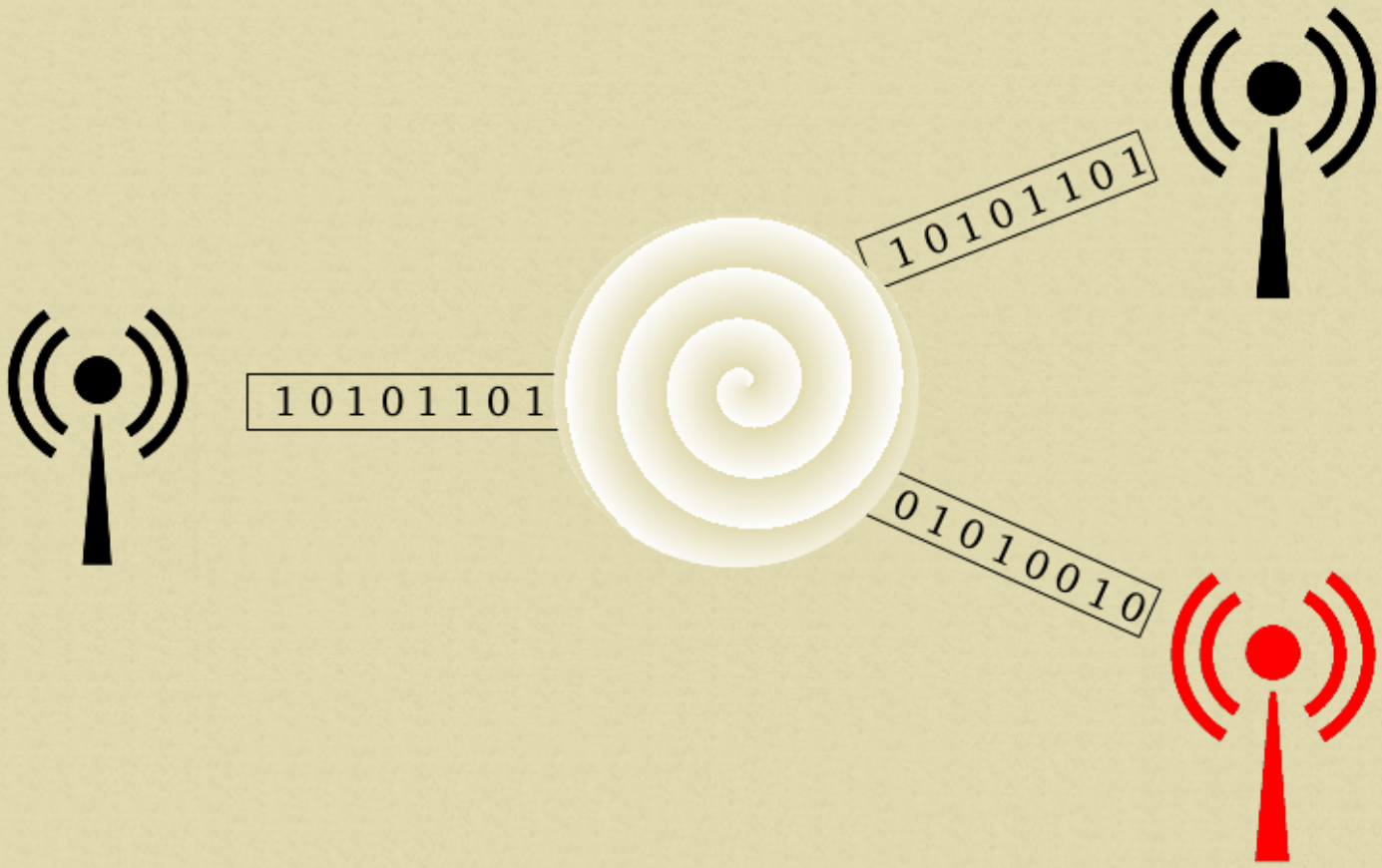


João Diogo Madeira Cristina Marques

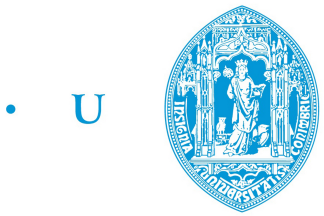
Jamming based on Channel Reciprocity for secrecy in wireless communications

Dissertação submetida para a satisfação parcial dos requisitos do grau de Mestre em Engenharia Eletrotécnica e de Computadores na especialidade de Telecomunicações

Setembro, 2017

· U C ·

Universidade de Coimbra



FCTUC FACULDADE DE CIÊNCIAS
E TECNOLOGIA
UNIVERSIDADE DE COIMBRA

Jamming based on Channel Reciprocity for secrecy in wireless communications

João Diogo Madeira Cristina Marques

Dissertação para obtenção do Grau de Mestre em
Engenharia Electrotécnica e de Computadores

Orientador:	Doutor Marco Alexandre Cravo Gomes
Co-Orientador:	Doutor João Paulo da Silva Machado Garcia Vilela

Júri

Presidente:	Doutora Lúcia Maria dos Reis Albuquerque Martins
Orientador:	Doutor Marco Alexandre Cravo Gomes
Vogal:	Doutor Jorge Nuno de Almeida e Sousa Almada Lobo

Setembro de 2017

There are no limits, except for those we impose upon ourselves.
- Dr. Walter Bishop

A person who never made a mistake never tried anything new.
- Albert Einstein

Acknowledgments

Gostaria de começar por agradecer ao Professor Marco Gomes e ao Professor João Vilela pela oportunidade de participar neste projeto. A orientação e disponibilidade dada pelos dois foi fundamental durante esta etapa e permitiu-me crescer a nível profissional e pessoal.

Aos meus pais e ao meu irmão por quem sinto uma enorme gratidão e por me darem o privilégio de os ter como modelo.

A todos os amigos com quem tive o prazer de percorrer esta importante etapa da minha vida.

A ti Mariana, por todo o apoio e carinho que me dás fazendo com que cada dia seja melhor que o anterior.

A todos,

Muito Obrigado.

Este trabalho é suportado pelo projeto SWING2 (PTDC/EEI-TEL/3684/2014), financiado pelos Fundos Europeus Estruturais e de Investimento (FEEI) através do Programa Operacional Competitividade e Internacionalização - COMPETE 2020 e por Fundos Nacionais através da FCT - Fundação para a Ciência e a Tecnologia no âmbito do projeto POCI-01-0145-FEDER-016753.

Abstract

Wireless communications had an impressive growth in the last decades. Nowadays, most daily tasks require the use of a device that resorts to wireless communications. For these reasons, a necessity to increase security in this type of communications has emerged.

Recently, physical-layer security techniques have raised interest, since they allow complementing, through a different approach, the already existent security protocols in higher logical layers.

In this dissertation, a method of physical layer security is implemented in an Orthogonal Frequency Division Multiplexing (OFDM) system. This method uses the channel characteristics to encode and decode the transmitted message. Taking as foundation the channel reciprocity principle it grants a common source of random information between the legitimate transmitter and receiver. The channel estimations are used to create a jamming signal that, when applied to the transmitted signal, will shift the modulated symbols phase, and, therefore, protect the message.

The OFDM system with the secrecy method was developed using the GNU Radio framework. Also, each module of the built system can be used with Software Defined-Radio (SDR) platforms for testing in a real world environment. Simulation tests were conducted considering a scenario based on the wiretap channel where there are two legitimate users (Alice and Bob) and an illegitimate eavesdropper (Eve); in the original wiretap model Alice sends a message to Bob while Eve tries to intercept it, where it is also assumed that, Bob's reception condition is advantageous over Eve's. The conducted simulations have considered different kinds of channel conditions to determine the effects on the system performance. Also, an actual real world environment experiment with SDR was made.

With the obtained results, it was shown that the employed method can increase security even in a configuration with an illegitimate channel with good conditions. The proposed technique may be important for future networks composed of low capability devices, such as in the Internet of Things, where it may be difficult to implement complex encryption techniques, and where physical-layer security methods may play an important role.

Keywords

Physical-layer Security, OFDM, Channel Reciprocity , Jamming, Secrecy

Resumo

Nas últimas décadas as comunicações sem fios tiveram um crescimento enorme. Hoje em dia, a maioria das tarefas diárias necessitam da utilização de um dispositivo que recorra a uma comunicação sem fios. Por isso, criou-se uma necessidade de aumentar a segurança nestas comunicações.

Recentemente, as técnicas de segurança na camada física têm gerado interesse, pois, permitem complementar, através de uma abordagem diferente, os protocolos de segurança já existente em outras camadas lógicas.

Nesta dissertação foi implementado, um método de segurança na camada física num sistema *Orthogonal Frequency Division Multiplexing* (OFDM). Este método utiliza as características do canal para codificar e descodificar a mensagem transmitida, tendo por base o princípio da reciprocidade do canal que permite obter uma fonte de informação aleatória comum entre transmissor e recetor legítimo. As estimações de canal feitas são utilizadas para criar sinais de interferência que irão mudar a fase dos símbolos modulados e, consequentemente, proteger a mensagem transmitida.

O sistema OFDM com o método de segurança foi desenvolvido utilizando a *framework* GNU Radio. Além disso, cada módulo constituinte do sistema pode ser utilizado com plataformas de rádio definido por software (SDR) para testes em ambiente real. Os testes realizados foram feitos considerando um cenário baseado no canal *wiretap* onde existem dois utilizadores legítimos (Alice e Bob) e um ilegítimo (Eve); no modelo original a Alice envia uma mensagem para o Bob enquanto a Eve tenta intercetá-la, além disso é assumido que as condições de transmissão do Bob são mais vantajosas do que as da Eve. As simulações realizadas consideraram diferentes condições de canal de forma a determinar os efeitos no desempenho do sistema. Além disso, foi realizada uma experiência em condições reais com SDR para validar a implementação.

Os resultados obtidos, constituem evidência que o método usado pode aumentar a segurança mesmo quando o canal ilegítimo tem boas condições. Isto pode ser importante nas futuras redes de dispositivos de baixa complexidade, como na Internet das coisas, onde é difícil a implementação de técnicas complexas de encriptação, e onde os métodos de segurança na camada física podem ter um papel importante.

Keywords

Physical-layer Security, OFDM, Channel Reciprocity , Jamming, Secrecy

Contents

List of Figures

List of Tables

List of Acronyms

AWGN Additive White Gaussian noise

BPSK Binary Phase Shift Key

C_s Secrecy Capacity

CP Cyclic Prefix

CRC Cyclic Redundancy Check

CS Cyclic Suffix

DAB Digital Audio Broadcasting

DFT Discrete Fourier Transform

DVB Digital Video Broadcasting

FFT Fast Fourier Transform

GRC GNU Radio Companion

ICI Inter-Carrier Interference

IDFT Inverse Discrete Fourier Transform

IFFT Inverse Fast Fourier Transform

ISI Inter Symbol Interference

JBRC Jamming Based on Channel Reciprocity

LSB Least Significant Bit

MMSE Minimum Mean Squared Error

OFDM Orthogonal Frequency Division Multiplexing

PDP Power Delay Profile

PLS Physical-layer Security

PMT Polymorphic type

QAM Quadrature Amplitude Modulation

QPSK Quadrature Phase Shift Key

RC Raised Cosine

RFID Radio-Frequency Identification

SDR Software-Defined Radio

SNR Signal-to-noise Ratio

SR Secrecy Ratio

TDD Time Division Duplexing

USRP Universal Software Radio Peripheral

WLAN Wireless Local Area Network

WMAN Wireless Metropolitan Area Network

ZF Zero Forcing

ZP Zero Padding

1

Introduction

This chapter introduces the topic explored, the motives that led to the development of this work, the tools used, sets the main goals to be accomplished and the main contributions.

1.1 Motivation and objectives

In the current days, wireless communications are an indispensable part of the daily routine. From military to civil applications wireless networks are a necessity that does not cease to grow. Cellular communication alone is accessible to 5 billion people, and this is only one branch of the several that have appeared in the last decades [1].

With the use of wireless systems in applications that handle personal and financial information, such as online banking or health services, secrecy became a relevant and important topic. Security has normally been implemented at the highest logical layers of the network, however, in some cases, the use of data encryption is hard due to issues of key management or with the high complexity that this process requires, for example in sensor or Radio-Frequency Identification (RFID) networks that are constituted by devices of very low capability [1].

For these reasons, Physical-layer security (PLS) has gained much attention lately, because it complements the existent security methods implemented in higher-logical layers through a different approach, since it aims to obtain secrecy through the use of physical layer characteristics [2] [3].

In [4], it is proposed a method to increase secrecy in a wireless system using uncorrelated channel estimations as the common random source between two legitimate users. A jamming signal is generated accordingly to the channel estimation phase and applied to the transmitted signal. This method takes as foundation the channel reciprocity principle which states that the effects applied to the signal will be equal downlink and uplink [5] [6].

The method fits well with the use of Orthogonal Frequency Division Multiplexing (OFDM) to perform channel estimation. This dissertation addresses this study through simulation, as well as, provides an experimental setup on Software-Defined Radio (SDR) platforms, for proof of concept. Furthermore, OFDM is the preferred multicarrier modulation technique for wireless communications systems due to the low computational complexity of the implementation, which raises the interest of PLS techniques applied to OFDM.

During the course of this dissertation PLS is taken from a functional perspective. The main goal is the development of a transmission system for real world communications with a method of PLS implemented. The security scheme implemented will use the channel characteristics as a common random source between a legitimate transmitter

and receiver to generate jamming signals, which will be applied during the transmission and eliminated on the receiver, and, therefore, protecting the signal from any possible eavesdropper while it is airborne.

1.2 Software & Hardware Framework

In order to accomplish the work on this dissertation, several software frameworks had to be learned, such as GNU Radio, and the specificities required for the programming of SDR testbeds. The software and hardware used in this dissertation is introduced in this section.

GNU Radio [7]:

It is a open source development framework that provides digital signal processing tools and allows implementing them in software defined radios (SDR). This framework provides a graphical interface, GNU Radio Companion [8], that allows programing through blocks. The development of applications in this framework can be done in C++ and Python. Even though this is a tool of election in the radio community, the documentation of some functions can be scarce which can bring difficulties to new users.

This framework was used on this dissertation for development, simulation and to deploy the system in a real world environment using SDR boards.

MATLAB [9]:

It is a widely used programing language in the scientific medium. Developed by Math Works, it allows programming highly complex mathematical algorithms in a simplified way compared to other programming languages. Besides that, it allows interfacing with other programing languages, such as Python and C++. Being such a functional environment for numerical calculations it was used to calculate the metrics and create graphics with the obtained results.

In the first stage of the work development this framework was also used in the programming of SDR, through the graphical editor Simulink which is integrated with MATLAB, however, due to a more versatility concerning SDR it was decided to change into the GNU Radio framework.

Universal Software Radio Peripheral [10]:

The Universal Software Radio Peripheral, USRP, is a software-defined radio platform created by Ettus Research. This kind of boards allows processing radio signals using software instead of electronics, which makes implementation much more flexible.

In this dissertation it was used 3 USRP's B210 with and antena VERT2450, which have the following specifications [11]:

USRP B210

- 2 Tx & 2RX, Half or Full Duplex;
- RF Coverage from 70 MHz to 6 GHz;
- USB 3.0 connectivity;
- Fully Capable 2x2 MIMO;
- Up to 56 MHz of bandwidth in 1x1.

Antenna VERT 2450

- Omnidirectional;
- Two bands of frequency : 2.4 to 2.48 and 4.9 to 5.9 GHz.

1.3 Main contributions

Taking into consideration this dissertation objectives and the developed worked it is possible to resume the main contributions as the following:

- Development of an OFDM system in simulation environment with Time Division Duplexing (TDD) in GNU Radio Companion;
- Development of a PLS method in GNU Radio, which uses jamming signals based on channel reciprocity;
- Implementation of a PLS method in an OFDM system with SDR;
- Analysis of the simulated OFDM system with the PLS method in terms of secrecy capacity under different channel conditions;
- Real world tests , using SDR platforms for proof of concept of the PLS method.

The obtained results of this work are a considerable contribution to the research of PLS techniques in the SDR field, which are two scientific areas with great interest in the modern days.

1.4 Dissertation outline

The remainder of this dissertation is organized as follows,

Chapter 2 - Preliminary concepts :

It introduces the fundamentals of OFDM, the channel estimation techniques used in this type of systems. It is also presented the concept of physical security, the wire-tap model, all the metrics used to evaluate secrecy in communications and the jamming method based on channel reciprocity.

Chapter 3 - Implementation of an OFDM system with jamming based on channel reciprocity:

Gives a detailed explanation of the used modules on the development of an OFDM system using a jamming method based on channel reciprocity implemented in GNU radio. It is also presented the modus operandi of the system.

Chapter 4 - Experimental results:

Presents the obtained results for the developed system in simulation environment, where the channel model parameters were varied to determine in what conditions this system would be more secure. Also, a real world experiment with USRP boards was conducted to determine the system viability.

Chapter 5 - Conclusion :

Presents the main conclusions drawn from this dissertation and suggests some future research lines.

2

Preliminary Concepts

In this chapter it is made an overview of background concepts important for the developed work comprehension. First ir is an explanation of the theory behind OFDM which will be relevant to understand the system presented in sections 3.1 and 3.2. Also, a brief explanation on PLS allows for a better understanding on the developed system and nomenclature through out this dissertation.

2.1 OFDM concepts

(OFDM) is a form of multi-carrier modulation where a high rate data stream is divided into N lower rate streams that are transmitted simultaneously and overlapped by guaranteeing orthogonality between transmitted streams [12]. This method has been adopted by several wireless standards such as Wireless Metropolitan Area Network (WMAN), Wireless Local Area Network (WLAN), Digital Audio Broadcasting (DAB) and Digital Video Broadcasting (DVB) [13], because it makes the communication system more resilient to frequency selective fading and narrowband interference, which are typical impairments that occur in wireless transmissions [14]. Additionally, the overlap of sub-channels leads to a considerable saving of the available spectrum, in comparison with classical multi-carrier transmission systems employing frequency division multiplexing where there is no overlap between sub-channels, Figure 2.1 illustrates this comparison.

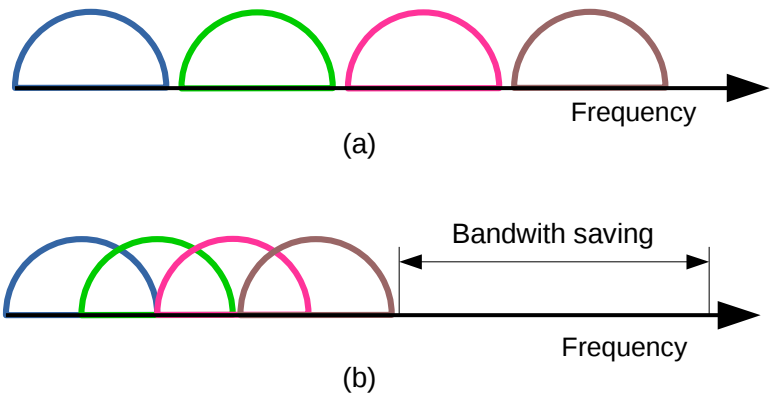


Figure 2.1: Bandwith used in (a) conventional multi-carrier technique that uses frequency multiplexing and in (b) OFDM

In order to illustrate an OFDM modulator it is presented Figure 2.2, the concept behind each of these blocks will be explained in section 2.1.1.

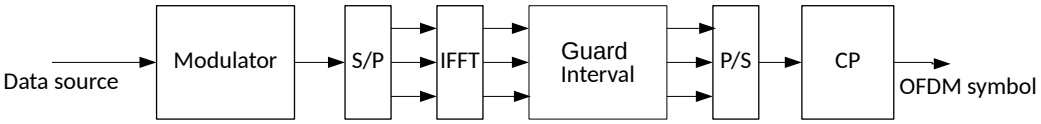


Figure 2.2: Block diagram of an OFDM modulator

2.1.1 OFDM Definition

An OFDM symbol is composed of several sub-carriers that are orthogonally overlapped in order to avoid Inter-Carrier Interference (ICI) [12]. Each of these N sub-carriers will contain a stream of complex symbols $S_k : \{k = 0, 1, ..., N-1\}$. S_k is obtained by mapping the initial data to a selected M-ary constellation, e.g., Quadrature Phase Shift Key(QPSK) or M-ary Quadrature Amplitude Modulation (M-QAM), at a specific rate $\frac{1}{T_s}$, where T_s is the M-ary modulated symbol duration time. Let s_n be the OFDM baseband signal representation in the discrete time domain, which can be written as

$$s_n = s(nT_s) = \sum_{k=0}^{N-1} S_k e^{j2\pi f_k nT_s} \quad , \quad 0 \leq nT_s \leq T_{sym} \tag{2.1}$$

where $T_{sym} = NT_s$ stands for the OFDM symbol period and $f_k = k/T_{symbol}$, represents the k^{th} sub-carrier in the OFDM signal.

Observe that equation 2.1 can be recognized as the Inverse Discrete Fourier Transform (IDFT), which means that

$$s_n = IDFT\{S_k\}. \tag{2.2}$$

The previous equation shows that an OFDM signal can be built using an IDFT and demodulated using a Discrete Fourier Transform (DFT). Both DFT and IDFT can be implemented with the low complexity Fast Fourier Transform (FFT) when N is a power of 2 [15]. For this reason and due to the fact that this method allows the implementation of simple equalization techniques on the frequency domain, such as Zero Forcing ZF, OFDM got very popular in modern communications [16].

Periodic discrete time signals are defined as orthogonal with each other if the sum of their product during a period is zero. In the of OFDM carriers signals this can be shown by looking at the discrete time domain exponential signal $c_k = \left\{e^{j2\pi f_k nT_s}\right\}_{k=0}^{N-1}$ where $0 \leq nT_s \leq T_{symbol}$ and $n = 0, 1, ..., i, ..., N-1$; in order to be orthogonal they must verify the following [14],

$$\frac{1}{N}\sum_{k=0}^{N-1} c_k c_k^* = \frac{1}{N}\sum_{n=0}^{N-1} e^{j2\pi\frac{k}{T_{sym}}nT_s} e^{-j2\pi\frac{i}{T_{sym}}nT_s} = \\ = \frac{1}{N}\sum_{n=0}^{N-1} e^{j2\pi\frac{k}{NT_s}nT_s} e^{-j2\pi\frac{i}{NT_s}nT_s} = \frac{1}{N}\sum_{n=0}^{N-1} e^{j2\pi\frac{k-i}{N}n} = \begin{cases} 1, k = i \\ 0, k \neq i \end{cases} \tag{2.3}$$

Equation 2.3 shows what must be verified so that different sub-carriers like k and i will not suffer from ICI.

2.1.2 Guard Interval

One of the main issues regarding wireless communications over time-dispersive channels is the possibility to receive delayed replicas of the transmitted signal, this phenomenon is designated as multipath effect [17]. These copies occur when the transmitted signal encounters obstacles, like buildings or trees, which may reflect, absorb or scatter the original signal and, consequently, creating delayed copies with different gains. On reception, these replicas can lead to Inter-Symbol Interference (ISI) as result of the overlap between the actual symbol tail with the delayed replica beginning, figure 2.3 illustrates this phenomenon.

In order to deal with delay spreads and the resulting effects , OFDM communications systems resort to guard bands that are appended to the OFDM symbols. There are three types of guard bands: cyclic prefix (CP) , cyclic suffix (CS) and Zero Padding (ZP) [18].

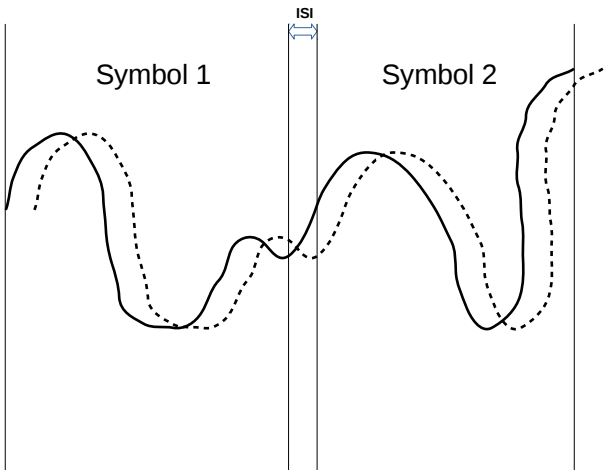


Figure 2.3: Representation of ISI due to multipath delay

2.1.2.A Cyclic Prefix

The CP scheme is an OFDM symbol cyclic extension where a copy of the tail bits is appended at the symbol beginning. This provides a certain periodicity to the symbol which, consequently, will allow to turn the linear convolution inherent to the reception into a circular convolution. Also, it ensures that there is always going to be a complete OFDM symbol in the FFT window. This creates an easy method to eliminate the consequences of ISI and ICI by keeping the orthogonality.

One problem with these methods is the waste of resources, since it increases the OFDM symbol length with non-useful data which, consequently, diminishes the effective transmission rate. Figure 2.4 represents an OFDM symbol with CP, where T'_{sym} is the OFDM symbol length at the transmitter output

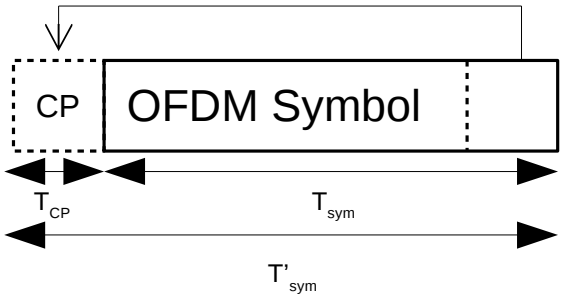


Figure 2.4: Representation of an OFDM symbol with CP

2.1.2.B Zero-Padding

The ZP scheme consists on appending a set of zeros at the OFDM symbol ending. By adding this group of null samples there is a component of redundancy introduced. However, if the subcarriers are affected by different delays then the effectiveness of this method is compromised, because ICI is generated and the orthogonality between subcarriers is lost. For these reasons, the CP method is preferable since it allows the elimination of ICI.

2.1.3 Equalization

When a signal, X_k, goes through a real channel, H_k, it suffers distortion from certain factors introduced by the channel. In order to mitigate these effects it is necessary to do an equalization of the signal.

Taking in consideration what was mentioned in Section 2.1.2.A, as long as the cyclic prefix length is larger than the channel impulse response after the FFT, i.e. in the frequency domain, then the linear convolution can be converted into a circular convolution, and, therefore, the received signal is given by Equation 2.4 , where N_k is AWGN.

$$Y_k = H_k X_k + N_k \tag{2.4}$$

There are two main methods to achieve signal equalization, Zero Forcing (ZF) [16] and the Minimum Mean Squared Error (MMSE) [18].

The ZF method takes an estimated signal, F_k, where $F_k = H_k^{-1}$ and applies it into the received signal, as shown in Equation 2.5

$$\begin{aligned} \tilde{X}_k &= F_k Y_k \\ &= F_k H_k X_k + F_k N_k \\ &= H_k^{-1} H_k X_k + H_k^{-1} N_k \end{aligned} \tag{2.5}$$

This method is popular since it has an implementation of low complexity. However, for carriers where the estimation signal, F_k, has low magnitude, i.e carriers in deep fading, then the factor $H_k^{-1} N_k$ increases the noise power. This problem can be avoided through the use of MMSE, since F_k is given by Equation 2.6.

$$F_k = \frac{H_k^{-1}}{\frac{1}{SNR} + 1} \tag{2.6}$$

Even though, it can decrease the noise influence in the received signal, the implementation of this method is complex, therefore, ZF is the most common and the one which was implemented in this dissertation.

2.1.4 OFDM Systems Channel Estimation

On a real world communication system the transmission channel is not ideal, which means that when a signal is sent through the air it is affected by phenomenons that will decrease the system performance, e.g. multi-path fading, noise and frequency offset. Therefore, to minimize these effects on reception, it is crucial to estimate the channel and perform an equalization. The channel estimation techniques can be classified as:

- **Blind channel estimation**: in these the channel state estimate is obtained without any knowledge of the transmitted signal [13] ;

- **Data-aided channel estimation**: these techniques require that the transmitted signal carries known information to determine the channel response [13] .

The blind estimation techniques do not need any known reference (training) signal, which means that there is more bandwidth available for useful information. However, since there are not any training signals, this technique will require to collect a lot of data, because this method exploits the received signal statistic behavior [19]. On the other hand, data-aided channel estimation techniques need known information which reduces the amount of actual data that can be transmitted but it allows to perform a quick and reliable estimation by comparing the received and training signal.

In this dissertation the focus will be on the data-aided channel estimation because of the speed and reliability. Amongst these, the two most common techniques are the channel estimation with training symbols, where OFDM symbols are filled with known information or useful data and pilot-aided channel estimation, in which the payload is sent on data sub-carriers and known information transmitted on pilot subcarriers, both these methods are illustrated in Figure 2.5. This allows to send both useful and known data on the same OFDM symbol.

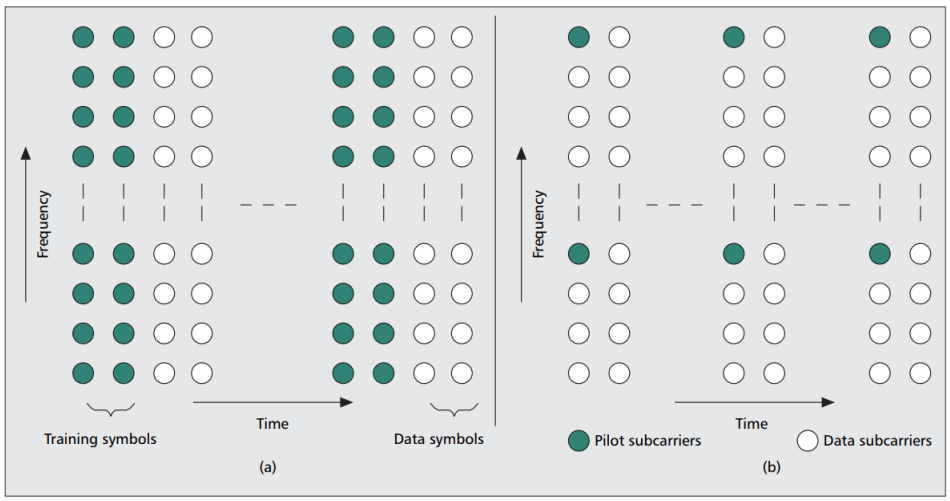


Figure 2.5: OFDM symbols structure using training symbols (a) and pilot subcarriers (b) [13]

The implementation done in this dissertation uses a channel estimation technique with training symbols, which will be presented in the next chapter.

2.2 Physical Layer Security

Nowadays, secrecy in wireless communications is a crucial issue considering that all kinds of sensible data are handled by modern wireless networks, e.g. personal, financial or even medical information. One of the main issues inherent to this type of communications is the shared medium that is used; the open nature of this resource allows for anyone within range of a vulnerable network to capture or record the digital traffic.

Security has traditionally been achieved in wireless communications through the use of encryption protocols implemented at the higher logical layers [20]. These methods relay on cryptography to hide the message, however in certain network architectures where the end devices are of low complexity, as in the envisioned Internet of Things, the employment of data encryption is a difficult task to accomplish. Also, a message encoded with these protocols can be clearly intercepted at the physical layer level, which means that the message secrecy will only last while the used cypher is not broken. For these reasons there has been much interest in developing physical layer security methods, since they use the channel characteristics to encode the message, and therefore provide an additional layer of security to the network [1] [21].

2.2.1 General Structure Conditions

The system model that is going to be used in this dissertation is know as the *wiretap channel* model, that was introduced in 1975 by Wyner [22] and it is presented in figure 2.6. This model is composed of two legitimate users, *Alice* and *Bob*, and an eavesdropper *Eve*. Alice's objective is to send a certain message M to Bob, while Eve will be trying to intercept Alice's signal. It is important to notice that Eve's channel is different from the main channel, since, for these two channels to be identical then Eve would have to be in the exact same location as Bob. Therefore, the signal received by Bob is given by

$$Y_B^n = H_B^n X^n. \tag{2.7}$$

Meanwhile Eve will receive Y_E^n, which is given by Equation 2.8

$$Y_E^n = H_E^n X^n \tag{2.8}$$

where n represents the time-slot channel index [4].

In this scenario, Bob and Eve will have the same instruments to decode the received message, which means that the difference between these two will be the channels H_B^n and H_E^n, and the different impairments inherent to each of them.

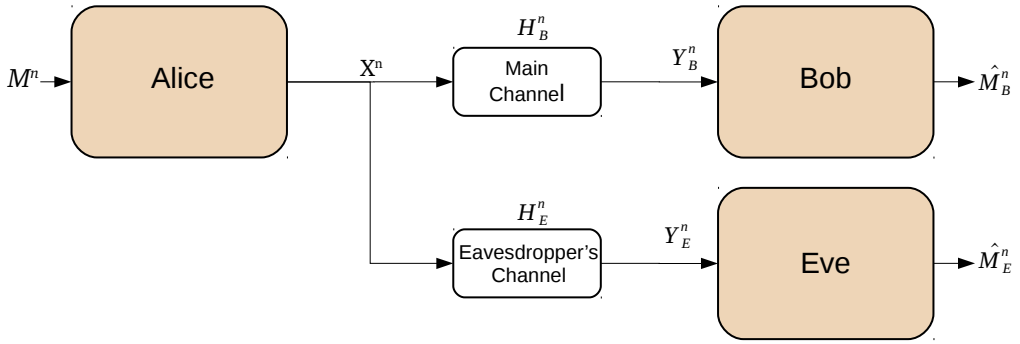


Figure 2.6: The wiretap channel

2.2.2 Security metrics

In order to evaluate the secrecy level of a certain system, a metric is necessary. In 1949, Shannon introduced the concept of *perfect secrecy* [23], where a code is considered safe if the mutual information, between the message M and the encoded signal X^n is zero, i.e., $I(M;X^n) = 0$. However, Shannon also concluded that to achieve perfect secrecy the key would have to be of the same size as M, which makes it highly restrictive [24] [25].

Wyner proposed in 1975 the secrecy metric know as *weak secrecy*. In this case it is not necessary for X^n to be free from any information present in M, however, it is required that the mutual information between M and the received signal by Eve, Y_E^n, is small enough so that a factor $\frac{1}{n}$ can turn it into zero as n approach's infinity, i.e., $\lim_{n\to\infty} \frac{1}{n} I(M;Y_E^n)$ [24].

These security methods ensure secrecy accordingly to information theory, however they are hard to implement which makes them impracticable when applied to realistic channel models [24]. Therefore, the metric used in this dissertation is the *secrecy capacity*, C_s, that will evaluate the rate between useful data received in a certain time interval by Bob (reliability) and the amount that was lost by Eve in the same time (security). This metric can be formulated as shown in the following equation

$$C_s = I(M^n, \hat{M}_B^n) - I(M^n, \hat{M}_E^n) \tag{2.9}$$

where $I(M^n, \hat{M}_B^n)$ and $I(M^n, \hat{M}_E^n)$ stand for the mutual information between the sent message and the messages received by Bob and Eve, respectively. Generally speaking, this metric allows knowing the amount of information contained in one variable, e.g. the received message by Bob, $\hat{M}_B^n$, by comparison with another variable, in this case the sent message M. Obviously, in order to have a good quality and secure system it is necessary to maximize the value of $I(M^n, \hat{M}_B^n)$ and minimize $I(M^n, \hat{M}_E^n)$ [4] [1].

2.2.3 Jamming Based on the Reciprocal Channel

In a wireless communication system, knowing the channel state information can be really advantageous since it allows the transmitter to adapt his message to the channel. This adaptation can be used for multiple purposes such as, improving the transmission quality through equalization or applying jamming signals to the transmitted message signal in order to increase the system secrecy.

In this dissertation the focus will be on a security scheme that employs this information as a common random source between Alice and Bob to generate jamming signals that will be added to the outgoing data by Alice and canceled by Bob on reception. However, since the legitimate and illegitimate channels are different, $H_B^n \neq H_E^n$, Eve will be affected by another jamming signal which will then make her unable to properly decode the message [4].

This method uses the channel reciprocity principle [4] which states that electromagnetic waves traveling in both directions will undergo the same physical perturbations, i.e. reflections, diffractions and refractions. For this reason, when the link operates at the same frequency it is safe to assume that the impulse response of the channel between any pair of antennas should be equal despite of the direction [5].

Considering the notation used in Figure 2.6 the jammed transmitted signal can be written as

$$X^n = M^n \theta_B^m \tag{2.10}$$

where θ_B^m is a function of the reciprocal channel H_B^n phase, being $m \neq n$ because m represents a time-slot channel index previous to n. Meanwhile the received signals at Bob and Eve are given by equations 2.11 and 2.12 respectively. Also, W_B^n and W_E^n designate the noise added by Bob's channel and Eve's, respectively.

$$Y_B^n = M^n \theta_B^m H_B^n + W_B^n \tag{2.11}$$

$$Y_E^n = M^n \theta_B^m H_E^n + W_E^n \tag{2.12}$$

The jamming signals θ^m will have four possible values as shown in Table 2.1, and they are determined by the channel estimate argument quadrant, i.e. if the argument is within the first quadrant then $\theta^m = 0$ radians, this is illustrated in Figure 2.7.

Since Bob's channel estimate is assumed to be equal regardless of being uplink or downlink, then he can easily cancel the jamming effect on the received signal by multiplying by the inverse of θ^m. However, in Eve's case the channel estimate will most likely

Channel estimation argument	θ^m
$0 \leq arg < \frac{\pi}{2}$	0
$\frac{\pi}{2} \leq arg < \pi$	$\frac{\pi}{2}$
$\pi \leq arg < \frac{3\pi}{2}$	π
$\frac{3\pi}{2} \leq arg < 2\pi$	$\frac{3\pi}{2}$

Table 2.1: Jamming signal mapping

be different and, therefore it will not be possible to decode the message as easily, since for each received symbol there will be 3 other wrong possibilities.

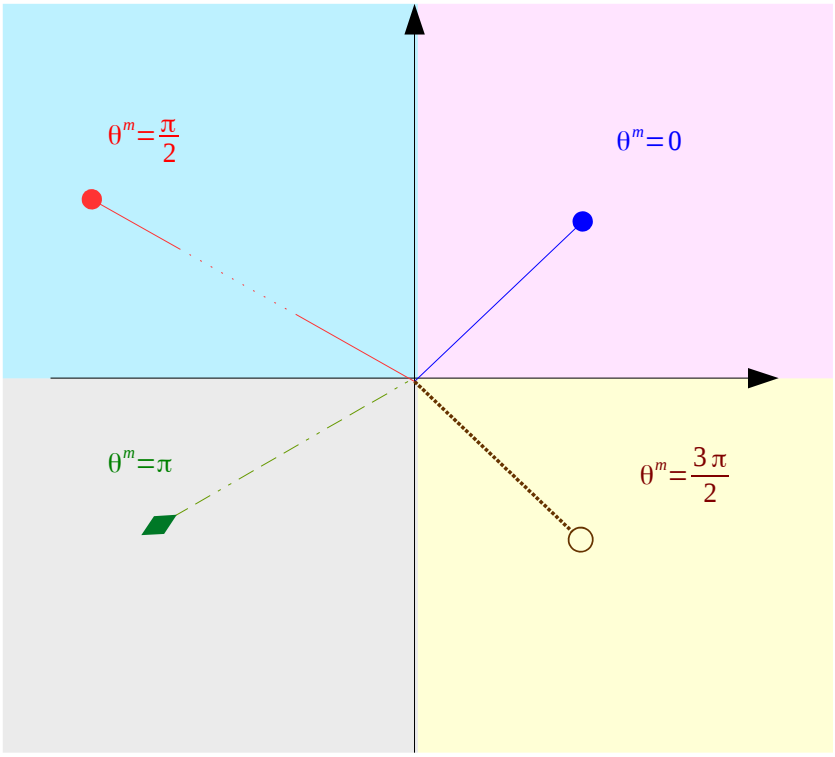


Figure 2.7: Possible values of θ^m considering different channel estimations represented by complex symbols with different colors and styles

In this dissertation, it is implemented an OFDM transmitter and receiver with a PLS method, which is evaluated through simulation and a real world environment experiment. The next chapter presents the developed work and the used modules description.

2.2.4 Other PLS approaches for secrecy

The JBRC method aims to increase the system secrecy by using the channel characteristics. As it was stated before, PLS is raising much interest, and, therefore, there are other methods to increase secrecy in wireless systems, such as techniques resorting to scramblers [26], using artificial noise [27] [28] or frequency hopping [29].

In [30] [31] it is proposed a PLS method a coding scheme based on interleaving with systematic channel codes. In this method a random key is generated and used to shuffle the message at the origin. The receiver will only be able to decode the message correctly if the SNR is above a certain level [30].

The interleaving and JBRC methods are both PLS techniques that allow an increase of secrecy in wireless communications, and, even though they both use the channel characteristics to protect the message this usage is different, since in the interleaving case the channel conditions, SNR, determine when it is possible do receive the message correctly. In the JBRC it is not used in the same way, since in this case the channel conditions can be understood as the key generator, and, therefore the main goal in this case is to have a different legitimate and illegitimate channel and not necessarily better.

3

Implementation of an OFDM system with jamming based on channel reciprocity

This chapter presents the implemented communications system in detail. In section 3.1 the OFDM transmitter developed in GNU Radio is split into stages and each stage is analyzed. Then, the OFDM receiver suffers a similar analysis in section 3.2. Finally, the developed model with the PLS method is described in section 3.3. Note that the nomenclature used in the previous chapter was also used in the implementation description.

3.1 GNU Radio OFDM Transmitter

The OFDM Transmitter [32] is built with blocks from the GNU Radio Companion (GRC) library taking as reference [33] [14]. In Figure 3.1 it is presented the flow-graph of this transmitter.

It begins with a stream of bytes, to which a stream tag is attached. The concept of stream tags will be presented in the following subsection.

3.1.1 Stream Tags

Stream tags [34] are isosynchronous mechanisms for data flow control developed by the GNU Radio project to create boundaries, which allows introducing the concept of packets. These components propagate data in parallel to the main stream by pairing the tag data with a designated item. Each tag has four components:

- The *key*, which has information on what the tag represents; for example in this case it has the value *packet len* which represents, as the name indicates, the packet length in items;
- The *value*, which is of a Polymorphic type (PMT) and therefore can handle any kind of data that the user intends to send downstream; in this transmitter it is set to 96 items;
- The *offset*, that represents the position on the stream of the item to whom the tag is attached;
- The *srcid* parameter that identifies in which block the tag was added.

Figure 3.2 illustrates the working procedure associated with a tagged stream [35]. In this illustration *Block 1* attaches a tag to the fourth byte of a stream, this stream of bytes will then pass through the *Dec 0* which will decimate the stream by a factor of 2, this allows understanding how tags behave when there is a change on the stream rate.

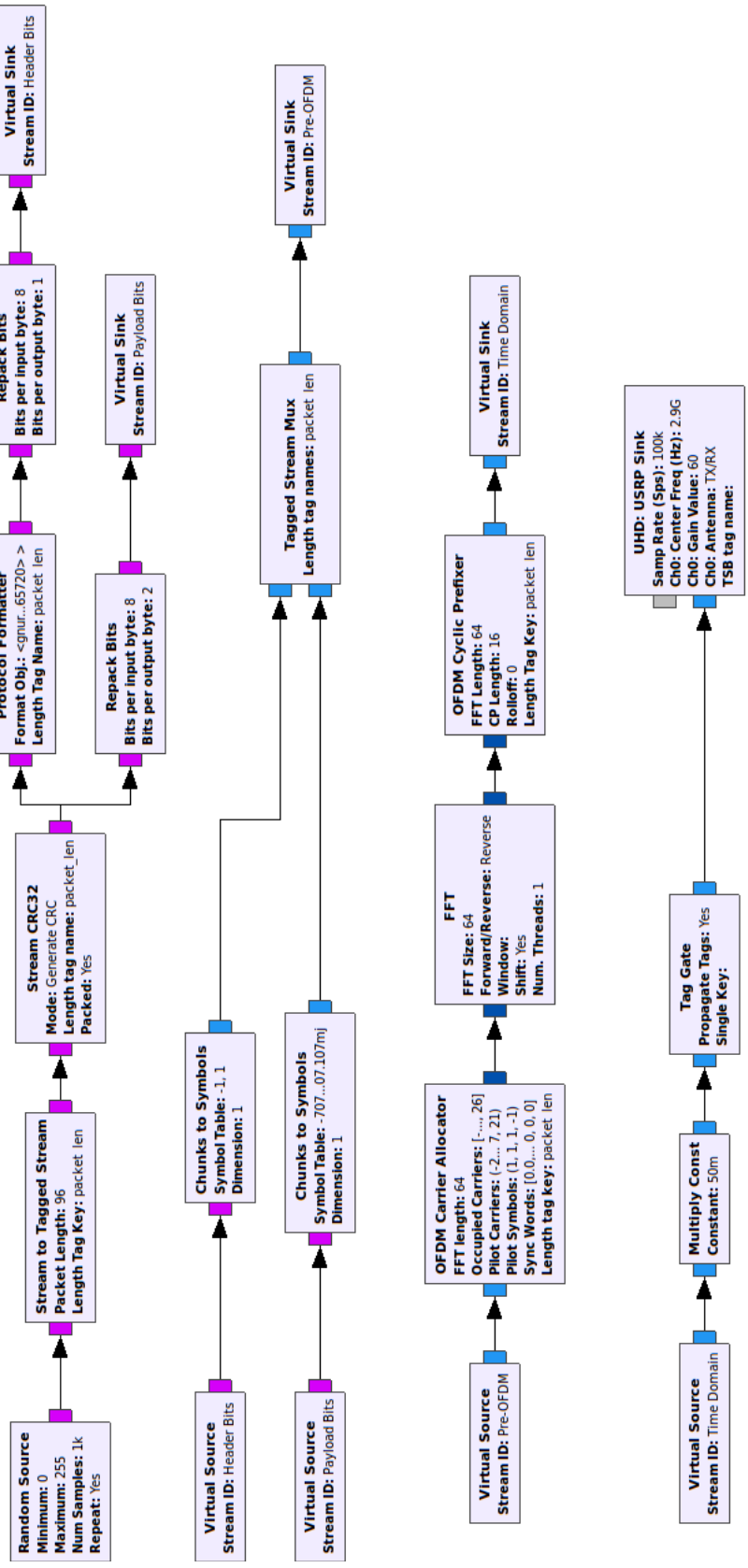


Figure 3.1: OFDM Transmitter GRC flow-graph

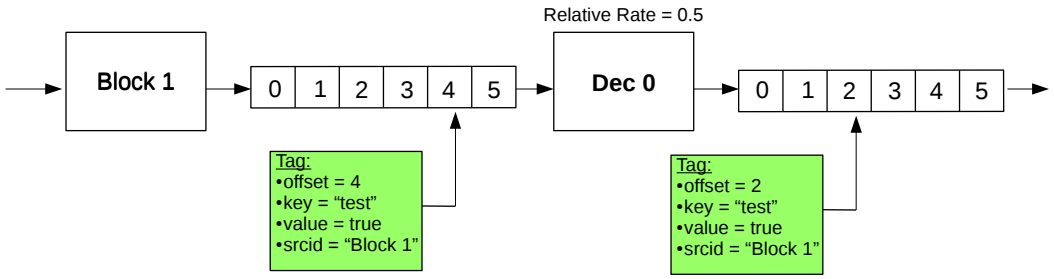


Figure 3.2: Stream tag procedure

Accordingly to Figure 3.1, now the stream is composed of bytes with a tag every 96 items, which can be understood as packets of bytes. The next step is to add a Cyclic Redundancy Check (CRC) for error detection to each packet, which will be presented in section 3.1.2.

3.1.2 Cyclic Redundancy Check

A CRC is an error-detecting code usually employed in digital communications to check for errors in the transmitted data [36]. This type of code is popular due to his simple implementation and because of his capability to detect both random and bursts of errors. A device using CRC will calculate a short fixed-length binary sequence, for every packet to be sent and will append it to the data, forming a *codeword*. On reception the device can conduct a CRC on the entire *codeword* and compare the result with a known constant, or alternatively, it may compare the check value with one newly calculated from the data packet. If the check values do not match, then some errors occurred during the transmission.

In the ”Stream CRC32” block it was implemented a CRC-32 accordingly to [37], which means that the check value is composed of 32 bits, or 4 bytes. Therefore, after appending the 4 bytes sequence, the parameter *value* of the corresponding tag will be updated to the new length, which is 100 bytes.

At this point, like it is showed in Figure 3.1, the tagged stream is split into the header and payload branch. The *modus operandi* of the header branch will be presented first.

3.1.3 Header branch

In this section all the relevant parts of the header procedure are presented, this is shown in Figure 3.3.

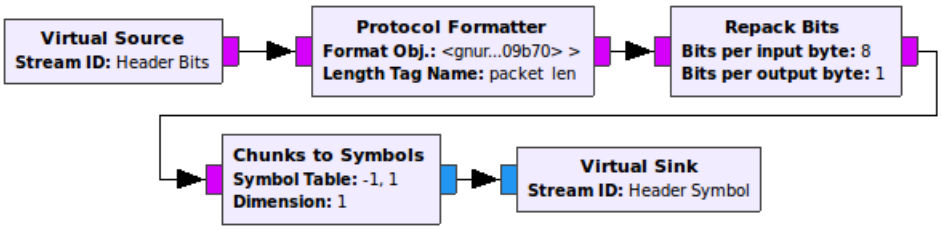


Figure 3.3: Header branch from the GNU Radio transmitter flowgraph

The *Protocol Formatter* block [38] starts the header branch procedure. It is responsible to select information in each tag from the main stream and format this into the structure showed in Figure 3.4.

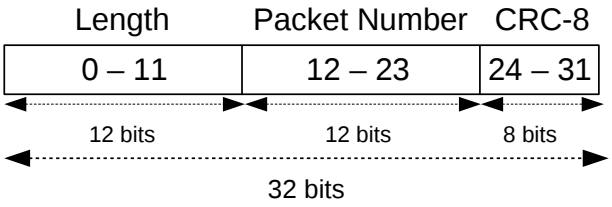


Figure 3.4: Implemented Header Structure

Taking into consideration the structure presented in Figure 3.4 it was expected a packet of 4 bytes after this block, however, the actual size is 6 bytes, that is because the block adds 2 bytes filled with zeros, this is done guarantee that the CRC works properly [38].

In order to apply a digital modulation to the tagged stream it is necessary to change the content in each byte to a binary form, the block *Repack Bits* is here for that purpose. In this specific case, it will read 8 bits from every input byte and write onto 1 bit of the output byte stream; the reading and writing both begin on the Least Significant Bit (LSB) [39]. A graphical explanation can be found in Figure 3.5.

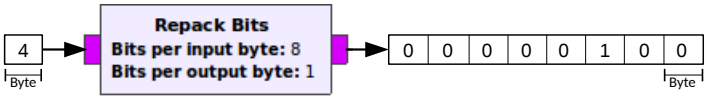


Figure 3.5: *Repack Bits* block example

Obviously, the packet length will change. Since there is an interpolation [40] of 8 the size will be 48 bytes after this reshape. Now the tagged stream items are ready to be digital modulated.

The *Chunks to Symbols* block is a symbol mapper, it will modulate the input byte stream into complex symbols mapped accordingly to a certain constellation, which is set in another GNU radio function. The header is modulated using the Binary Phase Shift key (BPSK) method, with the correspondence between bit and complex symbol as presented in Table 3.1. This modulation is used for the header to simplify the channel estimation and to improve the system reliability.

Bits	BPSK Symbol
0	$1 + 0j$
1	$-1 + 0j$

Table 3.1: BPSK modulation scheme

Next section will present the Payload branch and the remaining OFDM transmitter.

3.1.4 Payload Branch

The Payload branch procedure is analogous to the one presented previously, however, there are a few differences that will be explained. First of all, there is no *Protocol Formatter block*, as it is shown in Figure 3.6, that is because all the information on stream matters. The modulation scheme used in this branch is the Quadrature Phase Shift Key (QPSK) which correspondence is set as showed in Table 3.2.

As it can be seen in Table 3.2 this constellation takes 2 bits to form a complex symbol, which means the *Repack Bits* block in this branch will read 8 bits from every input byte and write onto 2 bits of each output byte, what leads to a interpolation of 4. Therefore, in terms of packet dimension it goes from 100 bytes to 400 bytes.

Obviously the Header and Payload will not be sent separately, therefore the *Tagged*

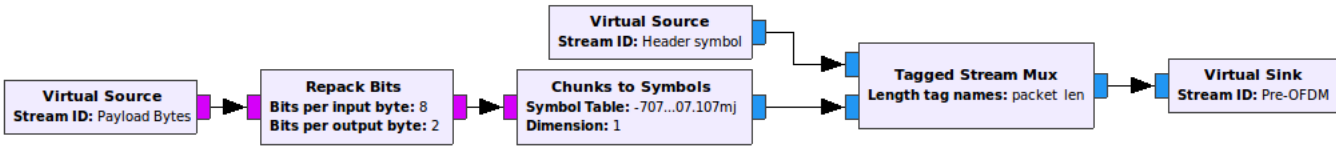


Figure 3.6: Payload branch flowgraph

Bits	QPSK Symbol
00	$1 - 1j$
01	$1 + 1j$
11	$-1 - 1j$
10	$-1 + 1j$

Table 3.2: QPSK modulation scheme

Stream Mux will combine these two streams into a single tagged stream of packets with 448 bytes. Afterwards it is showed the OFDM Modulation stage, this is where the stream will be shaped into an OFDM frame, Figure 3.7 represents this stage.

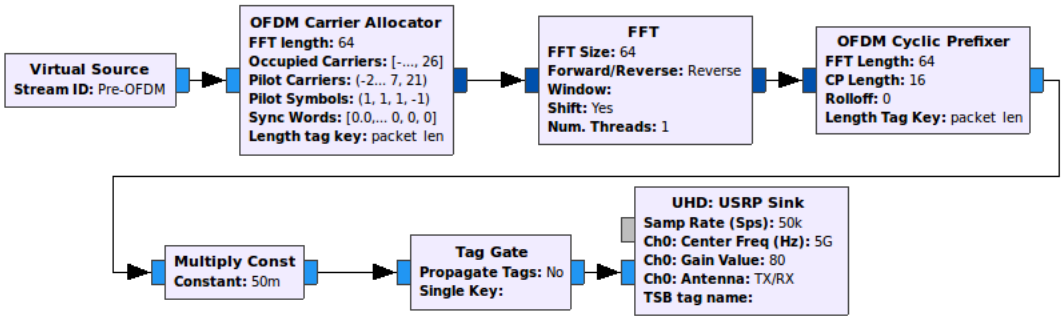


Figure 3.7: *OFDM* modulation stage

The *OFDM Carrier Alocator* block [41] is responsible for the construction of OFDM symbols by placing the incoming complex items onto the OFDM carriers. Additionally, it will place pilots symbols on designated positions and add the guard band. The OFDM symbols length is set accordingly to the specified FFT length , in this case it is 64. In Figure 3.8 the structure of an OFDM symbol without the guard band is presented. Note that ”P” corresponds to the pilot carriers.

The output of this block corresponds to a frame composed of OFDM symbols and OFDM preamble symbols that are appended to the frame for synchronization purposes. The structure of this output is shown in Figure 3.9.

In regard to the preamble symbols they are composed of a sequence that is characterized for the first half being identical to the second in the time domain.

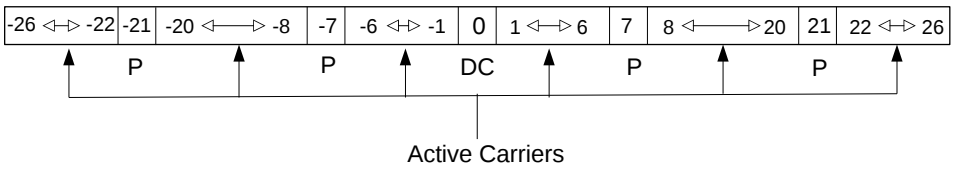


Figure 3.8: *OFDM* Symbol structure

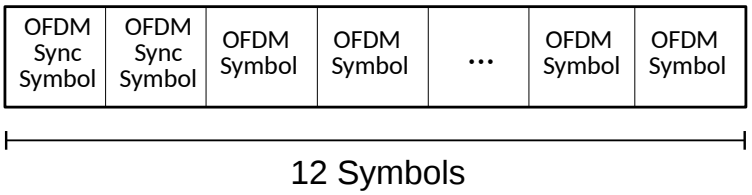


Figure 3.9: *OFDM* frame structure

At this point an Inverse Fast Fourier Transform (IFFT) is applied to the stream, from analysis of Figure 3.1 it is noticeable that this process occurs on the *FFT* block. This block can apply both the Fast Fourier Transform (FFT) and IFFT, if it is set to ”Forward” corresponds to a FFT and ”Reverse” to an IFFT.

After the IFFT has been applied to the signal, a cyclic prefix (CP) is added to each packet. Furthermore, the *OFDM Cyclic Prefixer* block will also conduct the pulse shaping, by applying a Raised Cosine RC on the time domain.

The OFDM signal is then multiplied by a constant in order to set the amplitude value to $[-1, 1]$, this is because if the USRP is fed samples of amplitude larger than $|1.0|$ then the signal will suffer from clipping. Finally, the OFDM signal will transmitted through an USRP.

3.2 GNU Radio OFDM Receiver

The OFDM Receiver [33] is built with the GRC framework. It uses a simple OFDM frame structure without the complexity of wireless protocols specifications. In Figure 3.10 the flowgraph of this receiver is split into four parts: Synchronization/Detection, Demux/Packetize, Header demodulation and Payload demodulation; these will be presented by that respective order.

3.2.1 Synchronization and Detection

The received signal is a stream of complex samples that is split into two identical signals. One of the signals suffers a delay of samples equal to the length of one packet minus the CP, this block delays the signal by inserting zeros at the input stream beginning equal to the defined value. Meanwhile, the other stream is sent to a synchronization block, this is shown in Figure 3.11.

The *Schmidl & Cox OFDM synch* implements the method introduced in [42] for rapid synchronization in an OFDM system using continuous or burst transmission. The acquisition is a two step process where a training sequence with the length of two symbols is used as a preamble. First, a search for a symbol with the first half equal to the second in the time domain is conducted to determine the frame timing. Then, the phase difference between the two halves of the first training symbol, $\hat{\phi} = angle(P(d))$, is used to determine the normalized carrier frequency offset, $\hat{\Delta f} = \frac{\hat{\phi}}{\pi T}$, which will consequently be used to correct the delayed signal frequency. Note that T stands for the useful time length of an OFDM symbol and d is a time sample corresponding to the first in a window with the training symbol length.

A timing metric, $M(d)$, is used to determine if a training sequence has arrived or not [42], and is given by Equation 3.1. The normalization done by this block is different from the one in [42],

$$M(d) = \frac{|P(d)|^2}{R(d)^2} \tag{3.1}$$

being $P(d)$, given by Equation 3.2) the sum of the product pairs, L the complex samples present in half of the first training symbol and r_m are the complex samples of the signal.

$$P(d) = \sum_{m=0}^{L-1} \left(r^*_{d+m} r_{d+m+L} \right) \tag{3.2}$$

The normalization factor , $R(d)$, is in this case

$$R(d) = \frac{1}{2} \sum_{k=0}^{V-1} |r_{k+d}|^2 \tag{3.3}$$

where $V-1$ is half of the useful subcarriers. Equation 3.3 can be understood as the energy estimate from both half-symbols. This avoids false detections at the burst final, where the energy suffers a rapid decrease [43].

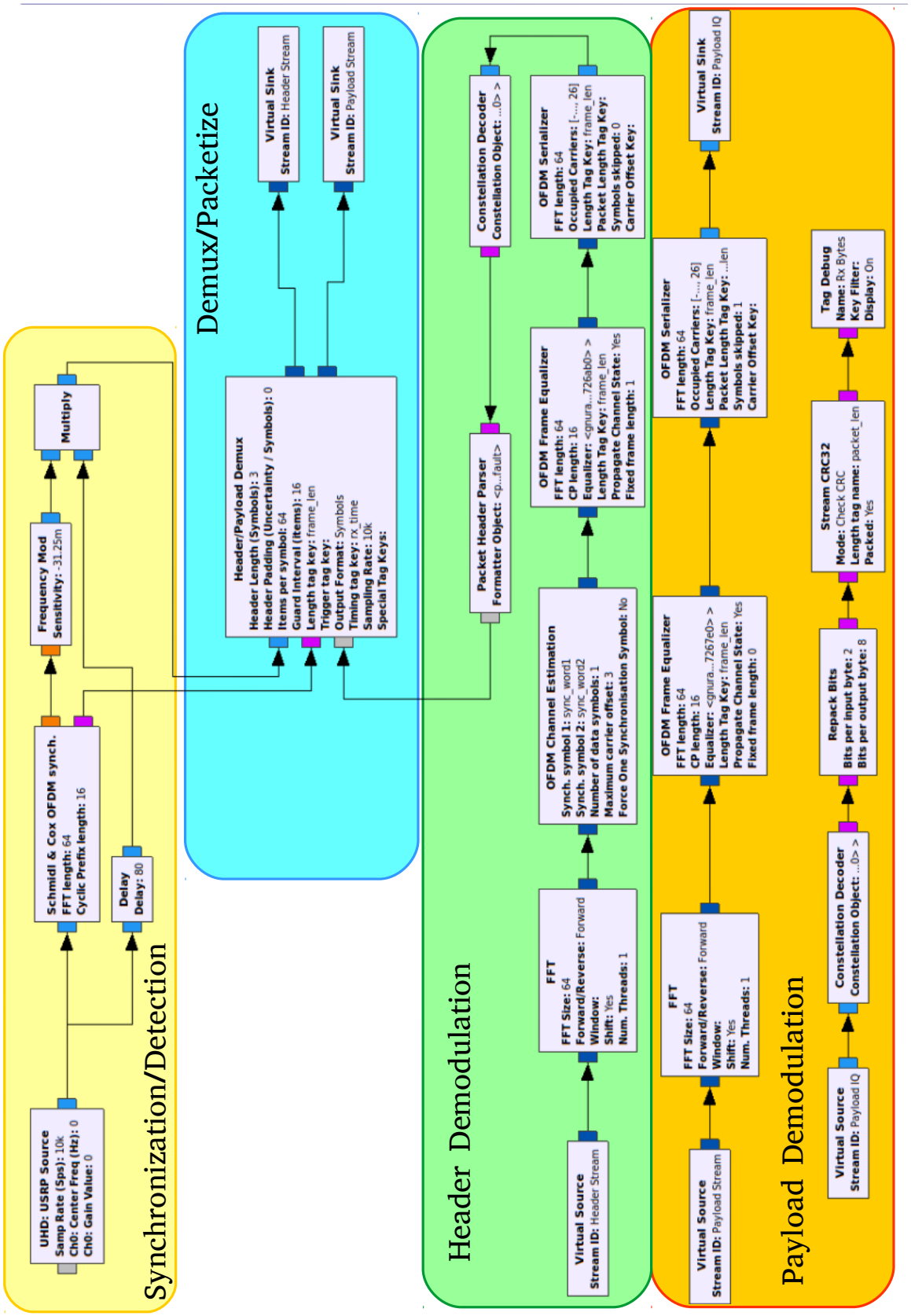


Figure 3.10: OFDM receiver GRC flow-graph [33]

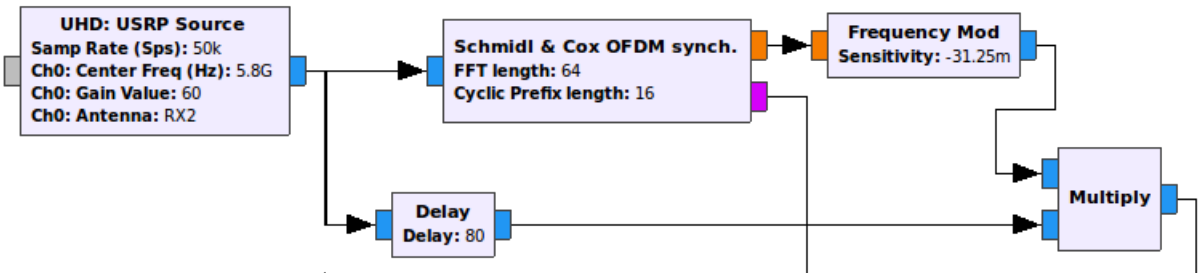


Figure 3.11: *OFDM* Receiver synchronization and detection

Also, a trigger signal is created, in which the beginning of the first OFDM symbol after the first doubled OFDM symbol is marked with a 1 [43] [42].

Since the offset, $\hat{\Delta}_f$, is in this case a stream of real samples it is necessary to turn it into a complex signal. The *frequency mod* block will receive the frequency offset , $\hat{\Delta}_f$, previously determined and will convert it into a complex in the baseband. The concept behind this block is the following: it takes a real signal $x_m[n]$ as input and converts it into a frequency modulated signal $y[n]$ the following way,

$$y[n] = e^{(jk\sum x[n])} \tag{3.4}$$

where k stands for the sensitivity and it is $k = 2\pi\frac{f_\Delta}{f_s}$, with f_Δ being the maximum frequency deviation and f_s the sampling frequency [44].

The delayed signal is then multiplied by the complex stream corresponding to the frequency offset correction, therefore, the signal that is sent into the next stage has already suffered a fine frequency correction.

3.2.2 Demux and Packetyze

The second part is responsible for the division of header and payload into two different packed streams, and it is presented in Figure 3.12.

All of this process happens on the *Header/Payload Demux*. This block takes as inputs the previous stage stream, the trigger signal from the *Schmidl & Cox OFDM synch* and a PMT dictionary from the *Header Parser*, which will also serve as trigger. Until a trigger is detected all of the samples that enter this block will be discarded.

After a trigger is detected a total of samples equal to the header length are copied into the first output. All the data from this output will be sent through the next stage (Header Demodulation) where it will be demodulated into bytes, in order to take the information contained in the header, and returned to the *Header/Payload Demux* block as a PMT dictionary [45].

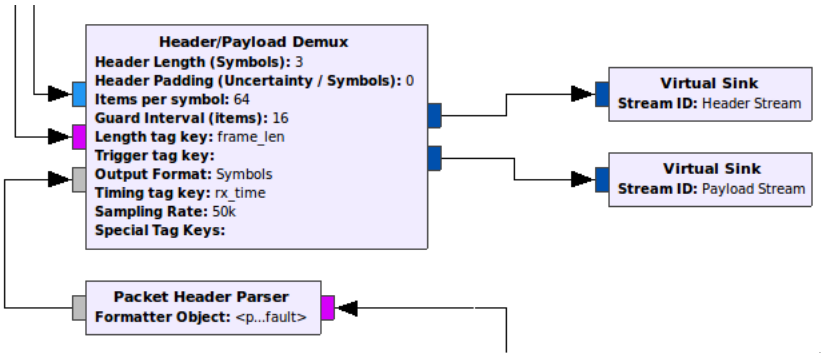


Figure 3.12: *OFDM* Receiver Demux and Packetyse

3.2.3 Header Demodulation

The next stage to be analyzed is the Header Demodulation, this is presented in Figure 3.13. The header stream is sent through a FFT which turns the OFDM symbols into the frequency domain. Even though this signal has already suffered a frequency correction it was not enough, therefore, it is done a coarse frequency offset estimation and a channel estimation in the *OFDM Channel Estimation* block.

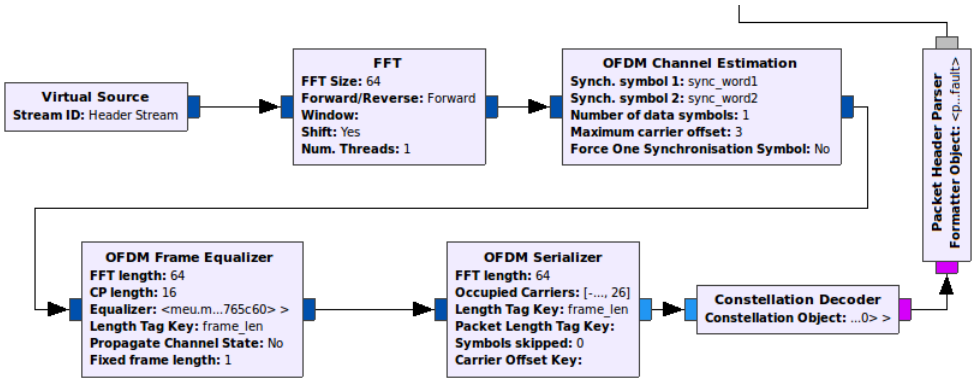


Figure 3.13: *OFDM* Receiver Header demodulation

The estimation concerning the channel state condition , H^n_{est}, is done through a zero-forcing algorithm, as show in Equation 3.5.

$$H^n_{est} = \frac{X^n_{rx}}{X^n_{train}} \tag{3.5}$$

where X^n_{rx} stands for the input stream and the X^n_{train} is a know training sequence equal to the transmitted preamble.

The coarse frequency estimate follows the algorithm described in [42] with some changes as presented in section 3.2.1. The estimates, which are complex numbers, are

then passed as tags, since the corrections are not applied in this block [46].

The signal is then propagated into an equalizer block, the *OFDM Frame Equalizer*, where the frequency offset and the channel effects are corrected using the previously calculated estimates.

To dismantle the OFDM symbol, i.e. to pass from several low rate bit streams in parallel into one high rate bit stream, and have the pilots removed, the signal is sent into the *OFDM Serializer*. This block will search for two different tags, the *len_tan* and the *packet_len*, which specify the number of OFDM symbols in the input frame and the amount of modulated complex symbols, respectively. If both tags are used then the *packet_len* will correspond to the maximum value of outputs and the other will set the number of inputs consumed by the block, Figure 3.14 illustrates this procedure.

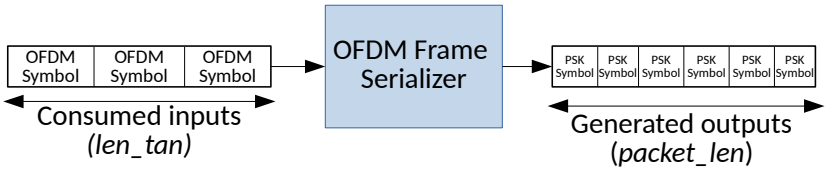


Figure 3.14: *OFDM Serializer* procedure

Finally, this stream is sent through the *Packet Header Parser* which will select the relevant information from the header in order to form a PMT dictionary. Afterwards, it is sent to the *Header/Payload Demux* block.

3.2.4 Payload Demodulation

The last stage is the Payload Demodulation. This is very similar to the Header Demodulation, except that there is no channel estimation done here. Therefore, the payload will use the estimations that were propagated through tags, from the Header Demodulation branch, to do the signal equalization. Also, there is a CRC block that will verify the content, if the package does not pass on the verification it will be dropped.

3.3 Model with Jamming Based on Reciprocal Channel

In the traditional wiretap channel model the transmission is done simply from Alice to Bob and, inadvertently, to Eve. In this section, we present the Jamming Based on Re-

ciprocal Channel (JBRC) security method that follows a different approach, in which it is necessary for Bob to transmit a know sequence to Alice, previously to Alice's first transmission. The reason for this necessary transmission is that Alice has to know the channel state information in order to shift the signal phase. Figure 3.15 illustrates the developed model with the JBRC technique.

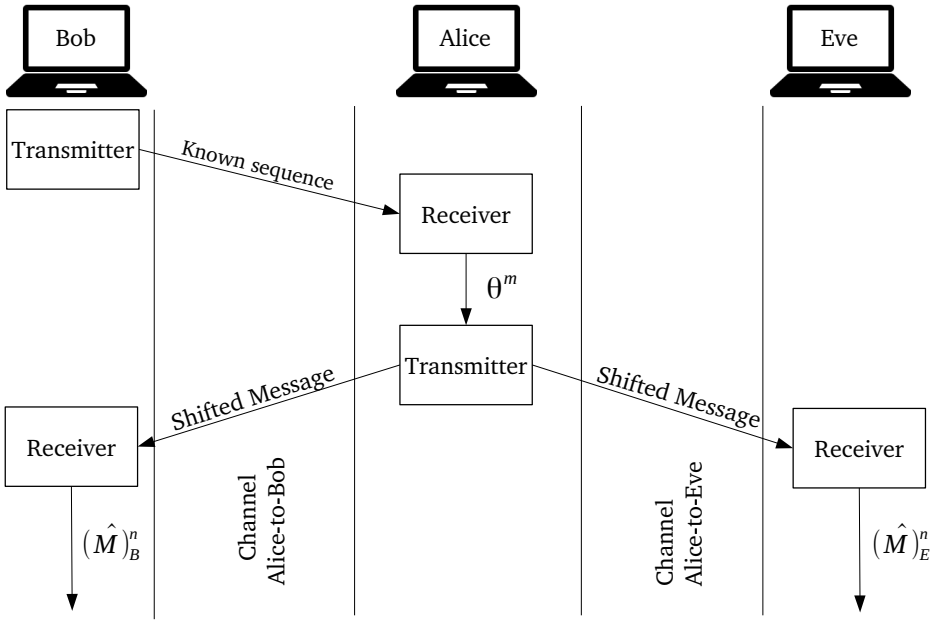


Figure 3.15: Work concept in time of the developed system

In Figure 3.15 the "Transmitter" and "Receiver" correspond to the GNU radio OFDM Transmitter and Receiver presented in the previous sections of this chapter, with a new added functionality, they were modified to apply or eliminate the phase shift on the transmitted signal.

The first stage of Figure 3.15 is the transmission of a known sequence by Bob. When this known sequence is received by Alice, she will estimate the channel and use that estimation to generate θ^m, which will then be sent, internally, to the transmitter. Alice will then use θ^m to shift the message and transmit the shifted message. On reception, both Bob and Eve, will do a channel estimation and use that estimate to eliminate the shift introduced by Alice. It is important to note that Eve does not estimate her own θ^m from the know sequence. This is because that estimation would correspond to the channel Bob-to-Eve and, therefore, it would be an estimate of a different channel. The estimation

used by Eve and Bob to eliminate the θ^m is done with the received shift message header information.

In order to determine the phase offset accordingly to the channel state it was necessary to develop new a GNU radio block. The implementation is described in the next subsection.

3.3.1 Phase offset finder

The *Phase offset finder* block is a component built with the GNU radio software. This block can be understood as a tagged stream interpolator. It takes as input a signal after the channel estimation has been made and propagated through the signal tags. Also, it will return a number of items equal to the length of the modulated QPSK frame. Each item created is a float type value corresponding to a phase offset determined by the channel state argument of a certain subcarrier.

As it was explained in section 3.2, each channel estimate is a complex number, therefore, by analyzing to which quadrant they belong a given angle is determined accordingly to the mapping presented in Figure 2.7; e.g. for a channel estimate of $-1-1j$ then the output would be π. The concept is illustrated in Figure 3.16.

It is important to note that the channel state is read from the tags, which are propagated once in the beginning of every frame, so, even though it takes the whole signal as input it will only be interested on the channel estimations passed in the tags.

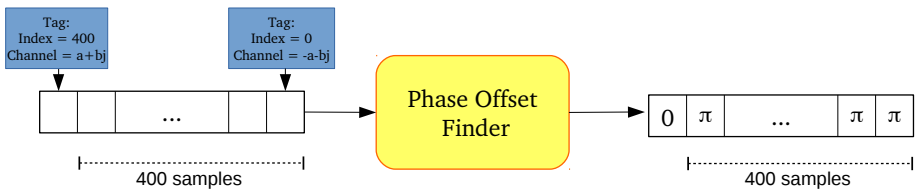


Figure 3.16: Phase offset finder modus operandi

The application of this block with the OFDM transmitter and receiver to form the desired system will be discussed in the next subsection.

3.3.2 Simulation system

In order to create the system described in Figure 3.15, it was necessary to apply the *Phase offset finder* block to the OFDM Transmitter and Receiver accordingly to the ne-

cessities.

In the first transmission (Bob to Alice) it was used a transmitter that was simplified to the components concerning the Header, since there is no actual payload message being sent. That is because, the transmitted known sequence is the information used for synchronization and channel estimation.

The first receiver (Alice) must determine a phase offset and send it to Alice's transmitter. Therefore, the *Phase offset finder* block was added as it can be seen in Figure 3.17.

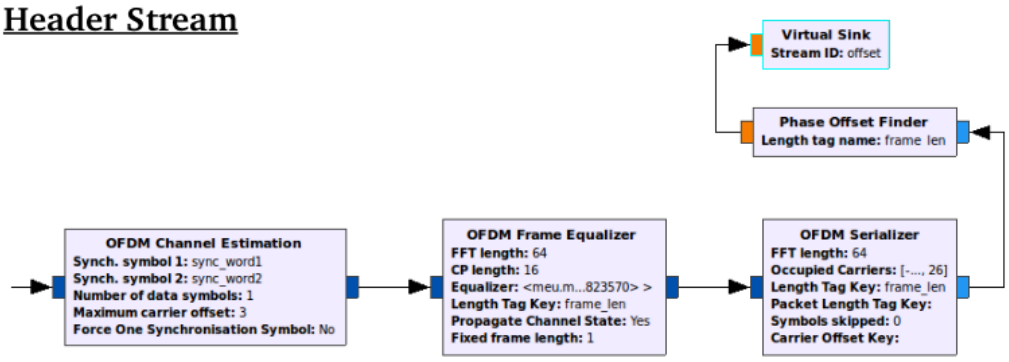


Figure 3.17: Phase offset finder in the OFDM receiver flow graph

After the phase offset is sent to Alice's transmitter, it will broadcast a modulated message, where the symbols were shifted with a determined offset. The phase shift on Alice's transmitter is done as show in Figure 3.18.

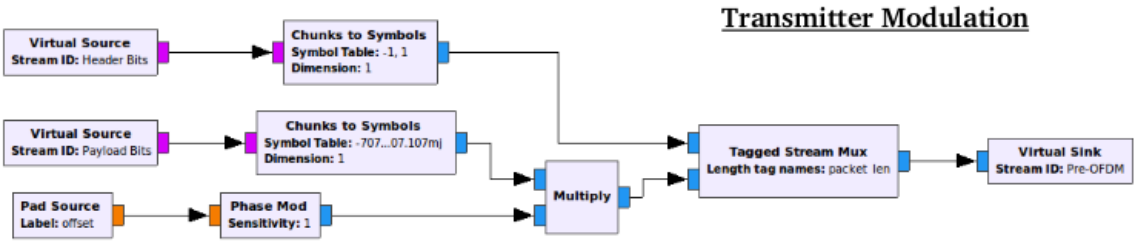


Figure 3.18: Phase offset finder in Alice's OFDM transmitter flow graph

As it can be seen, the flow graph (Figure 3.18) has three new blocks. The *Pad source*, which is used to create a new virtual input to the system flow graph, the *Phase Mod*, which takes a certain float number as input and returns a complex with argument equal to the input, also, it has a parameter, sensitivity, that can be understood as a gain. In this system the sensitivity is set to 1 on the modulator and to -1 on the demodulator, so that the receiver phase shift corresponds to the transmitter phase shift inverse and, therefore,

allows to eliminate the shift effects.

In order to receive the message correctly, this phase shift has to be eliminated. For that reason both Bob and Eve's receiver have been modified so that they can determine the channel estimate, just as it is shown in Figure 3.17, and eliminate it from the signal right before the QPSK demodulation, as it can be seen in Figure 3.19

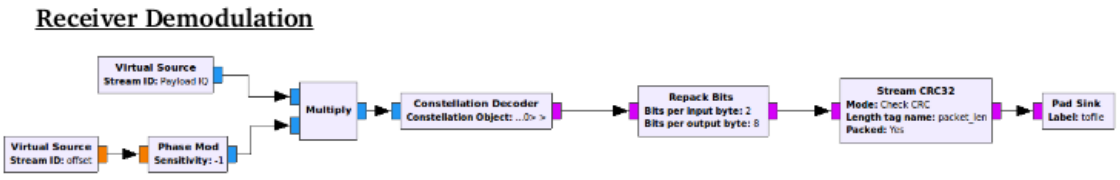


Figure 3.19: Phase offset finder in Bob's (or Eve's) OFDM receiver flow graph

In figure 3.20 the complete simulation system is presented. The flow graph is composed of several *hier* blocks. A *hier* block is a GNU radio companion method to represent a flow graph in the form of a block to another flow graph.

The nomenclature used in the *hier* blocks is the following: "TX" for a transmitter and "RX" for receivers + name of the user, for example the block *TXBOB* will be the modified OFDM transmitter used by Bob on the first transmission to Alice. The *RXBOB&EVE* block has that designation because the receiver for Bob and Eve is equal.

The *Channel Model* block, represented in Figure 3.20 allows to add impairments inherent to a wireless communications channel such as:

- Noise: Sets the noise level for a AWGN, also has a seed which determines the noise creation;
- Frequency Offset: Normalized frequency offset;
- Time Offset (Epsilon):This emulates the different rates between the samples clocks on the transmitter and receiver;
- Taps/Multipath: Taps of a Finite impulse response filter which create delayed replicas of the signal. [47]

In order to have the "same" channel between Alice and Bob the parameters will always be the same, obviously, the channel model corresponding to Eve's stream must be different.

In the next chapter this simulation system will be used to determine the secrecy capacity given by this method, accordingly to the channel specifications.

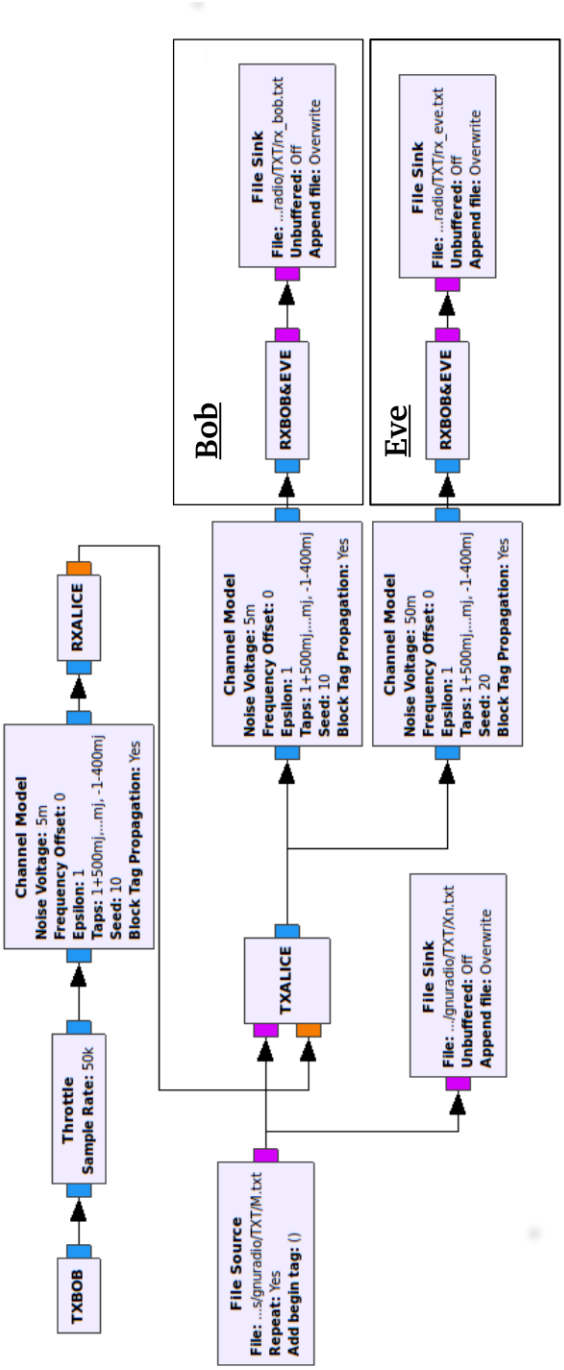


Figure 3.20: Simulation system

4

Experimental results

In this chapter the results obtained using the system described in the previous chapter are presented. Also, a real world approach using USRP is done to test the viability of this method in a real world environment.

4.1 Secrecy of an OFDM system in simulated environment

The simulation will be used to determine the secrecy under different types of channel impairments.

In order to measure the level of secrecy it was used a unit based on the, C_s, which will be called the secrecy ratio, SR. The SR is obtained by analyzing the amount of information that was received correctly with the quantity of sent data. The validation is done through the CRC, since every package that has an error is discarded, as explained on section 3.2.4. Equation 4.1 shows how this metric is determined.

$$SR = \frac{Bob's\ rx\ packages - Eve's\ rx\ packages}{Transmitted\ packages} \tag{4.1}$$

The two main effects analyzed were the noise and multipath distortion.

This study was conducted by changing the channel model parameters in order to determine the system behavior. Three scenarios were considered:

• **Scenario 1**

For the first setup a SNR of 65 dB was set for the legitimate transmission and 55 dB to the illegitimate one. Also, a Power Delay Profile (PDP) was set to the legitimate channel through the definition of a set of taps as shown in Table 4.1.

Taps	1.0+1.0j	1.0+0.5j	-0.1+0.5j	-1-0.4j

Table 4.1: PDP taps of channel Alice-to-Bob

The PDP gives the distribution of signal power received over a multipath channel as a function of the propagation delays, as shown in Equation 4.2 [48].

$$PDP(\tau) = \sum_{m=0}^{J-1} P_m \delta(\tau - T_m) \tag{4.2}$$

where J is the number of taps and P_m is the power corresponding to a certain tap; Figure 4.1 illustrates this PDP.

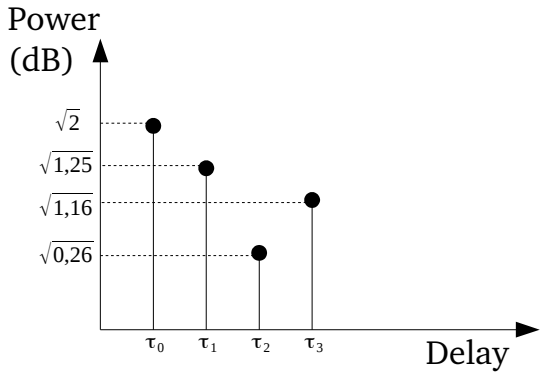


Figure 4.1: Defined PDP for the simulation scenario

The transmitted message was a text file with a size of 71 MBytes, this means that when the message was sent there were a total of 755320 packet transmissions; this value is obtained through Equation 4.3

$$PacketTransmissions = \frac{MessageLength}{PacketLength} \tag{4.3}$$

where the Packet length is 96 as described in section 3.1.

This conditions were maintained through all the iterations. Regarding the channel between Alice and Eve, the same number of taps were considered, although, their value may differ. The set of taps where there is a maximum of different PDP taps between legitimate and illegitimate channels is presented in Table 4.2.

Taps	1.0-1.0j	-1.0-0.5j	+0.1-0.5j	-1+0.4j

Table 4.2: PDP taps of channel Alice-to-Eve

In order to have a valid average, all the possible combinations of different taps between Tables 4.1 and 4.2 were considered, Table 4.3 presents the number of combinations for each scenario.

This is used to determine the influence of multipath distortion in the JBRC method.

The SR results presented in Table 4.4 indicate that the JBRC method increases the system security with the increase of different taps. This is expected, since the increase of different taps correspond to an illegitimate channel suffering from an increasingly more distinct multipath distortion, which, consequently, changes the channel estimates argument used to generate the θ^m.

Taps used in both channels	Different taps between channels	Combinations
1	1	$\binom{1}{1} = 1$
2	1	$\binom{1}{1} = 2$
	2	$\binom{2}{2} = 1$
3	1	$\binom{3}{1} = 3$
	2	$\binom{3}{2} = 3$
	3	$\binom{3}{3} = 1$
4	1	$\binom{4}{1} = 4$
	2	$\binom{4}{2} = 6$
	3	$\binom{4}{3} = 4$
	4	$\binom{4}{4} = 1$

Table 4.3: Number of simulations for each scenario

Taps used in both channels	Different taps between channels	*SR*
1	1	0.07
2	1	0,11
	2	0,24
3	1	0.25
	2	0.32
	3	0.35
4	1	0.30
	2	0.35
	3	0.33
	4	0.34

Table 4.4: Analysis of multipath effect on SR

• **Scenario 2**

The second scenario consider the same number of taps for the channels Alice-to-Eve and Alice-to-Bob but with different PDP taps. The SNR is in this case equal for both channels. The taps coefficients used are presented in Table 4.5. Each point is the average of at least 20 repetitions. Figure 4.2 present the SR achieved for this scenario as function of the SNR. Also, it is given the standard deviation for each average.

Channel	Taps
Alice-to-Bob	1.0+1.0j, 1.0 +0.5j, -0.1+0.5j, -1-0.4j
Alice-to-Eve	1.0-1.0j, 1.0 +0.5j, -0.1+0.5j, -1-0.4j

Table 4.5: Taps used for the second simulation scenario

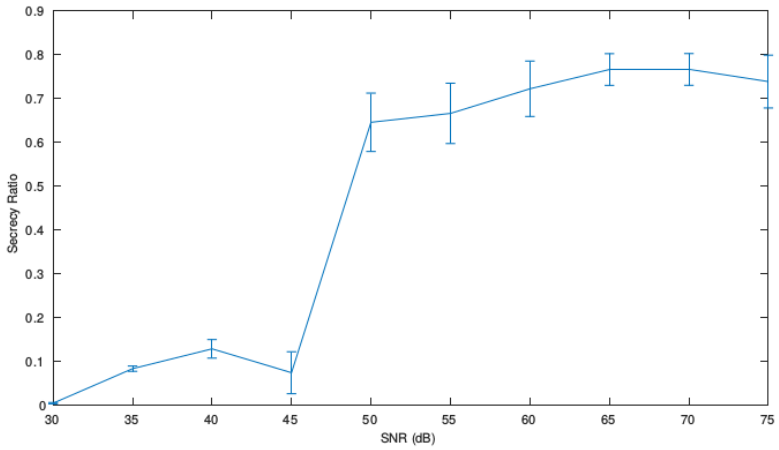


Figure 4.2: SNR influence on *SR* for the second simulation scenario

The results presented in Figure 4.2 indicate that for high values of SNR there is an increase of the SR. This can be understood as the noise influence on the transmission getting smaller, and, consequently, a decrease on the packets lost by Bob and Eve during the CRC validation. Which indicates that the main culprit for the different amount of information obtained by Bob and Eve, SR, is the multipath distortion. In order to do a deeper test of SNR influence, another scenario was considered.

- **Scenario 3**

The third simulation scenario considered sets a difference between Bob's and Eve's gains of 5 dB, i.e. when Eve has a gain of 30 dB then Bob has 35 dB. In each iteration the gain suffers an incrementation of 1 dB. Each point is the average of at least 9 repetitions. The number of simulations is different from the previous scenario due to time constraints. The SR results are presented in Figure 4.3.

The obtained results indicate that the SNR value does not have much influence in terms of secrecy. The increase of SR in Figure 4.2 and 4.3, is due to the a decrease of lost packages. However, it is also possible to see that there is a relevant value of SR for high values of SNR due to the effect of multipath distortion on the channel estimations used to generate θ^m. This indicates that even when Eve is listening in a relatively good channel, as long as it is different enough to change the channel estimation argument, a considerable level of secrecy is ensured by this method.

It is important to note that, in Figure 4.2 and 4.3, there is a rapid increase of the SR when raising the SNR, even though there was an increase in security, the growth was mostly due to the information lost by both Bob and Eve in the considered first levels

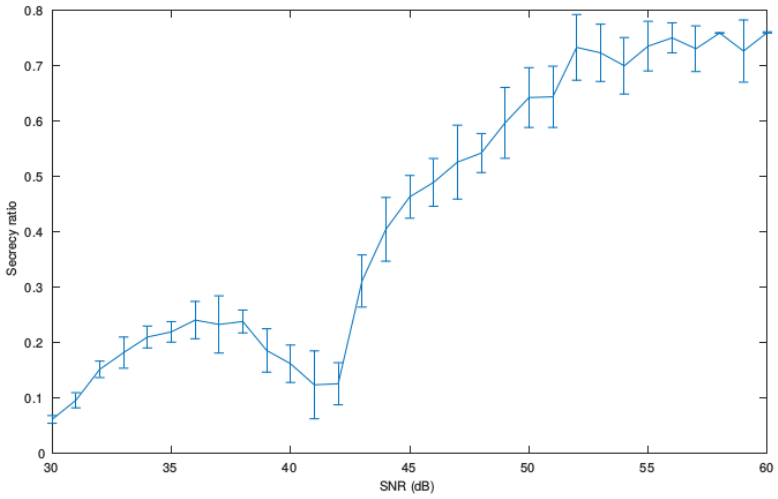


Figure 4.3: SNR influence on *SR* for the third simulation scenario

of SNR. Also, between 40 and 45 dB there is a decrease on the SR, this is due to the aleatory behavior of the noise, which allowed for an increase of Eve's packets validated by the CRC while Bob's packets did not suffer the same increase.

4.2 USRP approach

In order to determine the viability of this system in a real wireless environment, a SDR testbed was developed by using three boards USRP B210 from Ettus Research to emulate the three subjects of the communications scenario (Alice, Bob and Eve). The first objective of this dissertation was to develop the system presented in section 3.3.2, however, for this to be possible it was necessary to implement Time Division Duplexing, which turned out to be a difficult challenge. This difficulty was mostly due to the fact that it was necessary to stop Alice's receiver and start the transmitter in a period of time small enough that the channel estimate would not change, since the USRP takes some time to start working this issue could not be resolved. Therefore, the SDR system was to be implemented only on part.

So, in order to test in a real world environment the *Phase offset finder* block was tested with a simple communication, much like the first stage of Figure 3.15 where Bob sends a known sequence to Alice. The phase offsets determined from the channel estimations were analyzed to see if they would have enough changes to keep the message secret from Eve, even though there was no actual message sent.

4.2.1 Laboratory transmission

The setup used for this experiment was the following

Figure 4.4: Setup for Alice, Bob and Eve

The frequency used in all the experiments with SDR was 5.75 GHz; this frequency was chosen since the spectrum was free from other external transmissions, such as Wi-Fi. The distance from Alice to Bob is the same as in the case of Alice to Eve, just like the

gain used in Bob is equal to Eve's gain. That is because, the objective is to see if, in an apparently equal channel, Bob will have any advantage over Eve, taking in consideration the method developed in this dissertation. A segment, more precisely 3 seconds, of the determined offsets from Bob and Eve are presented in Figure 4.5.

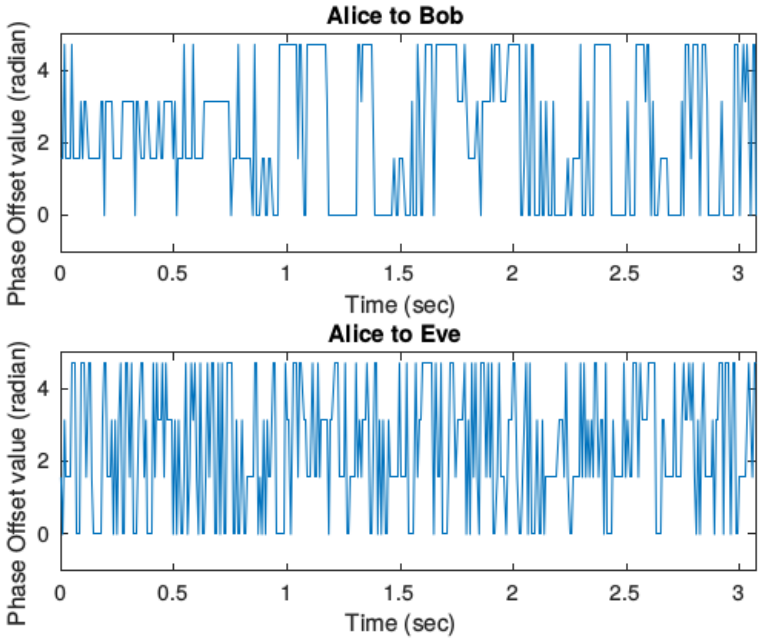


Figure 4.5: Segment of the phase offsets estimated with the previous setup

The phase offsets determined, for this specific interval, show that, in fact, it variates between channels. However, there are certain moments where Bob's offset is equal to Eve's so, in order to quantify the amount of time this undesirable phenomenon happens, it was performed the subtraction of Bob's offset values to Eve's. Figure 4.6 shows the obtained result, taking in consideration that all the values presented are absolute.

The desirable performance is when Figure 4.6 is different from zero. Therefore, all the offsets determined were analyzed to determine the amount of non-zero values existent. Table 4.6 presents the obtained results, that indicates that in this test the illegitimate user would not be able to decode the message at first attempt 71.51% of the time, even though Eve's channel has similar gain and distance compared to Bob's. Which allows a considerable introduction of secrecy to the transmission.

In order to increase the concept validity another room was used for testing.

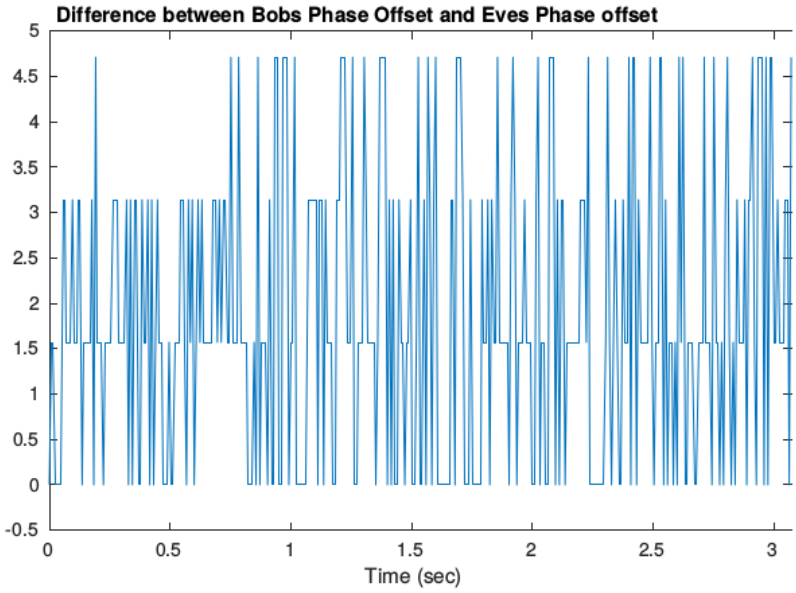


Figure 4.6: Differences between Bob's and Eve's estimated offset in a segment of the previous setup

	Offsets	Ratio
Zero	439	28.49 %
Non-zero	1102	71.51 %
Total	1541	100 %

Table 4.6: Offset analysis

4.2.2 Transmission in a wide room

The second room used is wider and has less obstructions, therefore, the results presented in this dissertation are not just specific to one environment. In this second room two experiments were made, which were designated as "Eve's changing gain Experiment" and "Moving Eve Experiment".

• **Eve's changing gain Experiment**

The first experiment maintained Alice, Bob and Eve in the same position through out all iterations, as showed in Figure 4.7. However, in every iteration Eve's gain had an increase of 5 dB, starting from 45 dB.

The obtained results are presented in Figure 4.8. Each point corresponds to the average of 4 repetitions. Also, it is given the standard deviation for each point.

The objective of this test is to determine the gain effect on the generation of θ^m and to see if in the case that Eve's transmission has a good condition the JBRC method concept is still valid.

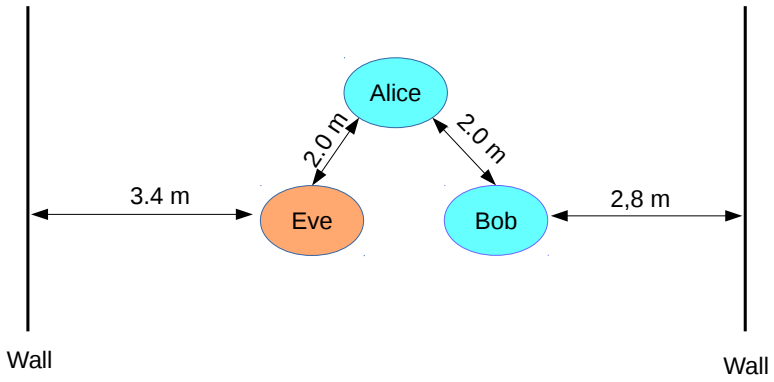


Figure 4.7: Setup for the ”Eve’s changing gain” experiment

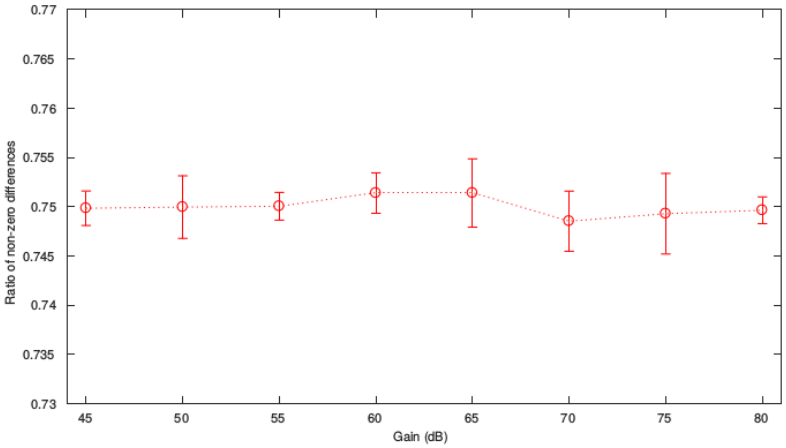


Figure 4.8: Ratio of non-zero differences for the ”Eve’s changing gain” experiment

- **Moving Eve Experiment**

In this experiment a gain of 70 dB was set to both Bob and Eve, while Alice’s gain was set to 90 dB in order to minimize the package loss. The distance from Alice to Bob is fixed at 9 meters, this time what changes is Eve’s position. Eve starts 1 meter away from Alice and will approximate Bob’s position 1 meter per iteration, Figure 4.9 illustrates this procedure.

The objective of this scenario is to approximate Eve’s channel to Bob’s, this is done by moving Eve iteratively closer to Bob. The effect of Eve’s increasingly proximity to Bob on the percentage of non-zero offsets is presented in Figure 4.10, where each point

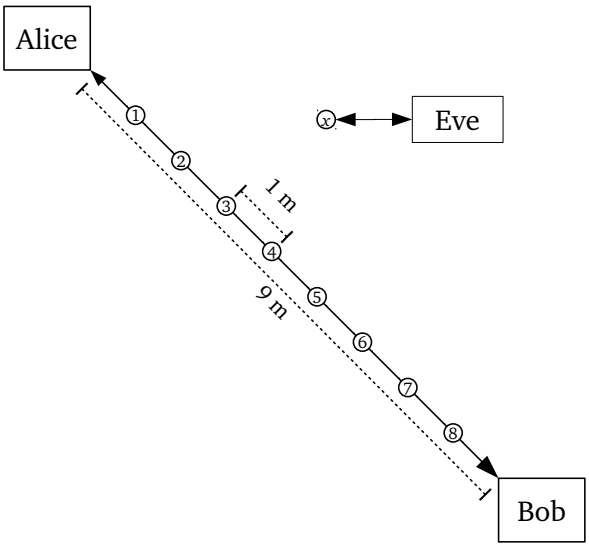


Figure 4.9: Setup for the second experiment with SDR

is the average of 4 repetitions. Also, it is given the standard deviation for each point.

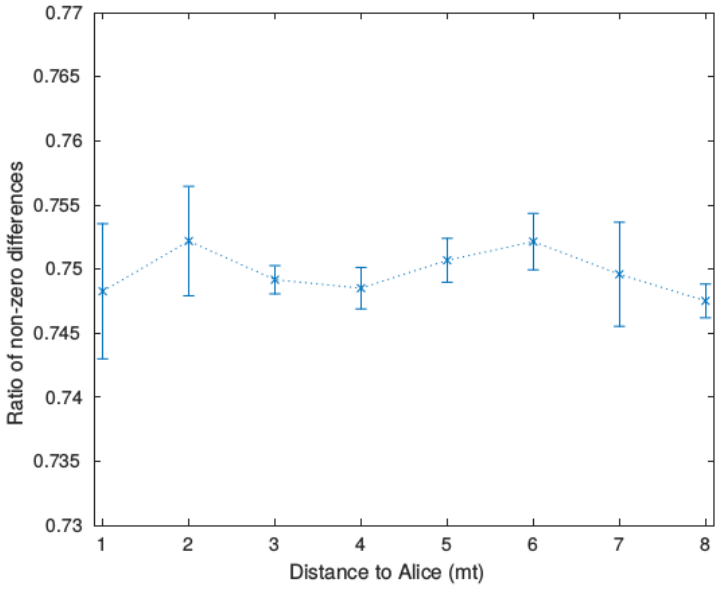


Figure 4.10: Ratio of non-zero differences for the ”Moving Eve” experiment

For both experiments, the results presented in Figures 4.8 and 4.10 show that on average 75% of the generated θ^m are different from Bob to Eve. This means that the JBRC

method can always introduce a considerable level of secrecy on the communication due to the aleatory configuration of the environment and because of the fact that Eve can not be in the same position as Bob. Also, it is important to note that the JBRC method performance is not dependent on the signal gain, this is because the only propriety used to generate θ_m is the estimation argument. In fact, the main responsible for the changes of θ_m is the multi-path distortion which depend on the environment configuration. This results are in accordance with the simulations scenario, which pointed out that the gain did not had much influence on this method, being the multipath distortion the main culprit for the secrecy increase.

5

Conclusion

In this dissertation, an OFDM system with a PLS method was developed. The motivation for this implementation arose from the growth of wireless communications which require different methods of security, such as PLS, in which the channel randomness is used as advantage to protect the transmitted information.

The JBRC method uses channel estimations to generate jamming signals which can be eliminated by the legitimate user. This takes as foundation the channel reciprocity principle. This procedure allows introducing a certain level of secrecy to the transmitted message, since it allows for the legitimate users to have a common random information source between them and different from the eavesdropper.

One of the objectives of this dissertation was to deploy the developed system in a real world environment through SDR. The implementation was done with the GNU radio framework which is a known tool of the radio community. The experiment using SDR, was conducted only in part due to the difficulties in implementing time-division duplexing in real-time, which ables to guarantee the coherence of the channel estimation in both directions of the legitimate transmission, justifying the channel reciprocity approach.

To surpass this constraint and to evaluate the new proposed method a new GNU radio block was built to determine a phase offset accordingly to the channel estimation argument.

The obtained results for each of those tests show that the JBRC method can, in fact, increase the transmitted message level of secrecy, since, it relays on the environment configuration, which is a highly random variable, to generate the phase shifts. That reason and the fact that Bob and Eve can not be in the same position at the same time, allows for this method to be able to introduce security to the transmission even when Eve has a channel with good conditions.

5.1 Future Work

The development of PLS methods has much room for improvement. In consideration to the method used in this dissertation, other type of channel estimation could be used to improve his reliability and a more restrictive approach to the creation of jamming signals could be adopted. Also, the full implementation of the system with time division duplexing would allow testing it in a real world environment and determine how certain restrictive factors behave.

Bibliography

[1] H. V. Poor, R. F. Schaefer, M. R. Bloch, and G. W. Wornell, "Wireless physical layer security," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 114, pp. 19–26.

[2] M. Atallah, G. Kaddoum, and L. Kong, "A Survey on Cooperative Jamming Applied to Physical Layer Security," in *2015 IEEE International Conference on Ubiquitous Wireless Broadband (ICUWB)*. IEEE, oct 2015, pp. 1–5.

[3] M. Bloch and J. Barros, *Physical-Layer Security*. Cambridge: Cambridge University Press, 2011.

[4] G. Anjos, D. Castanheira, A. Silva, and A. Gameiro, "Exploiting reciprocal channel estimations for jamming to secure wireless communications," *2017 Wireless Days, WD 2017*, pp. 136–142, 2017.

[5] J. Zhang, T. Q. Duong, A. Marshall, and R. Woods, "Key Generation from Wireless Channels: A Review," *IEEE Access*, vol. 4, pp. 614–626, 2016.

[6] M. Guillaud, D. T. M. Slock, and R. Knopp, "A practical method for wireless channel reciprocity exploitation through relative calibration," *Proceedings of the Eighth International Symposium on Signal Processing and Its Applications, 2005*, vol. 1, pp. 403–406, 2005.

[7] GNU Radio, "GNU Radio." [Online]. Available: https://www.gnuradio.org/

[8] ——, "GNU Radio Manual and C++ API Reference: Main Page." [Online]. Available: https://gnuradio.org/doc/doxygen/index.html

[9] Matlab, "MATLAB - MathWorks." [Online]. Available: https://www.mathworks.com/products/matlab.html

[10] Ettus Research, "USRP B210 USB Software Defined Radio (SDR) - Ettus Research." [Online]. Available: https://www.ettus.com/product/details/UB210-KIT

[11] ——, "Spec." [Online]. Available: https://www.ettus.com/content/files/b200-b210{_}spec{_}sheet.pdf

[12] R. Nee, Richard van and Prasad, *OFDM for Wireless Multimedia Communications*, 1st ed. Artech House, Inc., 2000.

[13] M. K. Ozdemir and H. Arslan, "Channel estimation for wireless ofdm systems," *IEEE Communications Surveys & Tutorials*, vol. 9, no. 2, pp. 18–48, 2007.

[14] Y. S. Cho, J. Kim, W. Y. Yang, and C. G. Kang, *MIMO-OFDM Wireless Communications with MATLAB®*. Chichester, UK: John Wiley & Sons, Ltd, aug 2010.

[15] E. Gopi, *Digital Signal Processing for Wireless Communication using Matlab*. Cham: Springer International Publishing, 2016.

[16] A. Goldsmith, *Wireless Communications*. Cambridge University Press, 2005.

[17] N. Marchetti, M. I. Rahman, S. Kumar, and R. Prasad, *OFDM: Principles and Challenges*. Boston, MA: Springer US, 2009, pp. 29–62.

[18] T.-D. Chiueh and P.-Y. Tsai, *OFDM Baseband Receiver Design for Wireless Communications*. Wiley Publishing, 2007.

[19] J.-J. van den Beek, O. Edfors, M. Sandell, S. K. Wilson, and P. O. Börjesson, "On channel estimation in OFDM systems," *Proceedings of Vehicular Technology Conference*, vol. 2, no. 1, pp. 815–819, 1995.

[20] B. Akkawi, "Physical layer secrecy channel coding," *LSU Master's Theses*, jan 2008.

[21] W. K. Harrison, J. Almeida, M. R. Bloch, S. W. McLaughlin, and J. Barros, "Coding for Secrecy: An Overview of Error-Control Coding Techniques for Physical-Layer Security," *IEEE Signal Processing Magazine*, vol. 30, no. 5, pp. 41–50, sep 2013.

[22] A. D. Wyner, "The Wire-Tap Channel," *Bell System Technical Journal*, vol. 54, no. 8, pp. 1355–1387, oct 1975.

[23] C. E. Shannon, "Communication Theory of Secrecy Systems.pdf," *Bell Labs Technical Journal*, vol. 28, no. 4, pp. 657–715, 1949.

[24] W. K. Harrison, D. Sarmento, J. P. Vilela, and M. Gomes, "Analysis of Short Block-length Codes for Secrecy," *CoRR*, vol. abs/1509.0, p. 10, sep 2015.

[25] B. Schneier, "Cryptographic design vulnerabilities," *Computer*, vol. 31, no. 9, pp. 29–33, 1998.

[26] M. Baldi, M. Bianchi, and F. Chiaraluce, "Coding with scrambling, concatenation, and HARQ for the AWGN wire-tap channel: A security gap analysis," *IEEE Transactions on Information Forensics and Security*, vol. 7, no. 3, pp. 883–894, 2012.

[27] S. Goel and R. Negi, "Guaranteeing secrecy using artificial noise," *IEEE Transactions on Wireless Communications*, vol. 7, no. 6, pp. 2180–2189, 2008.

[28] J. P. Vilela, M. Bloch, J. Barros, and S. W. McLaughlin, "Wireless Secrecy Regions With Friendly Jamming," *IEEE Transactions on Information Forensics and Security*, vol. 6, no. 2, pp. 256–266, jun 2011.

[29] J. S. Sousa and J. P. Vilela, "Uncoordinated Frequency Hopping for Wireless Secrecy Against Non-degraded Eavesdroppers," *IEEE Transactions on Information Forensics and Security*, vol. PP, no. 99, pp. 1–1, 2017.

[30] D. Sarmento, J. Vilela, W. K. Harrison, and M. Gomes, "Interleaved Coding for Secrecy with a Hidden Key," in *2015 IEEE Globecom Workshops (GC Wkshps)*. IEEE, dec 2015, pp. 1–6.

[31] J. P. Vilela, M. Gomes, W. K. Harrison, D. Sarmento, and F. Dias, "Interleaved Concatenated Coding for Secrecy in the Finite Blocklength Regime," *IEEE Signal Processing Letters*, vol. 23, no. 3, pp. 356–360, mar 2016.

[32] GNU Radio, "GNU Radio Manual and C++ API Reference: Components," 2016. [Online]. Available: https://gnuradio.org/doc/doxygen/page{_}components.html

[33] ——, "GNU Radio Manual and C++ API Reference: OFDM." [Online]. Available: https://gnuradio.org/doc/doxygen/page{_}ofdm.html

[34] ——, "GNU Radio Manual and C++ API Reference: Main Page - Tagged Stream Blocks," 2016. [Online]. Available: https://gnuradio.org/doc/doxygen/page{_}tagged{_}stream{_}blocks.html

[35] T. Rondeau, "Scheduler Details," pp. 1–56, 2013.

[36] S. Lin, D. J. Costello, and P. Prentice Hall, *Error Control Coding*. Prentice-hall Englewood Cliffs, 2004.

[37] ITU-T, "Series V: Data Communication over the Telephone Network. Error-correcting procedures for DCEs using asynchronous-to-synchronous conversion," *Rec. ITU-T v.42*, 2002.

[38] GNU Radio, "GNU Radio Manual and C++ API Reference: gr::digital::header_format_ofdm Class Reference." [Online]. Available: https://gnuradio.org/doc/doxygen/classgr{_}1{_}1digital{_}1{_}1header{_}{_}format{_}{_}ofdm.html

[39] ——, "GNU Radio Manual and C++ API Reference: gr::blocks::repack_bits_bb Class Reference."

[40] ——, "BlocksCodingGuide - GNU Radio." [Online]. Available: https://wiki.gnuradio.org/index.php/BlocksCodingGuide{#}Interpolation{_}Block

[41] ——, "GNU Radio Manual and C++ API Reference: gr::digital::ofdm_carrier_allocator_cvc Class Reference." [Online]. Available: https://gnuradio.org/doc/doxygen/classgr{_}1{_}1digital{_}1{_}1ofdm{_}{_}carrier{_}{_}allocator{_}{_}cvc.html

[42] T. Schmidl and D. Cox, "Robust frequency and timing synchronization for OFDM," *IEEE Transactions on Communications*, vol. 45, no. 12, pp. 1613–1621, 1997.

[43] GNU Radio, "GNU Radio Manual and C++ API Reference: gr::digital::ofdm_sync_sc_cfb Class Reference." [Online]. Available: https://gnuradio.org/doc/doxygen/classgr{_}1{_}1digital{_}1{_}1ofdm{_}{_}sync{_}{_}sc{_}{_}cfb.html

[44] ——, "GNU Radio Manual and C++ API Reference: gr::analog::frequency_modulator_fc Class Reference." [Online]. Available: https://gnuradio.org/doc/doxygen/classgr{_}1{_}1analog{_}1{_}1frequency{_}{_}modulator{_}{_}fc.html

[45] ——, "GNU Radio Manual and C++ API Reference: gr::digital::header_payload_demux Class Reference." [Online]. Available: https://gnuradio.org/doc/doxygen/classgr{_}1{_}1digital{_}1{_}1header{_}{_}payload{_}{_}demux.html

[46] ——, "GNU Radio Manual and C++ API Reference: gr::digital::ofdm_chanest_vcvc Class Reference." [Online]. Available: https://gnuradio.org/doc/doxygen/classgr{_}1{_}1digital{_}1{_}1ofdm{_}{_}chanest{_}{_}vcvc.html

[47] ——, “GNU Radio Manual and C++ API Reference: Channel Model Blocks.”

[48] H. Arslan, *Cognitive Radio , Software Defined Radio , and Adaptive Wireless Systems*. Springer, 2007.