2013

# Physical Layer Watermarking of Direct Sequence Spread Spectrum Signals

Xiang Li
*Cleveland State University*

# PHYSICAL LAYER WATERMARKING OF DIRECT SEQUENCE SPREAD SPECTRUM SIGNALS

**XIANG LI**

**Bachelor of Science in Computer Science and Technology**

Beijing University of Posts and Telecommunications

July, 2001

submitted in partial fulfillment of the requirements for the degree

**MASTERS OF SCIENCE IN ELECTRICAL ENGINEERING**

at the

**CLEVELAND STATE UNIVERSITY**

May 2013

This thesis has been approved for the

Department of **ELECTRICAL AND COMPUTER ENGINEERING**

and the College of Graduate Studies by

Thesis Committee Chairperson, Dr. Chansu Yu

Department/Date

Dr. Murad Hizlan

Department/Date

Dr. Ye Zhu

Department/Date

Dedicated to my family and friends

# ACKNOWLEDGMENTS

I would like to express my sincere gratitude to Dr. Chansu Yu, my academic advisor, for introducing me to the broad and interesting world of software defined radio, patiently guiding me through the entire thesis research, and offering me the opportunity to touch the edging techniques in the network security.

I would like to offer my special thanks to Dr. Murad Hizlan and Dr. Ye Zhu for serving on my thesis committee and providing invaluable advices for my thesis.

I would like to take this opportunity to thank to Robert Fiske and Dr. Young Sup Roh for their help in the composition of this thesis.

I would also like to thank my family and friends for their support and encouragement throughout my study.

# PHYSICAL LAYER WATERMARKING OF DIRECT SEQUENCE SPREAD SPECTRUM SIGNALS

**XIANG LI**

## ABSTRACT

Security services and mechanisms in wireless networks have long been studied and developed. However, compared to upper network layers, physical layer security did not play a significant role in the OSI security model. Thanks to the easier implementation and verification methods brought by the development of software defined radio (SDR) techniques, physical layer security mechanisms have recently drawn increasing interest from researchers. Digital watermarking is one of the popular security techniques that can fully utilize various exclusive characteristics of the physical layer.

This thesis proposes a physical layer watermarking technique named Watermarked Direct Sequence Spread Spectrum (DSSS) or WDSSS technique, which embeds authentication information into pseudonoise (PN) sequences of a DSSS system. The design and implementation of the WDSSS prototype system on the GNU Radio/USRP SDR platform is discussed, as well as two watermark embedding methods, the maximized minimum distance method and the sub-sequence method. Theoretical analysis and experimental results on the WDSSS prototype system are presented to evaluate the performances of both the content signal and the watermark signal. Results show that, for the 11-chip PN sequence, increasing artificial chip errors has a quantitatively predictable impact on the content signal, requiring 2 dB higher signal-to-noise ratio (SNR) to maintain an acceptable packet error rate (PER) for one additional flipped chip. In terms of the watermark signal, the two embedding methods demonstrated individual advantages in either PER or throughput. The maximized minimum distance method outperforms the sub-sequence embedding method with a 3 dB lower SNR requirement, while the latter provides 400% more throughput than the former with adequate SNR.

# TABLE OF CONTENTS

Page

ABSTRACT . . . . . . . . . . v

LIST OF TABLES . . . . . . . . . . ix

LIST OF FIGURES . . . . . . . . . . x

CHAPTER

I. INTRODUCTION . . . . . . . . . . 1

1.1 Motivation . . . . . . . . . . 1

1.2 Proposed Work . . . . . . . . . . 3

1.3 Organization . . . . . . . . . . 5

II. BACKGROUND . . . . . . . . . . 6

2.1 Watermarking . . . . . . . . . . 6

2.1.1 Digital Watermarking . . . . . . . . . . 6

2.1.2 Steganography . . . . . . . . . . 7

2.1.3 Physical Layer Watermarking . . . . . . . . . . 8

2.2 DSSS . . . . . . . . . . 9

2.2.1 DSSS System Model . . . . . . . . . . 9

2.2.2 DSSS Demodulator . . . . . . . . . . 9

2.3 GNU Radio/USRP SDR Platform . . . . . . . . . . 12

2.3.1 GNU Radio . . . . . . . . . . 12

2.3.2 USRP . . . . . . . . . . 13

2.3.3 Related SDR Projects with GNU Radio/USRP . . . . . . 16

III. RELATED WORKS . . . . . . . . . . 18

3.1 DSSS in Physical Layer Security . . . . . . . . . . 18

3.2 Works in Physical Layer Watermarking . . . . . . . . . . . . . . 20

3.3 Works in Physical Layer Steganography . . . . . . . . . . . . . . 25

3.4 Summary . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 28

IV. WATERMARKED DSSS . . . . . . . . . . . . . . . . . . . . . . . . 29

4.1 WDSSS Technique Overview . . . . . . . . . . . . . . . . . . . . 29

4.2 Watermark Embedding Methods . . . . . . . . . . . . . . . . . . 30

4.3 Prototype System Implementation . . . . . . . . . . . . . . . . . 35

4.3.1 Baseline DBPSK System . . . . . . . . . . . . . . . . . 35

4.3.2 Generic DSSS System . . . . . . . . . . . . . . . . . . . 38

4.3.3 WDSSS System . . . . . . . . . . . . . . . . . . . . . . 43

V. PERFORMANCE STUDY . . . . . . . . . . . . . . . . . . . . . . . . 47

5.1 Experimental Setup . . . . . . . . . . . . . . . . . . . . . . . . 47

5.2 Spectrum Analysis . . . . . . . . . . . . . . . . . . . . . . . . . 49

5.3 WDSSS Content Signal Performance . . . . . . . . . . . . . . . . 51

5.4 WDSSS Watermark Signal Performance . . . . . . . . . . . . . . 54

5.4.1 With Various Flipping Settings . . . . . . . . . . . . . . 54

5.4.2 With Different Embedding Methods . . . . . . . . . . . 56

VI. CONCLUSIONS AND FUTURE WORK . . . . . . . . . . . . . . . . . 59

6.1 Conclusions . . . . . . . . . . . . . . . . . . . . . . . . . . . . 59

6.2 Future Work . . . . . . . . . . . . . . . . . . . . . . . . . . . . 60

BIBLIOGRAPHY . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 62

# LIST OF TABLES

Table Page

I Illustration of the Relationship of Security Services and Layers [27] . 2

II Specification Comparison of USRP2 and USRP1 [8] . . . . . . . . . . . 15

III One expanded set of PN sequences for steganography in 802.15.14 [21] 27

IV Maximized Minimum Distance Method . . . . . . . . . . . . . . . . . . . 33

V Sub-sequence Watermark Embedding Method . . . . . . . . . . . . . 34

VI Packet Format . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 36

VII Explanations of Important Denotations . . . . . . . . . . . . . . . . . . 40

# LIST OF FIGURES

Figure Page

1 DSSS System Model . . . . . . . . . . . . . . . . . . . . . . . . . . . 10

2 DSSS Spreading Operation . . . . . . . . . . . . . . . . . . . . . . . . 10

3 GNU Radio Layered API . . . . . . . . . . . . . . . . . . . . . . . . . 14

4 GNU Radio Flow Graph and Shared Circular Buffer . . . . . . . . . . . 14

5 USRP2 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 16

6 Watermarking on M-PSK [17] . . . . . . . . . . . . . . . . . . . . . . 24

7 WDSSS System Model . . . . . . . . . . . . . . . . . . . . . . . . . . 31

8 Minimum Distance Illustration [26] . . . . . . . . . . . . . . . . . . . 33

9 DBPSK System Transmitter Flow Graph . . . . . . . . . . . . . . . . . 37

10 DBPSK System Receiver Flow Graph . . . . . . . . . . . . . . . . . . . 39

11 DSSS System Transmitter Flow Graph . . . . . . . . . . . . . . . . . . 41

12 DSSS System Receiver Flow Graph . . . . . . . . . . . . . . . . . . . . 43

13 WDSSS System Transmitter Flow Graph . . . . . . . . . . . . . . . . . 45

14 WDSSS System Receiver Flow Graph . . . . . . . . . . . . . . . . . . . 46

15 Spectrum Comparison between the DBPSK Signal without Spreading and the DSSS Signal . . . . . . . . . . . . . . . . . . . . . . . . . . 50

16 Spectrum Comparison among the DSSS Signal and the WDSSS Signals 51

17 Theoretical PER of DSSS Signal and WDSSS Content Signals . . . . 54

18 Experimental PER of DSSS Signal and WDSSS Content Signals . . . 54

19 PER of WDSSS Watermark Signal with Various Flipping Settings . . 55

20 Throughput of WDSSS Watermark Signal with Various Flipping Settings 56

21 Theoretical PER of WDSSS Watermark Signal with Different Embedding Methods . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 57

22 Experimental PER of WDSSS Watermark Signal with Different Embedding Methods . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 57

23 Throughput of WDSSS Watermark Signal with Different Embedding Methods . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 58

# CHAPTER I

# INTRODUCTION

## 1.1 Motivation

Wireless networks provide an extension upon wired networks due to their open communication environment. Wireless network connections facilitate information sharing, however, they also cause concerns about the privacy and security of communication contents. Therefore, security mechanisms to defeat hostile attacks in wireless networks have been explored in depth and breadth ever since wireless networks emerged. As defined in the ITU-T Recommendation X.800 [27], Security Architecture for Open Systems Interconnection (OSI), various security services are distributed across all layers to provide systematic protection. However, in this security model, the physical layer does not play an important role because the design of the model intended to minimize the duplication of services among layers, as shown in Table I.

Along with the growth of wireless communication systems, demands on the power efficiency of mobile devices have increased. There is also an increasing need for

| Service | Layer | | | | | | |
|---|---|---|---|---|---|---|---|
| | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| Peer entity authentication | . | . | Y | Y | . | . | Y |
| Data origin authentication | . | . | Y | Y | . | . | Y |
| Access control service | . | . | Y | Y | . | . | Y |
| Connection confidentiality | Y | Y | Y | Y | . | Y | Y |
| Connectionless confidentiality | . | Y | Y | Y | . | Y | Y |
| Selective field confidentiality | . | . | . | . | . | Y | Y |
| Traffic flow confidentiality | Y | . | Y | . | . | . | Y |
| Connection Integrity with recovery | . | . | . | Y | . | . | Y |
| Connection integrity without recovery | . | . | Y | Y | . | . | Y |
| Selective field connection integrity | . | . | . | . | . | . | Y |
| Connectionless integrity | . | . | Y | Y | . | . | Y |
| Selective field connectionless integrity | . | . | . | . | . | . | Y |
| Non-repudiation Origin | . | . | . | . | . | . | Y |
| Non-repudiation. Delivery | . | . | . | . | . | . | Y |

Table I: Illustration of the Relationship of Security Services and Layers [27]
Layer 1 -7 are physical layer, data link layer, network layer, transport layer, session layer, presentation layer and application layer, respectively.
“Y”means service provided; “.” means service not provided.

prompt security services in dynamic wireless environments. These requirements have drawn considerable attention to physical layer security research. A mobile device in power saver mode only keeps its physical layer and minimal MAC layer functions active [5]. If the physical layer can provide more security services, such as peer entity authentication, the security information exchange can be completed without activating the upper layers, supporting the application of power saving techniques as well as the rapidly changing wireless communication environment. In addition, SDR platforms with their low cost and flexible configuration, such as GNU Radio/USRP, ease the design and verification processes for new physical layer security methods.

Physical layer security in wireless networks can be implemented by integrating traditional security mechanisms with the unique properties of the physical layer. Digital watermarking is one of the most widely adopted security techniques in the physical layer, since many physical layer attributes, such as channel coding, modulation schemes and transmission power, are favorable candidates for the carrier of digital watermarking technique [24].

A well-known physical layer watermarking example is the direct sequence spread spectrum (DSSS) technique, which was originated in military applications. DSSS technique spreads out the spectrum of an information signal, by taking the modulo-2 sum of information bits with pseudonoise (PN) sequences. Benefiting from the pseudorandomness of PN sequences, the DSSS technique is able to offer jamming resistance, interference rejection, message privacy and a number of other desirable properties [18]. Specifically with the interference rejection property, DSSS is widely adopted in commercial wireless network standards, such as IEEE 802.11 [12] and IEEE 802.15.4 [13], to provide robust communications. However, these standards make the PN sequences available to public, and hence, expose a weakness that could be exploited by adversaries. To complement the systematic security for commercial DSSS systems, researchers proposed to encrypt scrambling sequences [19], employ uncoordinated PN sequences [25] or extend the set of PN sequences in use [23]. Another trend is to construct a covert channel on top of a plain DSSS system by manipulating the PN sequence [21] [35] [15].

## 1.2 Proposed Work

This thesis proposes a physical layer watermarking technique, which embeds watermark information into PN sequences of a DSSS system, named watermarked DSSS or WDSSS technique. WDSSS technique mainly utilizes the correlation procedure in the DSSS receiver. The correlation procedure in the DSSS receiver applies the maximum-likelihood decoding method to recover data bits from the spread signal. This is done by correlating the spread signal with a synchronized copy of the PN sequence and compares the correlation result to a threshold to decide the value of the data bit. The bit errors in the spread signal are ignored and do not affect the decoding decision as long as the threshold is satisfied.

The maximum-likelihood decoding algorithm enables WDSSS to go one step further than DSSS to supply additional physical layer security. The WDSSS transmitter flips chips on chosen positions in the PN sequence to represent authentication patterns, and then the receiver detects and recovers the authentication patterns by examining in-sequence positions of flipped chips; meanwhile the controlled number of flipped chips still allows the receiver to successfully recover the content data. In this manner, the authentication process can be completed at the physical layer without requiring extra bandwidth. The WDSSS technique is also useful in another scenario, where the transmitter and the receiver are not willing to expose their communications, so the transmitter only broadcasts a seemingly meaningless cover signal embedded with a secret signal, which can only be detected and recovered by the aware receiver.

This paper also develops a WDSSS prototype system on the GNU Radio/USRP platform, and implements two embedding methods, the maximized minimum distance method and the sub-sequence method. Both theoretical analysis and practical experiments are conducted to explore the capability of the WDSSS technique. Results show that, for the 11-chip PN sequence, increasing artificial chip errors has a quantitatively predictable impact on the content signal, requiring 2 dB higher signal-to-noise ratio (SNR) to maintain an acceptable packet error rate (PER) for one additional flipped chip. In terms of the watermark signal, the two embedding methods demonstrated individual advantages in either PER or throughput. The maximized minimum distance method outperforms the sub-sequence embedding method with a 3 dB lower SNR requirement, while the latter provides 400% more throughput than the former with adequate SNR.

## 1.3 Organization

The rest of the thesis is organized as follows. Chapter 2 introduces the background of the involved techniques and tools. Chapter 3 studies related works in physical layer security, physical layer watermarking, and physical layer steganography. Chapter 4 presents the proposed WDSSS technique and explains the implementation details of the WDSSS prototype system. Chapter 5 analyzes and discusses the performance of the WDSSS prototype system. Chapter 6 summarizes the thesis and discusses the possible future work with the WDSSS technique.

# CHAPTER II

# BACKGROUND

This chapter will introduce the techniques and tools involved in the development of WDSSS, including the watermarking techniques, DSSS and the GNU Radio/USRP platform, presented in Sections 2.1, 2.2 and 2.3, respectively.

## 2.1 Watermarking

### 2.1.1 Digital Watermarking

A traditional watermark is a hidden pattern in a sheet of paper, which is made by changing the fiber density of a certain area on the paper and can only be read when held against a light [1]. This technique is usually used to determine the authenticity of the carrier of the watermark, and it was introduced to provide copyright protection for digital products in 1990's, known as the digital watermarking technique [14].

In digital watermarking, a watermark signal can be embedded into a content signal or multiplexed with a content signal [4]. The multiplexing methods consume a portion of the channel capacity for carrying the dedicated watermark signal, and hence the watermark signal achieves the same quality as the content signal. On the other hand, embedding the watermark into the content signal fully utilizes channel capacity by sending the watermark signal and the content signal simultaneously, providing additional secrecy quality for the watermark signal, at the cost of degradation of content signal. In this sense, the watermark signal can be viewed as a secondary communication channel. Most digital watermarking techniques adopt the embedding method.

The digital watermarking technique has been widely adopted on the upper network layers. For example, digital watermarking can be applied on multimedia signals in the application layer, where copyright information is embedded into the least significant bits of multimedia data, concealing the existence of the hidden information as well as conserving the quality of the multimedia signal [14].

Digital watermarking is also applied on the transport layer for flow marking [34]. Flow marking is a method for network enforcement, where a watermark signal is superimposed on the communication flow at the transmitter side, and a detecting device is put at the suspicious receiver side to observe whether the watermark signal exists in the incoming traffic. The watermarking applications in the flow marking also requires the invisibility of the watermark signal and the integrity of the content signal in order to avoid alerting the suspicious receiver.

### 2.1.2 Steganography

Steganography is a security method that is often compared with watermarking. Steganography provides a covert channel for a pair of communication parties which need to conceal the existence of their communications. Redundancy in signals is the most exploitable feature for steganography, and hence this technique is popularly used in communication networks to utilize the unoccupied protocol fields or coding redundancy.

Both watermarking and steganography are information hiding techniques, and both of them can be visible or invisible. The main difference between these two techniques resides in whether the hidden information is related with its carrier. Steganography only utilizes the extra capacity of the carrier and conveys secret information irrelevant to the carrier, while the hidden information in watermarking carries descriptions about the carrier, such as identity, location or permission [3].

### 2.1.3 Physical Layer Watermarking

As the lowest level of the OSI network model [28], the physical layer provides the physical connections between communication parties and realizes the virtual data bits transmission. In wireless networks, the physical layer usually consists of hardware devices, modulation schemes, transmission medium and coding algorithms. Similar to the upper OSI layers, there exists a requirement for security services such as confidentiality, authentication and authorization on the physical layer.

Along with the development of SDR technique, which facilitates the implementation and verification of physical layer functions, the research of physical layer security has expanded greatly. Various physical layer properties are involved in the research of physical layer security. The physical layer watermarking specifically makes use of the randomness and the redundancy in the channel, signal and coding attributes of the physical layer, modifies one or more of the attributes to represent watermark signals, in order to provide signal tracing, additional authentication or access control in the physical layer.

## 2.2 DSSS

### 2.2.1 DSSS System Model

Figure 1 shows a generic DSSS system model [26]. In a typical DSSS system, the transmitter first modulates the data signal with a carrier signal, and then spreads the modulated signal by applying modulo-2 addition to it with a spreading signal. The spreading signal is generated from a PN sequence running periodically at a much higher rate than the original data signal. The spreading operation is shown in Figure 2. Each individual digit in the PN sequence is called a chip to be differentiated from the bit in data signal, and each period of the PN sequence is used to spread one data bit. Because the PN sequence is designed to resemble white noise, the spectrum of the original signal is spread out. Thus, the spectrum of the spread signal occupies a larger bandwidth and shows a lower power spectral density than that of the original signal.

Symmetrically, the receiver first performs a correlation process on the incoming signal, that is, it applies the modulo-2 addition to the incoming signal with a synchronized copy of the spreading signal. The receiver then obtains the underlying modulated signal, which is in turn demodulated to recover the original data signal. The duplicating modulo-2 addition provides interference rejection for the DSSS signal if the interference is narrow band, because modulo-2 addition of the narrow band interference with the spreading signal will spread out the power of the interference, and hence will increase receiving SNR of the signal of interest.

### 2.2.2 DSSS Demodulator

The DSSS demodulator applies the correlation operation in two processing phases: acquisition in synchronization establishment stage, and decoding in data

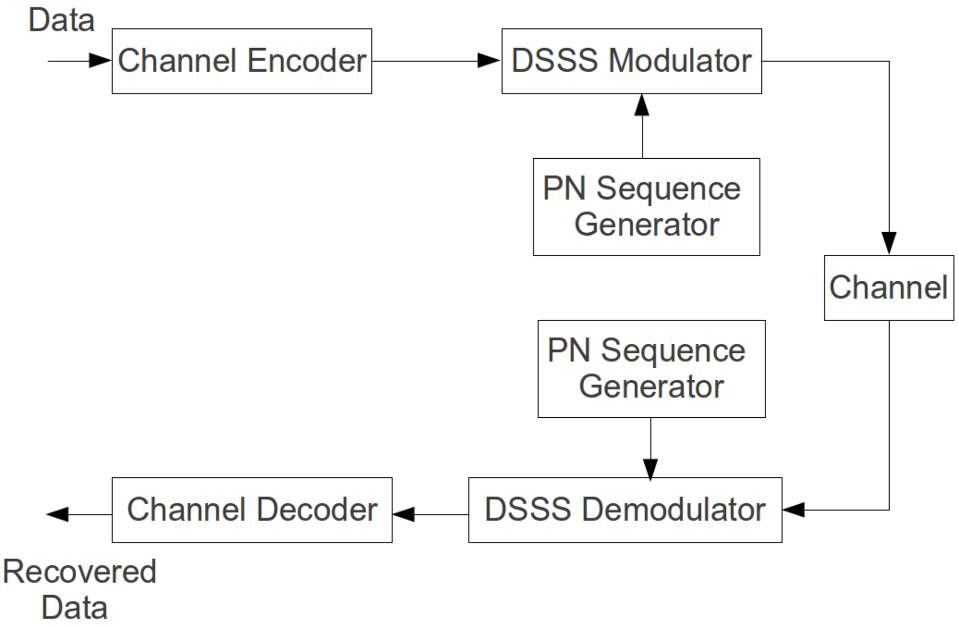


Figure 1: DSSS System Model

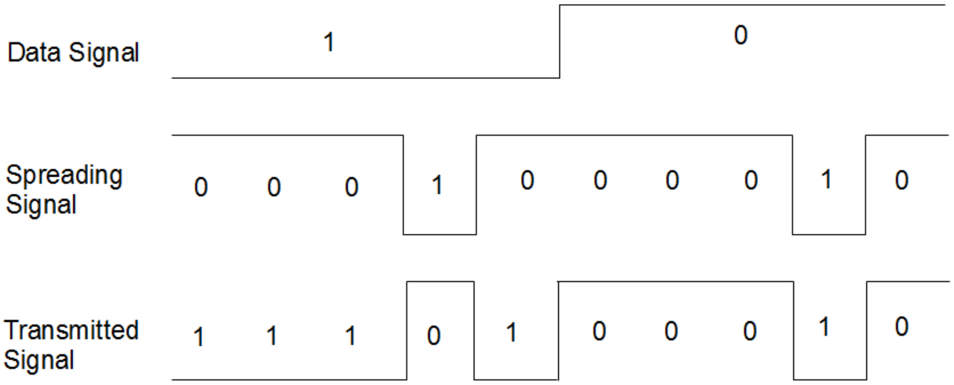


Figure 2: DSSS Spreading Operation. PN sequence in this example is 00010.

receiving stage.

In the acquisition phase, correlation can be performed in parallel or sequentially [31]. A parallel correlation system consists of a series of correlators. Each correlator corresponds to a phase shifting version of the PN sequence, apart from each other with 1/2 of the chip duration in time. The incoming signal is fed into this set of correlators simultaneously. A comparator circuit reads correlation results from these correlators and locates the largest one. The shifting version of PN sequence corresponding to the largest correlation result is then used to adjust the clock of the PN sequence generator and the synchronization is established. On the other hand, a sequential correlation system contains only one correlator, which correlates the incoming signal with a copy of the PN sequence and compares the correlation result to a threshold. If the threshold is satisfied, the synchronization is considered successful, otherwise the PN sequence shifts 1/2 of the chip duration, and the correlator performs the previous operations again, until the threshold is satisfied.

After the synchronization is established, the DSSS demodulator enters the data receiving stage and begins to decode the incoming signal. The correlation operation in this phase correlates the incoming signal with the spreading signal from the PN sequence generator, and compares the correlation result to a threshold to determine the value of a data bit.

In both phases, the correlation results are processed with a maximum-likelihood algorithm, so the incoming signal is handled as groups of chips in one period of PN sequence instead of each chip being handles individually, providing a possibility to hide information in the individual chips.

## 2.3 GNU Radio/USRP SDR Platform

SDR is a radio communication system consisting of the most essential signal processing components implemented in software form. Software defined components in SDR can function on a general purpose processor, and hence significantly reduce the cost of SDR compared to specialized hardware components. The programmability of software components also provides reconfigurability, which facilitates the compatibility of various communication standards and the development of new communication techniques, as well as reveals the prospect of cognitive communication systems [22].

### 2.3.1 GNU Radio

GNU Radio [7] is an open-source SDR development toolkit that provides software implementations of various digital signal processing components, such as filters, modulators, demodulators, equalizers and so forth. GNU Radio can be used independently as a simulation environment, where communication parties can be connected directly or via a simulating channel. GNU Radio is more powerful when coupled with simple external RF front end hardware to construct a radio communication system. In such scenarios, GNU Radio blocks are the software defined components of the SDR and can be configured with any sets of parameters to support any signal specific operations, and the external RF front end is thus only required to provide general purpose functions such as ADC, DAC, upsampling and downsampling. GNU Radio is so flexible in realizing new communication systems with low cost requirements that it is widely adopted in research, academic, and commercial environments.

The GNU Radio API is organized in a layered structure [29], shown in Figure 3. The core of GNU Radio architecture is the library of C++ classes, including performance-critical essential signal processing components, multi-threading scheduler, buffer and I/O managers, as well as data stream converters. The C++ classes can be accessed directly by software applications written in C++, or via SWIG, the Simplified Wrapper and Interface Generator. SWIG builds Python extension modules out of the C++ classes by processing interface definition files that contain declaration information of the C++ classes. The wrapper Python modules are then imported into Python applications or GNU Radio Companion GUI development environment to build flow graphs. A flow graph is the data structure defined in GNU Radio to represent a signal processing path.

A typical flow graph is shown in Figure 4. The blocks represent signal processing components and the edges represent the data stream passing along the signal processing path. The data stream is internally stored as a series of integers, float numbers or complex numbers in a shared circular buffer. The flow graph is actually driven by the data stream. Each block is the data producer to its downstream neighbor. At the flow graph construction stage, the GNU Radio scheduler allocates each block a thread, along with a read pointer and a write pointer to access the shared buffer. When the flow graph starts running, each block checks whether it has available output space and enough input items, and if the requirements are satisfied, that block starts its operation and continues until running out of its output allocation. When the source block stops producing output, it sends out a stopping signal and this signal is passed down the flow graph, and consequently all blocks stop after using up their input items, and at this point the flow graph is considered successfully executed.

### 2.3.2 USRP

A popular set of RF hardware that can be used with GNU Radio is the Universal Software Radio Peripheral (USRP), designed by Ettus Research [6]. Some USRP models are used as RF front end devices connected to a host computer via a USB or Gigabit Ethernet interface, while other models are built with an embedded processor

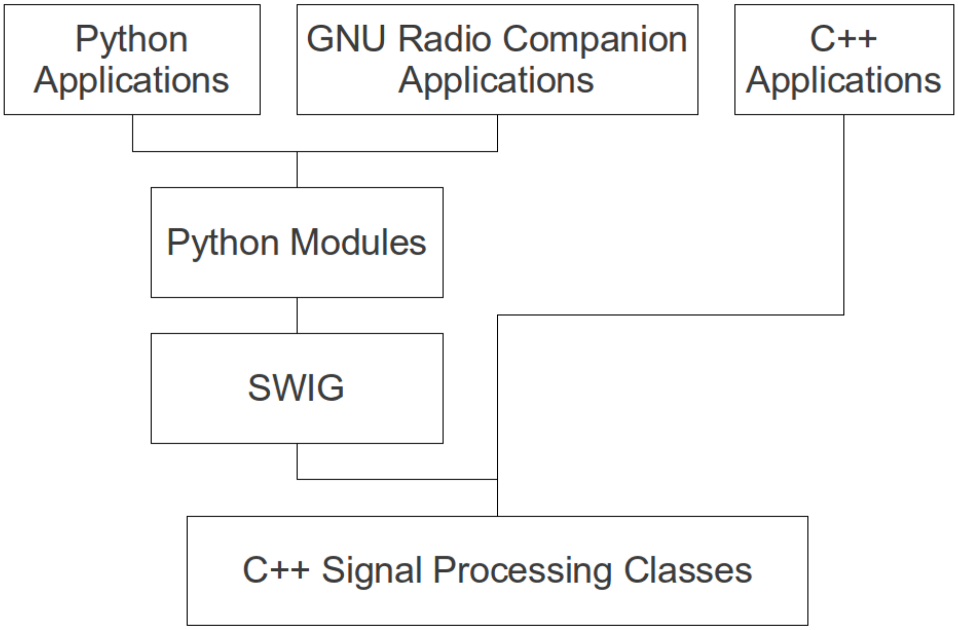

Python Applications
GNU Radio Companion Applications
C++ Applications
Python Modules
SWIG
C++ Signal Processing Classes

Figure 3: GNU Radio Layered API

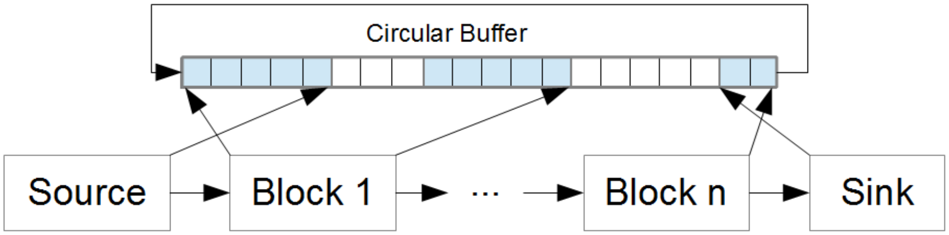


Figure 4: GNU Radio Flow Graph and Shared Circular Buffer

| Specifications | USRP2 | USRP1 |
|---|---|---|
| FPGA | Xilinx Spartan 3-2000 | Altera Cyclone EP1C12 |
| ADC | 14-bit, 100 MHz | 12-bit, 64 MHz |
| DAC | 16-bit, 400 MHz | 14-bit, 128 MHz |
| RF Bandwidth to/from host | 25MHz @ 16bits | 8MHz @ 16bits |
| Interface | Gigabit Ethernet | USB 2.0 |

Table II: Specification Comparison of USRP2 and USRP1 [8]

and memory and thus can be used as standalone devices. When the USRP is used as a RF front end, its complete setup includes a motherboard, with one or more daughterboards and antenna sets attached, as well as a USB or Gigabit Ethernet cable linking it to the host computer. The motherboard of the USRP is equipped with high speed signal processing hardware blocks, including FPGA, ADC and DAC, and can support sampling rates up to 100M samples per second. The daughterboards are the frequency specific front-ends in the range of 0 Hz to 6 GHz. USRP2 was the first evolution from the original USRP1. The comparison between USRP2 and the USRP1 on the important specifications are listed in Table II [8]. This thesis used USRP2 to implement the prototype WDSSS system given its higher sampling rate and wider bandwidth. Figure 5 shows a picture of one of the USRP2 units used for the prototype implements.

There are several important parameters required to configure the USRP2 so that it can function correctly. First, a USRP2 need to know the target center frequency to tune its antenna. Then, an interpolation rate is necessary for a USRP2 in transmission mode. The DAC in USRP2 handles samples at a constant rate, 100M samples per second, which is usually more than adequate for samples generated from user programs in the host computer. The USRP2 therefore applies an interpolator to convert the samples from the host computer to accommodate the clock rate of DAC. Similarly, a USRP2 in receiving mode requires a decimation rate in order to reduce the sample rate before transferring the received signal to the host computer.

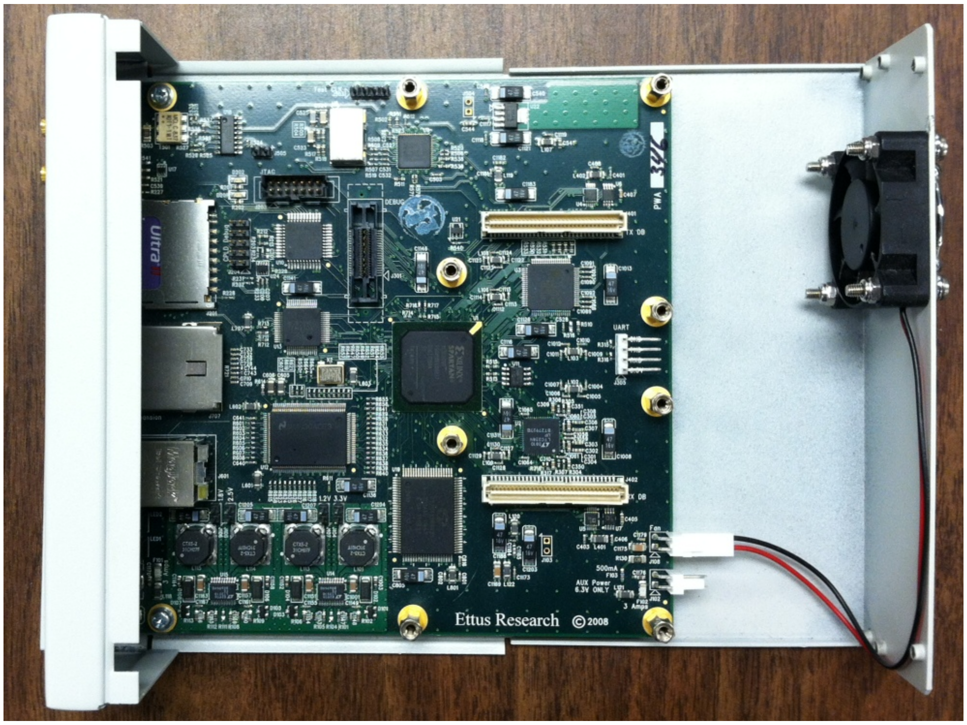


Figure 5: USRP2

### 2.3.3 Related SDR Projects with GNU Radio/USRP

DSSS has been implemented in several SDR projects on the GNU Radio/USRP platform, such as the SPAN 802.11b receiver, the BBN 802.11b project and the UCLA ZigBee PHY implementation. The WDSSS prototype system is different from the above projects, in that it treats every single chip individually, and hence makes use the coding redundancy of the PN sequence. The following is the introduction to the related DSSS projects.

The SPAN 802.11b receiver project implements the matched filter for the correlation processing in the FPGA of USRP [20]. The USB interface between USRP and the host computer is not capable of providing sufficient bandwidth for the 802.11b signal, since the DSSS-based 802.11b signal requires 22 MHz bandwidth while the maximum bandwidth for the USB interface is only 8 MHz. The authors thus placed the correlation operation in the FPGA of the USRP, so that the sample rate can be reduced to a sustainable value for the USB interface. Their implementation reduced the sample rate and computational complexity for the host computer and meanwhile provided a full bandwidth 802.11b receiver.

The BBN 802.11b project includes a transmitter and a receiver [32]. A previous version of this project was an 802.11b receiver implemented on USRP. To overcome the same bandwidth problem that the authors of the SPAN 802.11b receiver faced, this project applies a bandwidth reduction method, decimating the incoming sample rate to 8M Hz prior to feeding them into the USB interface in the USRP. The resulting signal is distorted due to the bandwidth loss and only the 1M bps signal can survive. The updated version is implemented with USRP2, which eliminates the bandwidth limitation for the 802.11b signals. The spreading and correlation operations are implemented as the matching filter combining with a rate converter in the transmitter and the receiver, respectively. The 11-chip Barker Sequence is used as the taps of the matching filters in both sides. The USRP2 version of BBN 802.11b project provided a complete 802.11b system built on the GNU Radio/USRP platform, and is widely adopted in 802.11b signal research.

The UCLA ZigBee PHY project implemented a transmitter and a receiver in the 2.4G Hz band of the IEEE 802.15.4 [2]. Their work deals with the set of 16 32-chip PN sequences and the underlying OQPSK modulation scheme. The DSSS spreading operation maps a 4-bit symbol into one of the 16 PN sequences, and sends the PN sequence instead of the symbol. Conversely, the correlation process in the receiver compares the incoming chips to all of the 16 PN sequences, and the PN sequence with the minimum Hamming distance to the chip set is chosen. Meanwhile the distance is compared to a threshold, and if the threshold is satisfied, the index of the PN sequence is rendered as the symbol value.

# CHAPTER III

# RELATED WORKS

This chapter reviews existing works in physical layer security methods utilizing DSSS, physical layer watermarking techniques, and physical layer steganography schemes.

## 3.1 DSSS in Physical Layer Security

A lot of research endeavors have looked at physical layer security in commercial DSSS systems, managing to provide additional security service. The authors of [19] proposed to encrypt the scrambling sequence of CDMA systems with the Advanced Encryption Standard (AES), in order to hinder the potential detection and recovery of PN sequences via intercepting sufficient transmission signals. Several individual works considered expanding the PN sequence set used in a DSSS system, such as uncoordinated PN sequences [25] and time hopping PN sequence management [10] and [23].

The authors of [25] aimed to unleash the PN sequences from being a shared secret. In their system, the PN sequences are placed in a public set, from which the transmitter randomly chooses one to spread its data signal, and consequently, the receiver needs to search the whole set to locate the one for its received signal. The major drawback for their method is the low efficiency in the receiver program. Thus, they suggested to employ multiple transmitters to individually send the same data signal with different PN sequences through a band of channels. In this way, the receiver gains some time efficiency by trading the channel efficiency.

Both [10] and [23] applied a hopping scheme to manage an expanded set of PN sequences. Compared to the totally uncoordinated PN sequences technique [25], these two techniques allow the transmitter and the receiver to share a PN sequence hopping scheme, which provides certain level of synchronization, and hence facilitate the receiver in the decoding process. The authors of [23] took one step further from [10] to implement the suggested PN sequences hopping scheme on the GNU Radio/USRP platform. In their prototype system, the synchronization protocol generates the same set of PN sequences at both sides and designates the hopping sequence. The transmitter has a Code-Life-Time to determine active PN sequence for spreading, and the receiver first decodes the incoming signal with the current active PN sequence and will move on to the next sequence if current one does not produce the satisfied result. The authors also assigned different preamble sequences corresponding to different sets of PN sequences to further blur the well known samples. Their design specifically addressed issues in performance loss, security gain and synchronization. Their experimental results demonstrated 1 dB and 2 dB in performance loss in the scenarios of a 60dB attenuating cable connection and wireless channel, respectively. In addition, the authors designed an interception attack algorithm to evaluate the security property of the system.

## 3.2 Works in Physical Layer Watermarking

The works in this section combined the watermarking technique with diverse physical layer properties to provide authentication service in the physical layer.

The authors of [9] designed a physical layer watermarking approach for licensed user authentication in the Digital TV (DTV) spectrum. The coexistence of licensed DTV stations and broadband users necessitates the research for methods to maintain the spectrum access priority of licensed users. The authors managed to integrate a fingerprinting technique with typical receiver preprocessing algorithms such as equalization. Their watermark signal is composed of the finite length synthetic FIR channel response, which equivalently introduces a slight linear inter-symbol interference to the content signal. Consequently, the receiver equalizer applies two individual tap weight vectors corresponding to the synthetic FIR channel responses and the real FIR channel components separately, and recovers the watermark signal by removing the estimate of the real FIR channel components. They also presented a cryptographic watermark design method by generating a digital signature with a hash function in terms of time, location and frequency of the transmitter. The authors introduced two properties to evaluate their authentication signal, namely the ratio of the authentication symbol rate to the primary signal rate (ASR) and the ratio of the authentication signal power to the primary signal power (ASP). Their Monte Carlo simulation results showed that embedding a watermark signal has little effect on the bit error rate (BER) of the content signal. The BER of the watermark signal is improved by decreasing the ASR, while the channel estimation mean square error is only slightly increased. In terms of power ratio, the ASP is required to exceed 0.002 for a watermark signal to outperform the content signal, while the ASP is determined by the properties of the equalizer.

Paper [33] proposed a framework to superimpose a watermark signal into a content signal in a way that would not consume extra bandwidth. The watermark signal in their system is generated by the following steps. The first step is applying a function to the content signal and the secret key to generate a sequence. The sequence from the first step is then padded to match the length of the content signal. Finally the watermark signal is combined with the content signal using a small portion of the total transmission power. The generating function used in the first step is responsible for the security property of the watermark signal, so it should be chosen to establish an uncorrelated relationship between the content signal and the watermark signal, but still needs to relate the watermark signal with the length of the content signal. The aware receiver can apply the same generating function used by the transmitter to the recovered content signal and the known secret key to recover the watermark signal. The authors suggested to use robust hash functions as the generating function to overcome the decoding errors in the recovered content signal.

Two physical layer watermarking techniques were proposed in [16], applying on the payload or the cyclic time of an orthogonal frequency-division multiplexing (OFDM) signal, respectively. The first method, named constellation dither (CD), spreads QPSK modulated watermark bits with a Gaussian distributed PN sequence and then superimposes them onto the OFDM payload data at a relatively low power level. It is equivalently to add an additive white Gaussian noise with a low power level to the OFDM payload symbols. The receiver then detects the watermark signal by applying a match filter and a QPSK modulator to the received signal, with the knowledge of the power allocation of the watermark signal and the content signal. Thus, the detection performance of CD technique depends on the spreading processing gain and the power allocation parameter.

The second method, baud dither (BD), uses a positive or a negative cyclic time shift to represent a watermark bit of 1 or 0, which is encoded with the Manchester coding scheme, and hence it can be viewed as intentionally introducing time jitter to the OFDM symbols after attaching the cyclic prefix. The receiver deduces the watermark signal by looking at the shifted version of OFDM symbol. If the last sample appears at the beginning of the symbol, the shift is considered positive. On the other hand, if the first sample appears at the end of the symbol, the shift is considered negative. The positive or negative sign is then interpreted into a watermark bit. The orthogonality of OFDM is still preserved because the largest multi-path delay is less than the difference between the guard interval length and the introduced time shift. The detection performance of BD technique depends on the received SNR and the number of pilots used in the receiver tracking circuit.

The authors implemented a testbed for the CD watermark technique, and provided analytical and simulation results for both CD and BD watermark techniques. Analytical BER is calculated using the literature BER equation for BPSK and QPSK. According to the results, if the power portion for the CD watermark signal is less than or equal to 0.01, there is only slight degradation of the content signal's BER. The BD watermark signal's BER is much better than the OFDM content signal's BER, while the BD watermark signal has negligible influence to the OFDM content signal's BER. Both methods can achieve 10s kbits/sec capability for the watermark signal and hence can provide enhanced authentication services for dynamic network environments such as roaming ad hoc networks.

The author of [17] proposed a physical layer watermarking method utilizing the $M$-ary phase-shift keying (M-PSK) modulation schemes. This work adopted the adversary model, configurable features and measurement merits of stealth, robustness and security from [33], and can be viewed as a type of constellation dithering similar to [16], but was implemented on the M-PSK constellation instead. The principle behind the design is that the M-PSK modulator encodes $\log_2 M$ bits of data into one symbol, which is a point on a constellation map. Different values of M result in a different number of bits per symbol and different phase distances between points on the constellation map. An M-PSK demodulator then matches each received symbol to the points on the constellation map, and locates the point that has a minimum distance with the received symbol, considering the effect from channel fading, additive noise and other interference. The physical layer watermarking method presented in this paper uses a fraction of the phase distance of the content signal to embed the watermark signal. Content bits and watermark bits are encoded together for a single transmitted symbol. The combined signal is then transmitted using a PSK scheme with higher M value. The watermark signal appears as noise to unaware users who only know and apply the content signal PSK to analyze received signal. Meanwhile the watermark signal can be detected and recovered by aware users who decode the incoming signal with the higher M-PSK scheme and analyze the relative shift of the recovered signal to the content signal. There is a trade-off between the SNRs of the content signal and the watermark signal when choosing the shifted phase distance for the watermark signal. The author suggested that this method can be extended to QAM systems by altering the QAM order.

The author developed the method on the GNU Radio/USRP platform, along with the use of Matlab. GNU Radio and USRP were responsible for the communication processing, and Matlab was used to encode the content signal with the watermark signal as well as to analyze the received data. The modulation pair of DBPSK and D8PSK was used in his implementation, as shown in Figure 6. The transmitter encodes symbols of 3 bits with D8PSK, and then unaware users decodes 1 bit with DBPSK, while aware users decode 3 bits with D8PSK.

The author evaluated his work by first deriving the BER performance from the theoretical BER of PSK generated from the BERTool GUI in Matlab. The analysis

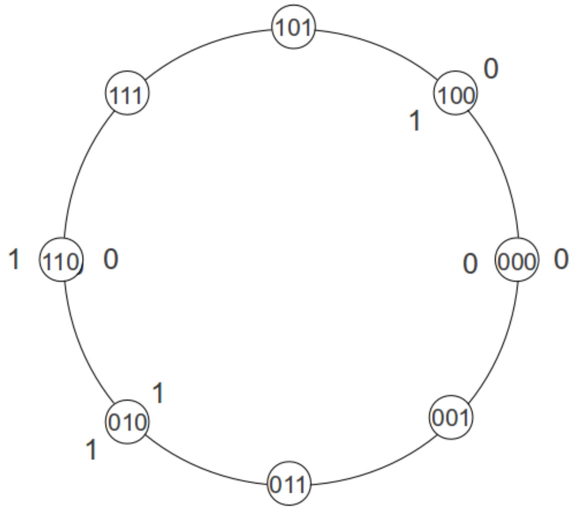


Figure 6: Watermarking on M-PSK [17]: the bits outside the large circle are content data bits; the bits inside the large circle are watermark data bits; the bit patterns in the small circles are the constellation points for D8PSK, which is used to encode the embedded signal.

concluded that if only PSK is considered, the separation phase between two symbols can be used directly to deduce the effective power when calculating bit energy to noise density ratio. Because the proposed system was implemented with DPSK, where the demodulator decodes data by using the previous symbol as a phase reference for the current symbol, the rotation caused by noise or interference would introduce extra bit error. The estimated watermark BER is roughly 1.5 times of the BER of an ordinary D8PSK signal. The author then created two Monte Carlo simulations to further estimate the BER of the proposed watermarking method. Finally, the BER of the content signal and the watermark signal were measured in a real-world wireless environment with various distances between the transmitter and the receiver, for the unaware and the aware receptions. BER values between different pairs of PSK modulation schemes were also compared. The observed BER values in all scenarios were compared with the estimated BER curves and showed overall better results, despite the existence of a BER floor caused by Wi-Fi signals. When in the identical environment, the content signal with watermark had worse performance than when the content signal was transmitted without watermark.

The authors of [30] presented another watermarking algorithm, in which watermarking bits are bit-wise multiplied with spreading codes and the results are summed, the sum is then multiplied by an intensity parameter alpha, and then added selected modulated data bits. The watermark bits are extracted from the transmission by cross correlating selected bits with the spreading codes, producing a result with positive or negative sign indicating the value of watermark bits.

## 3.3 Works in Physical Layer Steganography

Steganography is another technique that can be used to provide physical layer security services. Paper [15] explored all permutations of $2^{32}$ PN sequences used in the IEEE 802.15.4 standard and found that since the minimum distance is 12 and maximum distance is 20, there are $145,742,202$ chip sequences can be mapped to each symbol. The author thus exploited this encoding redundancy, and assigned a certain amount of chips in a PN sequence to represent the data of the steganographic channel. The embedding process for the PN sequence is achieved by applying the following formula: $newchipseq = chipseq \oplus stegmask$; where *newchipseq* is the chip sequence used to spread the data symbol, *chipseq* is the original 32-chip sequence for the data symbol in the IEEE 802.15.4 standard, and *stegmask* is the codeword representing the embedded steganographic data.

The security for the system was mainly realized by generating the *stegmask* using a counter-mode block cipher with a counter value of the Absolute Slot Number (ASN), which can count the number of time-division slots since network creation, concatenated with a 0-based counter, so that the *stegmask* appears to be a random error to a third party. The performance of the steganographic channel with various capacities was evaluated by simulation. According to the simulation results, a steganographic channel with the same data rate as the content signal can maintain the content signal performance while providing a relatively high data rate for the secret signal.

A similar technique is proposed in [21]. This paper mainly presented the encoding scheme of steganographic data bits and sought a method to minimize the effect of the steganographic signal on the underlying 802.15.4 data signal. The authors focused on an AWGN channel and considered the effect of signal coding with a given modulation scheme. According to the observations of the authors, when data bits are encoded with a larger code words, the energy per bit to noise ratio required for a certain SNR value is smaller, and this implies a higher noise tolerance ability. The larger the Hamming distance between the symbol code words, the higher the noise tolerance. Specifically in IEEE 802.15.4, where the PN sequences are the symbol code words, the minimal Hamming distance between the PN sequences is 12, which provides a minimum of 6 chip error tolerance to ensure symbol recovery. The authors then deduced the minimum acceptable SNR as -2.2 dB, considering the allowable maximum symbol error rate of $1.9 \times 10^{-4}$ and a coherent QPSK detector.

The authors aimed to convey secret information through an 802.15.4 system, to achieve maximum steganographic channel capacity, and meanwhile to maintain the performance of the original 802.15.4 signal. Thus, they proposed an encoding scheme, which expands each original 802.15.4 PN sequence into a set of PN sequences in such a way that each derived set of PN sequences are closer to the corresponding original PN sequence than to any other PN sequences in the original 802.15.4 set. This encoding scheme allows an unaware receiver to recover the 802.15.14 signal with normal correlation procedure. In addition, the authors designed two ways for the aware receiver to decode the embedded steganographic data. One way is to recover the underlying 802.15.4 signal first and then to decode the steganographic

| | |
|---|---|
| C0.0 =C0 = d9c3522e | C0.16 = 5bd350aa |
| C0.1 = 19e3da2a | C0.17 = 5ce3d02e |
| C0.2 = 31f3522e | C0.18 = 91d2da2e |
| C0.3 = 45c35a2f | C0.19 = c1b3d22e |
| C0.4 = 4983da6e | C0.20 = c3c3d06e |
| C0.5 = 49d74a2e | C0.21 = d0d3d42e |
| C0.6 = 515b526e | C0.22 = d1c38b2e |
| C0.7 = 51935a3e | C0.23 = d1d3d2a2 |
| C0.8 = 51d3512f | C0.24 = d1eb1a2a |
| C0.9 = 51e7c22e | C0.25 = d1f35246 |
| C0.10 = 53e3126e | C0.26 = d5c3926a |
| C0.11 = 55c3546e | C0.27 = d913d82e |
| C0.12 = 58f35a26 | C0.28 = d9c7c82a |
| C0.13 = 59c3196e | C0.29 = d9d39a24 |
| C0.14 = 59d35c2c | C0.30 = d9f2580e |
| C0.15 = 59e34a4e | C0.31 = ddd3d00c |

Table III: One expanded set of PN sequences for steganography in 802.15.14 [21]: C0 is the original PN sequence for symbol value 0; C0.1 - C0.31 are the derived PN sequences from the original PN sequence; the minimum distance in this PN sequence set is 6.

data by searching the corresponding set of PN sequences. The other way is to use the expanded set of PN sequences to recover the 802.15.4 signal and the steganographic data at the same time. Receivers using these two different decoding methods are referred to as a hierarchical receiver and a full receiver, respectively. There is a trade-off between the decoding ability of 802.15.4 data and steganographic data due to the distance among the set of PN sequences for one particular original PN sequence. Table III shows one example of the expanded PN sequences set.

Their simulation results for the proposed method showed that the minimum acceptable SNR was increased to 1.95dB for the underlying 802.15.4 content signal. For the steganographic data decoding, when considering same amount of sent symbols, the full receiver only requires a 1.5 dB minimum SNR, while the hierarchical receiver shows a SNR closer to the underlying 802.15.4 content signal. The full receiver can achieve an effectively lower symbol error rate because fewer symbols are required for a complete packet, and this led to a 1.2 dB minimum SNR. In summary, the

SNR increased 3.4 dB compared to the original 802.15.4 signal while the data rate is increased 3.5 dB, so this yields a 0.1 dB decrease in energy per bit to noise ratio.

The authors of [35] proposed a steganography scheme based on [15] as well as adopting the Hamming distance maximization method in [21]. They provided additional discussion in terms of error correcting capability in the code words design. They suggested to utilize chip positions to carry steganogaphy information. The steganogaphy information is then recovered by obtaining altered chip positions information after comparing the received signal to the standard sequences. Thus the size of the covert communication alphabet is equal to the sum of the combinations of each possible number of altered chips. However, they deduced that the optimal number of altered chips is five for the purposes of interference resistance. According to the simulation results of this paper, the BER and receiver sensitivity are bound by the number of altered chips, and an optimal embedding scheme is 4 steganographic bits for each PN sequence.

## 3.4 Summary

The WDSSS system in this thesis employs a similar technique to modify PN sequences as in [15], [21] and [35]. However, instead of only providing simulation results, this thesis implements a prototype system and obtains experimental performance results for the proposed WDSSS technique. Another similarity can be found in [23], since the modified PN sequences in WDSSS system can be viewed as an expanded code set. The major difference between [23] and this thesis is that the work in [23] only needs to decode the underlying system and ignores any chip errors, while this thesis utilizes the extra PN sequences to carry additional information when maintaining the underlying system function.

# CHAPTER IV

# WATERMARKED DSSS

## 4.1 WDSSS Technique Overview

The goal for designing the WDSSS technique is to embed authentication information in the PN sequences of the DSSS technique in an imperceptible manner, so as to complement the physical layer security for a DSSS system with publicly known PN sequences. Two fundamental premises for the proposed watermarked DSSS technique are the coding redundancy of PN sequences and the correlation procedure of the DSSS receiver. In order to provide robust communications, PN sequences are designed with high error correcting capability, which implies coding redundancy in good communication environments. The coding redundancy can consequently be exploited to carry additional information without requiring extra bandwidth. The correlation procedure of the DSSS receiver also reassures the viability of embedding secret information in the PN sequences, since only the correlation results of the incoming signal with the entire PN sequence will be used to recover the data bits and the artificial chip errors are overlooked. In this context, PN sequences are no longer secret keys and become the hosts for authentication information.

The WDSSS technique consists of two major operations, watermark embedding and extraction, which reside in a transmitter and an aware receiver, respectively. Figure 7 shows a communication system model realized with the WDSSS technique. A watermark embedding processing block and an extraction processing block are added in the classic DSSS signal processing paths.

In the transmitter, the watermark embedding processing block flips chips in the PN sequence at positions indicated by the watermark information, and then the PN sequence with flipped chips is used to spread the content bit. In the aware receiver, the demodulator correlates incoming signals with the original PN sequence and determines content bit values by comparing the correlation results with a threshold. The unmatched chip positions are then passed to the watermark extraction processing block, which in turn translates the position information into watermark bits. On the other hand, in an unaware receiver without the watermark extraction processing block, the demodulator still can recover the content bit values based on the correlation results, but ignores the specific positions of the error chips since it does not examine the individual chips.

## 4.2 Watermark Embedding Methods

The WDSSS technique embeds watermark information by flipping chips on designated positions and extracts watermark information by identifying flipped chip positions. Thus, the key point for the WDSSS technique is to establish a mapping relationship between the watermark data bits and the chip positions in the PN sequence.

One simple method is to exhaustively use all combinations of chip positions to represent watermark data bits. For a PN sequence of length $n$, if $m$ chips can be

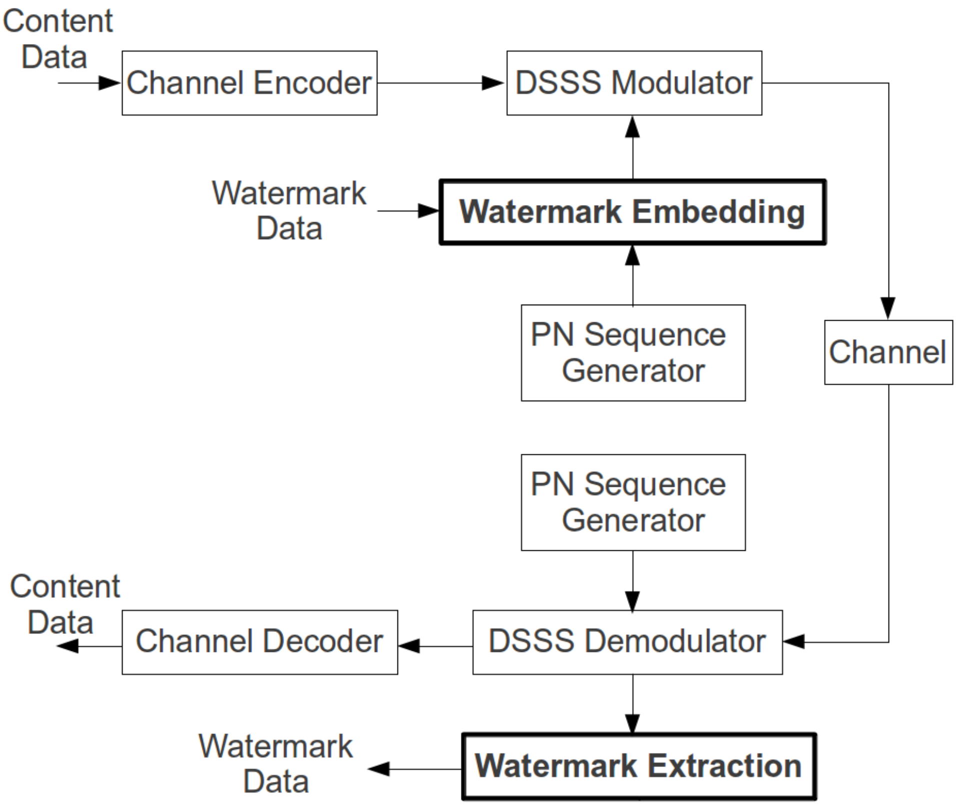


Figure 7: WDSSS System Model

flipped, the number of position combinations is $\binom{n}{m}$. Thus, the number of watermark bits that can be represented by one modified PN sequence is $\lfloor \log_2 \binom{n}{m} \rfloor$. Because one PN sequence is equivalent to one content data bit, this embedding method can provide a much higher watermark data rate than the content data rate. However, this simple method has several weaknesses. First, it can not provide security for watermark information since the combinations of chip positions are straightforward. Second, the extraction process is time consuming because the number of combinations increases exponentially when the number of flipped chips increases. Third, this method is vulnerable to noise because it does not provide error correcting capability. In order to provide more advantages in terms of security of watermark data, processing efficiency, robustness, and embedding capability, two other embedding methods were designed and implemented, namely the maximized minimum distance method and the subsequence method.

The maximized minimum distance method views the set of modified PN sequences with flipped chips as a set of code words. According to coding theory [26], the error correcting capability $t$ of one set of code words is determined by the minimum Hamming distance $d_{min}$ of the set:

$$t = \lfloor (d_{min} - 1)/2 \rfloor \tag{4.1}$$

As shown in Figure 8, the centers of the circles are chosen to be the code words with maximized minimum distance in this code space, the radius of each circle is half of the minimum distance. All code words within the circle can be decoded as the center code word, that is, any errors that cause the center code word to transform into any code word within the circle can be corrected. Therefore, maximized minimum distance can provide optimal error correcting capability. Specifically in the WDSSS system, the generating method for such a set of PN sequences is to flip chips at non-overlapped positions. Table IV shows the sets of modified PN sequences for the

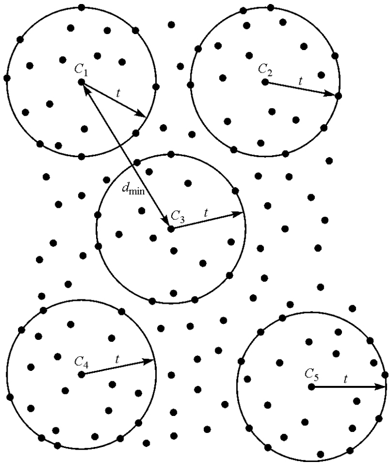


Figure 8: Minimum Distance Illustration [26]

| Watermark Bit Value | Modified PN Sequence | Flipped Positions | Total Embedding Capability (bits) |
|---|---|---|---|
| 0 | 01001100101 | 9, 7, 3 | 1 |
| 1 | 00010101000 | 6, 2, 0 | |

Table IV: Maximized minimum distance method: A Set of Modified PN Sequences for the 11-chip Barker Sequence (00011101101) with 3 Flipped Chips

11-chip Barker Sequence (00011101101) with 3 flipped chips. The resulting set has a minimum distance of 6.

As a result, both the embedding and extraction processes for this method are performed by a simple mapping table lookup. This method provides security for watermark information because the set of PN sequences is the shared secret for the transmitter and the aware receiver. This method also provides optimal error correcting capability with the maximized minimum distance. The drawback of this method is its low embedding capability, because there is a limited number of PN sequences that can satisfy the maximized minimum distance requirement.

The other method designed for this paper is the sub-sequence method. The basis of the sub-sequence method is to divide the PN sequence into sub-sequences, flip one chip in each sub-sequence and then combine the flipped chips together to rep-

| Flipped Chips | Sub-sequence ID | Starting Position | Embedding Capability (bits) | Total Embedding Capability (bits) |
|---|---|---|---|---|
| 1 | 0 | 0 | 3 | 3 |
| 2 | 0 | 0 | 2 | 4 |
| | 1 | 4 | 2 | |
| 3 | 0 | 0 | 1 | 5 |
| | 1 | 2 | 2 | |
| | 2 | 6 | 2 | |
| 4 | 0 | 0 | 1 | 5 |
| | 1 | 2 | 1 | |
| | 2 | 4 | 1 | |
| | 3 | 6 | 2 | |

Table V: Sub-sequence Watermark Embedding Method for 11-chip PN Sequence

resent a watermark value. Table V shows an example of the sub-sequence embedding method which flips up to 4 chips in the 11-chips Barker Sequence. The embedding process for the sub-sequence method first divides the watermark bytes into bit groups corresponding to sub-sequences. Then for each bit group, the position of the chip to flip is calculated as the sum of the start position of each sub-sequence and the bit value. Finally the chips at the determined positions are flipped. The modified PN sequence is then sent to the DSSS modulator for spreading. On the receiver side, the watermark extraction block first records the flipped positions from the DSSS demodulator, and then for each flipped position, calculates the bit value by subtracting the start position of the corresponding sub-sequence from the flipped position, and concatenates all bit values to recover the watermark byte.

This method can also provide security for the watermark information because the sub-sequence dividing scheme is the shared secret between the transmitter and the aware receiver. The embedding and extraction processes are efficient because the processing time is linear with the number of flipped chips. As shown in Table V, this method provides high embedding capability. However, this method does not provide error correcting capabilities for the watermark signal because the minimum distance is 2 in the sets of modified PN sequences.

The sub-sequence method and the maximized minimum distance method are implemented in the prototype WDSSS system of this thesis. Their performance results are analyzed and compared in the next chapter. In terms of the security property of the watermark signal, these embedding methods can provide different security levels to meet the requirements of different applications. The maximized distance method is suitable for critical watermark information, which is worth the price of a lower watermark bit rate. On the other hand, the sub-sequence method can be applied to less important watermark information with more tolerance to decoding errors.

## 4.3 Prototype System Implementation

The WDSSS prototype system was developed using an incremental software building methodology. First, the digital communication example in the GNU Radio package was adopted to construct the underlying DBPSK system. Then a generic DSSS system was built on top of the DBPSK system. Finally, the WDSSS system was implemented by expanding the DSSS transmitter program and the DSSS receiver program to include watermark embedding and extraction processing blocks. The rest of this section will explain the implementation details.

### 4.3.1 Baseline DBPSK System

The baseline DBPSK communication system provides the fundamental communication functions. Important processing blocks along the transmission path of the DBPSK system are shown in Figure 9. Their functions are described as follows:

| Preamble<br>2 bytes | Access code<br>8 bytes | Packet length<br>4 bytes | Packet number<br>2 bytes | Payload<br>variable | CRC32<br>4 bytes |
|---|---|---|---|---|---|

Table VI: Packet Format

- Data source - This block generates data packets. The packet format is shown in Table VI. This processing block reads in arbitrary number of bytes from a random binary file. A packet number is then assigned for that batch of bytes. Next the CRC field is then calculated. The payload and CRC fields are then masked with the output of a 15-LFSR, and prefixed them with the preamble, access code and packet length.

- Byte unpacker - This block treats the packets from the data source as a byte stream and decomposes each byte into a stream of bits.

- Symbol mapper - This block maintains a symbol mapping table and converts data bits into symbol bits using gray code. The conversions for this DBPSK system are 1 to 1 and 0 to 0, since each symbol only contains one bit.

- Differential encoder - This block calculates output bit $y[n]$ based on input bit $x[n]$ and one previous output bit $y[n-1]$ with the equation:

$$y[n] = x[n] \oplus y[n-1] \tag{4.2}$$

- Constellation encoder: This block converts symbol bits into complex constellation symbols. The conversions for this DBPSK system are 1 to $-1+0j$ and 0 to $1+0j$.

- Matched filter - This block implements a root raised cosine matched filter with roll-off factor 0.35, and also produces complex samples out of constellation symbols.

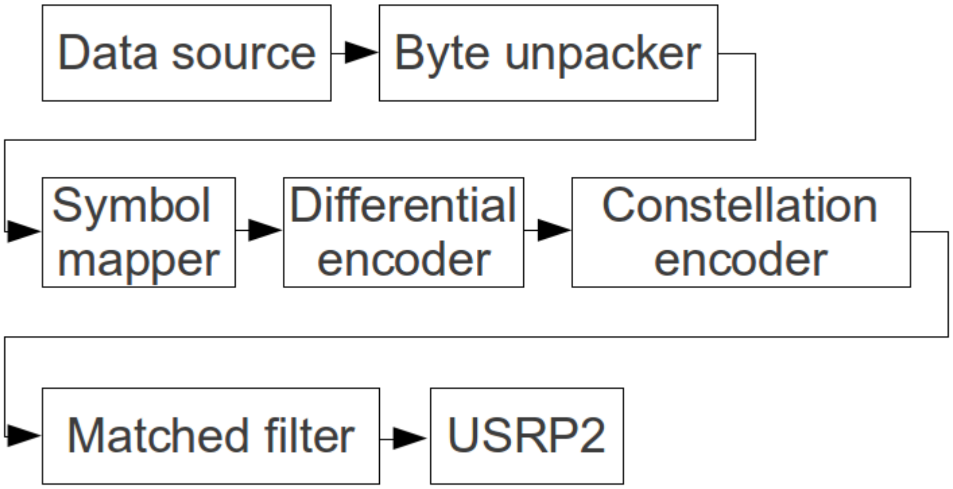


Figure 9: DBPSK System Transmitter Flow Graph

- USRP2 - This block represents signal processing occurring in the USRP2 platform. USRP2 draws samples from the host computer, upsamples them to accommodate the clock frequency of the DAC, and finally transmits the resulting analog signal via antenna.

As shown in Figure 10, processing blocks along the receiving path of the DBPSK system provide symmetric functions to blocks in the transmitter. Their detailed functions are described as follows:

- USRP2 - This block represents signal processing occurring in the USRP2 receiving path. USRP2 tunes the antenna to the same center frequency as that in the transmitting USRP2, receives analog signals via antenna, converts them into digital samples, decimates samples to the sample rate required by the host computer, and sends the samples to the host computer via the Ethernet interface.

- Channel filter - This block implements a low pass filter to obtain samples from the desired channel.

- Matched filter - This block is a root raised cosine filter with roll-off factor 0.35

to match the transmitter root raised filter so as to align samples .

- MPSK receiver - This block processes samples in one symbol interval to estimate the symbol value. This block also tracks the carrier phase and frequency errors with a Costas loop, and corrects symbol timing errors with a modified Mueller and Muller algorithm.
- Differential decoder - This block calculates output complex symbol $y[n]$ based on the phase difference between the current input symbol $x[n]$ and one previous input symbol $x[n-1]$ using the equation:

$$y[n] = x[n] \times x^*[n-1] \tag{4.3}$$

- Constellation decoder - This block determines the correct symbol value by selecting the constellation point that has the minimum Euclidean distance with the estimated symbol value, and also converts complex symbol values into symbol bits.
- Symbol mapper - This block reverses gray encoding and converts symbol bits into data bits.
- Byte packer - This block groups data bits into bytes.
- Frame sink - This block correlates the byte stream with known preamble and access code to reconstruct packets, and inspects these packets with CRC32 to determine whether the packets are correct.

### 4.3.2 Generic DSSS System

In order to implement a DSSS system on top of the baseline DBPSK system, a DSSS modulator and a DSSS demodulator are added to the transmitter and the

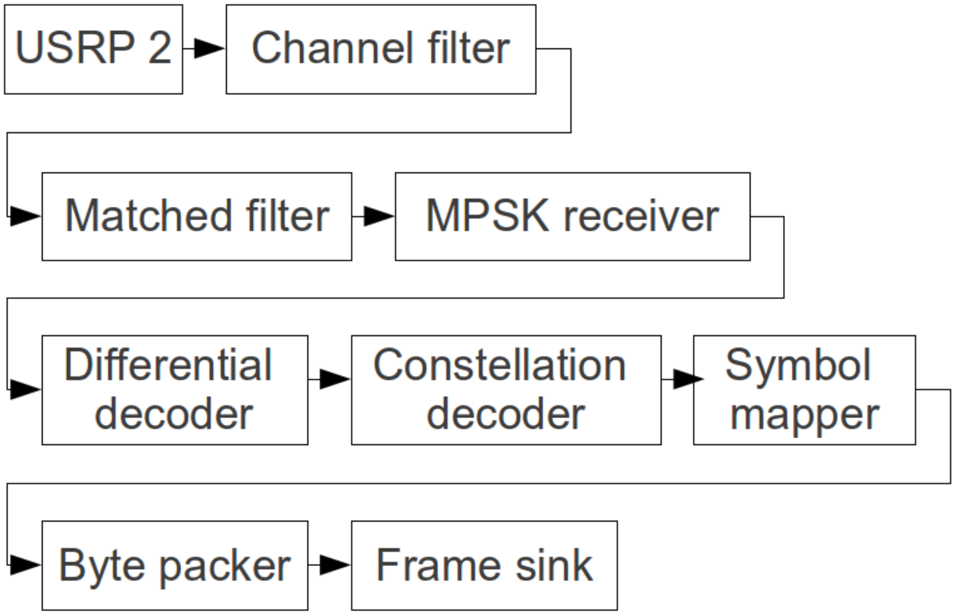


Figure 10: DBPSK System Receiver Flow Graph

receiver respectively. A rate converter is also included in both paths to deal with the conversion of the sampling rate and the USRP2 parameters caused by the switching of bit rate and chip rate. The explanation of the important denotations involved are shown in Table VII.

The flow graph of the DSSS transmitter is shown in Figure 11. Functional blocks relevant to DSSS are explained below:

- DSSS modulator - This block applies modulo-2 addition to each differential encoded data bit with every chip of the PN sequence. The $i$th altered bit $a_i$ is produced with the equation:

$$a_i = b_j \oplus c_k, \quad j = \lfloor i/n \rfloor \quad and \quad k = i \bmod n, \quad i = 0, 1, 2, \ldots \tag{4.4}$$

  where, $n$ is the length of the PN sequence, $b_j$ is the $j$th original bit and $c_k$ is the $k$th chip in the PN sequence. $\{a_i\}$ substitutes for $\{b_j\}$ in the subsequent processing blocks, and hence the bit rate needs to be adjusted accordingly.

- Rate converter - This block performs a series of rate conversions to obtain the

| Denotations | In DSSS Transmitter | In DSSS Receiver |
|---|---|---|
| $a_i$ | $i$th altered bit | $i$th received bit |
| $\{a_i\}$ | Altered bit stream | Received bit stream |
| $R_a$ | Altered bit rate | Received bit rate |
| $b_j$ | $j$th original bit | $j$th recovered bit |
| $\{b_j\}$ | Original bit stream | Recovered bit stream |
| $R_b$ | Original bit rate | Recovered bit rate |
| $c_k$ | $k$th chip | $k$th chip |
| $n$ | Length of PN sequence | Length of PN sequence |
| $spb$ | Samples per symbol | Samples per symbol |
| $R_s$ | Sample rate generated by DSSS | Sample rate accepted by DSSS |
| $R_t$ | Targeted sample rate | Expected sample rate |
| $interp$ | Interpolation rate for USRP2 | n/a |
| $F_{DAC}$ | Sample rate of USRP2 DAC | n/a |
| $decim$ | n/a | Decimation rate for USRP2 |
| $F_{ADC}$ | n/a | Sample rate of USRP2 ADC |

Table VII: Explanations of Important Denotations

interpolation rate for USRP2 configuration, and then applies rational resampling if necessary. First, the original bit rate $R_b$ is converted to the altered bit rate $R_a$ for $\{a_i\}$ with the equation:

$$R_a = R_b \times n \tag{4.5}$$

The number of samples for one symbol *spb* can be customized with a command line option, and since the symbol rate is the same as bit rate for the DBPSK modulation, the sample rate $R_s$ will be:

$$R_s = R_a \times spb \tag{4.6}$$

If $R_s$ is an integer factor of $F_{DAC}$, it is directly used to calculate the interpolation rate *interp* for the USRP2:

$$interp = F_{DAC}/R_s \tag{4.7}$$

If the result of $F_{DAC}/R_s$ is a fraction, *interp* needs to be chosen as the largest integer factor of $F_{DAC}$ that is smaller than $F_{DAC}/R_s$, and the resulting sample

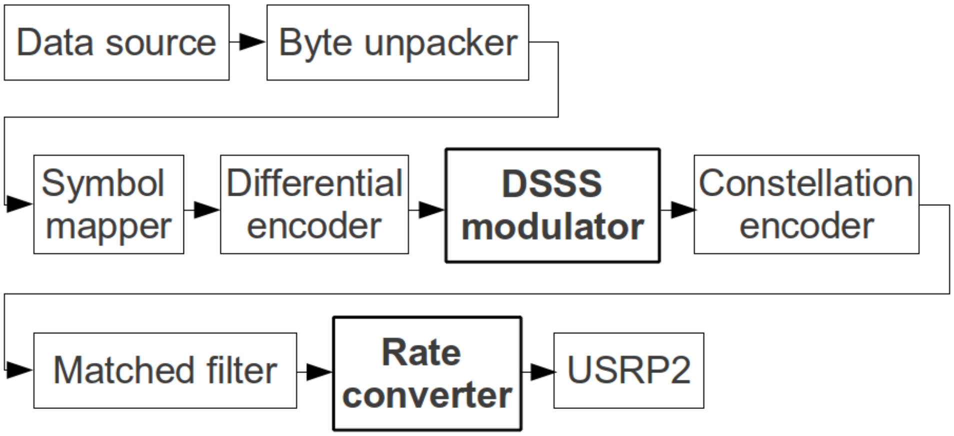


Figure 11: DSSS System Transmitter Flow Graph: blocks in bold boxes represent additional DSSS processing blocks added to the baseline DBPSK system

rate is assigned to be the targeted sample rate $R_t$:

$$R_t = F_{DAC}/interp \tag{4.8}$$

A rational resampler function is then called to convert $R_s$ to $R_t$ before samples are sent to the USRP2.

The corresponding DSSS processing blocks in the receiver include:

- Rate converter - This block performs similar rate conversions as the counterpart block in the transmitter. First, the expected recovered bit rate $R_b$ is converted to the received bit rate $R_a$ for $\{a_i\}$ with the equation:

$$R_a = R_b \times n \tag{4.9}$$

The number of samples for one symbol *spb* is also customized with a command line option, so the sample rate $R_s$ will be:

$$R_s = R_a \times spb \tag{4.10}$$

If $R_s$ is an integer factor of $F_{ADC}$, it is directly used to calculate the decimation rate *decim* for the USRP2:

$$decim = F_{ADC}/R_s \tag{4.11}$$

If the result of $F_{ADC}/R_s$ is a fraction, *decim* needs to be chosen as the largest integer factor of $F_{ADC}$ that is smaller than $F_{ADC}/R_s$, and the resulting sample rate is assigned to be the expected sample rate $R_t$:

$$R_t = F_{ADC}/decim \tag{4.12}$$

A rational resampler is responsible for converting $R_t$ to $R_s$ as soon as it receives samples from the USRP2.

- Constellation decoder - This block replaces the differential decoder that is used in the DSSS receiving path. It is used to convert incoming complex values into binary bits whose corresponding complex values have the minimum Euclidean distances to the incoming complex values.

- DSSS demodulator - This block implements a sliding correlator [26], combining the two correlation processes in the synchronization and data decoding stages. The sliding correlator first correlates the received bit stream $\{a_i\}$ from the constellation decoder with chips of the PN sequence to produce a correlation result *sum*:

$$sum = \sum_{k=1}^{n} a_i \oplus c_k, \quad i = n \times j + k, \quad j = 0, 1, 2, \ldots \tag{4.13}$$

  Then two thresholds, $threshold_0$ and $threshold_1$, are used to determine the value of $b_j$. If $sum$ is less than or equal to $threshold_0$, then $b_j$ is determined to be 0. If $sum$ is larger than or equal to $threshold_1$, then $b_j$ is determined to be 1. The sliding correlator then advances one PN sequence period. If $sum$ falls between the two thresholds, it is considered to be a synchronization error, and the sliding correlator advances only one chip.

- Differential decoder - This block performs differential decoding on the bit stream from the DSSS demodulator, which actually contains the differential decoded

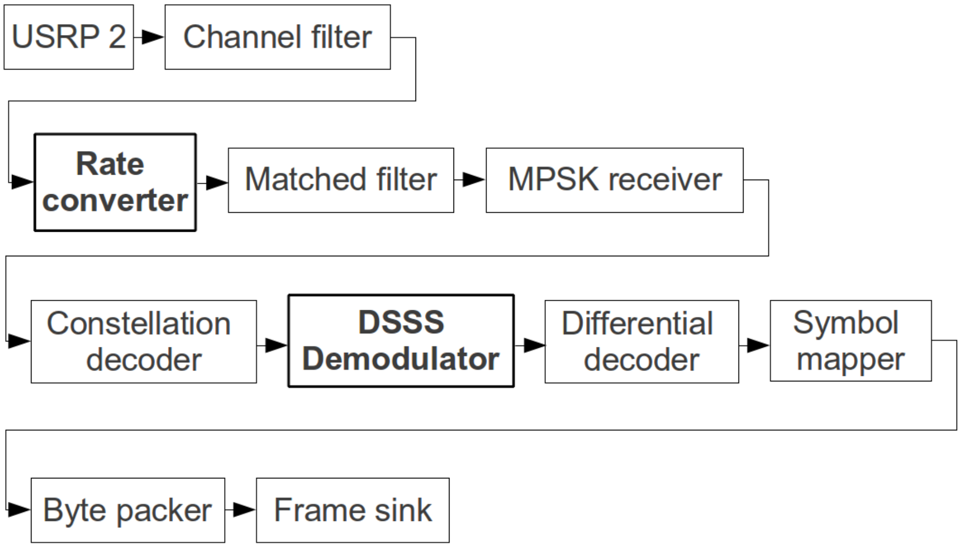


Figure 12: DSSS System Receiver Flow Graph: blocks in bold boxes represent additional DSSS processing blocks to the baseline DBPSK system

bits. The operation is to calculate the output bit $y[n]$ based on the difference between the current input bit $x[n]$ and the previous input bit $x[n-1]$ using the equation:

$$y[n] = x[n] \oplus x[n-1] \tag{4.14}$$

### 4.3.3 WDSSS System

Implementation of the WDSSS technique requires several additional important components, including the watermark embedding processor, the extraction processor, and a number of auxiliary blocks, which handle the watermark data preparation and interpretation. Processing blocks for WDSSS transmitter are shown in Figure 13. Their functions are described below:

- Watermark data source - This block is adopted from the data source block in the DBPSK transmitter program and applies the same packet format to organize watermark data into packets with preamble, access code, and CRC check fields.

- Byte coordinator - This block unpacks watermark data bytes into bits, and concatenates a designated number of bits to form new bytes for embedding.

- Watermark embedding - This block reads output from the byte coordinator, and embeds the byte value into the PN sequence according to specific embedding methods.

  If the sub-sequence method is used, this block constructs a table corresponding to the section for the designated number of chips to flip in Table V, at the block establishment stage. During the block operation stage, each iteration of the watermark embedding process starts with one copy of the original PN sequence and one incoming byte. The incoming byte is divided into bit groups corresponding to each sub-sequence. The value of each bit group is then added to the starting position of its sub-sequence to obtain a chip position. The chip at this location is then flipped in the copy of the PN sequence. After all bit groups in one byte are processed and all of the designated chips are flipped, the modified PN sequence copy is sent to the DSSS modulator for subsequent processing.

  If the maximized distance method is used, this block constructs a table similar to Table IV at the block establishment stage. During the block operation stage, each iteration of the watermark embedding process starts with one incoming byte. The value of the incoming byte is used as the table index to fetch the corresponding modified PN sequences. The modified PN sequence copy is then sent to the DSSS modulator for subsequent processing.

Processing blocks for the WDSSS receiver are shown in Figure 14. Their functions are described below:

- DSSS demodulator - This block add one more step to the original DSSS correlation process. After a differential encoded bit is successfully recovered, the group

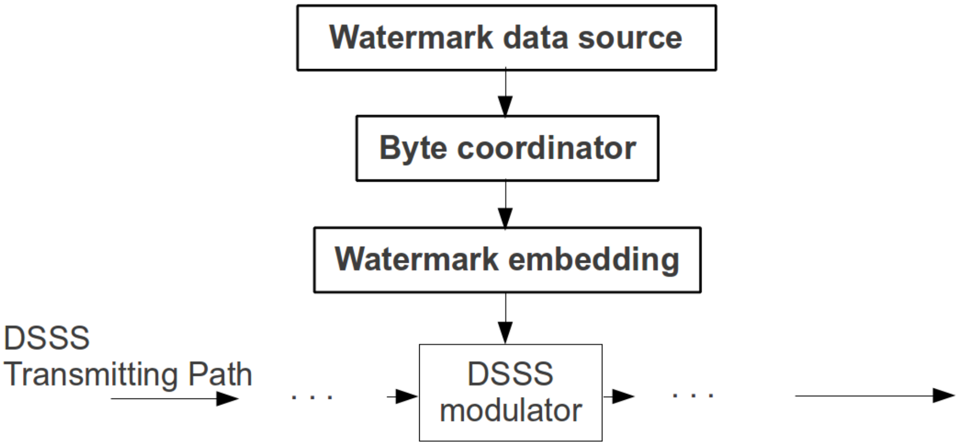


Figure 13: WDSSS System Transmitter Flow Graph: blocks in bold boxes represent additional WDSSS processing blocks added to the generic DSSS system

of incoming bits corresponding to that bit are sent to the watermark extraction processor.

- Watermark extraction - This block also performs watermark extraction according to specific embedding methods.

  If the sub-sequence method is used, this block constructs a table corresponding to the section for the expected number of chip errors in Table V, at the block establishment stage. During the block operation stage, each iteration of the watermark extraction process compares the incoming bits with the PN sequence to find the positions of flipped chips. Each flipped chip position is compared with the starting position of the corresponding sub-sequence, and the difference is then represented by a group of bits, which contains the same number of bits as required by the sub-sequence. After all positions of the flipped chips are processed, the bit groups are packed together to form an extracted byte, which is sent to the byte coordinator for subsequent processing.

  If the maximized distance method is used, this block constructs a table similar to Table IV at the block establishment stage. During the block operation stage, each iteration of the watermark extraction process counts the Hamming

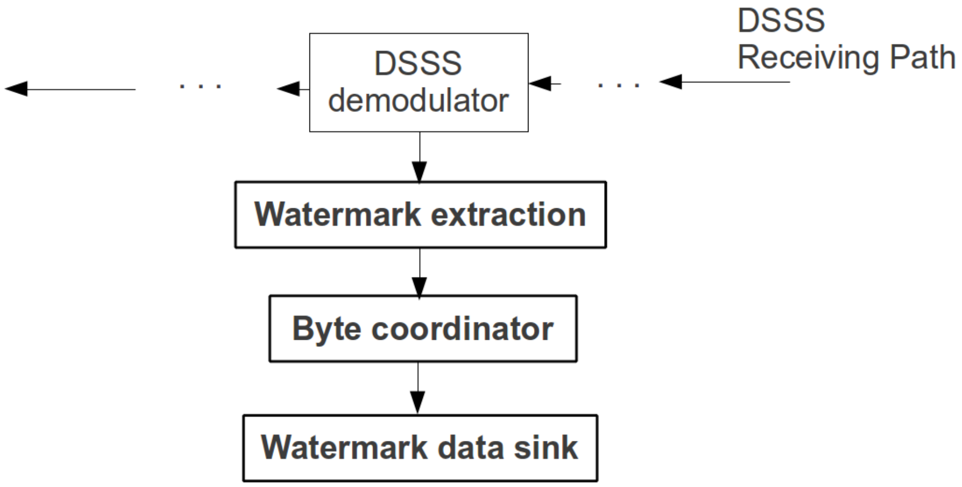


Figure 14: WDSSS System Receiver Flow Graph: blocks in bold boxes represent additional WDSSS processing to the generic DSSS system

distances between the incoming bits and every stored modified PN sequence, and compares the Hamming distances to get the sequence with the minimum distance to the incoming bits. The distance is then compared to a threshold to decide whether it is acceptable. If the result is acceptable, the table index of the corresponding PN sequence is the extracted byte, which is sent to the byte coordinator for subsequent processing.

- Byte coordinator - This block unpacks and packs bytes in the order reverse to that of its counterpart in the transmitter. It unpacks the value part of the extracted bytes into a designated number of bits, and packs every 8 bits to form one watermark data byte.

- Watermark data sink - This block is adopted from the frame sink block in the DBPSK receiver program. It correlates the byte stream with the known preamble and access code to reconstruct packets, and inspects packets with CRC32 to determine whether the packets are correct.

# CHAPTER V

# PERFORMANCE STUDY

This chapter first explains the experimental setup, and then analyzes the expected theoretical PER performance of the content signal and the watermark signal, which is compared with the experimental results.

## 5.1 Experimental Setup

The transmitter and receiver programs of the WDSSS prototype system are installed separately on two HP Compaq 8000 Elite desktop computers. Both computers are running Ubuntu 11.04 as the operating system with GNU Radio 3.4.2 installed. Two USRP2s are used as the front-ends and each device is connected to one of the two desktop computers via a Gigabit Ethernet cable. Each USRP2 is equipped with one RFX 1200 daughterboard and one VERT400 antenna, supporting a carrier frequency range of 1150 to 1450 MHz. The two USRP2s are placed about 10 feet away from each other. In addition, an Agilent N9000A CXA Signal Analyzer is used to observe the signal spectrum.

Both the transmitter and the receiver can choose the operating mode as DBPSK, DSSS or WDSSS. For the WDSSS mode, chip flipping options are 1-chip, 2-chip, 3-chip and 4-chip for flipping 1 chip, 2 chips, 3 chips and 4 chips of the original PN sequence, respectively. The PN sequence used in these experiments is the 11-chip Barker Sequence.

All experiments were run at a center frequency of 1200MHz, with a content data rate of 100K bits per second. The parameter of samples per symbol is set to 5. When the system is running in DBPSK mode, the resulting sample rate is 500K Hz, and hence the interpolation rate or the decimation rate for USRP2 is 200. If the operating mode is DSSS or WDSSS, the DBPSK signal is spread with the 11-chip Barker Sequence, yielding a sample rate of 5.5M Hz. Because 5.5M is not a factor of the DAC/ADC sample rate of the USRP2, which is 100M Hz, the program calculates an interpolation rate and a decimation rate as 16, producing a sample rate of 6.25M Hz for the interface between the USRP2 and GNU Radio, and applies a rational resampler to handle the sample rate conversion.

In the transmitter program, the data sources of the content signal and watermark signal are two different binary files. The amplitude of the transmitted signal was tuned to values in the range of $[0.002, 0.012]$ to simulate varying SNR values. This range of values is chosen, because the PER for all chip flipping option is 100% when the amplitude is below 0.002, and the PER approximates 0% when the amplitude is above 0.015. The corresponding SNR values are obtained from the measured results of the signal analyzer. Based on observations through the signal analyzer, the noise floor is shown at -96 dbm, the discernible minimum signal power is -92 dbm for the amplitude of 0.005, and the readings increase in logarithmic scale when the amplitude is increased. Therefore, the SNR value corresponding to each amplitude value can be calculated with the equation:

$$SNR = (-92 - (-96)) + 20\log(\frac{amplitude}{0.005}) \tag{5.1}$$

which yields the range for SNR values as [-4, 12] dB, and the incremental step for tuning the amplitude is set to approximately 1 dB.

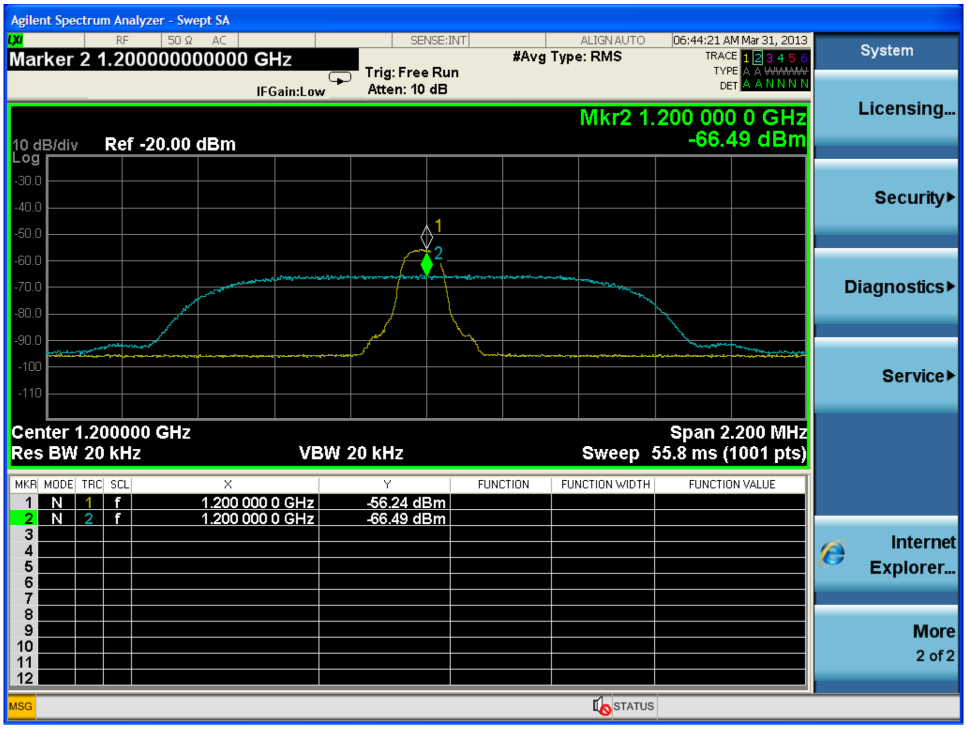
In the receiver program, the two decoding thresholds for the correlation operation in the DSSS modulator are set to 4 and 7 for bit value of 0 and 1, respectively, in order to accommodate with flipping of up to 4 chips.

## 5.2 Spectrum Analysis

The first object of this batch of experiments is to compare the spectrum of the DBPSK signal with or without the spreading operation to verify the spreading effect of the DSSS system. The spreading effect should be demonstrated in the DSSS signal with expanded bandwidth and a reduced power spectra density compared to the DBPSK signal without spreading. The bandwidth expansion ratio of the spread signal to the original signal in a DSSS system is called the processing gain, and it is equal to the length of the PN sequence. The processing gain is also applied to the power spectra density reduction ratio of the spread signal to the original signal.

In this experiment, the length of the PN sequence is 11, so the processing gain is 11, and the bandwidth expansion ratio should be 11 accordingly. In terms of the reduction ratio, because it is shown as a logarithmic scale in the spectrum analyzer display, the amount of reduced power spectra density in dB should be $10\log 11$ or $10.41dB$.

The measured result from the signal analyzer is shown in Figure 15. Trace 1 is the spectrum of the DBPSK signal without spreading, and the value of Marker 1 records the peak value of the spectrum. Trace 2 is the spectrum of the DSSS signal,



RF 50 Ω AC SENSE:INT ALIGN AUTO 06:44:21 AM Mar 31, 2013

Marker 2 1.200000000000 GHz

IFGain:Low

Trig: Free Run
Atten: 10 dB

#Avg Type: RMS

Mkr2 1.200 000 0 GHz
-66.49 dBm

Ref -20.00 dBm

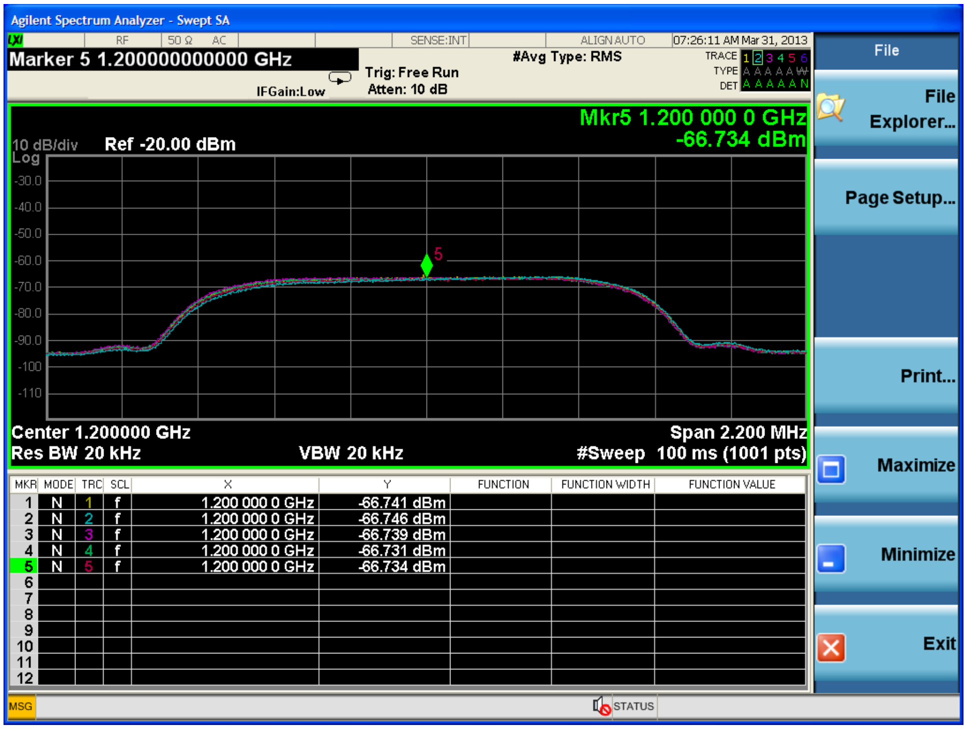
Center 1.200000 GHz
Res BW 20 kHz
VBW 20 kHz
Span 2.200 MHz
Sweep 55.8 ms (1001 pts)

| MKR | MODE | TRC | SCL | X | Y | FUNCTION | FUNCTION WIDTH | FUNCTION VALUE |
|---|---|---|---|---|---|---|---|---|
| 1 | N | 1 | f | 1.200 000 0 GHz | -56.24 dBm | | | |
| 2 | N | 2 | f | 1.200 000 0 GHz | -66.49 dBm | | | |
| 3 | | | | | | | | |
| 4 | | | | | | | | |
| 5 | | | | | | | | |
| 6 | | | | | | | | |
| 7 | | | | | | | | |
| 8 | | | | | | | | |
| 9 | | | | | | | | |
| 10 | | | | | | | | |
| 11 | | | | | | | | |
| 12 | | | | | | | | |

Figure 15: Spectrum Comparison between the DBPSK Signal without Spreading and the DSSS Signal

and the value of Marker 2 records the peak value of the spectrum. The bandwidth of the DSSS signal is approximately 11 times that of the DBPSK signal without spreading, and the peak value of the DSSS signal is 10.25 dB less than that of the DBPSK signal without spreading. Both results are close to expected results.

The second spectrum comparison is between the DSSS signal without watermark and the WDSSS signals with all options of chip flipping. Carrying embedded watermark data bits in the flipped chips does not require extra bandwidth, and hence, the spectra of the WDSSS signals with various flipped chips should have the same bandwidth and same power spectra density as the spectrum of the DSSS signal. Measurements from the signal analyzer in Figure 16 show the expected results. Trace 1 and Marker 1 belong to the DSSS signal without watermark. Trace 2 though 5 and Marker 2 through 5 are the spectra measurements for the WDSSS signals with flipping option of 1-chip, 2-chip, 3-chip and 4-chip, respectively. All traces and markers



Agilent Spectrum Analyzer - Swept SA

RF 50 Ω AC SENSE:INT ALIGN AUTO 07:26:11 AM Mar 31, 2013

Marker 5 1.200000000000 GHz

IFGain:Low

Trig: Free Run
Atten: 10 dB

#Avg Type: RMS

TRACE 1 2 3 4 5 6
TYPE A A A A A W
DET A A A A A N

Mkr5 1.200 000 0 GHz
-66.734 dBm

10 dB/div Ref -20.00 dBm

Center 1.200000 GHz
Res BW 20 kHz
VBW 20 kHz
Span 2.200 MHz
#Sweep 100 ms (1001 pts)

| MKR | MODE | TRC | SCL | X | Y | FUNCTION | FUNCTION WIDTH | FUNCTION VALUE |
|---|---|---|---|---|---|---|---|---|
| 1 | N | 1 | f | 1.200 000 0 GHz | -66.741 dBm | | | |
| 2 | N | 2 | f | 1.200 000 0 GHz | -66.746 dBm | | | |
| 3 | N | 3 | f | 1.200 000 0 GHz | -66.739 dBm | | | |
| 4 | N | 4 | f | 1.200 000 0 GHz | -66.731 dBm | | | |
| 5 | N | 5 | f | 1.200 000 0 GHz | -66.734 dBm | | | |
| 6 | | | | | | | | |
| 7 | | | | | | | | |
| 8 | | | | | | | | |
| 9 | | | | | | | | |
| 10 | | | | | | | | |
| 11 | | | | | | | | |
| 12 | | | | | | | | |

Figure 16: Spectrum Comparison among the DSSS Signal and the WDSSS Signals with different chip flipping options

overlap.

## 5.3 WDSSS Content Signal Performance

Performance of a digital communication system is usually evaluated by bit error rate (BER). The BER of the WDSSS content signal is related to the BER of the underlying DSSS signal. This paper implemented the DSSS spreading operation after the differential encoding in the transmitter, and placed the DSSS correlation process before the differential decoding in the receiver, therefore, the actual transmitted and received signal is modulated and demodulated with BPSK. The BER of BPSK is [26]:

$$P_{BPSK} = Q(\sqrt{\frac{2\varepsilon_b}{N_0}}) = Q(\sqrt{2SNR}) \tag{5.2}$$

which determines the DSSS decoding error probability $P$ by treating the PN sequence as a block code of length $n$ [11] using the equation [26]:

$$P = \sum_{i=t+1}^{n} P(i,n) \tag{5.3}$$

where $t$ is the DSSS decoding threshold, $P(i,n)$ is the probability of $i$ errors in a block code of length $n$:

$$P(i,n) = \binom{n}{i} {P_{BPSK}}^{i} (1 - P_{BPSK})^{(n-i)} \tag{5.4}$$

Because the implementation of DSSS involves 2 thresholds to determine bit value 0 and 1 individually, when the number of errors falls between the two thresholds, the DSSS decoding error is taken to be approximately 50%. Thus, the DSSS decoding error probability $P$ can be estimated as:

$$P = [\sum_{i=t_0+1}^{t_1-1} P(i,n)]/2 + \sum_{i=t_1}^{n} P(i,n) \tag{5.5}$$

where $t_0$ and $t_1$ are the two DSSS decoding thresholds for bit value 0 and 1, respectively.

The DSSS decoding error probability in turn is used to estimate the differential decoding error probability, which is the BER of the DSSS signal:

$$P_{DSSS} = 2P(1 - P) \tag{5.6}$$

The BER of the WDSSS content signal can be calculated from separate considerations for the two parts of the original PN sequence, the flipped part and the non-flipped part. The flipped part contains the artificial chip errors and hence effects the DSSS decoding error probability differently. If the artificial chip errors are received correctly, they increase the DSSS decoding error probability. On the other hand, if they are corrupted when received, they do not effect the DSSS decoding error probability. Because the DSSS decoding error still occurs when the total chip errors exceed the DSSS decoding threshold $t$, the new decoding error probability $P^{'}$ for a flipping option of $c$-chip is:

$$P^{'} = \sum_{i=0}^{c} [P(i,c) \times \sum_{j=(t+1)-(c-i)}^{n-c} P(n-c,j)] \tag{5.7}$$

Similarly, the $P^{'}$ can be estimated more accurately with consideration for the number of errors falling between two thresholds, by combining (5.5) and (5.7).

Therefore, the BER of the WDSSS content signal is:

$$P_{WDSSS} = 2P^{'}(1-P^{'}) \tag{5.8}$$

In order to be consistent with the measured experimental results, the theoretical BER is converted into packet error rate (PER):

$$PER = 1 - (1 - BER)^{8s} \tag{5.9}$$

where $s$ is the packet size in bytes, which is set to 1520 in experiments of this paper. The theoretical PERs of the DSSS signal and WDSSS content signals with various chip flipping options are plotted in Figure 17. To maintain PER less than 10% [12], the required SNR increases 1.25 dB for 1-chip flipping, increases another 1.5 dB for 2-chip flipping, increases another 2 dB for 3-chip flipping, and increases another 3.1 dB for 4-chip flipping. On average, the extra SNR required for each additional flipped chip is 1.96 dB.

Experimental results corresponding to the theoretical analysis are shown in Figure 18. Because the estimated SNR values used in the experiments are different from the actual SNR values from the USRP2 platform, there is approximately 2 dB SNR difference between the theoretical results and the experimental results. Except for this SNR offset, the experimental results show that to maintain the acceptable 10% PER, the required SNR increases 1.45 dB for 1-chip flipping, increases another 1.85 dB for 2-chip flipping, increases another 2.4 dB for 3-chip flipping, and increases

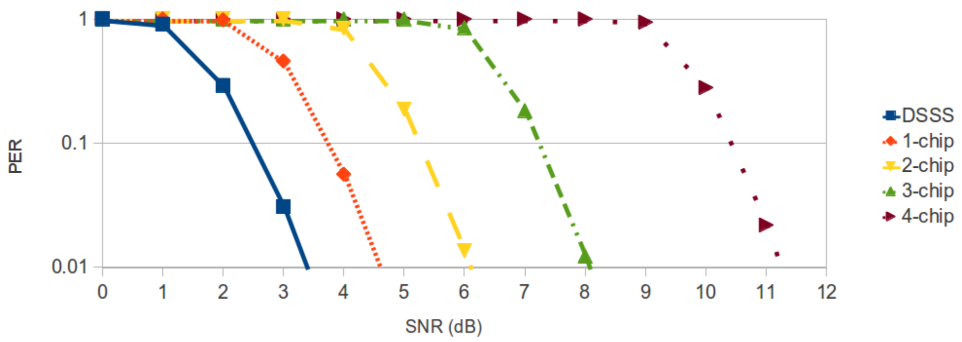


Figure 17: Theoretical PER of DSSS Signal and WDSSS Content Signals

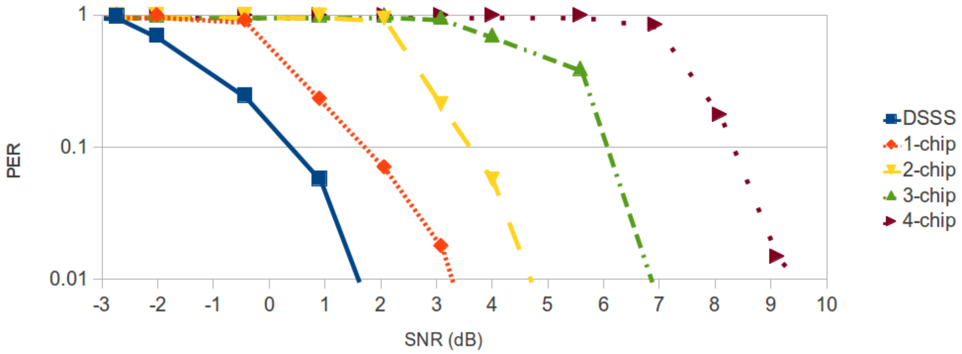


Figure 18: Experimental PER of DSSS Signal and WDSSS Content Signals

another 2.25 dB for 4-chip flipping. On average, the extra SNR required for each additional flipped chip is 1.99 dB, which is similar to the theoretical result.

## 5.4 WDSSS Watermark Signal Performance

The performance of WDSSS watermark signal was evaluated in terms of PER and throughput, and compared among various chip flipping options as well as between the two embedding methods.

### 5.4.1 With Various Flipping Settings

To measure the WDSSS watermark signal with various chip flipping options, the sub-sequence method shown in Table V is used. Because the sub-sequence embed-

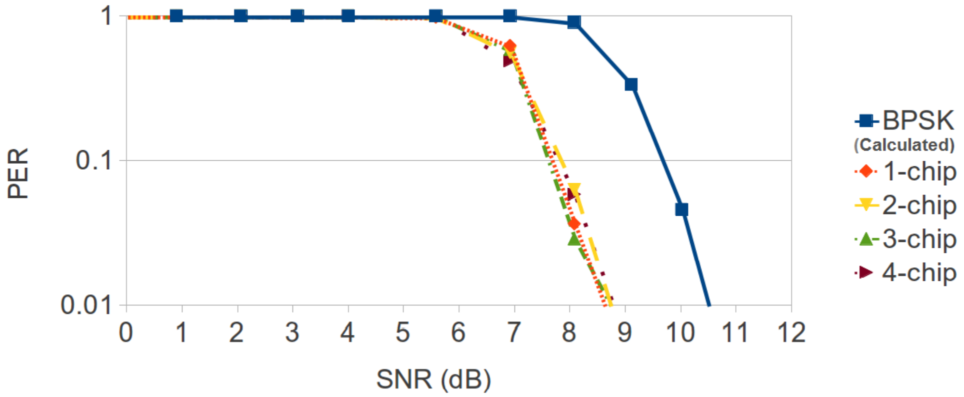


Figure 19: PER of WDSSS Watermark Signal with Various Flipping Settings

ding method dose not provide error correcting capability for the watermark signal, the BER of the WDSSS watermark signal is equal to the BER of BPSK modulation. PER is then calculated from BER with equation (5.9). Figure 19 shows the calculated BPSK PER and the measured PER for the WDSSS watermark signal. The measured PER with various chip flipping options overlap each other. Except for the 2 dB SNR offset, the measured PER matched the calculated PER.

As shown in Table V, various chip flipping options provide different embedding capabilities, which lead to different throughput for the WDSSS watermark signal. With embedding capability $w$, and content data rate $R$, the throughput of the WDSSS watermark signal can be calculated as:

$$Throughput = (1 - PER) \times w \times R \tag{5.10}$$

The measured throughput for WDSSS watermark signals with various chip flipping options is shown in Figure 20. With adequate SNR, 2-chip flipping embeds 33% more than 1-chip flipping; 3-chip and 4-chip flipping embed 67% more than 1-chip flipping, and 25% more than 2-chip flipping. Comparing this result with the previous PER result of the WDSSS content signal, there exists a trade-off between the performance of the content signal and the throughput of the watermark signal.

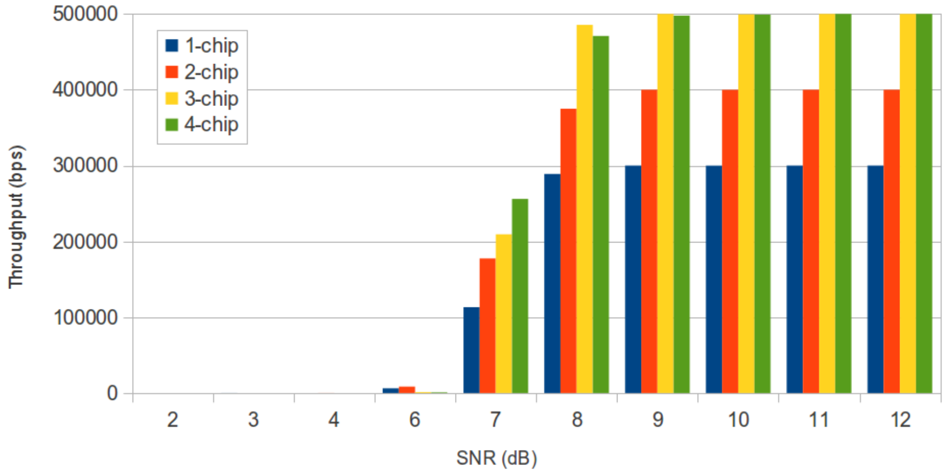


Figure 20: Throughput of WDSSS Watermark Signal with Various Flipping Settings

### 5.4.2 With Different Embedding Methods

In this batch of experiments, 3-chip flipping option is used with the two embedding methods. The sub-sequence dividing scheme is the same as in the Table V, and the set of modified PN sequences used in the maximized minimum distance method is shown in Table IV. Therefore, the BER for the sub-sequence method is still equal to the BER of BPSK. On the other hand, since the maximized minimum distance method offers error correcting capability to the watermark signal, the BER for the maximized minimum distance method is calculated from equation (5.3), where $t$ is calculated as $t = \lfloor (d_{min} - 1)/2 \rfloor$. The converted PER is plotted in Figure 21, which shows that 4 dB less SNR is required for the maximized minimum distance method to maintain PER less than 10%. The measured results are shown in Figure 22, which shows that 3 dB less SNR is required for the maximized minimum distance method to maintain PER less than 10%, with 1 dB SNR difference to the theoretical analysis.

The measured throughput for the WDSSS watermark signal with different embedding methods is shown in Figure 23. With adequate SNR, the sub-sequence method can provide 400% more throughput with adequate SNR.

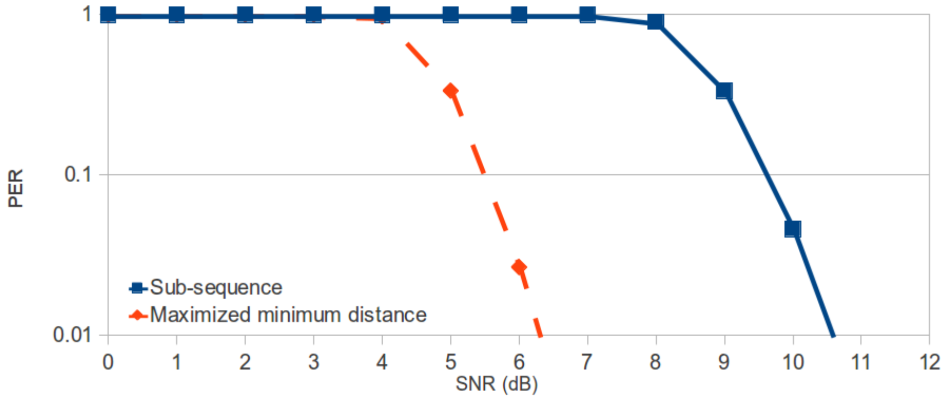


Figure 21: Theoretical PER of WDSSS Watermark Signal with Different Embedding Methods

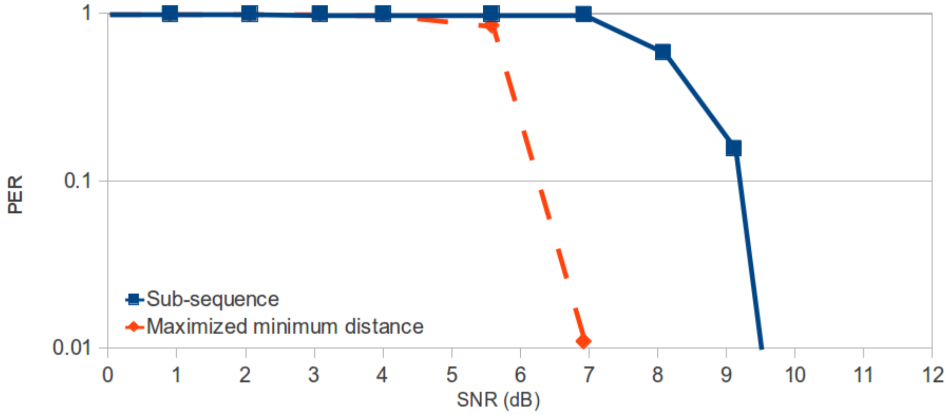


Figure 22: Experimental PER of WDSSS Watermark Signal with Different Embedding Methods

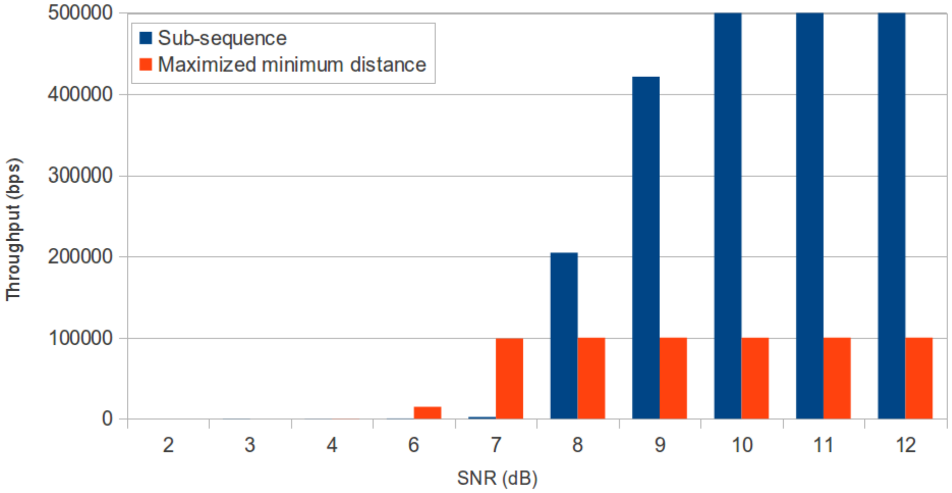


Figure 23: Throughput of WDSSS Watermark Signal with Different Embedding Methods

# CHAPTER VI

# CONCLUSIONS AND FUTURE WORK

## 6.1 Conclusions

This thesis proposed a watermarked DSSS (WDSSS) technique. The WDSSS technique flips chips on designated positions in the PN sequence to convey authentication information, and hence provides additional physical layer security to the DSSS system without requiring extra bandwidth. This thesis also developed a WDSSS prototype system on the GNU Radio/USRP SDR platform and presented theoretical analysis and experimental results for the performances of the content signal and watermark signal.

Experimental results verified the theoretical analysis and demonstrated the fundamental functions of the WDSSS technique. The impact of flipped chips to the performance of content signal was quantitatively measured, and indicated that, for the 11-chip PN sequence, an approximate 2 dB extra SNR is required for each additional flipped chip. On the other hand, increasing number of flipped chips provides increased throughput for the watermark signal. Two different watermark embedding methods were implemented, with different embedding capability and different error correcting capability. The maximized minimum distance method outperformed the sub-sequence method in terms of PER with 3 dB improvement, while the sub-sequence method provided up to 400% extra channel capacity for the watermark signal in an optimal communication environment. The trade-off between the robustness of the watermark signal and the capacity of the watermark channel implies that watermark embedding methods with different strengths can be customized for different levels of security requirements.

## 6.2 Future Work

Considering that the performance of the WDSSS system is closely related to the channel quality, an adaptive chip flipping scheme can be designed to balance content signal quality and the watermark signal throughput. For example, according to the experimental results in Figure 18, when 1% PER degradation can be tolerated and measured SNR is larger than 3.5 dB, 1-chip watermark can be used; when measured SNR increases to 5 dB, then 2-chip watermark can be used; and so forth. If the measured SNR dropped back to 3 dB, then no watermark can be used.

With this scheme, chip flipping options rise when channel quality reaches a set of thresholds, and reduces when channel quality deteriorates. At the communication setup stage, the transmitter and the receiver can establish an agreement on a set of thresholds by using public key or other security scheme. First the watermark signal is generated according to the current SNR value and the threshold. During the information exchange stage, both sides keep monitoring the channel quality and adjust the chip flipping option accordingly. There can be a transition period between two different chip flipping options when a SNR variance is observed, the receiver will continue to use the previous chip flipping option to decode incoming signals, and at the same time, save the incoming signal into a buffer. If the watermark decoding process fails continuously for a certain number of samples, the new chip flipping option is put in use.

When chip flipping option is managed in this way, the watermark signal gives priority to the content signal when the channel quality is suboptimal, so that the system performance can be maintained at an acceptable level. On the other hand, when the channel quality is excellent, more chips can be flipped so that more watermark data bits can be embedded, and hence the capacity of the watermark channel can be fully utilized.

# BIBLIOGRAPHY

[1] C. J. Biermann. *Handbook of pulping and papermaking.* Academic press, 1996.

[2] L. Choong. Multi-channel ieee 802.15. 4 packet capture using software defined radio. *University of California. http://nesl.ee.ucla.edu/fw/thomas/leslie_choong_multichannel_ieee802154.pdf*, 2009.

[3] I. Cox, M. Miller, J. Bloom, J. Fridrich, and T. Kalker. *Digital Watermarking and Steganography.* Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2 edition, 2008.

[4] I. J. Cox, M. L. Miller, and A. L. McKellips. Watermarking as communications with side information. *Proceedings of the IEEE*, 87(7):1127–1141, 1999.

[5] B. daCosta, D. Desch, and C. Read. Method and system for power save mode in wireless communication system, November 2004.

[6] Ettus Research. http://www.ettus.com/, accessed April 2013.

[7] GNU Radio Website. http://www.gnuradio.org, accessed April 2013.

[8] GNU Radio Website. What are the differences between the usrp1 and usrp2? http://gnuradio.org/redmine/projects/gnuradio/wiki/USRP2GenFAQ#What-are-the-differences-between-the-USRP1-and-USRP2, accessed April 2013.

[9] N. Goergen, T. Clancy, and T. Newman. Physical layer authentication watermarks through synthetic channel emulation. In *New Frontiers in Dynamic Spectrum, 2010 IEEE Symposium on*, pages 1–7. IEEE, 2010.

[10] F. Hermanns. Cryptographic cdma code hopping (ch-cdma) for signal security and anti-jamming. *EMPS 2004*.

[11] IEEE. Coexistence of Wireless Personal Area Networks with Other Wireless Devices Operating in Unlicensed Frequency Bands. *IEEE Std 802.15.2-2003*, 1997.

[12] IEEE. Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications. *IEEE Std 802.11-1997*, 1997.

[13] IEEE. Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (WPANs). *IEEE Std 802.15.4-2006*, 2006.

[14] S. Katzenbeisser and F. A. Petitcolas, editors. *Information Hiding Techniques for Steganography and Digital Watermarking.* Artech House, Inc., Norwood, MA, USA, 1st edition, 2000.

[15] T. Kho. Steganography in the 802.15. 4 physical layer. *UC Berkeley*, 2007.

[16] J. Kleider, S. Gifford, S. Chuprun, and B. Fette. Radio frequency watermarking for ofdm wireless networks. In *Acoustics, Speech, and Signal Processing, 2004. Proceedings.(ICASSP'04). IEEE International Conference on*, volume 5, pages V–397. IEEE, 2004.

[17] B. Lebold. *Physical Layer Watermarking of Binary Phase-shift Keyed Signals using Standard GNU Radio Blocks.* PhD thesis, Oklahoma State University, 2011.

[18] J. S. Lee and L. E. Miller. *CDMA Systems Engineering Handbook.* Artech House, Inc., 1998.

[19] T. Li, J. Ren, Q. Ling, and A. Jain. Physical layer built-in security analysis and enhancement of cdma systems. In *Military Communications Conference, 2005. MILCOM 2005. IEEE*, pages 956–962. IEEE, 2005.

[20] D. Maas, M. H. Firooz, J. Zhang, N. Patwari, and S. K. Kasera. Channel sounding for the masses: Low complexity gnu 802.11b channel impulse response estimation. *Wireless Communications, IEEE Transactions on*, 11(1):1–8, 2012.

[21] A. M. Mehta, S. Lanzisera, and K. Pister. Steganography in 802.15. 4 wireless communication. In *Advanced Networks and Telecommunication Systems, 2008. ANTS'08. 2nd International Symposium on*, pages 1–3. IEEE, 2008.

[22] J. Mitola. The software radio architecture. *Communications Magazine, IEEE*, 33(5):26–38, 1995.

[23] B. Muntwyler, V. Lenders, F. Legendre, and B. Plattner. Obfuscating ieee 802.15. 4 communication using secret spreading codes. In *Wireless On-demand Network Systems and Services (WONS), 2012 9th Annual Conference on*, pages 1–8. IEEE, 2012.

[24] R. L. Olesen, P. R. Chitrapu, J. D. Kaewell, B. A. Chiang, R. D. Herschaft, J. E. Hoffmann, S.-H. Shin, and A. Reznik. Watermarks/signatures for wireless communications, October 2005.

[25] C. Popper, M. Strasser, and S. Capkun. Anti-jamming broadcast communication using uncoordinated spread spectrum techniques. *Selected Areas in Communications, IEEE Journal on*, 28(5):703–715, 2010.

[26] J. G. Proakis. *Digital Communications*. McGraw-hill New York, 4 edition, 2001.

[27] Rec, ITU-T. X. 800 security architecture for open systems interconnection for ccitt applications. *ITU-T (CCITT) Recommendation*, 1991.

[28] Recommendation, ITUTX. 200 (1994)— iso/iec 7498-1: 1994. *Information technology–Open Systems Interconnection–Basic Reference Model: The basic model*.

[29] T. Rondeau. Exposing gnu radio: Developing and debugging. `http://www.trondeau.com/gr-tutorial/`, 2012.

[30] M. Samee, J. Geldmacher, and J. Gotze. Authentication and scrambling of radio frequency signals using reversible watermarking. In *Communications Control and Signal Processing (ISCCSP), 2012 5th International Symposium on*, pages 1–4. IEEE, 2012.

[31] B. Sklar. *Digital Communications*. Prentice Hall, 2 edition, 2001.

[32] G. D. Troxel, E. Blossom, S. Boswell, A. Caro, I. Castineyra, A. Colvin, T. Dreier, J. B. Evans, N. Goffee, K. Z. Haigh, et al. Adaptive dynamic radio open-source intelligent team (adroit): Cognitively-controlled collaboration among sdr nodes. In *Networking Technologies for Software Defined Radio Networks, 2006. SDR'06.1 st IEEE Workshop on*, pages 8–17. IEEE, 2006.

[33] P. Yu, J. Baras, and B. Sadler. Physical-layer authentication. *Information Forensics and Security, IEEE Transactions on*, 3(1):38–51, 2008.

[34] W. Yu, X. Fu, S. Graham, D. Xuan, and W. Zhao. Dsss-based flow marking technique for invisible traceback. In *Security and Privacy, 2007. SP'07. IEEE Symposium on*, pages 18–32. IEEE, 2007.

[35] E. Zielinska and K. Szczypiorski. Direct sequence spread spectrum steganographic scheme for ieee 802.15. 4. In *Multimedia Information Networking and Security (MINES), 2011 Third International Conference on*, pages 586–590. IEEE, 2011.