group congress

[13] Lamoussa Sanogo, Eric Alata, Alexandru Takacs, Daniela Dragomirescu, “Intrusion Detection System for IoT : Analysis of PSD robustness”, 17ème Colloque du GDR SoC^2, 12-14 Juin 2023, Campus de la Doua, 3 Rue Enrico Fermi, 69100 Villeurbanne.

[14] Lamoussa Sanogo, Eric Alata, Alexandru Takacs, Gaël Loubet, Daniela Dragomirescu, “Polarization Shift Keying: A New Approach to Secure IoT Communications”, 18ème Colloque du GDR SoC^2, 10-12 Juin 2024, Campus de l’INSA Toulouse.

References

[1] J. C. Maxwell, “VIII. A dynamical theory of the electromagnetic field,” *Philos. Trans. R. Soc. Lond.*, vol. 155, pp. 459–512, Jan. 1997, doi: 10.1098/rstl.1865.0008.

[2] IoT Analytics, “State of IoT 2024: Number of connected IoT devices growing 13% to 18.8 billion globally.” Accessed: Oct. 02, 2024. [Online]. Available: https://iot-analytics.com/number-connected-iot-devices/

[3] Berg Insight, “Berg Insight says 76.7 million patients worldwide are remotely monitored.” Accessed: Oct. 01, 2024. [Online]. Available: https://www.berginsight.com/berg-insight-says-767-million-patients-worldwide-are-remotely-monitored

[4] Statista, “Smart Home - Worldwide | Statista Market Forecast.” Accessed: Oct. 01, 2024. [Online]. Available: https://www.statista.com/outlook/cmo/smart-home/worldwide

[5] Transforma Insights, “Smart Cities.” Accessed: Oct. 01, 2024. [Online]. Available: https://transformainsights.com/smart-cities

[6] IoT Business News, “Global smart meters to double to 3.4B by 2033, generating USD 40B in annual revenue.” Accessed: Oct. 01, 2024. [Online]. Available: https://iotbusinessnews.com/2024/08/16/60017-global-smart-meters-to-double-to-3-4b-by-2033-generating-usd-40b-in-annual-revenue/

[7] Berg Insight, “The installed base of wireless devices in industrial automation reached 56.5 million in 2023.” Accessed: Oct. 01, 2024. [Online]. Available: https://www.berginsight.com/the-installed-base-of-wireless-devices-in-industrial-automation-reached-565-million-in-2023

[8] IoT Analytics, “Industrial Connectivity Market Report 2024–2028.” Accessed: Oct. 01, 2024. [Online]. Available: https://iot-analytics.com/product/industrial-connectivity-market-report-2024-2028/

[9] Berg Insight, “The Global Remote Tank Monitoring Market.” Accessed: Oct. 01, 2024. [Online]. Available: https://www.berginsight.com/the-global-remote-tank-monitoring-market

[10] G. Loubet, A. Sidibe, P. Herail, A. Takacs, and D. Dragomirescu, “Autonomous Industrial IoT for Civil Engineering Structural Health Monitoring,” *IEEE Internet Things J.*, vol. 11, no. 5, pp. 8921–8944, Mar. 2024, doi: 10.1109/JIOT.2023.3321958.

[11] Berg Insight, “The installed base of tracking devices for trailers, containers, rail freight wagons, ULDs and pallets reached 8.8 million in 2022.” Accessed: Oct. 02, 2024. [Online]. Available: https://www.berginsight.com/the-installed-base-of-tracking-devices-for-trailers-containers-rail-freight-wagons-ulds-and-pallets-reached-88-million-in-2022

[12] Transforma Insights, “Child & Pet Tracking: 19.5 million connected devices by 2030, generating USD1.2 billion - Reports & Insights.” Accessed: Oct. 02, 2024. [Online]. Available: https://transformainsights.com/research/reports/child-pet-tracking

[13] Transforma Insights, “AgTech.” Accessed: Oct. 02, 2024. [Online]. Available: https://transformainsights.com/ag-tech

[14] NETGEAR, “The 2024 IoT Security Landscape Report - NETGEAR.” Accessed: Oct. 15, 2024. [Online]. Available: https://www.netgear.com/hub/network/2024-iot-threat-report/

[15] B. Krebs, “KrebsOnSecurity Hit With Record DDoS – Krebs on Security,” KrebsOnSecurity. Accessed: Oct. 15, 2024. [Online]. Available: https://krebsonsecurity.com/2016/09/krebsonsecurity-hit-with-record-ddos/

[16] H. Sinanović and S. Mrdovic, “Analysis of Mirai malicious software,” in *2017 25th International Conference on Software, Telecommunications and Computer Networks (SoftCOM)*, Sep. 2017, pp. 1–5. doi: 10.23919/SOFTCOM.2017.8115504.

[17] J. Margolis, T. T. Oh, S. Jadhav, Y. H. Kim, and J. N. Kim, “An In-Depth Analysis of the Mirai Botnet,” in *2017 International Conference on Software Security and Assurance (ICSSA)*, Jul. 2017, pp. 6–12. doi: 10.1109/ICSSA.2017.12.

[18] G. Kambourakis, C. Kolias, and A. Stavrou, “The Mirai botnet and the IoT Zombie Armies,” in *MILCOM 2017 - 2017 IEEE Military Communications Conference (MILCOM)*, Oct. 2017, pp. 267–272. doi: 10.1109/MILCOM.2017.8170867.

[19] M. Baezner and P. Robin, “Hotspot Analysis: Stuxnet,” Center for Security Studies (CSS), ETH Zürich, Oct. 2017.
[20] J. Rudie, Z. Katz, S. Kuhbander, and S. Bhunia, “Technical Analysis of the NSO Group’s Pegasus Spyware,” in *2021 International Conference on Computational Science and Computational Intelligence (CSCI)*, Dec. 2021, pp. 747–752. doi: 10.1109/CSCI54926.2021.00188.
[21] R. Cayre, F. Galtier, G. Auriol, V. Nicomette, M. Kaâniche, and G. Marconato, “WazaBee: attacking Zigbee networks by diverting Bluetooth Low Energy chips,” in *2021 51st Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*, Jun. 2021, pp. 376–387. doi: 10.1109/DSN48987.2021.00049.
[22] National Instruments, “Software Defined Radios.” Accessed: Nov. 26, 2024. [Online]. Available: https://www.ni.com/en-us/shop/category/software-defined-radios.html
[23] Great Scott Gadgets, “HackRF One - Great Scott Gadgets.” Accessed: Nov. 26, 2024. [Online]. Available: https://greatscottgadgets.com/hackrf/one/
[24] GNU Radio, “GNU Radio.” Accessed: Dec. 19, 2024. [Online]. Available: https://www.gnuradio.org/
[25] D.-I. J. Pohl, *jopohl/urh*. (Dec. 19, 2024). Python. Accessed: Dec. 19, 2024. [Online]. Available: https://github.com/jopohl/urh
[26] J. Pohl and A. Noack, “Universal Radio Hacker: A Suite for Analyzing and Attacking Stateful Wireless Protocols,” presented at the 12th USENIX Workshop on Offensive Technologies (WOOT 18), 2018. Accessed: May 05, 2025. [Online]. Available: https://www.usenix.org/conference/woot18/presentation/pohl
[27] Flipper Devices Inc., “Flipper Zero — Portable Multi-tool Device for Geeks.” Accessed: Jan. 15, 2025. [Online]. Available: https://flipperzero.one
[28] L. Calderon and G. T. Salvador, “Detection and Analysis of Flipper Zero Deauthentication Signals Using HackRF One Software-Defined Radio,” in *2024 International Conference on Innovation and Intelligence for Informatics, Computing, and Technologies (3ICT)*, Nov. 2024, pp. 798–804. doi: 10.1109/3ict64318.2024.10824254.
[29] M. E. Bouazzati, R. Tessier, P. Tanguy, and G. Gogniat, “A Lightweight Intrusion Detection System against IoT Memory Corruption Attacks,” in *2023 26th International Symposium on Design and Diagnostics of Electronic Circuits and Systems (DDECS)*, May 2023, pp. 118–123. doi: 10.1109/DDECS57882.2023.10139718.
[30] N. Ye, B. Kang, B. Qi, X. Bu, and J. An, “Towards High-Security Ubiquitous Iot Networks Using AI,” in *2023 IEEE International Conference on Acoustics, Speech, and Signal Processing Workshops (ICASSPW)*, Jun. 2023, pp. 1–5. doi: 10.1109/ICASSPW59220.2023.10193693.
[31] M. Gupta, C. Akiri, K. Aryal, E. Parker, and L. Praharaj, “From ChatGPT to ThreatGPT: Impact of Generative AI in Cybersecurity and Privacy,” *IEEE Access*, vol. 11, pp. 80218–80245, 2023, doi: 10.1109/ACCESS.2023.3300381.
[32] H. Xu, Y. Li, O. Balogun, S. Wu, Y. Wang, and Z. Cai, “Security Risks Concerns of Generative AI in the IoT,” *IEEE Internet Things Mag.*, vol. 7, no. 3, pp. 62–67, May 2024, doi: 10.1109/IOTM.001.2400004.
[33] Y. M. Pa Pa, S. Tanizaki, T. Kou, M. van Eeten, K. Yoshioka, and T. Matsumoto, “An Attacker’s Dream? Exploring the Capabilities of ChatGPT for Developing Malware,” in *Proceedings of the 16th Cyber Security Experimentation and Test Workshop*, in CSET ’23. New York, NY, USA: Association for Computing Machinery, Aug. 2023, pp. 10–18. doi: 10.1145/3607505.3607513.
[34] N. Begou, J. Vinoy, A. Duda, and M. Korczyński, “Exploring the Dark Side of AI: Advanced Phishing Attack Design and Deployment Using ChatGPT,” in *2023 IEEE Conference on Communications and Network Security (CNS)*, Oct. 2023, pp. 1–6. doi: 10.1109/CNS59707.2023.10288940.
[35] V. Palazzi *et al.*, “Radiative Wireless Power Transfer: Where We Are and Where We Want to Go,” *IEEE Microw. Mag.*, vol. 24, no. 2, pp. 57–79, Feb. 2023, doi: 10.1109/MMM.2022.3210145.
[36] O. Seller, “LoRaWAN Security,” *J. ICT Stand.*, vol. 9, no. 1, pp. 47–60, 2021, doi: 10.13052/jicts2245-800X.915.
[37] L. Sanogo, E. Alata, A. Takacs, and D. Dragomirescu, “Intrusion Detection System for IoT: Analysis of PSD Robustness,” *Sensors*, vol. 23, p. 2353, Feb. 2023, doi: 10.3390/s23042353.

[38] S. G. Abbas, F. Hashmat, and G. A. Shah, “A Multi-layer Industrial-IoT Attack Taxonomy: Layers, Dimensions, Techniques and Application,” in *2020 IEEE 19th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom)*, Dec. 2020, pp. 1820–1825. doi: 10.1109/TrustCom50675.2020.00249.

[39] S. Khanam, I. B. Ahmedy, M. Y. Idna Idris, M. H. Jaward, and A. Q. Bin Md Sabri, “A Survey of Security Challenges, Attacks Taxonomy and Advanced Countermeasures in the Internet of Things,” *IEEE Access*, vol. 8, pp. 219709–219743, 2020, doi: 10.1109/ACCESS.2020.3037359.

[40] X. Liang and Y. Kim, “A Survey on Security Attacks and Solutions in the IoT Network,” in *2021 IEEE 11th Annual Computing and Communication Workshop and Conference (CCWC)*, Jan. 2021, pp. 0853–0859. doi: 10.1109/CCWC51732.2021.9376174.

[41] P. Mann, N. Tyagi, S. Gautam, and A. Rana, “Classification of Various Types of Attacks in IoT Environment,” in *2020 12th International Conference on Computational Intelligence and Communication Networks (CICN)*, Sep. 2020, pp. 346–350. doi: 10.1109/CICN49253.2020.9242592.

[42] F. Meneghello, M. Calore, D. Zucchetto, M. Polese, and A. Zanella, “IoT: Internet of Threats? A Survey of Practical Security Vulnerabilities in Real IoT Devices,” *IEEE Internet Things J.*, vol. 6, no. 5, pp. 8182–8201, Oct. 2019, doi: 10.1109/JIOT.2019.2935189.

[43] G. Rajendran, R. S. Ragul Nivash, P. P. Parthy, and S. Balamurugan, “Modern security threats in the Internet of Things (IoT): Attacks and Countermeasures,” in *2019 International Carnahan Conference on Security Technology (ICCST)*, Oct. 2019, pp. 1–6. doi: 10.1109/CCST.2019.8888399.

[44] R. Samrin and D. Vasumathi, “Review on anomaly based network intrusion detection system,” in *2017 International Conference on Electrical, Electronics, Communication, Computer, and Optimization Techniques (ICEECCOT)*, Dec. 2017, pp. 141–147. doi: 10.1109/ICEECCOT.2017.8284655.

[45] V. K. Pandey, S. Prakash, A. Pandey, T. Yang, and R. Singh Rathore, “Optimizing Intrusion Detection in WSNs: A Data-Driven Analysis of Signature-Based Intrusion Detection System,” in *2024 International Conference on Control, Computing, Communication and Materials (ICCCCM)*, Aug. 2024, pp. 772–775. doi: 10.1109/ICCCCM61016.2024.11039969.

[46] R. Fu, “Design and Implementation of Network Intrusion Detection System based on Machine Learning,” in *2025 International Conference on Intelligent Systems and Computational Networks (ICISCN)*, Jan. 2025, pp. 1–6. doi: 10.1109/ICISCN64258.2025.10934502.

[47] J. Roux, É. Alata, G. Auriol, V. Nicomette, and M. Kâaniche, “Toward an Intrusion Detection Approach for IoT Based on Radio Communications Profiling,” in *2017 13th European Dependable Computing Conference (EDCC)*, Sep. 2017, pp. 147–150. doi: 10.1109/EDCC.2017.11.

[48] J. Roux, É. Alata, G. Auriol, M. Kaâniche, V. Nicomette, and R. Cayre, “RadIoT: Radio Communications Intrusion Detection for IoT - A Protocol Independent Approach,” in *2018 IEEE 17th International Symposium on Network Computing and Applications (NCA)*, Nov. 2018, pp. 1–8. doi: 10.1109/NCA.2018.8548286.

[49] S. Bhadauria and T. Mohanty, “Hybrid Intrusion Detection System using an Unsupervised method for Anomaly-based Detection,” in *2021 IEEE International Conference on Advanced Networks and Telecommunications Systems (ANTS)*, Dec. 2021, pp. 1–6. doi: 10.1109/ANTS52808.2021.9936919.

[50] B. Chatterjee, D. Das, S. Maity, and S. Sen, “RF-PUF: Enhancing IoT Security Through Authentication of Wireless Nodes Using In-Situ Machine Learning,” *IEEE Internet Things J.*, vol. 6, no. 1, pp. 388–398, Feb. 2019, doi: 10.1109/JIOT.2018.2849324.

[51] B. Gassend, D. Clarke, M. van Dijk, and S. Devadas, “Silicon physical random functions,” in *Proceedings of the 9th ACM conference on Computer and communications security*, in CCS ’02. New York, NY, USA: Association for Computing Machinery, Nov. 2002, pp. 148–160. doi: 10.1145/586110.586132.

[52] B. Gassend, D. Clarke, M. van Dijk, and S. Devadas, “Controlled physical random functions,” in *18th Annual Computer Security Applications Conference, 2002. Proceedings.*, Dec. 2002, pp. 149–160. doi: 10.1109/CSAC.2002.1176287.

[53] X. Wang, Y. Zhang, H. Zhang, X. Wei, and G. Wang, “Identification and authentication for wireless transmission security based on RF-DNA fingerprint,” *EURASIP J. Wirel. Commun. Netw.*, vol. 2019, no. 1, p. 230, Sep. 2019, doi: 10.1186/s13638-019-1544-8.
[54] G. Vaidya, A. Nambi, T. V. Prabhakar, V. Kumar T, and S. Sudhakara, “IoT-ID: A Novel Device-Specific Identifier Based on Unique Hardware Fingerprints,” in *2020 IEEE/ACM Fifth International Conference on Internet-of-Things Design and Implementation (IoTDI)*, Apr. 2020, pp. 189–202. doi: 10.1109/IoTDI49375.2020.00026.
[55] M. Köse, S. Taşcioğlu, and Z. Telatar, “RF Fingerprinting of IoT Devices Based on Transient Energy Spectrum,” *IEEE Access*, vol. 7, pp. 18715–18726, 2019, doi: 10.1109/ACCESS.2019.2896696.
[56] M. F. Bari, M. R. Chowdhury, B. Chatterjee, and S. Sen, “Detection of Rogue Devices using Unintended Near and Far-field Emanations with Spectral and Temporal Signatures,” in *2022 IEEE/MTT-S International Microwave Symposium - IMS 2022*, Jun. 2022, pp. 591–594. doi: 10.1109/IMS37962.2022.9865347.
[57] N. Soltanieh, Y. Norouzi, Y. Yang, and N. C. Karmakar, “A Review of Radio Frequency Fingerprinting Techniques,” *IEEE J. Radio Freq. Identif.*, vol. 4, no. 3, pp. 222–233, Sep. 2020, doi: 10.1109/JRFID.2020.2968369.
[58] S. Benedetto and P. Poggiolini, “Polarization shift keying: an efficient coherent optical modulation,” in *SBT/IEEE International Symposium on Telecommunications*, Sep. 1990, pp. 14–20. doi: 10.1109/ITS.1990.175565.
[59] S. Benedetto and P. Poggiolini, “Theory of polarization shift keying modulation,” *IEEE Trans. Commun.*, vol. 40, no. 4, pp. 708–721, Apr. 1992, doi: 10.1109/26.141426.
[60] S. Benedetto and P. T. Poggiolini, “Multilevel polarization shift keying: optimum receiver structure and performance evaluation,” *IEEE Trans. Commun.*, vol. 42, no. 234, pp. 1174–1186, Feb. 1994, doi: 10.1109/TCOMM.1994.580226.
[61] Versawave Technologies, “Polarization Modulators.” Accessed: Nov. 24, 2024. [Online]. Available: https://www.versawave.ca/polarization-modulators/
[62] S. Sibecas, C. A. Corral, S. Emami, G. Stratis, and G. Rasor, “Polarization state techniques for wireless communications,” US20040264592A1, Dec. 30, 2004 Accessed: Nov. 24, 2024. [Online]. Available: https://patents.google.com/patent/US20040264592A1/en
[63] L. Arend, R. Sperber, M. Marso, and J. Krause, “Polarization shift keying over satellite - Implementation and demonstration in Ku-band,” in *2014 7th Advanced Satellite Multimedia Systems Conference and the 13th Signal Processing for Space Communications Workshop (ASMS/SPSC)*, Sep. 2014, pp. 165–169. doi: 10.1109/ASMS-SPSC.2014.6934539.
[64] X. Wu, T. G. Pratt, and T. E. Fuja, “Polarization Shift Keying for Wireless Communication,” *IEEE Trans. Wirel. Commun.*, vol. 18, no. 10, pp. 4927–4941, Oct. 2019, doi: 10.1109/TWC.2019.2930973.
[65] W. De Wilde *et al.*, “Authentication by Polarization: A Powerful Anti-Spoofing Method,” presented at the Proceedings of the 31st International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS+ 2018), Sep. 2018, pp. 3643–3658. doi: 10.33012/2018.15917.
[66] E. Ibrahim, R. Nilsson, and J. Van De Beek, “Binary Polarization Shift Keying With Reconfigurable Intelligent Surfaces,” *IEEE Wirel. Commun. Lett.*, vol. 11, no. 5, pp. 908–912, May 2022, doi: 10.1109/LWC.2022.3149618.
[67] E. Basar, M. Di Renzo, J. De Rosny, M. Debbah, M.-S. Alouini, and R. Zhang, “Wireless Communications Through Reconfigurable Intelligent Surfaces,” *IEEE Access*, vol. 7, pp. 116753–116773, 2019, doi: 10.1109/ACCESS.2019.2935192.
[68] H. L. Wang, H. F. Ma, and T. J. Cui, “A Polarization-Modulated Information Metasurface for Encryption Wireless Communications,” *Adv. Sci.*, vol. 9, no. 34, p. 2204333, 2022, doi: 10.1002/advs.202204333.
[69] Z. Ying. Ong, A. Saidatul, and Z. Ibrahim, “Power Spectral Density Analysis for Human EEG-based Biometric Identification,” in *2018 International Conference on Computational Approach in Smart Systems Design and Applications (ICASSDA)*, Aug. 2018, pp. 1–6. doi: 10.1109/ICASSDA.2018.8477604.

[70] W. C. Suski II, M. A. Temple, M. J. Mendenhall, and R. F. Mills, “Using Spectral Fingerprints to Improve Wireless Network Security,” in *IEEE GLOBECOM 2008 - 2008 IEEE Global Telecommunications Conference*, Nov. 2008, pp. 1–5. doi: 10.1109/GLOCOM.2008.ECP.421.
[71] F. Galtier, R. Cayre, G. Auriol, M. Kâaniche, and V. Nicomette, “A PSD-based fingerprinting approach to detect IoT device spoofing,” in *2020 IEEE 25th Pacific Rim International Symposium on Dependable Computing (PRDC)*, Dec. 2020, pp. 40–49. doi: 10.1109/PRDC50213.2020.00015.
[72] W. Tranter, K. Shanmugan, T. Rappaport, and K. Kosbar, *Principles of communication systems simulation with wireless applications*, First. USA: Prentice Hall Press, 2003.
[73] A. Sidibe, G. Loubet, A. Takacs, and D. Dragomirescu, “A Multifunctional Battery-Free Bluetooth Low Energy Wireless Sensor Node Remotely Powered by Electromagnetic Wireless Power Transfer in Far-Field,” *Sensors*, vol. 22, no. 11, Art. no. 11, Jan. 2022, doi: 10.3390/s22114054.
[74] G. Loubet, A. Sidibe, A. Takacs, and D. Dragomirescu, “Battery-free Bluetooth Low Energy Sensing Nodes for Structural Health Monitoring of Concretes,” in *Proceedings of the 13th International Workshop on Structural Health Monitoring*, Destech Publications, Inc., Mar. 2022. doi: 10.12783/shm2021/36247.
[75] P. Welch, “The use of fast Fourier transform for the estimation of power spectra: A method based on time averaging over short, modified periodograms,” *IEEE Trans. Audio Electroacoustics*, vol. 15, no. 2, pp. 70–73, Jun. 1967, doi: 10.1109/TAU.1967.1161901.
[76] Bluetooth SIG, Inc., “Bluetooth Low Energy -It Starts with Advertising.” Accessed: Feb. 04, 2025. [Online]. Available: https://www.bluetooth.com/blog/bluetooth-low-energy-it-starts-with-advertising/
[77] Bluetooth SIG, Inc., “Bluetooth® Core 5.3 Feature Enhancements.” Accessed: Feb. 04, 2025. [Online]. Available: https://www.bluetooth.com/bluetooth-resources/bluetooth-core-5-3-feature-enhancements/
[78] Bluetooth SIG, Inc., “Understanding Reliability in Bluetooth® Technology.” Accessed: Feb. 04, 2025. [Online]. Available: https://www.bluetooth.com/bluetooth-resources/understanding-reliability-in-bluetooth-technology/
[79] Z. A. Abbood, I. Khaleel, and K. Aggarwal, “Challenges and Future Directions for Intrusion Detection Systems Based on AutoML,” *Mesopotamian J. CyberSecurity*, vol. 2021, pp. 16–21, Jan. 2021, doi: 10.58496/MJCS/2021/004.
[80] S. Wang, M. Shi, D. Li, and T. Du, “A Survey of Time Synchronization Algorithms for Wireless Sensor Networks,” in *2019 Chinese Control Conference (CCC)*, Jul. 2019, pp. 6392–6397. doi: 10.23919/ChiCC.2019.8866385.
[81] A. Leva, F. Terraneo, L. Rinaldi, A. V. Papadopoulos, and M. Maggio, “High-Precision Low-Power Wireless Nodes’ Synchronization via Decentralized Control,” *IEEE Trans. Control Syst. Technol.*, vol. 24, no. 4, pp. 1279–1293, Jul. 2016, doi: 10.1109/TCST.2015.2483559.
[82] D. Gao *et al.*, “Time Synchronization Based on Cross-Technology Communication for IoT Networks,” *IEEE Internet Things J.*, vol. 10, no. 22, pp. 19753–19764, Nov. 2023, doi: 10.1109/JIOT.2023.3282202.
[83] L.-A. Phan and T. Kim, “Fast Consensus-Based Time Synchronization Protocol Using Virtual Topology for Wireless Sensor Networks | IEEE Journals & Magazine | IEEE Xplore.” Accessed: Dec. 17, 2024. [Online]. Available: https://ieeexplore.ieee.org/document/9261366
[84] Ettus Research, a National Instruments Brand, “USRP B210 USB Software Defined Radio (SDR).” Accessed: Dec. 18, 2024. [Online]. Available: https://www.ettus.com/all-products/ub210-kit/
[85] Mini-Circuits, “2 Ways Resistive Power Splitter, DC - 10000 MHz, 50Ω | ZX10R-14-S+ | Mini-Circuits.” Accessed: Dec. 18, 2024. [Online]. Available: https://www.minicircuits.com/WebStore/dashboard.html?model=ZX10R-14-S%2B&srsltid=AfmBOorB5fXXI0vyXGhlyFM5EnC4nGcV_Dgc0OHCHBUlQT7TVbI9iUj5
[86] Mouser Electronics, Inc., “EV1HMC647ALP6 Analog Devices | Mouser.” Accessed: Dec. 18, 2024. [Online]. Available: https://www.mouser.fr/ProductDetail/Analog-Devices/EV1HMC647ALP6?qs=ILgNtqsyH23k9B5gU48l3w%3D%3D

[87] C. A. Balanis, *Antenna Theory: Analysis and Design.*, Fourth Edition. John Wiley & Sons, Inc., 2016. [Online]. Available: www.wiley.com/go/antennatheory4e
[88] Tektronix, Inc., “Calculating RF Power from IQ Samples.” Accessed: Mar. 05, 2025. [Online]. Available: https://www.tek.com/en/blog/calculating-rf-power-iq-samples
[89] L. Sanogo, E. Alata, T. E. Djidjekh, G. Loubet, A. Takacs, and D. Dragomirescu, “Polarization Shift Keying for Device Authentication in Wireless Sensor Network,” in *2024 54th European Microwave Conference (EuMC)*, Sep. 2024, pp. 296–299. doi: 10.23919/EuMC61614.2024.10732078.
[90] L. Sanogo, E. Alata, G. Loubet, T. E. Djidjekh, A. Takacs, and D. Dragomirescu, “A Hardware Add-On for IoT Secure Communications: Polarization Shift Keying Smart Antenna System,” in *2024 IEEE 10th World Forum on Internet of Things (WF-IoT)*, Nov. 2024, pp. 810–815. doi: 10.1109/WF-IoT62078.2024.10811243.
[91] L. SANOGO, E. ALATA, G. LOUBET, T. E. DJIDJEKH, A. TAKACS, and D. DRAGOMIRESCU, “Circular Polarization Shift Keying for IoT Communications Security: A Proof of Concept,” in *2024 31st IEEE International Conference on Electronics, Circuits and Systems (ICECS)*, Nov. 2024, pp. 1–4. doi: 10.1109/ICECS61496.2024.10848847.
[92] “Embedded Python Block - GNU Radio.” Accessed: May 10, 2025. [Online]. Available: https://wiki.gnuradio.org/index.php/Embedded_Python_Block
[93] “Creating Your First Block - GNU Radio.” Accessed: May 10, 2025. [Online]. Available: https://wiki.gnuradio.org/index.php?title=Creating_Your_First_Block
[94] Premier Farnell Limited, “DTJ1524D05 XP POWER, Isolated Chassis Mount DC/DC Converter, ITE, Module, 4:1, 15 W, 2 Output, 5 V | Farnell France.” Accessed: Feb. 24, 2025. [Online]. Available: https://fr.farnell.com/en-FR/xp-power/dtj1524d05/dc-dc-converter-2-o-p-15w/dp/3248175
[95] “File transfer using Packet and BPSK - GNU Radio.” Accessed: Aug. 07, 2025. [Online]. Available: https://wiki.gnuradio.org/index.php?title=File_transfer_using_Packet_and_BPSK

Appendix 1: Theoretical Model of PoSK

The Polarization Shift Keying can be presented as a three-signal model:

- $\mathrm{m(t)}$: the signal of the message,
- $\mathrm{s(t)}$: the switching signal,
- $\mathrm{y(t)}$: the signal resulting from PoSK, i.e., the signal resulting from the switching, i.e., the switched signal.

Where t represents time.

Switching according to $s(t)$ the polarization of the radio wave of the message signal $m(t)$ throughout its transmission can be mathematically expressed as a multiplication of these two signals in the time domain.

$$\mathrm{y(t) = m(t).s(t)} \qquad \text{(i).}$$

Since multiplication in the time domain corresponds to convolution in the frequency domain, relation (i) is equivalent to the following relation in the frequency domain:

$$\mathrm{Y(f) = M(f) * S(f)} \qquad \text{(ii).}$$

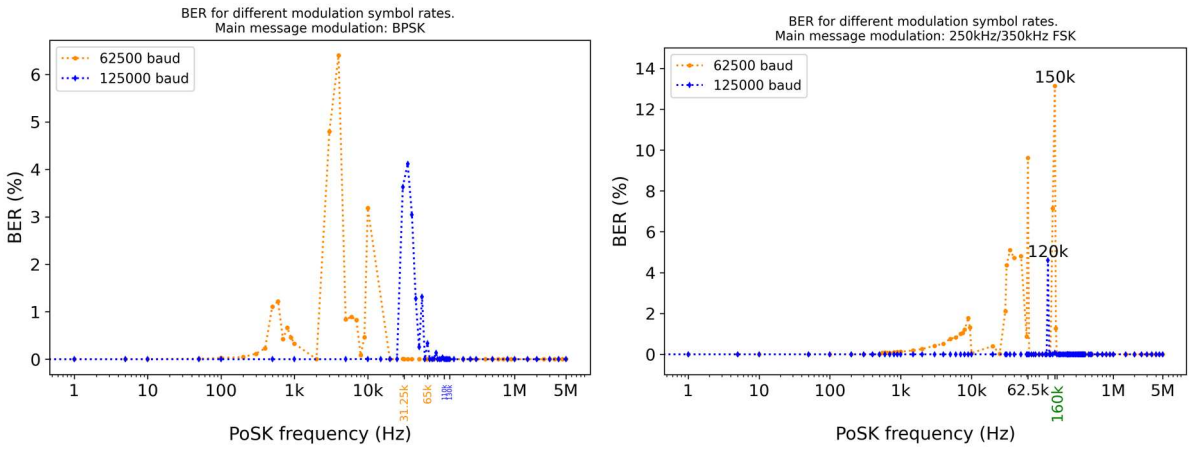
Where f represents the frequency, $\mathrm{M(f)}$ is the spectrum of the message, $\mathrm{S(f)}$ is the spectrum of the switching signal, the symbol $*$ represents the convolution operator, and $\mathrm{Y(f)}$ is the spectrum of the signal resulting from PoSK, i.e., the spectrum of the switched signal.
Relation (ii) means that PoSK corresponds to a spectral convolution.

Appendix 2: BER as a Function of Polarization Keying Frequency

Figure 63 shows the experimental results of the study of the impact of PoSK on the BER. In Figure 63(a), the message m(t) used BPSK modulation, while in Figure 63(b) it used 2-FSK modulation. In both cases, two symbol rates were used: 62,500 baud and 125,000 baud. The choice of these values is empirical; nevertheless, they can be considered realistic values in IoT. The two tones used in 2-FSK are 250 kHz and 350 kHz, giving a Δf of 50 kHz. A periodic square wave signal was used as an experimental cryptographic output model, s(t), to be able to experiment over a wide frequency band (up to 5 MHz) with a manually controlled waveform generator.

Now let's look at the BER profile, which is the same in both cases, BPSK and 2-FSK. This profile can be divided into two spectral zones: the first zone corresponds to the interval]0, F_Z[where F_Z is the frequency at which the BER definitively falls to 0. For example, in the case of BPSK, for R_s = 62,500 baud, F_Z = 65 kHz and for R_s = 125,000 baud, F_Z = 130 kHz. In this zone, the BER is negatively and randomly affected by the PoSK. This is the worst zone for the BER. This is the zone in which the polarization keying frequency F_{PoSK} should not be chosen. The second spectral zone corresponds to the interval [F_Z, 5 MHz]. The BER is zero in this zone, meaning that PoSK has no impact on the transmission quality. This is the zone in which the polarization keying frequency F_{PoSK} must be chosen. This work was limited to 5 MHz, but the BER should remain at zero well beyond this value.

Now let's explain, why BER have this two-zone spectral profile, just like the PER (Packet Error Rate) presented previously in this manuscript.



(a) **(b)**

Figure 63. The BER as a function of polarization keying frequency F_{PoSK}; **(a)** BPSK-modulated message transmission. **(b)** 2-FSK-modulated message transmission.

This two-region profile can be explained by the fact that, when F_{PoSK} is less than the bandwidth of the message spectrum M(f), some of the spectral lines resulting from the convolution of M(f) with S(f) fall within the band where M(f) is located in Y(f). These spectral lines thus become inseparable from M(f) by filtering at reception. This auto-interference degrades the quality of the transmission. On the other hand, when F_{PoSK} is greater than the bandwidth of M(f), none of the spectral lines resulting from the convolution of M(f) with S(f) fall within the band where

M(f) is located in Y(f), because the spectral lines are sufficiently spaced apart that the space between two consecutive lines is large enough to contain the entire M(f). Thus, M(f) alone can be recovered by filtering at the receiver. As a result, PoSK no longer has any impact on the quality of the transmission; the BER falls definitively to zero.
As a reminder: s(t) being a periodic square wave, S(f) is composed of a main power peak at its fundamental frequency and spectral lines (harmonics) at each even (or odd) multiple of the fundamental frequency. The higher the fundamental frequency, the more spaced out the harmonics are.
A Pseudo-Random Binary Sequence (PRBS) is more representative as a cryptographic output model than a periodic square wave. We also simulated the use of a PRBS as s(t) and came to the conclusion that the rule "F_{PoSK} must be greater than the bandwidth of M(f)" remains valid. As a reminder: the spectrum of the PRBS spreads over a bandwidth equal to the fundamental frequency.
The optimal polarization keying frequency is the result of a good trade-off between message bandwidth, the transmission duration of the message, the desired length of the authentication key (which can be shorter or longer depending on the sensitivity of the message content), energy consumption and interference with adjacent channels.