



NAVAL POSTGRADUATE SCHOOL

MONTEREY, CALIFORNIA

DISSERTATION

**DESIGN OF A TAILORABLE PHYSICAL LAYER
SECURITY FRAMEWORK INCLUDING
ALTERNATING SPACE-TIME CODING AND GRAY
CODE HOPPING**

by

Michael R. Cribbs

June 2021

Dissertation Supervisor:

Ric Romero

Approved for public release. Distribution is unlimited.

THIS PAGE INTENTIONALLY LEFT BLANK

REPORT DOCUMENTATION PAGE			<i>Form Approved OMB No. 0704-0188</i>	
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instruction, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188) Washington, DC, 20503.				
1. AGENCY USE ONLY (Leave blank)		2. REPORT DATE June 2021	3. REPORT TYPE AND DATES COVERED Dissertation	
4. TITLE AND SUBTITLE DESIGN OF A TAILORABLE PHYSICAL LAYER SECURITY FRAMEWORK INCLUDING ALTERNATING SPACE-TIME CODING AND GRAY CODE HOPPING			5. FUNDING NUMBERS RENUT	
6. AUTHOR(S) Michael R. Cribbs				
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Postgraduate School Monterey, CA 93943-5000			8. PERFORMING ORGANIZATION REPORT NUMBER	
9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES) NAVWAR (PMW-130), San Diego, CA 92110			10. SPONSORING / MONITORING AGENCY REPORT NUMBER	
11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.				
12a. DISTRIBUTION / AVAILABILITY STATEMENT Approved for public release. Distribution is unlimited.			12b. DISTRIBUTION CODE A	
13. ABSTRACT (maximum 200 words) A physical layer security framework is developed that may be tailored to a particular wireless communications link. Alternating space-time coding and Gray code hopping techniques are presented that may be implemented individually or concurrently. The first technique may be utilized in any multiple-input multiple-output system with choice of single- or multi-space-time code configurations. Orthogonal, non-orthogonal, and spatially-multiplexed space-time codes are included to achieve a desired diversity-multiplexing balance in addition to demonstrated security benefits. Gray code hopping may be used in any system with common digital modulation schemes to "hop" between alternative binary reflected Gray code mappings of bits to complex-valued modulation symbols. Unlike similar constellation remapping techniques, Gray code hopping provides improved physical layer security against plaintext attacks without sacrificing bit error rate performance. To facilitate these schemes, we present two series of algorithms that systematically build large sets of unique alternative space-time codes and modulation mappings. Decoding methods are discussed to allow for channel equalization and symbol co-phasing when employing an alternating space-time code. Equivalent modulation/demodulation approaches are provided for Gray code hopping. Information-theoretic and computational security analysis, Monte Carlo simulations, and over-the-air testing are completed as confirmation of proposed methods.				
14. SUBJECT TERMS computational security, Gray coding, information-theoretic security, multiple-input multiple-output, physical layer security, space-time coding			15. NUMBER OF PAGES 171	
			16. PRICE CODE	
17. SECURITY CLASSIFICATION OF REPORT Unclassified	18. SECURITY CLASSIFICATION OF THIS PAGE Unclassified	19. SECURITY CLASSIFICATION OF ABSTRACT Unclassified	20. LIMITATION OF ABSTRACT UU	

THIS PAGE INTENTIONALLY LEFT BLANK

Approved for public release. Distribution is unlimited.

**DESIGN OF A TAILORABLE PHYSICAL LAYER SECURITY FRAMEWORK
INCLUDING ALTERNATING SPACE-TIME CODING AND GRAY CODE
HOPPING**

Michael R. Cribbs
Lieutenant Commander, United States Navy
BS, University of Kansas, 2009
MS, Electrical Engineering, Naval Postgraduate School, 2015
EE, Electrical Engineer, Naval Postgraduate School, 2015

Submitted in partial fulfillment of the
requirements for the degree of

DOCTOR OF PHILOSOPHY IN ELECTRICAL ENGINEERING

from the

**NAVAL POSTGRADUATE SCHOOL
June 2021**

Approved by:	Ric Romero Department of Electrical and Computer Engineering Dissertation Supervisor	Tri T. Ha Department of Electrical and Computer Engineering Dissertation Co-Supervisor
	David C. Jenn Department of Electrical and Computer Engineering	Frank E. Kragh Department of Electrical and Computer Engineering
	Thor Martinsen Department of Applied Mathematics	Ric Romero Department of Electrical and Computer Engineering Dissertation Chair
Approved by:	Douglas J. Fouts Chair, Department of Electrical and Computer Engineering	
	Orrin D. Moses Vice Provost of Academic Affairs	

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

A physical layer security framework is developed that may be tailored to a particular wireless communications link. Alternating space-time coding and Gray code hopping techniques are presented that may be implemented individually or concurrently. The first technique may be utilized in any multiple-input multiple-output system with choice of single- or multi-space-time code configurations. Orthogonal, non-orthogonal, and spatially-multiplexed space-time codes are included to achieve a desired diversity-multiplexing balance in addition to demonstrated security benefits. Gray code hopping may be used in any system with common digital modulation schemes to “hop” between alternative binary reflected Gray code mappings of bits to complex-valued modulation symbols. Unlike similar constellation remapping techniques, Gray code hopping provides improved physical layer security against plaintext attacks without sacrificing bit error rate performance. To facilitate these schemes, we present two series of algorithms that systematically build large sets of unique alternative space-time codes and modulation mappings. Decoding methods are discussed to allow for channel equalization and symbol co-phasing when employing an alternating space-time code. Equivalent modulation/demodulation approaches are provided for Gray code hopping. Information-theoretic and computational security analysis, Monte Carlo simulations, and over-the-air testing are completed as confirmation of proposed methods.

THIS PAGE INTENTIONALLY LEFT BLANK

Table of Contents

1	Introduction	1
1.1	Literature Review	2
1.2	Objectives	13
1.3	Organization	14
2	Source Set Building	15
2.1	Space-Time Code Sets	15
2.2	Modulation Mapping Sets	31
3	Transmission and Reception Support Methods	37
3.1	Individual Space-Time Code Building	37
3.2	Decoding Methods for Alternating Space-Time Codes.	39
3.3	Individual Mapping Building	53
3.4	Gray Code Hopping Modulation and Demodulation.	55
4	Proposed Physical Layer Security Framework	59
4.1	Alternating Space-Time Coding	59
4.2	Gray Code Hopping	66
4.3	Combining Techniques	71
5	Security Analysis	73
5.1	Security System Nomenclature	73
5.2	Alternating Space-Time Code Security.	87
5.3	Gray Code Hopping Security	100
5.4	Combined Security	104
6	Results	105
6.1	Alternating Space-Time Coding Bit Error Rate	105

6.2	Evaluation of Key Residue Class Decoding	111
6.3	Gray Code Hopping Bit Error Rate Comparison	119
6.4	Gray Code Hopping Hard versus Soft Decision Decoding	121
7	Conclusion	123
7.1	Significant Contributions	123
7.2	Future Work	126
Appendix A	Proof of Maximal Ratio Combining Sequence	129
Appendix B	Additional Key Residues	135
List of References		141
Initial Distribution List		149

List of Figures

Figure 1.1	Illustration of two-way MIMO wire-tap channel. Source: [39].	4
Figure 1.2	Illustration of using precoding matrix to optimize reception at only the intended receiver. Source: [36].	5
Figure 1.3	Illustration of CMO. Source: [19].	8
Figure 1.4	Illustration of common physical layer network model with addition of artificial interference. Source: [20].	9
Figure 1.5	Illustration of a secure OFDM transmission scheme based on chaos mapping. Source: [27].	11
Figure 1.6	Illustration of applying a pseudorandom antipodal sequence to the TX antennas of a MIMO system. Source: [25].	13
Figure 2.1	Full permutations array for a vector with three elements.	24
Figure 2.2	Illustration of all equivalent BRGC mappings for QPSK with colors as shown in Table 2.3.	34
Figure 3.1	Block diagrams illustrating decoding steps for non-orthogonal and SM STCs. (a) STCs without transmit diversity. (b) STCs with transmit diversity.	47
Figure 3.2	Example implementations of a GCH modulator and demodulator. (a) Internal mapping function or LUT implementation. (b) Equivalent two-step method implementation.	55
Figure 4.1	Basic elements for each phase of the alternating STC PLS scheme.	60
Figure 4.2	Illustration of indexing into permutations array containing elements of $\mathbf{v}_{boso}$ in lexicographic order with row index = 1639.	63
Figure 4.3	Basic elements for each phase of the GCH PLS scheme.	67
Figure 4.4	Illustration of indexing into permutations array containing elements of $\mathbf{v}_{bomo}$ in lexicographic order with row index = 1.	69

Figure 5.1	Illustration of USSs within the 16QAM scheme when $a = 1$	82
Figure 5.2	Illustration of USSs within the 64QAM scheme when $a \geq 2$	83
Figure 5.3	Template for Alamouti message and cryptogram residue classes with $f = 2$ employing 16QAM data symbols and STCs from a single KRC with $a = 1$. Adapted from [13].	86
Figure 5.4	Template for Alamouti message and cryptogram residue classes with $f = 1$ employing 16QAM data symbols and STCs from a single KRC with $a = 1$. Adapted from [13].	87
Figure 5.5	Probability mass function of the Q_f metric for 16QAM with $k = 2$	90
Figure 5.6	Probability mass function of the Q_f metric for 64QAM with $k = 2$	90
Figure 5.7	Probability mass function of the Q_f metric for 256QAM with $k = 2$	91
Figure 5.8	Message equivocation given STCs with various number of data symbols per codeword and modulation schemes.	92
Figure 5.9	Mutual information between $\mathbf{s}$ and $\mathbf{s}_c$ given STCs with various number of data symbols per codeword and modulation schemes.	93
Figure 5.10	Theoretical plot of (5.36) with various number of data symbols per codeword and modulation schemes including high SNR Monte Carlo simulations for the Alamouti, SPO334, and MPO468 STCs.	99
Figure 6.1	Comparison of decoding techniques for SM441, NO442, and Alamouti STCs using various modulations and DO of four. (a) QPSK and 64QAM. (b) 16QAM and 256QAM.	107
Figure 6.2	Comparison of decoding techniques for SM441, NO442, and Alamouti STCs using various modulations and DO of eight. (a) QPSK and 64QAM. (b) 16QAM and 256QAM.	108
Figure 6.3	Illustration of OTA frame structure.	109
Figure 6.4	Block diagram illustration of GNU Radio flowgraph for transmission.	109
Figure 6.5	Illustration of MISO OTA test setup.	110
Figure 6.6	Block diagram illustration of GNU Radio flowgraph for reception and data export.	111

Figure 6.7	OTA testing SNR measurement.	111
Figure 6.8	OTA primary channel BER results for various values of a with baseline and LB theoretical curves.	112
Figure 6.9	Probability of correct KRC determination when employing the Alamouti STC with QPSK and various values of a based on Monte Carlo simulations and OTA testing of a MISO system.	114
Figure 6.10	Probability of correct KRC determination when employing the Alamouti STC with QPSK for various values of a and N_R based on Monte Carlo simulations.	115
Figure 6.11	Expectation of x_c based on simulations and OTA testing.	117
Figure 6.12	Expected value of key equivocation based on simulations and OTA testing.	118
Figure 6.13	Cumulative distribution function of x_c based on Monte Carlo simulations at E_b/N_0 of 32 dB.	118
Figure 6.14	Cumulative distribution function of x_c based on OTA testing at E_b/N_0 values nearest 32 dB.	119
Figure 6.15	Theoretical baseline, GCH, alternating CD-PHY, and eavesdropper BER comparison.	120
Figure 6.16	BER comparison of LDPC encoded 64QAM using hard and soft decision decoding with and without GCH.	121

THIS PAGE INTENTIONALLY LEFT BLANK

List of Tables

Table 2.1	Variable vectors for each base STC given in Section 2.1.1.	30
Table 2.2	Set cardinality of base STCs for various values of a	32
Table 2.3	All equivalent BRGC mappings for QPSK.	33
Table 2.4	Set cardinality of various baseline modulation mappings.	36
Table 5.1	Tabular form KR for the Alamouti variant code shown in (5.2). . .	77
Table 5.2	Tabular form KR for the NO442 variant code shown in (3.30). . .	78
Table 5.3	Tabular form KR for the SM221 variant code shown in (3.21). . .	79
Table 5.4	Number of KRCs for base STCs when incorporating PR algorithms with various values of a	81
Table 5.5	Entropy in $\mathbf{G}$, key equivocation, and mutual information between $\mathbf{G}$ and $\mathbf{G}_{KR}$ in bits for various STCs with $a = 9$	94
Table 5.6	Number of codewords and possible output frame combinations for a given frame length when performing a search using the message equivocation of each codeword.	95
Table 5.7	Expected number of bit errors per correctly assigned symbol for vari- ous modulations when $a \geq 2$	98
Table 5.8	Exhaustive key search attack complexity for alternating STC PLS based on number of possible combinations for a few example systems.	100
Table 5.9	Exhaustive key search attack complexity for GCH PLS based on num- ber of possible combinations for a few example systems.	103
Table 5.10	Exhaustive key search attack complexity for both proposed PLS tech- niques combined based on number of possible combinations for a few example systems with $a = 9$	104
Table 6.1	OTA test parameters.	113

Table B.1	Tabular form KR for the MPO334 variant code shown in (3.18). . .	136
Table B.2	Tabular form KR for the MPO334 variant code shown in (B.2). . .	138
Table B.3	Tabular form KR for the NO442 base code shown in (2.10).	140

List of Acronyms and Abbreviations

5G	5th generation
5G NR	5G New Radio
ARB	arbitrary waveform file
AWGN	additive white Gaussian noise
BER	bit error rate
BPSK	binary phase-shift keying
BRGC	binary reflected Gray code
BS	base station
CBMD	channel-based mapping diversity
CD-PHY	constellation diversity based physical layer security
CMO	constellation mapping obfuscation
CSAI-ES	constellation selection and artificial interference for eavesdropping suppression
CSI	channel state information
DO	diversity order
E_b/N_0	energy per bit to noise power spectral density ratio
GCH	Gray code hopping
IQ	in-phase and quadrature
KR	key residue
KRC	key residue class
LB	lower bound
LDPC	low density parity check
LLR	log-likelihood ratio
LUT	lookup table

MIMO	multiple-input multiple-output
MISO	multiple-input single-output
ML	maximum-likelihood
MMSE	minimum mean square error
MPO334	multi-pair orthogonal 3-3-4
MPO434	multi-pair orthogonal 4-3-4
MPO468	multi-pair orthogonal 4-6-8
MPSK	M -ary phase-shift keying
MQAM	M -ary quadrature amplitude modulation
MRC	maximal ratio combining
NMI	next mapping index
NMS	next mapping set
NO442	non-orthogonal 4-4-2
NSI	next STC index
NSS	next STC set
OFDM	orthogonal frequency-division multiplexing
OSPR	original symbol phase rotated
OTA	over-the-air
PLS	physical layer security
PMI	precoding matrix index
PR	phase rotation
PRN	pseudorandom number
PRNG	pseudorandom number generator
PT	plaintext
QPSK	quaternary or quadrature phase-shift keying
RF	radio frequency

RSSI	received signal strength indicator
RX	receive
SD	sphere decoding
SESD	Schnorr-Euchner sphere decoder
SGI	symbol group index
SIC	successive interference cancellation
SIM	subscriber identity module
SM	spatially-multiplexed
SM221	SM 2-2-1
SM441	SM 4-4-1
SMSTC	SM STC
SNR	signal-to-noise ratio
SOK_m	mapping seed obfuscation key
SOK_s	STC seed obfuscation key
SPO334	single-pair orthogonal 3-3-4
SQRD	sorted QR decomposition
ST	space-time
STC	space-time (ST) code
TDS-OFDM	time-domain synchronous OFDM
TX	transmit
UB	upper bound
UE	user equipment
USRP	Universal Software Radio Peripheral
USS	unique symbol set
UT	user terminal
$\mathbf{v}_{acp}$	allowed column permutations vector

$\mathbf{v}_{ait}$	alternative iterations vector
$\mathbf{v}_{asc}$	allowed symbol conjugations vector
$\mathbf{v}_{asp}$	allowed symbol permutations vector
$\mathbf{v}_{asr}$	allowed symbol PRs vector
$\mathbf{v}_{bmi}$	base mapping index vector
$\mathbf{v}_{bomo}$	base order of mapping operations vector
$\mathbf{v}_{bosso}$	base order of STC operations vector
$\mathbf{v}_{bsi}$	base STC index vector
$\mathbf{v}_{div}$	divisor vector
$\mathbf{v}_{it}$	iterations vector
$\mathbf{v}_{obm}$	original base mapping vector
$\mathbf{v}_{obs}$	original base STC vector
$\mathbf{v}_{ol}$	operation limits vector
$\mathbf{v}_{oo}$	order of operations vector
$\mathbf{v}_{tbm}$	temporary base mapping vector
$\mathbf{v}_{tbmi}$	temporary $\mathbf{v}_{bmi}$
$\mathbf{v}_{tbs}$	temporary base STC vector
$\mathbf{v}_{tbsi}$	temporary $\mathbf{v}_{bsi}$
$\mathbf{v}_{tomo}$	temporary order of mapping operations vector
$\mathbf{v}_{toso}$	temporary order of STC operations vector
$\mathbf{v}_{ubm}$	unique base mapping vector
$\mathbf{v}_{ubs}$	unique base STC vector
VSG	vector signal generator
XOR	exclusive-or

Acknowledgments

First, I would like to thank God for providing a path for me to achieve this lifelong dream of completing my doctoral degree. His wisdom, strength, and encouragement over these many years allowed me to never doubt this day would come.

Second, I would like to express my deepest gratitude to my wife, Ingrid, and children for their grace and understanding when dealing with me in my often-engrossed and sleep-deprived state throughout this journey. Thank you for your consistent encouragement and for giving me the time and space to finish this work.

I would also like to thank my advisor and co-advisor, Professors Ric Romero and Tri Ha, for sharing and working with me on this research. I appreciate your steady devotion of time and support throughout the process. It was an incredibly fun and rewarding experience.

To my committee members, thank you for your time, thoughtful reviews, and willingness to help.

To anyone who may read this dissertation, I hope you find this research to be as interesting and enjoyable to read as it was to accomplish. May it spur you on in your own endeavors!

“And whatever you do, do it heartily, as to the Lord and not to men” —Colossians 3:23

THIS PAGE INTENTIONALLY LEFT BLANK

CHAPTER 1:

Introduction

Wireless networks such as Wi-Fi and cellular technologies have become essential and ubiquitous tools of our everyday life. Maintaining security of the information shared over those networks is of the utmost importance. Although communications security is often performed via higher layer cryptography, the expectation that quantum computing may allow for certain types of cryptography to be rapidly broken has led to a dramatic increase in physical layer security (PLS) research in recent years to provide for a more holistic approach of “layered security” [1]–[3]. Further impetus for PLS research is found with its intended use in emergent 5th generation (5G) cellular networks [4]–[7].

To begin this chapter, we review some common approaches toward PLS from the literature to provide a better understanding of the solution space identified for contributions by this work. We then define the objectives of our research and provide a summary of the organization of this dissertation.

The contents of this dissertation have been revised from previous work already published or in publication by the author. Specifically, portions of Sections 2.1, 2.2.3, 5.1.4, and 5.2.2 are revised from “Orthogonal STBC Set Building and Physical Layer Security Application” by Michael Cribbs, Ric Romero, and Tri Ha published in *2020 IEEE 21st International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)* in May 2020 ©2020 IEEE¹ [8]. Portions of Sections 1.1, 2.1, 3.1, 3.2, 3.3, 4.1, 4.2, 5.1, 5.2, 5.3, and 7.1 and Appendices A and B are revised from “Physical Layer Security for Multiple-Input Multiple-Output Systems by Alternating Orthogonal Space-Time Block Codes” by Michael Cribbs, Ric Romero, and Tri Ha published in *IEEE Open Journal of the Communications Society* in September 2020, used under CC BY [9]. Portions of Sections 2.1, 3.2, 5.1, 5.2,

¹In reference to IEEE copyrighted material which is used with permission in this dissertation, the IEEE does not endorse any of Naval Postgraduate School’s products or services. Internal or personal use of this material is permitted. If interested in reprinting/republishing IEEE copyrighted material for advertising or promotional purposes or for creating new collective works for resale or redistribution, please go to http://www.ieee.org/publications_standards/publications/rights/rights_link.html to learn how to obtain a License from RightsLink. If applicable, University Microfilms and/or ProQuest Library, or the Archives of Canada may supply single copies of the dissertation.

6.1.1, and 7.1 and Appendices A and B are revised from “Alternative Codes and Phase Rotation Extensions for Alternating Space-Time Coding-Based Physical Layer Security” by Michael Cribbs, Ric Romero, and Tri Ha accepted for publication in *IEEE Open Journal of the Communications Society*, used under CC BY [10]. Portions of Sections 1.1.2, 2.2, 3.4, 5.3, 6.3, 6.4, and 7.1 are revised from “Modulation-Based Physical Layer Security via Gray Code Hopping” by Michael Cribbs, Ric Romero, and Tri Ha accepted for publication in *35th IEEE International Workshop Technical Committee on Communications Quality and Reliability* ©2021 IEEE¹ [11].

1.1 Literature Review

Although security in wireless networks covers many areas, including authentication, non-repudiation, confidentiality and access control, and integrity and availability [12], the focus of our research is data confidentiality. Broadly speaking, techniques aiming to provide confidentiality can be divided into two groups: information-theoretic security and computational security [7]. Information-theoretic security is generally understood to rely on properties of the physical channel between transmitter and intended receiver and is based on Shannon’s mathematical theory of communications [7], [13]. It is often based on the assumption that these properties are unknown to an eavesdropper, colloquially referred to as “Eve,” and/or their effects cannot be undone by Eve. On the other hand, computational security is based on the amount of effort or time required to break a code [1], [7], [14].

PLS solutions have traditionally been viewed as purely information-theoretic security approaches [5]–[7], [12], [15]; however, through examination of PLS techniques from the literature, several can be found that incorporate elements of computational security [4], [16]–[28]. This is considered to be true based on the understanding that computational security is a component of any communications system where it is possible for Eve to obtain the transmitted information, with a reasonable error rate, by decoding the system through enumeration of all values within a countable, albeit extremely large, set of possibilities [1], [6].

We now review a few PLS approaches from the literature, grouping them by whether they contain elements of computational security. This distinction is made in light of the fact that contributions made by this research also incorporate elements of computational security;

thus, it is desired that the reader is aware that the proposed solution does not stand alone in this aspect.

1.1.1 Techniques without Elements of Computational Security

In his pioneering work in this area of research, Wyner [29] introduced the wire-tap channel where secrecy capacity was achieved assuming the signal received over the wire-tap channel was a degraded version of that received over the main channel. Further research loosened this assumption to allow for security without such disparity between channels [30]. Many PLS approaches employ the use of forward error-correcting codes that may be randomized or punctured to provide security [15], [31]–[33]. Other common approaches over multiple-input multiple-output (MIMO) channels include use of precoding matrices or insertion of artificial noise [34]–[38]. Some of these techniques are now briefly discussed to provide a background of this vast area of research.

Artificial Noise

In [37], the transmitter estimates the channel state information (CSI) to the intended receiver and generates an additive artificial noise matrix, $\mathbf{w}$, that lies within the null space of the receiver’s channel matrix, $\mathbf{H}_{AB}$. The transmitter then sends the original signal, $\mathbf{s}$, plus the noise matrix, and the receiver obtains

$$\mathbf{y}_R = \mathbf{H}_{AB}(\mathbf{s} + \mathbf{w}) + \mathbf{n} \approx \mathbf{H}_{AB}\mathbf{s} + \mathbf{n}, \quad (1.1)$$

where $\mathbf{n}$ is an additive white Gaussian noise (AWGN) vector. On the other hand, Eve obtains

$$\mathbf{y}_E = \mathbf{H}_{AE}(\mathbf{s} + \mathbf{w}) + \mathbf{n} = \mathbf{H}_{AE}\mathbf{s} + \mathbf{H}_{AE}\mathbf{w} + \mathbf{n}, \quad (1.2)$$

where $\mathbf{H}_{AE}$ is Eve’s channel matrix. Thus, addition of artificial noise has little to no impact on the intended receiver but degrades the signal-to-noise ratio (SNR) over Eve’s channel.

Cooperative Jamming

In [39], two entities, commonly known as “Alice” and “Bob,” communicate to one another simultaneously over a full-duplex MIMO wireless communications link referred to as the two-way MIMO wire-tap channel, as shown in Fig. 1.1. With knowledge of their own

transmission, each entity is able to remove corresponding interference in order to extract the transmission from the other entity; however, Eve does not have this ability due to the lack of knowledge of either transmission information. This scheme suffers if either channel from Alice to Eve or Bob to Eve is sufficiently degraded in relation to the other. This creates a single-user decodable condition in which the information transmitted over the dominant channel is received by Eve without interference of the secondary channel.

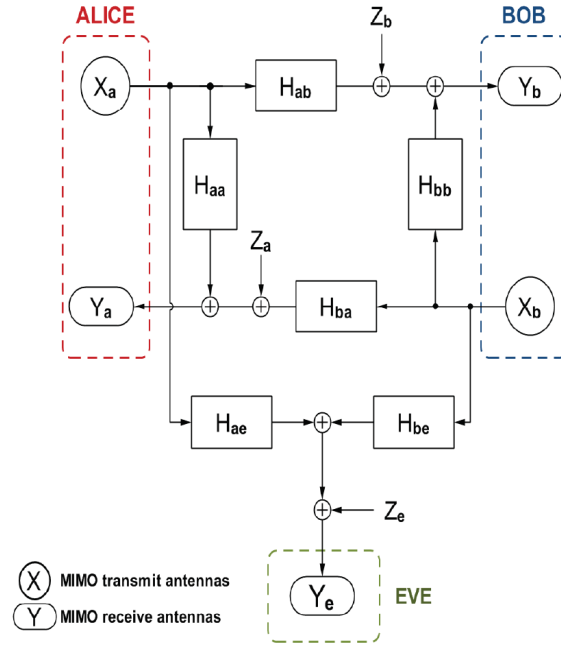


Figure 1.1. Illustration of two-way MIMO wire-tap channel. Source: [39].

A similar cooperative jamming technique was presented in [40] for a different type of two-way wire-tap channel.

Precoding and Beamforming Methods

Several methods employ CSI at the transmitter or a handshaking procedure to establish precoding matrices or beamforming vectors that optimize reception at only the intended receiver [2], [36], [41], [42]. One system utilizing a precoding matrix in this manner is shown in Fig. 1.2.

The handshaking procedure is sometimes employed to come to an agreement regarding the best suboptimal precoding matrix index (PMI) for use from a codebook rather than

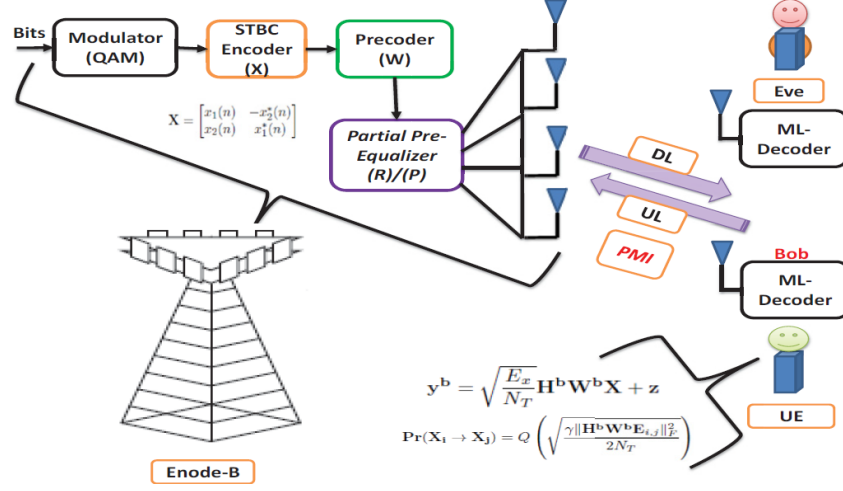


Figure 1.2. Illustration of using precoding matrix to optimize reception at only the intended receiver. Source: [36].

requiring performance of singular value decomposition to obtain the optimal precoding matrix solution. In [2], this handshaking procedure is exploited to establish a quantum secure shared secret key.

Error-Correcting Codes

Error-correcting codes such as the low density parity check (LDPC) and Polar codes have received extensive use in PLS schemes [15], [31]–[33]. These methods generally rely on the original assumption made by Wyner that the wire-tap channel is degraded more than the primary channel; thus, errors may be intentionally inserted by the transmitter and corrected only by the intended receiver. In [32], secret information was transferred from transmitter to receiver within the punctured bits of an LDPC code and not actually transmitted over the channel at all. Channel conditions were assumed such that Eve could not recover the punctured bits.

1.1.2 Techniques with Elements of Computational Security

There are a number of techniques that aim to achieve PLS via randomizations, rotations, or other modifications to either the complex-valued symbols or bit-to-symbol mappings

associated with standard digital modulation schemes. Other approaches apply phase rotations (PRs) to traditional space-time (ST) codes (STCs) or randomization of orthogonal frequency-division multiplexing (OFDM) subcarrier assignments. We now provide a brief description of these approaches including elements of computational security.

Constellation Remapping

A technique was presented in [16] to remap an M -ary quadrature amplitude modulation (MQAM) constellation using a process called mirror-mapped encoding that rearranges a “Gray code sequence by mirroring the arrangement of subsequences” [16]. This technique between the transmitter and receiver, referred to as the base station (BS) and user equipment (UE) from this point forward, starts by determining the CSI of the mutual channel between them. A quantization of the phase of the determined CSI is used to create a vector, $\mathbf{C}$, of b -bits, where $b = \log_2(M)$ is the number of bits represented by each of the M symbols in the MQAM scheme. This vector is then employed by the mirror-mapped encoding process to create a minimum distance mapping of the MQAM constellation from nested quaternary or quadrature phase-shift keying (QPSK) constellations [16].

This process is represented mathematically as

$$X = \sum_{i=1}^{b/2} X^{(i)}, \quad (1.3)$$

where

$$X^{(i)} = 2^{b/2-i} \left[(-1)^{p_1 + \mathbf{C}(i)} + j(-1)^{p_2 + \mathbf{C}(b/2+i)} \right], \quad (1.4)$$

$$p_1 = \sum_{l=1}^i \mathbf{B}(l), \text{ and} \quad (1.5)$$

$$p_2 = \sum_{l=1}^i \mathbf{B}\left(\frac{b}{2} + l\right). \quad (1.6)$$

X is a complex-valued modulation symbol, $X^{(i)}$ represents scaled QPSK symbols summed together to produce X , and $\mathbf{B}$ is the b -bit message, *aka* binary label, corresponding to X [16].

Applying this technique for all possible phase angles of CSI produces a subset of alternative

binary reflected Gray code (BRGC) mappings of size 2^b . Within this subset, there is only a single mapping that pairs each constellation symbol with each binary label; therefore, Eve can determine the entire mapping with only a single symbol of known plaintext (PT) data. Without mention of a constellation remapping frequency within [16], it is assumed that the same mapping is used throughout a communications session; thus, an attack employing only a single symbol of known PT would allow Eve to decode an entire communications session.

A similar technique called channel-based mapping diversity (CBMD) was proposed in [17] for remapping M -ary phase-shift keying (MPSK) constellations with Gray coding constraints by employing channel gains as the shared secret between BS and UE. As in [16], only a subset of alternative BRGC mappings is used in [17], and the remapping frequency is also not clearly stated. It is assumed that the same mapping is employed for numerous data symbols within a communications session leaving this technique equally vulnerable to a known PT attack.

In [18], a PLS technique called CD-PHY was presented based on what was called *constellation diversity*. The basic idea of CD-PHY is that BS and UE employ a secret random mapping of the constellation symbols to bit sequences. It was stated in [18] that the eavesdropper would have to attempt all $M!$ mappings to find the correct decoding; thus, CD-PHY allows for any mapping to be employed without requiring that binary labels with a Hamming distance of 1 be mapped to symbols separated by the minimum distance within the constellation. As [43] confirmed the optimality of BRGC mappings for common signal constellations, CD-PHY results in degraded bit error rate (BER) performance between BS and UE when a mapping other than a BRGC mapping is selected. Additionally, [18] did not mention how frequently the secret random mapping would be updated; thus, it is assumed that the same mapping is used throughout a communications session.

Another scheme referred to as constellation mapping obfuscation (CMO) was presented in [19] that likewise allowed for any of the $M!$ mappings to be employed between BS and UE as shown in Fig. 1.3. Thus, implementation of CMO will also lead to degraded BER performance.

As all $M!$ mappings are permitted by these two schemes, there are $(M - 1)!$ mappings that pair each symbol with each binary label. Thus, a known PT attack including $n \leq M$ unique

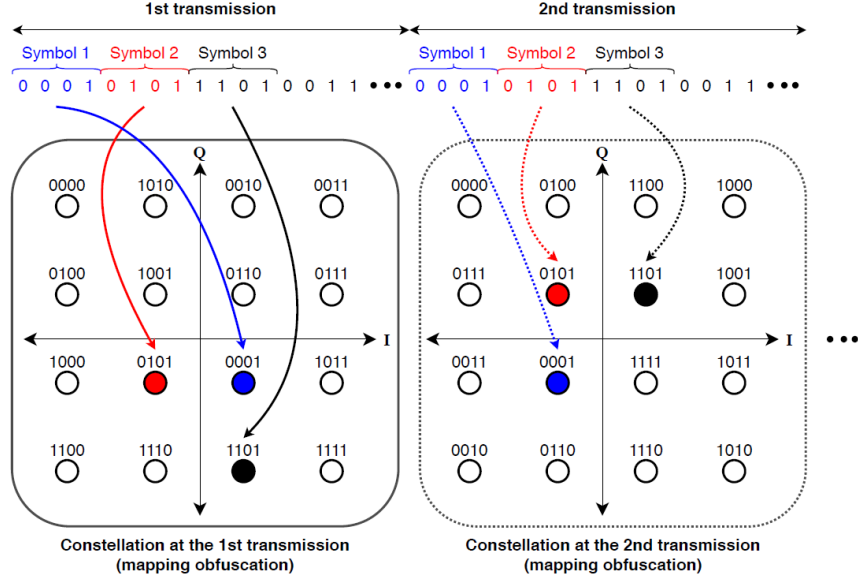


Figure 1.3. Illustration of CMO. Source: [19].

constellation symbols reduces the uncertainty of the static secret mapping to $(M - n)!$ possibilities corresponding to all mappings that pair the n received symbols with the corresponding n known PT binary labels.

Even without known PT data, all constellation remapping techniques presented utilize a countable set of possible mappings and thereby an element of computational security.

Constellation Selection and Artificial Interference

In [20], a scheme called constellation selection and artificial interference for eavesdropping suppression (CSAI-ES) was presented whereby a set of modulation modes (e.g., QPSK, 16QAM, 64QAM) and constellation structures (e.g., circular or rectangular) is employed along with artificial interference to enhance PLS. The network model given in [20] and shown in Fig. 1.4 is representative of a common technique of employing artificial noise or interference to degrade Eve's channel; however, adding interference may also degrade the BER performance between BS and UE up to a pre-determined threshold [20]. This technique did not choose to add interference within the null space of UE's channel matrix as discussed in [37].

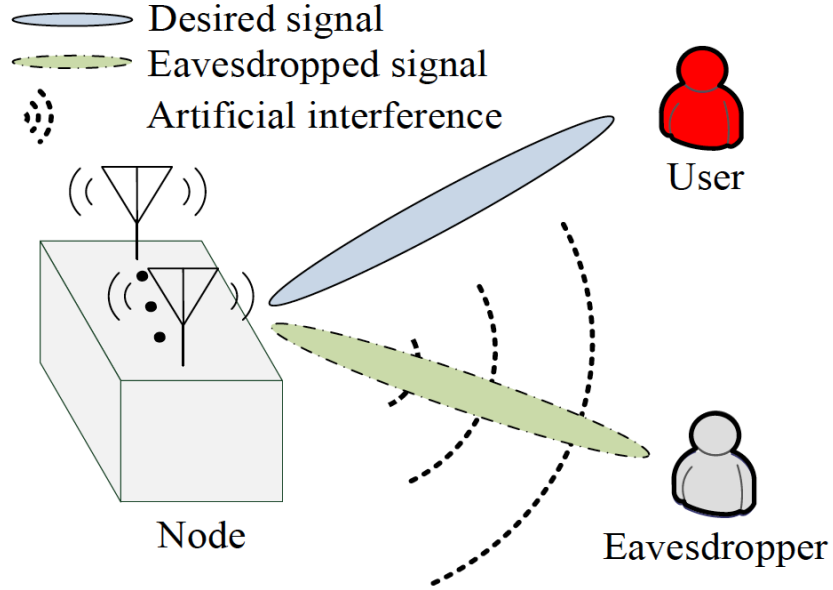


Figure 1.4. Illustration of common physical layer network model with addition of artificial interference. Source: [20].

A mechanism was proposed in [20] for BS to inform UE of the chosen modulation mode and structure prior to transmission. Without any mention of repetition frequency for this mechanism, it is assumed that the same mode and structure are used throughout a communications session. Modulation identification techniques could likely be used by Eve to determine the mode and structure in use [44]; however, with only a limited set of possibilities, Eve could also enumerate all possible modes and structures to break this scheme. Although the addition of artificial interference will result in degraded performance, Eve may decode an entire communications session once the static mode and structure in use have been determined.

Constellation Rotation

A technique referred to as original symbol phase rotated (OSPR) secure transmission was introduced in [4] whereby the symbols transmitted by a massive MIMO BS to K legitimate user terminals (UTs) are phase-rotated by an amount equal to the angle of one channel tap estimate for each UT. It was shown how a broadcast reference signal could be transmitted by the BS to each UT in order to disseminate the secret PR angle without giving that

information away to a passive eavesdropper. It was discussed that the reference signal could be rebroadcast to enhance secrecy performance [4]; however, this obviously results in increased overhead unless the reference signal is also required for other purposes.

Using OSPR leads to eavesdropper uncertainty between equal energy symbols within the chosen constellation; however, no inherent uncertainty exists between constellation points of different energies other than that of channel noise. Additionally, all symbols for a given UT are rotated the same amount until the reference signal broadcasting step is repeated; thus, a single symbol of known PT would allow Eve to decode an entire communications session between repetitions of the reference signal broadcast step. Without PT data, Eve could also break this scheme through enumeration as a finite alphabet of symbols exists within any chosen modulation scheme, and even fewer symbols have equal energy. Thus, only a relatively small number of PRs would need to be attempted to obtain the correct output data frame.

Constellation Rotation and Amplitude Randomization

In [21], an independent pseudorandom PR was applied to each transmitted symbol in a time-domain synchronous OFDM (TDS-OFDM) system. The symbol-specific rotation angle was generated by a pseudorandom number generator (PRNG). A weak artificial noise signal was subsequently added to further degrade the performance of the eavesdropper's channel. Later, a technique was developed based on [21] where a more detailed amplitude randomization algorithm was devised to reduce the ability for Eve to identify the modulation scheme in use [22]. Although QPSK performance provided in [22] indicated insignificant BER performance degradation between BS and UE, similar results for MQAM were not provided.

Similar to OSPR, this scheme leads to eavesdropper uncertainty between equal energy symbols within the chosen constellation; however, due to the addition of artificial noise, this scheme also introduces uncertainty to both Eve and UE regarding symbols within neighboring energy levels. It is suspected that BER performance between BS and UE would be degraded for higher-order MQAM schemes using this technique due to this added uncertainty.

Unlike the schemes previously discussed, any known PT would only allow Eve to determine

the PR applied to the corresponding symbols. It is assumed in [21] that details of the PRNG implementation are unknown to Eve; thus, Eve is unable to determine the rotation applied to symbols before or after the known PT portion of the transmission. If Eve is able to gather some information about the PRNG with exception of the state at the beginning of transmission, Eve could attempt to break this scheme by enumerating the state space of the PRNG. Whereas this enumeration is likely a larger task than with previous techniques, the state space is still countable; thus, with sufficient time and computational effort, this scheme may be broken with slightly reduced BER performance due to the weak artificial noise insertion.

Constellation Rotation and Subcarrier Randomization

In [27], a pseudorandom chaos sequence was employed to phase-rotate modulation symbols and provide a chaos mapping of OFDM subcarriers as shown in Fig. 1.5. Computational security was evaluated within [27] stating that the number of possible secret keys for the proposed scheme was $1.8 \cdot 10^{39}$, which translates to approximately 131 bit security by taking the base two logarithm of the number of possible keys.

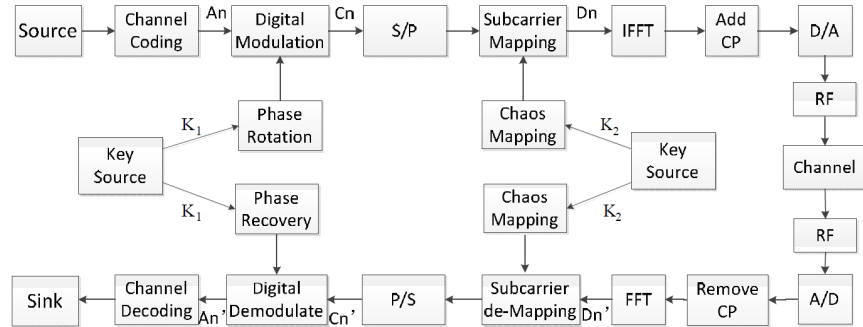


Figure 1.5. Illustration of a secure OFDM transmission scheme based on chaos mapping. Source: [27].

A revised version of this scheme was presented in [45] that also allowed for modulation amplitude randomization. This revision increased the number of possible keys to $1.5 \cdot 10^{98}$, or 327 bit security, but incurred a cost in BER performance.

Space-Time Coding Schemes

In [25], a technique named “hidden OSTBC” was introduced where a pre-shared secret pseudorandom antipodal sequence was applied across the transmit (TX) antennas of a MIMO system after encoding with a STC as shown in Fig. 1.6. The primary STC considered in [25] is a rate 1 real-valued orthogonal STC represented as

$$\mathbf{G} = \begin{bmatrix} +s_1 & +s_2 & +s_3 & +s_4 & +s_5 & +s_6 & +s_7 & +s_8 \\ -s_2 & +s_1 & +s_4 & -s_3 & +s_6 & -s_5 & -s_8 & +s_7 \\ -s_3 & -s_4 & +s_1 & +s_2 & +s_7 & +s_8 & -s_5 & -s_6 \\ -s_4 & +s_3 & -s_2 & +s_1 & +s_8 & -s_7 & +s_6 & -s_5 \\ -s_5 & -s_6 & -s_7 & -s_8 & +s_1 & +s_2 & +s_3 & +s_4 \\ -s_6 & +s_5 & -s_8 & +s_7 & -s_2 & +s_1 & -s_4 & +s_3 \\ -s_7 & +s_8 & +s_5 & -s_6 & -s_3 & +s_4 & +s_1 & -s_2 \\ -s_8 & -s_7 & +s_6 & +s_5 & -s_4 & -s_3 & +s_2 & +s_1 \end{bmatrix}, \quad (1.7)$$

using eight TX antennas to transmit eight data symbols, s_1 through s_8 , over eight symbol time periods. A few extensions of this STC were offered up to a rate 1/8 real-valued orthogonal STC using 64 TX antennas to transmit eight data symbols over 64 symbol time periods. It was stated in [25] that the same pseudorandom sequence could be used for multiple transmissions; thus, this technique is considered to employ a static STC throughout the entire communications session.

In [26], the received signal strength indicator (RSSI) was used as a secret key to seed a PRNG used to generate PRs to be applied to each TX antenna of a MIMO system after encoding with either the Alamouti STC from [46] or the 3/4 rate Octonion orthogonal STC from [47]. Unlike in [25], the PRs produced by this scheme change after every codeword, or one-time usage of the STC.

Whereas both of these schemes apply a unit-energy rotation to the TX antenna data streams after encoding with a standard STC, neither approach performs this same operation across symbol time periods or data symbols themselves. Likewise, neither technique applies any permutation or conjugation to the data streams, time periods, or data symbols. In terms of computational security, Eve could attempt to enumerate the state space of the PRNG employed within [26] to break this approach, as previously discussed with the [21] scheme.

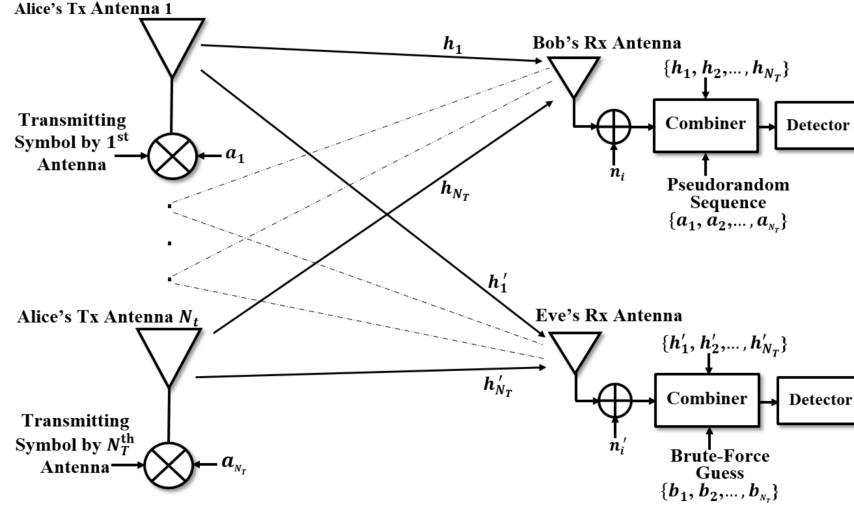


Figure 1.6. Illustration of applying a pseudorandom antipodal sequence to the TX antennas of a MIMO system. Source: [25].

Since the antipodal sequence employed by [25] is static throughout a communications session, Eve could enumerate the possible antipodal sequences to break this scheme.

1.2 Objectives

Based on our literature review, we found that no existing constellation remapping approach utilizes all alternative BRGC mappings to maximize the uncertainty of an eavesdropper while maintaining the BER performance of the intended receiver. Additionally, none of the ST coding-based PLS schemes make use of all available degrees of freedom within the STC structure to maximize security. With the solution space identified for contribution, we now state our objectives.

The primary objective for our research is to develop a tailorable PLS framework including optional techniques designed to alternate the STC and/or modulation mapping in use over a wireless communications link. To that end, details of these techniques are investigated including

- various types of STCs and digital modulation schemes,
- algorithms to build alternative STCs and mappings,
- single- versus multi-STC alternation methods,

- information-theoretic security offered in terms of message and key equivocation and mutual information,
- and computational security offered.

Monte Carlo simulations and over-the-air (OTA) testing are completed as confirmation of the proposed methods.

1.3 Organization

Due to the optimality of BRGC mappings for common signal constellations as confirmed in [43], all modulation schemes evaluated employ BRGC mappings of bits to complex-valued symbols. Alternating between these various mappings with consecutive modulation symbols is referred to as Gray code hopping (GCH) from this point forward.

The remainder of this dissertation is organized as follows: In Chapter 2, base STCs and mappings are presented along with operation descriptions and algorithms to systematically build large sets of unique alternative STCs and modulation mappings. In Chapter 3, transmission and reception support methods are provided to facilitate alternating STCs and GCH. In Chapter 4, the alternating STC and GCH PLS schemes are proposed. In Chapter 5, nomenclature is provided to allow for characterization of ST coding-based PLS schemes. Security offered by the proposed techniques when employed individually and in unison is then discussed. In Chapter 6, results of Monte Carlo simulations and OTA testing are provided. Finally, in Chapter 7, significant contributions made by this work are discussed along with possibilities for future work in this area of research.

CHAPTER 2:

Source Set Building

In order to develop techniques to alternate the STC in use and/or perform GCH over a wireless communications link, we must identify alternative STCs and Gray code mappings. To begin this task, we briefly discuss common STCs and modulation constellations. We then examine operations that may be performed to construct alternatives. Finally, we present a series of algorithms that systematically implement these operations to build large sets of unique alternative STCs and modulation mappings. These tasks are completed for STCs first and then for bit to complex-valued symbol mappings.

This chapter includes adaptations from previous work already published or in publication by the author. Specifically, portions of Section 2.1 are revised from [8]–[10]. Portions of Section 2.2 are revised from [8] and [11].

2.1 Space-Time Code Sets

Before presenting STC sets, we first establish common terminology and symbols. A STC matrix is represented as $\mathbf{G}$. For all STCs, let c represent the number of columns in $\mathbf{G}$. This also represents the number of TX antennas employed as each column is a TX antenna data stream. Likewise, r is used to represent the number of rows in $\mathbf{G}$, which also represents the number of symbol time periods. The number of symbols per codeword is represented by k , where codeword refers to a single block of encoded symbols as dictated by the chosen STC matrix. The data symbol vector for any given STC refers to $\mathbf{s} = [s_1 \ \dots \ s_k]^T$, where T represents the transpose operation. An extended data symbol vector, $\mathbf{s}_{ext} = [\mathbf{s}^T \ \mathbf{s}^\dagger]^T$, is also referenced, where $\dagger$ represents conjugate transpose operation. These definitions remain consistent throughout this dissertation.

2.1.1 Base Space-Time Codes

Three different types of base STCs are investigated for use with this research: orthogonal, non-orthogonal, and spatially-multiplexed (SM) STCs. We now present base STCs from each of these three types.

Orthogonal Space-Time Codes

The first type of STC investigated is the orthogonal STC. This type employs transmit diversity and exhibits orthogonality between each TX antenna data stream. The Alamouti STC is a well-known, complex rate 1 orthogonal STC represented as

$$\mathbf{G} = \begin{bmatrix} s_1 & s_2 \\ -s_2^* & s_1^* \end{bmatrix}, \quad (2.1)$$

using two TX antennas to transmit two data symbols, s_1 and s_2 , over two symbol time periods [46]. The asterisk, (*), represents complex conjugate operation.

The base code employed in [8], referred to as single-pair orthogonal 3-3-4 (SPO334), is a complex rate 3/4 orthogonal STC represented as

$$\mathbf{G} = \begin{bmatrix} s_1 & s_2 & s_3 \\ -s_2^* & s_1^* & 0 \\ -s_3^* & 0 & s_1^* \\ 0 & -s_3^* & s_2^* \end{bmatrix}, \quad (2.2)$$

using three TX antennas to transmit three data symbols, s_1 through s_3 , over four symbol time periods.

The base code employed in [9], referred to as multi-pair orthogonal 4-6-8 (MPO468), is a complex rate 6/8 orthogonal STC represented as

$$\mathbf{G} = \begin{bmatrix} s_1 & s_2 & s_3 & 0 \\ -s_2^* & s_1^* & 0 & s_6 \\ -s_3^* & 0 & s_1^* & -s_5 \\ 0 & -s_3^* & s_2^* & s_4 \\ s_4 & s_5 & s_6 & 0 \\ -s_5^* & s_4^* & 0 & s_3 \\ -s_6^* & 0 & s_4^* & -s_2 \\ 0 & -s_6^* & s_5^* & s_1 \end{bmatrix}, \quad (2.3)$$

using four TX antennas to transmit six data symbols, s_1 through s_6 , over eight symbol time periods. MPO468 is based on a STC constructed in [48] but reordered to show how

it extends SPO334. Although the code rate of these two STCs is equivalent, the additional columns, data symbols, and rows in MPO468 allows for a much larger set cardinality.

An alternative code, referred to as multi-pair orthogonal 3-3-4 (MPO334), is a complex rate 3/4 orthogonal STC represented as

$$\mathbf{G} = \begin{bmatrix} s_1 & s_2 & \frac{s_3}{\sqrt{2}} \\ -s_2^* & s_1^* & \frac{s_3}{\sqrt{2}} \\ \frac{s_3^*}{\sqrt{2}} & \frac{s_3^*}{\sqrt{2}} & \frac{(-s_1-s_1^*+s_2-s_2^*)}{2} \\ \frac{s_3^*}{\sqrt{2}} & -\frac{s_3^*}{\sqrt{2}} & \frac{(s_2+s_2^*+s_1-s_1^*)}{2} \end{bmatrix}, \quad (2.4)$$

using three TX antennas to transmit three data symbols, s_1 through s_3 , over four symbol time periods.

A variation of MPO334, referred to as multi-pair orthogonal 4-3-4 (MPO434), is a complex rate 3/4 orthogonal STC represented as

$$\mathbf{G} = \begin{bmatrix} s_1 & s_2 & \frac{s_3}{\sqrt{2}} & \frac{s_3}{\sqrt{2}} \\ -s_2^* & s_1^* & \frac{s_3}{\sqrt{2}} & \frac{-s_3}{\sqrt{2}} \\ \frac{s_3^*}{\sqrt{2}} & \frac{s_3^*}{\sqrt{2}} & \frac{(-s_1-s_1^*+s_2-s_2^*)}{2} & \frac{(-s_2-s_2^*+s_1-s_1^*)}{2} \\ \frac{s_3^*}{\sqrt{2}} & -\frac{s_3^*}{\sqrt{2}} & \frac{(s_2+s_2^*+s_1-s_1^*)}{2} & \frac{(-s_1-s_1^*-s_2+s_2^*)}{2} \end{bmatrix}, \quad (2.5)$$

using four TX antennas to transmit three data symbols, s_1 through s_3 , over four symbol time periods. MPO334 and MPO434 were designed using the theory of amicable designs from [49] and originally referred to as sporadic 3/4 rate STCs when given in [50].

A slight modification in representation is required for both MPO334 and MPO434 in order to use the set building algorithms that follow. These STCs must be translated into three-dimensional matrices that are subsequently summed along the third dimension to return to the original forms given in (2.4) and (2.5) prior to transmission. By translating each matrix into three dimensions, each element of the resulting matrix contains only a single weighted

data symbol variation. This representation allows for the same set building algorithms to be used with any base STC discussed.

As MPO334 and MPO434 contain matrix elements with sums of four symbol variations, the third-dimension depth, denoted as d from this point forward, for the resulting matrices is four. Thus, MPO334 is translated into a three-dimensional matrix with slices of

$$\mathbf{G}_{3D}(:, :, 1) = \begin{bmatrix} s_1 & s_2 & \frac{s_3}{\sqrt{2}} \\ -s_2^* & s_1^* & \frac{s_3}{\sqrt{2}} \\ \frac{s_3^*}{\sqrt{2}} & \frac{s_3^*}{\sqrt{2}} & \frac{-s_1}{2} \\ \frac{s_3^*}{\sqrt{2}} & \frac{-s_3^*}{\sqrt{2}} & \frac{s_2}{2} \end{bmatrix}, \quad (2.6)$$

$$\mathbf{G}_{3D}(:, :, 2) = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & \frac{-s_1^*}{2} \\ 0 & 0 & \frac{s_2^*}{2} \end{bmatrix}, \quad (2.7)$$

$$\mathbf{G}_{3D}(:, :, 3) = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & \frac{s_2}{2} \\ 0 & 0 & \frac{s_1}{2} \end{bmatrix}, \text{ and} \quad (2.8)$$

$$\mathbf{G}_{3D}(:, :, 4) = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & \frac{-s_2^*}{2} \\ 0 & 0 & \frac{-s_1^*}{2} \end{bmatrix}. \quad (2.9)$$

Non-Orthogonal Space-Time Codes

The second type of STC investigated is the non-orthogonal STC. Although this type also employs transmit diversity, not all antenna data streams are orthogonal. The one and only non-orthogonal base STC researched is referred to as non-orthogonal 4-4-2 (NO442). It is a complex rate 2 non-orthogonal STC introduced in [51] represented as

$$\mathbf{G} = \begin{bmatrix} s_1 & s_2 & s_3 & s_4 \\ -s_2^* & s_1^* & -s_4^* & s_3^* \end{bmatrix}, \quad (2.10)$$

using four TX antennas to transmit four data symbols, s_1 through s_4 , over two symbol time periods.

Spatially-Multiplexed Space-Time Codes

The last type of STC investigated is the SM STC (SMSTC). This type maximizes use of spatial multiplexing and does not employ transmit diversity. For all SMSTCs, data symbols are inserted into every STC matrix element without repetition. The first base STC of this type is referred to as SM 2-2-1 (SM221). It is a complex rate 2 SMSTC represented as

$$\mathbf{G} = \begin{bmatrix} s_1 & s_2 \end{bmatrix}, \quad (2.11)$$

using two TX antennas to transmit two data symbols, s_1 and s_2 , over one symbol time period.

The last base STC of this type is referred to as SM 4-4-1 (SM441). It is a complex rate 4 SMSTC represented as

$$\mathbf{G} = \begin{bmatrix} s_1 & s_2 & s_3 & s_4 \end{bmatrix}, \quad (2.12)$$

using four TX antennas to transmit four data symbols, s_1 through s_4 , over one symbol time period.

2.1.2 Alternative Code Operations

To construct an alternative STC from a given base code, operations must be applied in a way that maintains the properties of the original code. Thus, if the base code is orthogonal, the alternative STC is also orthogonal. The operations researched to obtain alternative STCs

include PRs, permutations, and conjugations. We now discuss each of these operations.

Phase Rotations

All base codes discussed in Section 2.1.1 contain complex-valued symbols. All elements within chosen rows and/or columns of these base codes may be phase-rotated by various amounts to produce alternative STCs. These operations are referred to as row and column PRs, respectively. Additionally, for those base codes exhibiting transmit diversity, all variants of a chosen data symbol may be phase-rotated to obtain an alternative code. This is referred to as a symbol PR. There are a few details that must be considered when performing PR operations.

Selection of allowed PR angles must first be determined. For the purpose of simplicity, we propose a PR scheme with selection of equally-spaced PR angles around the complex-plane. In order to do this, we define a variable $a \geq 1$ such that 2^a equally-spaced PR angles are produced of the form

$$\frac{2\pi x}{2^a} \quad (2.13)$$

radians, where $x \in \{0, \dots, 2^a - 1\}$ is the PR angle index. When $a = 1$, this PR scheme is equivalent to negation as only 0 and π radian PRs are allowed in this case.

When performing symbol PRs, it must be understood that orthogonal STC matrices consist of conjugated and unconjugated symbol variations. In order to maintain orthogonality while performing symbol PRs, these dissimilarly-conjugated symbol variations must receive opposing PRs. Without loss of generality, if unconjugated variations of a given data symbol are rotated by ϕ radians, conjugated variations of that same data symbol must be rotated by $-\phi$ radians.

Permutations

Rows, columns, and symbols may also be permuted to build alternative STCs. As a reminder of the standard structure for STC matrices, rows within the STC matrix represent symbol time periods; thus, row permutations are a simple reordering of the symbol time periods. Likewise, column permutations are a reordering of the TX antenna data streams. Symbol permutations involve switching all variations of a given data symbol for the same variations of a different data symbol according to a selected permutation vector.

Example 2.1.1. A symbol-permuted version of (2.3) is given as

$$\mathbf{G} = \begin{bmatrix} s_1 & s_2 & s_3 & 0 \\ -s_2^* & s_1^* & 0 & s_5 \\ -s_3^* & 0 & s_1^* & -s_6 \\ 0 & -s_3^* & s_2^* & s_4 \\ s_4 & s_6 & s_5 & 0 \\ -s_6^* & s_4^* & 0 & s_3 \\ -s_5^* & 0 & s_4^* & -s_2 \\ 0 & -s_5^* & s_6^* & s_1 \end{bmatrix}, \quad (2.14)$$

which is built by swapping symbols s_5 and s_6 (i.e., permutation vector of [1 2 3 4 6 5]).

Conjugations

All variants of a chosen data symbol may be conjugated to produce an alternative STC. Row and column conjugations are not performed as they generally do not maintain orthogonality of antenna data streams. There are select combinations of row and column conjugations that do maintain orthogonality; however, these combinations can be seen as equivalent to performing a particular combination of symbol conjugations. Thus, this operation is limited to symbol conjugations alone.

2.1.3 Space-Time Code Set Building Algorithms

Operations from the previous section are now implemented in a systematic manner to build a set, $\mathbf{S}_s$, of unique alternative STCs from a given base code from Section 2.1.1. Algorithms S0 - S7 are provided in this section for this purpose. Algorithm S0 is performed first to initialize the set. Algorithms S1 - S7 are then performed in any order. Depending on the chosen base code, applicable operation limits must be applied to prevent duplicate codes from being built.

Algorithm Notes

Values of c , k , r , and d for the chosen STC are known to all algorithms in this dissertation. If a two dimensional STC is employed, then $d = 1$. All text within curly brackets are comments

Algorithm S0 - STC Set Initialization

Inputs: $\mathbf{G}$ {Base STC}, $|\mathbf{S}_s|$ {STC set cardinality},

$\mathbf{K}_{perm}$ {Symbol permutation of base STC}

Outputs: $\mathbf{S}_s$ {Set of STCs}, PC_o {# of populated codes in output $\mathbf{S}_s$ },

$\mathbf{A}_{syms}$ {Symbols record array}

- 1: $PC_o = 1$
 - 2: $\mathbf{S}_s = 4$ -dimensional array of 0's of size r -by- c -by- d -by- $|\mathbf{S}_s|$
 - 3: $\mathbf{S}_s(:, :, :, 1) = \mathbf{G}$ {Populate set index 1 with Base STC}
 - 4: **for** $mi = 1$ to $|\mathbf{S}_s|$ **do**
 - 5: $\mathbf{A}_{syms}(mi, :) = \mathbf{K}_{perm}$ {Initialize each row of $\mathbf{A}_{syms}$ with $\mathbf{K}_{perm}$ }
 - 6: **end for**
 - 7: **return** $\mathbf{S}_s, PC_o, \mathbf{A}_{syms}$
-

to aid the reader. The $de2bi(i, n, p)$ MATLAB function is used to create a length n , little-endian (*aka* right most significant digit) row vector of the decimal value i represented in base p [52]. The $diag(\mathbf{v})$ MATLAB function is used to create a square diagonal matrix with elements of vector $\mathbf{v}$ on the main diagonal [53]. Length of any given vector $\mathbf{v}$ is represented as $|\mathbf{v}|$. A slightly modified version of the $perms(n)$ MATLAB function is used to create an $n!$ -by- n array containing all permutations of the integers from 1 to n in lexicographic order, where each row contains a different permutation [54]. The $[\mathbf{B}, \mathbf{I}] = sort(\mathbf{A})$ MATLAB function is used to sort the elements of vector $\mathbf{A}$ in ascending order, where $\mathbf{B}$ holds the sorted elements, and $\mathbf{I}$ holds the original indices of the elements of $\mathbf{A}$ as they appear within $\mathbf{B}$ such that $\mathbf{B} = \mathbf{A}(\mathbf{I})$ [55]. The $\sim$ symbol in place of $\mathbf{B}$ indicates that $\mathbf{B}$ is not used. Lastly, $\times$ is used to indicate matrix multiplication whereas $\circ$ indicates element-wise multiplication, and $==$ indicates logically equal whereas $=$ indicates assignment.

Operation Limits

For all base STCs given in Section 2.1.1 with the exception of SMSTCs, the operation limits for row PRs, column PRs, and row permutations are consistent. All 2^{ar} combinations of row PRs are permitted, and all $r!$ row permutations are permitted. Column PRs are limited to $2^{a(c-1)}$ to prevent building duplicate codes within the set.

The operation limit for the symbol PR algorithm is dependent on the base code selected. An allowed symbol PRs vector ($\mathbf{v}_{asr}$) is used for this purpose within the symbol PR algorithm.

Algorithm S1 - Row Phase Rotations

Inputs: $\mathbf{S}_s, PC_i$ {# of populated codes in input $\mathbf{S}_s$ }, $\mathbf{A}_{syms}$

Outputs: $\mathbf{S}_s, PC_o, \mathbf{A}_{syms}$

```
1:  $PC_o = PC_i \cdot 2^{ar}$  {Operation limit =  $2^{ar}$ }
2: for  $it = 1$  to  $2^{ar} - 1$  do
3:    $\mathbf{v}_{row} = de2bi(it, r, 2^a)$  {Row PR vector}
4:   In  $\mathbf{v}_{row}$ , replace each element,  $x$ , with  $e^{j2\pi x/2^a}$ 
5:    $\mathbf{T}_{row} = diag(\mathbf{v}_{row})$ 
6:   for  $mi = 1$  to  $PC_i$  do
7:      $\mathbf{A}_{syms}(PC_i \cdot it + mi, :) = \mathbf{A}_{syms}(mi, :)$  {Update symbols record array}
8:     for  $di = 1$  to  $d$  do
9:        $\mathbf{S}_s(:, :, di, PC_i \cdot it + mi) = \mathbf{T}_{row} \times \mathbf{S}_s(:, :, di, mi)$ 
10:    end for
11:  end for
12: end for
13: return  $\mathbf{S}_s, PC_o, \mathbf{A}_{syms}$ 
```

Although all $c!$ column permutations are permitted for the majority of STCs investigated, similarity of particular columns within MPO334 and MPO434 requires the column permutations algorithm to be limited to prevent duplicate codes. For the MPO334 STC, column permutations that are similar to a previously allowed permutation but with only the first and second columns swapped are not permitted. For the MPO434 STC, column permutations that are similar to a previously allowed permutation but with only the first and second columns swapped and/or third and fourth columns swapped are not permitted. Due to this fact, an allowed column permutations vector ($\mathbf{v}_{acp}$) is employed within the column permutations algorithm.

Example 2.1.2. The full permutations array for a vector with three elements is shown in Fig. 2.1 using the modified *perms*(n) MATLAB function as previously described [54]. Using Fig. 2.1 as a reference since $c = 3$ for the MPO334 STC, only permutation indices one, two, and five are permitted (i.e., $\mathbf{v}_{acp} = [1 \ 2 \ 5]$). This is determined as permutation three is similar to permutation one with only the first and second columns swapped. Likewise, permutation four is similar to permutation two, and permutation six is similar to permutation five.

Algorithm S2 - Column Phase Rotations

Inputs: $\mathbf{S}_s, PC_i, \mathbf{A}_{syms}$
Outputs: $\mathbf{S}_s, PC_o, \mathbf{A}_{syms}$

```

1:  $PC_o = PC_i \cdot 2^{a(c-1)}$  {Operation limit =  $2^{a(c-1)}$ }
2: for  $it = 1$  to  $2^{a(c-1)} - 1$  do
3:    $\mathbf{v}_{col} = de2bi(it, c, 2^a)$  {Column PR vector}
4:   In  $\mathbf{v}_{col}$ , replace each element,  $x$ , with  $e^{j2\pi x/2^a}$ 
5:    $\mathbf{T}_{col} = diag(\mathbf{v}_{col})$ 
6:   for  $mi = 1$  to  $PC_i$  do
7:      $\mathbf{A}_{syms}(PC_i \cdot it + mi, :) = \mathbf{A}_{syms}(mi, :)$  {Update symbols record array}
8:     for  $di = 1$  to  $d$  do
9:        $\mathbf{S}_s(:, :, di, PC_i \cdot it + mi) = \mathbf{S}_s(:, :, di, mi) \times \mathbf{T}_{col}$ 
10:    end for
11:  end for
12: end for
13: return  $\mathbf{S}_s, PC_o, \mathbf{A}_{syms}$ 

```

Permutation Index

1	1	2	3
2	1	3	2
3	2	1	3
4	2	3	1
5	3	1	2
6	3	2	1

Figure 2.1. Full permutations array for a vector with three elements.

Symbol permutations are not required for all STCs. Row and column permutations are sufficient for the Alamouti and SPO334 STCs due to the rows and columns of $\mathbf{G}$ in (2.1) and (2.2) containing all possible combinations of data symbols for those codes. For all other STCs investigated, symbol permutations are required to provide all possible symbol combinations; however, not all symbol permutations are permitted in order to ensure uniqueness of codes within the set. An allowed symbol permutations vector ($\mathbf{v}_{asp}$) is used for this purpose within the symbol permutations algorithm. For most base codes, $\mathbf{v}_{asp}$ is empirically derived by performing each of the $k!$ symbol permutations in lexicographic order and recording the first of each set of permutations that yields a new combination of symbols per row, or per

Algorithm S3 - Symbol Phase Rotations

Inputs: $S_s, PC_i, A_{symS}, \mathbf{v}_{asr}$ {Allowed symbol PRs vector}	14:	ki = symbol index of $S_s(ri, ci, di, mi)$ {e.g., $ki = 3$ for $S_s(ri, ci, di, mi) == \frac{s_3}{\sqrt{2}}$ }
Outputs: S_s, PC_o, A_{symS}	15:	if $\mathbf{v}_{csr}(ki) \neq 0$ then
1: $PC_o = PC_i \cdot 2^d \mathbf{v}_{asr} $ {Operation limit = $2^d \mathbf{v}_{asr} $ }	16:	if $S_s(ri, ci, di, mi)$ is conjugated then
2: for $it = 1$ to $2^d \mathbf{v}_{asr} - 1$ do	17:	$g = -1$ {PR angle must be negated}
3: $\mathbf{v}_{sym} = de2bi(it, \mathbf{v}_{asr} , 2^d)$ {Symbol PR vector}	18:	else
4: for $mi = 1$ to PC_i do	19:	$g = 1$
5: $A_{symS}(PC_i \cdot it + mi, :) = A_{symS}(mi, :)$	20:	end if
{Update symbols record array}	21:	$x = \mathbf{v}_{csr}(ki)$ {Symbol PR angle index}
6: $\mathbf{v}_{psr} = A_{symS}(mi, \mathbf{v}_{asr})$	22:	$T(ri, ci) = e^{j2\pi gx/2^d}$
{Permuted allowed symbol PRs vector}	23:	end if
7: $\mathbf{v}_{csr}$ = vector of 0's of size 1-by- k	24:	end if
{Current iteration full-length symbol PR vector}	25:	end for
8: $\mathbf{v}_{csr}(\mathbf{v}_{psr}) = \mathbf{v}_{sym}$	26:	end for
{Insert $\mathbf{v}_{sym}$ into $\mathbf{v}_{csr}$ in permuted locations}	27:	$S_s(:, :, di, PC_i \cdot it + mi) = S_s(:, :, di, mi) \circ T$
9: for $di = 1$ to d do	28:	end for
10: T = 2-dimensional array of 1's of size r -by- c	29:	end for
11: for $ri = 1$ to r do	30:	end for
12: for $ci = 1$ to c do	31:	return S_s, PC_o, A_{symS}
13: if $S_s(ri, ci, di, mi) \neq 0$ then		

Algorithm S4 - Row Permutations

Inputs: S_s, PC_i, A_{syms} **Outputs:** S_s, PC_o, A_{syms}

```
1:  $PC_o = PC_i \cdot r!$  {Operation limit =  $r!$ }
2:  $R_{perms} = perms(r)$  {Row permutations array}
3: for  $it = 1$  to  $r! - 1$  do
4:   for  $mi = 1$  to  $PC_i$  do
5:      $A_{syms}(PC_i \cdot it + mi, :) = A_{syms}(mi, :)$  {Update symbols record array}
6:      $S_s(:, :, :, PC_i \cdot it + mi) = S_s(R_{perms}(it + 1, :), :, :, mi)$ 
7:   end for
8: end for
9: return  $S_s, PC_o, A_{syms}$ 
```

Algorithm S5 - Column Permutations

Inputs: $S_s, PC_i, A_{syms}, v_{acp}$ {Allowed column permutations vector}**Outputs:** S_s, PC_o, A_{syms}

```
1:  $PC_o = PC_i \cdot |v_{acp}|$  {Operation limit =  $|v_{acp}|$ }
2:  $C_{perms} = perms(c)$  {Column permutations array}
3: for  $it = 1$  to  $|v_{acp}| - 1$  do
4:   for  $mi = 1$  to  $PC_i$  do
5:      $A_{syms}(PC_i \cdot it + mi, :) = A_{syms}(mi, :)$  {Update symbols record array}
6:      $S_s(:, :, :, PC_i \cdot it + mi) = S_s(:, C_{perms}(v_{acp}(it + 1), :), :, mi)$ 
7:   end for
8: end for
9: return  $S_s, PC_o, A_{syms}$ 
```

column, of G .

Example 2.1.3. For the MPO468 STC, v_{asp} is empirically determined to be

$$v_{asp} = [1 \ 2 \ 3 \ 4 \ 5 \ 6 \ 7 \ 8 \ 9 \ 11 \ 13 \ 15 \ 31 \ 32 \ 37]. \quad (2.15)$$

Whereas 15 allowed symbol permutations, including the original, are recorded in (2.15), it can be shown that there are $k!/15 - 1 = 720/15 - 1 = 47$ other permutations for each of the entries in v_{asp} that produces the same combination of symbols per row of MPO468.

Algorithm S6 - Symbol Permutations

Inputs: $\mathbf{S}_s, PC_i, \mathbf{A}_{syms}$, $\mathbf{v}_{asp}$ {Allowed symbol permutations vector}, $\mathbf{s}_{ext}$ {Extended data symbol vector}	
Outputs: $\mathbf{S}_s, PC_o, \mathbf{A}_{syms}$	
1: $PC_o = PC_i \cdot \mathbf{v}_{asp} $ {Operation limit = $ \mathbf{v}_{asp} $ }	
2: $\mathbf{K}_{perms} = perms(k)$ {Symbol permutations array}	
3: for $it = 1$ to $ \mathbf{v}_{asp} - 1$ do	16: for $ci = 1$ to c do
4: for $mi = 1$ to PC_i do	17: if $\mathbf{S}_s(ri, ci, di, mi) \neq 0$ then
5: $\mathbf{S}_s(:, :, :, PC_i \cdot it + mi) = \mathbf{S}_s(:, :, :, mi)$	18: $ki = \text{symbol index of } \mathbf{S}_s(ri, ci, di, mi)$
6: {Initialize STC}	{e.g. $ki = 2$ for $\mathbf{S}_s(ri, ci, di, mi) == s_2$ }
7: $[\sim, \mathbf{v}_{kpi}] = sort(\mathbf{A}_{syms}(mi, :))$	19: if $\mathbf{v}_{kp}(ki) \neq ki$ then
8: { $\mathbf{v}_{kpi}$ holds indices of sorted symbol permutation}	20: $w = \text{weight applied to } \mathbf{S}_s(ri, ci, di, mi)$
9: $\mathbf{v}_{kp} = \mathbf{K}_{perms}(\mathbf{v}_{asp}(it + 1), \mathbf{v}_{kpi})$	{e.g., $w = \frac{1}{\sqrt{2}}$ for $\mathbf{S}_s(ri, ci, di, mi) == \frac{-s_1}{\sqrt{2}}$ }
10: {Symbol permutation vector}	$\phi = \text{PR angle applied to } \mathbf{S}_s(ri, ci, di, mi)$
11: if $\mathbf{v}_{kp} == [1 \dots k]$ then	{e.g., $\phi = \pi$ for $\mathbf{S}_s(ri, ci, di, mi) == \frac{-s}{\sqrt{2}}$ }
12: $\mathbf{A}_{syms}(PC_i \cdot it + mi, :) = [1 \dots k]$	if $\mathbf{S}_s(ri, ci, di, mi)$ is conjugated then
13: {Update symbols record array}	$g = 1$
14: $\mathbf{v}_{kp} = \mathbf{v}_{kpi}$	{Permutated symbol must be conjugated}
15: {Reassign symbol permutation vector}	else
16: else	$g = 0$
17: $\mathbf{A}_{syms}(PC_i \cdot it + mi, :) = \mathbf{K}_{perms}(\mathbf{v}_{asp}(it + 1), :)$	end if
18: {Update symbols record array}	$\mathbf{S}_s(ri, ci, di, PC_i \cdot it + mi) =$
19: end if	$w \cdot e^{j\phi} \cdot \mathbf{s}_{ext}(\mathbf{v}_{kp}(ki) + g \cdot k)$
20: for $di = 1$ to d do	end if
21: for $ri = 1$ to r do	end for
22: end for	end for
23: end for	end for
24: end for	end for
25: end for	end for
26: end for	end for
27: end for	end for
28: end for	end for
29: end for	end for
30: end for	end for
31: end for	end for
32: end for	end for
33: end for	end for
34: end for	end for
35: end for	end for
36: end for	end for
37: end for	end for
38: end for	end for
39: end for	end for
40: end for	end for
41: end for	end for
42: end for	end for
43: end for	end for
44: end for	end for
45: end for	end for
46: end for	end for
47: end for	end for
48: end for	end for
49: end for	end for
50: end for	end for
51: end for	end for
52: end for	end for
53: end for	end for
54: end for	end for
55: end for	end for
56: end for	end for
57: end for	end for
58: end for	end for
59: end for	end for
60: end for	end for
61: end for	end for
62: end for	end for
63: end for	end for
64: end for	end for
65: end for	end for
66: end for	end for
67: end for	end for
68: end for	end for
69: end for	end for
70: end for	end for
71: end for	end for
72: end for	end for
73: end for	end for
74: end for	end for
75: end for	end for
76: end for	end for
77: end for	end for
78: end for	end for
79: end for	end for
80: end for	end for
81: end for	end for
82: end for	end for
83: end for	end for
84: end for	end for
85: end for	end for
86: end for	end for
87: end for	end for
88: end for	end for
89: end for	end for
90: end for	end for
91: end for	end for
92: end for	end for
93: end for	end for
94: end for	end for
95: end for	end for
96: end for	end for
97: end for	end for
98: end for	end for
99: end for	end for
100: end for	end for
101: end for	end for
102: end for	end for
103: end for	end for
104: end for	end for
105: end for	end for
106: end for	end for
107: end for	end for
108: end for	end for
109: end for	end for
110: end for	end for
111: end for	end for
112: end for	end for
113: end for	end for
114: end for	end for
115: end for	end for
116: end for	end for
117: end for	end for
118: end for	end for
119: end for	end for
120: end for	end for
121: end for	end for
122: end for	end for
123: end for	end for
124: end for	end for
125: end for	end for
126: end for	end for
127: end for	end for
128: end for	end for
129: end for	end for
130: end for	end for
131: end for	end for
132: end for	end for
133: end for	end for
134: end for	end for
135: end for	end for
136: end for	end for
137: end for	end for
138: end for	end for
139: end for	end for
140: end for	end for
141: end for	end for
142: end for	end for
143: end for	end for
144: end for	end for
145: end for	end for
146: end for	end for
147: end for	end for
148: end for	end for
149: end for	end for
150: end for	end for
151: end for	end for
152: end for	end for
153: end for	end for
154: end for	end for
155: end for	end for
156: end for	end for
157: end for	end for
158: end for	end for
159: end for	end for
160: end for	end for
161: end for	end for
162: end for	end for
163: end for	end for
164: end for	end for
165: end for	end for
166: end for	end for
167: end for	end for
168: end for	end for
169: end for	end for
170: end for	end for
171: end for	end for
172: end for	end for
173: end for	end for
174: end for	end for
175: end for	end for
176: end for	end for
177: end for	end for
178: end for	end for
179: end for	end for
180: end for	end for
181: end for	end for
182: end for	end for
183: end for	end for
184: end for	end for
185: end for	end for
186: end for	end for
187: end for	end for
188: end for	end for
189: end for	end for
190: end for	end for
191: end for	end for
192: end for	end for
193: end for	end for
194: end for	end for
195: end for	end for
196: end for	end for
197: end for	end for
198: end for	end for
199: end for	end for
200: end for	end for
201: end for	end for
202: end for	end for
203: end for	end for
204: end for	end for
205: end for	end for
206: end for	end for
207: end for	end for
208: end for	end for
209: end for	end for
210: end for	end for
211: end for	end for
212: end for	end for
213: end for	end for
214: end for	end for
215: end for	end for
216: end for	end for
217: end for	end for
218: end for	end for
219: end for	end for
220: end for	end for
221: end for	end for
222: end for	end for
223: end for	end for
224: end for	end for
225: end for	end for
226: end for	end for
227: end for	end for
228: end for	end for
229: end for	end for
230: end for	end for
231: end for	end for
232: end for	end for
233: end for	end for
234: end for	end for
235: end for	end for
236: end for	end for
237: end for	end for
238: end for	end for
239: end for	end for
240: end for	end for
241: end for	end for
242: end for	end for
243: end for	end for
244: end for	end for
245: end for	end for
246: end for	end for
247: end for	end for
248: end for	end for
249: end for	end for
250: end for	end for
251: end for	end for
252: end for	end for
253: end for	end for
254: end for	end for
255: end for	end for
256: end for	end for
257: end for	end for
258: end for	end for
259: end for	end for
260: end for	end for
261: end for	end for
262: end for	end for
263: end for	end for
264: end for	end for
265: end for	end for
266: end for	end for
267: end for	end for
268: end for	end for
269: end for	end for
270: end for	end for
271: end for	end for
272: end for	end for
273: end for	end for
274: end for	end for
275: end for	end for
276: end for	end for
277: end for	end for
278: end for	end for
279: end for	end for
280: end for	end for
281: end for	end for
282: end for	end for
283: end for	end for
284: end for	end for
285: end for	end for
286: end for	end for
287: end for	end for
288: end for	end for
289: end for	end for
290: end for	end for
291: end for	end for
292: end for	end for
293: end for	end for
294: end for	end for
295: end for	end for
296: end for	end for
297: end for	end for
298: end for	end for
299: end for	end for
300: end for	end for
301: end for	end for
302: end for	end for
303: end for	end for
304: end for	end for
305: end for	end for
306: end for	end for
307: end for	end for
308: end for	end for
309: end for	end for
310: end for	end for
311: end for	end for
312: end for	end for
313: end for	end for

Algorithm S7 - Symbol Conjugations

Inputs: $S_s, PC_i, A_{syms}, v_{asc}$ {Allowed symbol conjugations vector}, s_{ext}

Outputs: S_s, PC_o, A_{syms}

```
1:  $PC_o = PC_i \cdot 2^{|v_{asc}|}$  {Operation limit =  $2^{|v_{asc}|}$ }
2: for  $it = 1$  to  $2^{|v_{asc}|} - 1$  do
3:    $v_{sym} = de2bi(it, |v_{asc}|, 2)$  {Symbol conjugation vector}
4:   for  $mi = 1$  to  $PC_i$  do
5:      $S_s(:, :, :, PC_i \cdot it + mi) = S_s(:, :, :, mi)$  {Initialize STC}
6:      $A_{syms}(PC_i \cdot it + mi, :) = A_{syms}(mi, :)$  {Update symbols record array}
7:      $v_{psc} = A_{syms}(mi, v_{asc})$  {Permuted allowed symbol conjugations vector}
8:      $v_{csc} =$  vector of 0's of size 1-by- $k$ 
      {Current iteration full-length symbol conjugation vector}
9:      $v_{csc}(v_{psc}) = v_{sym}$  {Insert  $v_{sym}$  into  $v_{csc}$  in permuted locations}
10:    for  $di = 1$  to  $d$  do
11:      for  $ri = 1$  to  $r$  do
12:        for  $ci = 1$  to  $c$  do
13:          if  $S_s(ri, ci, di, mi) \neq 0$  then
14:             $ki =$  symbol index of  $S_s(ri, ci, di, mi)$ 
              {e.g.,  $ki = 1$  for  $S_s(ri, ci, di, mi) == s_1$ }
15:            if  $v_{csc}(ki) \neq 0$  then
16:               $w =$  positive weight applied to  $S_s(ri, ci, di, mi)$ 
                {e.g.,  $w = \frac{1}{\sqrt{2}}$  for  $S_s(ri, ci, di, mi) == \frac{-s_1^*}{\sqrt{2}}$ }
17:               $\phi =$  PR angle applied to  $S_s(ri, ci, di, mi)$ 
                {e.g.,  $\phi = \pi$  for  $S_s(ri, ci, di, mi) == \frac{-s_1^*}{\sqrt{2}}$ }
18:              if  $S_s(ri, ci, di, mi)$  is conjugated then
19:                 $g = 0$ 
20:              else
21:                 $g = 1$ 
22:              end if
23:               $S_s(ri, ci, di, PC_i \cdot it + mi) = w \cdot e^{j\phi} \cdot s_{ext}(ki + g \cdot k)$ 
24:            end if
25:          end if
26:        end for
27:      end for
28:    end for
29:  end for
30: end for
31: return  $S_s, PC_o, A_{syms}$ 
```

For the MPO334 and MPO434 codes, $\mathbf{v}_{asp}$ is derived by determining the first of each set of permutations that yields a new combination of symbol weights.

Example 2.1.4. For both MPO334 and MPO434, $\mathbf{v}_{asp}$ is empirically determined to be

$$\mathbf{v}_{asp} = [1 \ 2 \ 6]. \quad (2.16)$$

Elements within $\mathbf{v}_{asp}$ represent allowed permutation indices. Using Fig. 2.1 as a reference since $k = 3$ for these STCs, it can be seen that elements given in (2.16) allow for the original symbol permutation (i.e., $[1 \ 2 \ 3]$), swapping symbols s_2 and s_3 (i.e., $[1 \ 3 \ 2]$), or swapping symbols s_1 and s_3 (i.e., $[3 \ 2 \ 1]$) for MPO334 and MPO434.

All 2^k combinations of symbol conjugations may be performed for the majority of STCs investigated; however, only the first 2^{k-1} combinations are permitted for the Alamouti and NO442 STCs to prevent building duplicate codes. Thus, an allowed symbol conjugations vector ($\mathbf{v}_{asc}$) is used for this purpose within the symbol conjugations algorithm.

For all SMSTCs, data symbols are inserted into every STC matrix element without repetition; thus, k is equal to the number of matrix elements. Due to this structure, a set built using any SMSTC as the base code may be produced using Algorithm S0 along with Algorithms S3, S6, and S7 alone. Thus, the operation limit for row and column PRs and permutations is equal to 1 to indicate that these operations are not performed. For the symbol PRs algorithm, $\mathbf{v}_{asr} = [1 \ \dots \ k]$. For the symbol permutations algorithm, $\mathbf{v}_{asp} = [1 \ \dots \ k!]$, and for the symbol conjugations algorithm, $\mathbf{v}_{asc} = [1 \ \dots \ k]$.

The variable vectors employed to provide operation limits for each of the base codes given in Section 2.1.1 are given in Table 2.1.

Tracking Symbol Permutations

The symbols record array, denoted as $\mathbf{A}_{syms}$, is used for tracking symbol permutations throughout performance of each algorithm. When the $\mathbf{G}$ input to Algorithm S0 is a non-symbol-permuted version of the original base code for which $\mathbf{v}_{asp}$ is empirically derived, then $\mathbf{K}_{perm} = [1 \ \dots \ k]$ is input to Algorithm S0. For this case, symbol permutation tracking is only necessary when Algorithm S6 is performed prior to Algorithms S3 or S7. This serves to permute $\mathbf{v}_{asr}$ and $\mathbf{v}_{asc}$ as performed in Algorithms S3 and S7, respectively.

Table 2.1. Variable vectors for each base STC given in Section 2.1.1.

STC	c	k	r	$\mathbf{v}_{asr}$	$\mathbf{v}_{acp}$	$\mathbf{v}_{asp}$	$\mathbf{v}_{asc}$
Alamouti	2	2	2	[]	[1 2]	[1]	[1]
SPO334	3	3	4	[]	[1 ... 6]	[1]	[1 2 3]
MPO468	4	6	8	[1 2]	[1 ... 24]	(2.15)	[1 ... 6]
MPO334	3	3	4	[1]	[1 2 5]	[1 2 6]	[1 2 3]
MPO434	4	3	4	[1]	[1 3 4 13 14 17]	[1 2 6]	[1 2 3]
NO442	4	4	2	[1]	[1 ... 24]	[1 3 6]	[1 2 3]
SM221	2	2	1	[1 2]	[1]	[1 2]	[1 2]
SM441	4	4	1	[1 2 3 4]	[1]	[1 ... 24]	[1 2 3 4]

On the contrary, when a symbol-permuted version of the base code is used as the $\mathbf{G}$ input to Algorithm S0, the $\mathbf{K}_{perm}$ input to Algorithm S0 is vital to properly initialize the symbols record array. $\mathbf{A}_{syms}$ is then used within Algorithm S6 to determine the correct symbol permutation vector, $\mathbf{v}_{kp}$, to prevent duplicate codes from being built.

Example 2.1.5. If the symbol-permuted version of the MPO468 STC given in (2.14) is used as the base code for set building, then $\mathbf{K}_{perm} = [1 \ 2 \ 3 \ 4 \ 6 \ 5]$ is input to Algorithm S0.

2.1.4 Set Cardinalities

The cardinality of $\mathbf{S}_s$, denoted as $|\mathbf{S}_s|$, including the base code is equal to the product of the operation limits employed when building the set. For all base STCs given in Section 2.1.1 with the exception of SMSTCs, the set cardinality is

$$|\mathbf{S}_s| = 2^{(a(r+c+|\mathbf{v}_{asr}|-1)+|\mathbf{v}_{asc}|)} \cdot r! \cdot |\mathbf{v}_{acp}| \cdot |\mathbf{v}_{asp}|. \quad (2.17)$$

Since the row and column PR and permutation algorithms are not employed, the cardinality

of any set built using a SMSTC with k data symbols as the base code is given as

$$|\mathbf{S}_s| = 2^{k(a+1)} \cdot k!. \quad (2.18)$$

Cardinality of sets built with each base STC is provided in Table 2.2 for various values of a .

2.2 Modulation Mapping Sets

We now look at base modulation mappings and methods to produce alternative mappings from each.

2.2.1 Base Mappings

There are six modulation schemes specified for the 5G New Radio (5G NR) including binary phase-shift keying (BPSK), $\pi/2$ -BPSK, QPSK, 16QAM, 64QAM, and 256QAM [56]. A modulation mapper function is provided in [56, Section 5.1] for each of these schemes. These functions represent BRGC mappings from bits to complex-valued modulation symbols. The BPSK schemes are not utilized for this research. The remaining 5G NR modulation mapper functions are considered to be baseline mappings.

2.2.2 Alternative Mapping Operations

It was stated in [57] and again in [43] that operations such as bit permutations and inversions can transform BRGC mappings into alternative mappings exhibiting the same expected BER. Thus, if a bit permutation or bit inversion, *aka* exclusive-or (XOR) bit mask, is uniformly applied to the binary labels of all constellation points, the minimum distance properties of the original BRGC mapping are preserved.

For a given modulation scheme with M symbols and $b = \log_2(M)$ bits per symbol, there are $b!$ bit permutations and 2^b bit inversions leading to a total of

$$2^b \cdot b! \quad (2.19)$$

alternative BRGC mappings with equivalent expected BER performance. Upon comparison

Table 2.2. Set cardinality of base STCs for various values of a .

STC	Alamouti	SPO334	MPO468	MPO334	MPO434	NO442	SM221	SM441
$ \mathbf{V}_{ast} $	0	0	2	1	1	1	2	4
$ \mathbf{V}_{acp} $	2	6	24	3	6	24	1	1
$ \mathbf{V}_{asp} $	1	1	15	3	3	3	2	24
$ \mathbf{V}_{asc} $	1	3	6	3	3	3	2	4
$ \mathbf{S}_s $ ($a = 1$)	64	73, 728	$7.61 \cdot 10^{12}$	221, 184	884, 736	73, 728	32	6, 144
$ \mathbf{S}_s $ ($a = 2$)	512	4, 718, 592	$6.23 \cdot 10^{16}$	$2.83 \cdot 10^7$	$2.26 \cdot 10^8$	4, 718, 592	128	98, 304
$ \mathbf{S}_s $ ($a = 3$)	4, 096	$3.02 \cdot 10^8$	$5.11 \cdot 10^{20}$	$3.62 \cdot 10^9$	$5.80 \cdot 10^{10}$	$3.02 \cdot 10^8$	512	1, 572, 864
$ \mathbf{S}_s $ ($a = 4$)	32, 768	$1.93 \cdot 10^{10}$	$4.18 \cdot 10^{24}$	$4.64 \cdot 10^{11}$	$1.48 \cdot 10^{13}$	$1.93 \cdot 10^{10}$	2, 048	$2.52 \cdot 10^7$
$ \mathbf{S}_s $ ($a = 5$)	262, 144	$1.24 \cdot 10^{12}$	$3.43 \cdot 10^{28}$	$5.94 \cdot 10^{13}$	$3.80 \cdot 10^{15}$	$1.24 \cdot 10^{12}$	8, 192	$4.03 \cdot 10^8$
$ \mathbf{S}_s $ ($a = 6$)	2, 097, 152	$7.92 \cdot 10^{13}$	$2.81 \cdot 10^{32}$	$7.60 \cdot 10^{15}$	$9.73 \cdot 10^{17}$	$7.92 \cdot 10^{13}$	32, 768	$6.44 \cdot 10^9$
$ \mathbf{S}_s $ ($a = 9$)	$1.07 \cdot 10^9$	$2.08 \cdot 10^{19}$	$1.54 \cdot 10^{44}$	$1.59 \cdot 10^{22}$	$1.63 \cdot 10^{25}$	$2.08 \cdot 10^{19}$	2, 097, 152	$2.64 \cdot 10^{13}$

of these alternative mappings, it can be shown that there are $b!$ mappings that pair each of the constellation symbols with each of the binary labels.

Table 2.3 provides all equivalent BRGC mappings for the QPSK constellation with $M = 4$ and $b = 2$. Using (2.19), there are $2^2 \cdot 2! = 8$ mappings for QPSK. The symbols s_1, s_2, s_3 , and s_4 refer to the constellation points $+1 + j1, +1 - j1, -1 + j1$, and $-1 - j1$ each scaled by $1/\sqrt{2}$, respectively. These equivalent mappings are also shown in Figure 2.2 with colored labels corresponding to the associated mapping index as shown in Table 2.3.

Table 2.3. All equivalent BRGC mappings for QPSK.

BRGC Mapping Index	1	2	3	4	5	6	7	8
Bit Mask	[0 0]	[1 0]	[0 1]	[1 1]	[0 0]	[1 0]	[0 1]	[1 1]
Bit Permutation	[1 2]	[1 2]	[1 2]	[1 2]	[2 1]	[2 1]	[2 1]	[2 1]
s_1 Label	$\begin{bmatrix} 0 & 0 \end{bmatrix}$	$\begin{bmatrix} 1 & 0 \end{bmatrix}$	$\begin{bmatrix} 0 & 1 \end{bmatrix}$	$\begin{bmatrix} 1 & 1 \end{bmatrix}$	$\begin{bmatrix} 0 & 0 \end{bmatrix}$	$\begin{bmatrix} 0 & 1 \end{bmatrix}$	$\begin{bmatrix} 1 & 0 \end{bmatrix}$	$\begin{bmatrix} 1 & 1 \end{bmatrix}$
s_2 Label	$\begin{bmatrix} 0 & 1 \end{bmatrix}$	$\begin{bmatrix} 1 & 1 \end{bmatrix}$	$\begin{bmatrix} 0 & 0 \end{bmatrix}$	$\begin{bmatrix} 1 & 0 \end{bmatrix}$	$\begin{bmatrix} 1 & 0 \end{bmatrix}$	$\begin{bmatrix} 1 & 1 \end{bmatrix}$	$\begin{bmatrix} 0 & 0 \end{bmatrix}$	$\begin{bmatrix} 0 & 1 \end{bmatrix}$
s_3 Label	$\begin{bmatrix} 1 & 0 \end{bmatrix}$	$\begin{bmatrix} 0 & 0 \end{bmatrix}$	$\begin{bmatrix} 1 & 1 \end{bmatrix}$	$\begin{bmatrix} 0 & 1 \end{bmatrix}$	$\begin{bmatrix} 0 & 1 \end{bmatrix}$	$\begin{bmatrix} 0 & 0 \end{bmatrix}$	$\begin{bmatrix} 1 & 1 \end{bmatrix}$	$\begin{bmatrix} 1 & 0 \end{bmatrix}$
s_4 Label	$\begin{bmatrix} 1 & 1 \end{bmatrix}$	$\begin{bmatrix} 0 & 1 \end{bmatrix}$	$\begin{bmatrix} 1 & 0 \end{bmatrix}$	$\begin{bmatrix} 0 & 0 \end{bmatrix}$	$\begin{bmatrix} 1 & 1 \end{bmatrix}$	$\begin{bmatrix} 1 & 0 \end{bmatrix}$	$\begin{bmatrix} 0 & 1 \end{bmatrix}$	$\begin{bmatrix} 0 & 0 \end{bmatrix}$

These alternative mappings are created by applying the associated bit mask and bit permutation, in that order, to the baseline labels given for BRGC mapping index 1. This baseline corresponds to the 5G NR modulation mapper for QPSK specified in [56, Section 5.1.3] as

$$d(i) = \frac{1}{\sqrt{2}} \left[(1 - 2b(2i)) + j(1 - 2b(2i + 1)) \right]. \quad (2.20)$$

The left-hand label bit shown for each symbol in Table 2.3 for BRGC mapping index 1 corresponds to $b(2i)$ in (2.20). Likewise, the right-hand label bit for each symbol corresponds to $b(2i + 1)$, and symbols s_1 through s_4 correspond to the produced complex-valued modulation symbol $d(i)$.

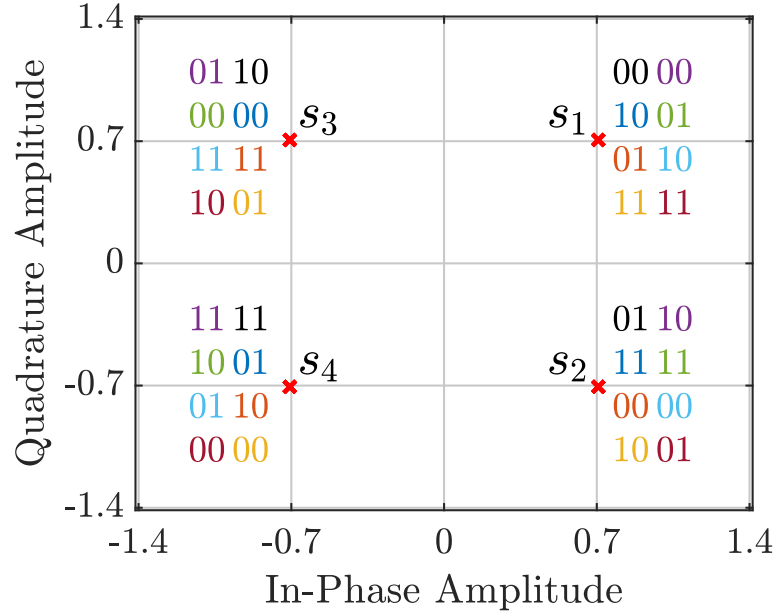


Figure 2.2. Illustration of all equivalent BRGC mappings for QPSK with colors as shown in Table 2.3.

2.2.3 Mapping Set Building Algorithms

Operations from the previous section are now implemented in a systematic manner to build a set, $\mathbf{S}_m$, of unique alternative mappings. Algorithm M0 is performed first followed by Algorithms M1 - M2 in any order.

Algorithm Notes

As in Section 2.1.3, values of b and M for the chosen mapping are known to all algorithms in this dissertation. All text within curly brackets are comments to aid the reader. The $de2bi(i, n, p)$ MATLAB function and slightly modified version of the $perms(n)$ MATLAB function previously described in Section 2.1.3 are also employed here [52], [54]. Lastly, the $\oplus$ symbol is used to denote XOR operation.

The base mapping input to Algorithm M0 is an M -by- b array, denoted as $\mathbf{B}_m$, where each row contains the bit label associated with the corresponding symbol. Thus, the label bits corresponding to symbol s_1 are contained in the first row, and the label bits corresponding to symbol s_M are contained in the M th and final row of $\mathbf{B}_m$.

Example 2.2.1. The base mapping input to Algorithm M0 for BRGC mapping index 1 given in Table 2.3 is represented as

$$\mathbf{B}_m = \begin{bmatrix} 0 & 0 \\ 0 & 1 \\ 1 & 0 \\ 1 & 1 \end{bmatrix}. \quad (2.21)$$

Algorithm M0 - Mapping Set Initialization

Inputs: $\mathbf{B}_m$ {Base Mapping}, $|\mathbf{S}_m|$ {Mapping set cardinality}

Outputs: $\mathbf{S}_m$ {Set of Mappings}, PM_o {# of populated mappings in output $\mathbf{S}_m$ }

- 1: $PM_o = 1$
 - 2: $\mathbf{S}_m = 3$ -dimensional array of 0's of size M -by- b -by- $|\mathbf{S}_m|$
 - 3: $\mathbf{S}_m(:, :, 1) = \mathbf{B}_m$ {Populate set index 1 with Base Mapping}
 - 4: **return** $\mathbf{S}_m, PM_o$
-

Algorithm M1 - Bit Inversions

Inputs: $\mathbf{S}_m, PM_i$ {# of populated mappings in input $\mathbf{S}_m$ }

Outputs: $\mathbf{S}_m, PM_o$

- 1: $PM_o = PM_i \cdot 2^b$ {Operation limit = 2^b }
 - 2: **for** $it = 1$ to $2^b - 1$ **do**
 - 3: $\mathbf{v}_{bm} = de2bi(it, b, 2)$ {Bit mask vector}
 - 4: **for** $mi = 1$ to PM_i **do**
 - 5: **for** $ri = 1$ to M **do**
 - 6: **for** $ci = 1$ to b **do**
 - 7: $\mathbf{S}_m(ri, ci, PM_i \cdot it + mi) = \mathbf{S}_m(ri, ci, mi) \oplus \mathbf{v}_{bm}(ci)$
 - 8: **end for**
 - 9: **end for**
 - 10: **end for**
 - 11: **end for**
 - 12: **return** $\mathbf{S}_m, PM_o$
-

Algorithm M2 - Bit Permutations

Inputs: S_m, PM_i **Outputs:** S_m, PM_o

```
1:  $PM_o = PM_i \cdot b!$  {Operation limit =  $b!$ }
2:  $\mathbf{B}_{perms} = perms(b)$  {Bit permutations array}
3: for  $it = 1$  to  $b! - 1$  do
4:    $\mathbf{v}_{bp} = \mathbf{B}_{perms}(it + 1, :)$  {Bit permutation vector}
5:   for  $mi = 1$  to  $PM_i$  do
6:     for  $ri = 1$  to  $M$  do
7:        $S_m(ri, :, PM_i \cdot it + mi) = S_m(ri, \mathbf{v}_{bp}, mi)$ 
8:     end for
9:   end for
10: end for
11: return  $S_m, PM_o$ 
```

2.2.4 Set Cardinalities

Using (2.19) for the baseline mappings discussed in Section 2.2.1, the cardinality of each set is calculated and recorded in Table 2.4.

Table 2.4. Set cardinality of various baseline modulation mappings.

Modulation	QPSK	16QAM	64QAM	256QAM
$ S_m $	8	384	46,080	10,321,920

CHAPTER 3:

Transmission and Reception Support Methods

To support alternating STCs and GCH, methods are required to facilitate transmission and reception. Here we present methods to build the next STC to be employed, decode various types of STCs, and perform GCH modulation and demodulation.

This chapter includes adaptations from previous work already published or in publication by the author. Specifically, portions of Sections 3.1, 3.2, and 3.3 are revised from [9]. Portions of Section 3.2 are revised from [10]. Portions of Section 3.4 are revised from [11].

3.1 Individual Space-Time Code Building

In schemes that perform beamforming, precoding, or encoding by selecting a code for use from a known codebook, *aka* code set, it is common to store the entire codebook for use as a lookup table (LUT) [41], [58]; however, when the size of the codebook is very large, this may not be practical, especially for low-storage devices. With this in mind, it would be desirable to build specific selected STCs from the codebook for use as needed without storing the entire codebook. In order to do this, three pieces of information are needed:

1. the base code and
2. order of operations/algorithms used to build the intended codebook and
3. the index value of the desired code.

The base code must be known as it is the first indexed code in the set. The order of operations conducted must also be known as building the set with different order of operations effectively reindexes the entire set with the exception of the first code. Naturally, the set is indexed from 1 to $|\mathbf{S}_s|$. The index value must be known as it specifies the iteration of each operation to be performed in order to build the selected code, where the iteration of an operation refers to the value of the *it* variable in each one of the algorithms provided. A fourth piece of information, namely the operation limits discussed in Section 2.1.3, must also be known but can generally be considered as common knowledge for a given code set.

With these pieces of information, a specific STC can be built by starting with the base code,

determining the specified iteration of each operation to perform based on the given index value, and performing each operation in the given order. The process for determining the specified iteration of each operation starts by defining a 1-by- l divisor vector, $\mathbf{v}_{div}$, with elements

$$\mathbf{v}_{div}(i) = \begin{cases} 1 & i = 1 \\ \prod_{j=1}^{i-1} \mathbf{v}_{ol}(\mathbf{v}_{oo}(j)) & 1 < i \leq l, \end{cases} \quad (3.1)$$

where $\mathbf{v}_{ol}$ is a row vector containing the operation limits for Algorithms S1 - S7 in that order, $\mathbf{v}_{oo}$ is an order of operations row vector containing the number of each algorithm in the order in which they are performed to build a code, and l is the length of each row vector, which equals seven when all of the provided algorithms are performed. Upon definition of $\mathbf{v}_{div}$, the iteration of each operation can be contained within a 1-by- l iterations vector, $\mathbf{v}_{it}$, with elements

$$\mathbf{v}_{it}(i) = \begin{cases} \left\lfloor \frac{IV-1}{\mathbf{v}_{div}(i)} \right\rfloor \bmod \mathbf{v}_{ol}(\mathbf{v}_{oo}(i)) & 1 \leq i < l \\ \left\lfloor \frac{IV-1}{\mathbf{v}_{div}(i-1)} \right\rfloor / \mathbf{v}_{ol}(\mathbf{v}_{oo}(i-1)) & i = l, \end{cases} \quad (3.2)$$

where IV represents the index value of the desired code, “mod” represents the modulo operator, and $\lfloor \cdot \rfloor$ represents the floor function. With these vectors defined, it can be seen that

$$IV = \mathbf{v}_{it} \times \mathbf{v}_{div}^T + 1, \quad (3.3)$$

where +1 is required due to set indexing starting at 1.

Once the iterations have been determined, the desired code is built by performing Algorithm S0 first with the $|\mathbf{S}_s|$ input argument set to 1 since only 1 code will be built. Algorithms S1 - S7 are performed next in the order specified by $\mathbf{v}_{oo}$. To modify each of these algorithms for individual STC building instead of set building, any reference to $\mathbf{S}_s$ is replaced with $\mathbf{G}$, the values of PC_i and PC_o are set to 1, the it variable is set equal to the value contained in $\mathbf{v}_{it}$ for that algorithm, and all index values of $PC_i \cdot it + mi$ are replaced with 1. For any element of $\mathbf{v}_{it}$ equal to zero, the corresponding algorithm is not performed.

Example 3.1.1. Using the MPO468 base code with $\mathbf{v}_{oo} = [4 \ 6 \ 2 \ 1 \ 3 \ 5 \ 7]$, $IV = 3,824,537,371,943$, $a = 1$, variable vectors as given in Table 2.1, and operation limits

specified in Algorithms S1 - S7, i.e.,

$$\mathbf{v}_{ol} = \begin{bmatrix} 2^{ar} & 2^{a(c-1)} & 2^{a|\mathbf{v}_{asr}|} & r! & |\mathbf{v}_{acp}| & |\mathbf{v}_{asp}| & 2^{|\mathbf{v}_{asc}|} \end{bmatrix} \quad (3.4)$$

$$= \begin{bmatrix} 256 & 8 & 4 & 40320 & 24 & 15 & 64 \end{bmatrix},$$

$$\mathbf{v}_{div} = \begin{bmatrix} 1 \\ 40320 \\ 604800 \\ 4838400 \\ 1238630400 \\ 4954521600 \\ 118908518400 \end{bmatrix}^T, \quad (3.5)$$

$$\mathbf{v}_{it} = \begin{bmatrix} 20902 & 12 & 7 & 182 & 3 & 3 & 32 \end{bmatrix}, \quad (3.6)$$

and the built STC is

$$\mathbf{G} = \begin{bmatrix} -s_2 & -s_6^* & 0 & -s_5 \\ s_3^* & 0 & -s_6^* & -s_1^* \\ -s_1 & s_4 & 0 & -s_3 \\ s_4^* & s_1^* & -s_5 & 0 \\ 0 & s_5^* & s_1 & -s_6 \\ -s_6 & s_2^* & -s_3 & 0 \\ 0 & s_3^* & s_2 & s_4^* \\ -s_5^* & 0 & -s_4 & s_2^* \end{bmatrix}. \quad (3.7)$$

It should be noted that the order of operations conducted to build this STC directed Algorithm S6 to be performed prior to Algorithm S3; therefore, $\mathbf{A}_{syms} = [1 \ 3 \ 4 \ 2 \ 5 \ 6]$ for this STC was used to permute $\mathbf{v}_{asr} = [1 \ 2]$ to $\mathbf{v}_{psr} = [1 \ 3]$.

3.2 Decoding Methods for Alternating Space-Time Codes

As the STC in use alternates, the receiver, *aka* UE, must be able to perform channel equalization, co-phase symbol copies as necessary, and sort data symbols into the correct location before demodulation. The processes employed by the receiver for the different

types of STCs presented in Section 2.1.1 are now discussed starting with the description of a communications link to aid in the discussion.

3.2.1 Alternating STC Communications Link

A potential MIMO communications link between BS and UE begins with a common digital modulation scheme such as QPSK or MQAM to convert a binary source to data symbols. These symbols are inserted into the next pseudorandomly built STC and transmitted across the wireless interface using c TX antennas over r symbol time periods.

Without loss of generality, the received sample array, $\mathbf{Z}$, is of size r -by- N_R and represented as

$$\mathbf{Z} = \mathbf{G} \times \mathbf{h}_{BU} + \mathbf{n}, \quad (3.8)$$

where N_R represents the number of receive (RX) antennas, $\mathbf{G}$ is the next alternating STC, $\mathbf{h}_{BU}$ is the c -by- N_R channel tap array between BS and UE, and $\mathbf{n}$ is an r -by- N_R AWGN array. The term array is used here to indicate a component that may be a vector or matrix depending on the STC in use and value of N_R .

UE is assumed to have perfect CSI and knowledge of the STC used with each codeword. UE performs the applicable decoding method depending on the STC in use to equalize and decode the data symbols and demodulates the symbols to binary data using the known digital modulation scheme.

3.2.2 Orthogonal Space-Time Codes

When an orthogonal STC is employed, UE performs maximal ratio combining (MRC) to equalize and decode the data symbols prior to symbol demodulation.

The general MRC process applicable to all orthogonal STCs given in Section 2.1.1 can be described by two steps performed in sequence for each RX antenna to obtain an estimated data symbol array, $\hat{\mathbf{s}}$. Each column of $\hat{\mathbf{s}}$ contains the estimated data symbol vector for the corresponding RX antenna. When $N_R > 1$, a weighted average of the columns of $\hat{\mathbf{s}}$ is performed to calculate the final estimated data symbol vector, $\hat{\mathbf{s}}$.

The two-step sequence for the i th RX antenna can be seen as calculating

$$\hat{\mathbf{s}}_{int} = \mathbf{H}_{CBU}^\dagger \times \mathbf{Z}(:, i) \quad (3.9)$$

followed by

$$\hat{\mathbf{s}}(x, i) = \frac{\hat{\mathbf{s}}_{int}(x) + \hat{\mathbf{s}}_{int}^*(x + k)}{\|\mathbf{h}_{BU}(:, i)\|^2} \quad (3.10)$$

for $x \in \{1, \dots, k\}$, where $\|\mathbf{h}_{BU}(:, i)\|^2$ is the square of the Euclidean norm of $\mathbf{h}_{BU}(:, i)$, $\hat{\mathbf{s}}_{int}$ is a $2k$ -by-1 intermediate vector, and $\mathbf{H}_{CBU}$ is an r -by- $2k$ combining matrix corresponding to $\mathbf{G}$ such that

$$\mathbf{H}_{CBU} \times \mathbf{s}_{ext} = \mathbf{G} \times \mathbf{h}_{BU}(:, i). \quad (3.11)$$

The intermediate vector represents a weighted version of $\mathbf{s}_{ext}$ with noise plus interference due to spreading of data symbols; however, completion of (3.10) cancels all interfering terms to produce the estimated data symbol vector for the i th RX antenna. This can be shown to be true according to the sufficient statistics

$$\hat{\mathbf{H}} = \|\mathbf{h}_{BU}(:, i)\|^2 \cdot \mathbf{I}_k, \quad (3.12)$$

where $\mathbf{I}_k$ is a k -by- k identity matrix,

$$\begin{aligned} \hat{\mathbf{H}}(x, y) = & \mathbf{H}_{int}(x, y) + \mathbf{H}_{int}^*(x + k, y + k) \\ & + \mathbf{H}_{int}(x, y + k) + \mathbf{H}_{int}^*(x + k, y) \end{aligned} \quad (3.13)$$

for $x, y \in \{1, \dots, k\}$, and

$$\mathbf{H}_{int} = \mathbf{H}_{CBU}^\dagger \times \mathbf{H}_{CBU}. \quad (3.14)$$

This represents a systematic approach to finding k vectors to solve for each of the k data symbols contained within any complex orthogonal STC using MRC. These sufficient statistics meet those given in [59, (10.272)].

When $N_R > 1$, these steps are repeated for each RX antenna. The final estimated data

symbol vector is obtained by taking the weighted average of the columns of $\hat{\mathbf{s}}$ as

$$\hat{\mathbf{s}} = \hat{\mathbf{s}} \times \begin{bmatrix} ||\mathbf{h}_{BU}(:, 1)||^2 / h_{BU} \\ \vdots \\ ||\mathbf{h}_{BU}(:, N_R)||^2 / h_{BU} \end{bmatrix}, \quad (3.15)$$

where

$$h_{BU} = \sum_{i=1}^{N_R} ||\mathbf{h}_{BU}(:, i)||^2 \quad (3.16)$$

is the summation of the squared Euclidean norms for each RX antenna channel tap vector.

3.2.3 Combining Matrix Determination

The process for determining the combining matrix, $\mathbf{H}_{CBU}$, for any given complex orthogonal STC matrix, $\mathbf{G}$, and channel tap vector, $\mathbf{h}_{BU}(:, i)$, can be seen as solving the system of equations given in (3.11) for the terms of $\mathbf{H}_{CBU}$ with appropriate constraints. In general, the terms of $\mathbf{H}_{CBU}$ are constrained to zeros, weighted terms from $\mathbf{h}_{BU}(:, i)$, and sums of these weighted terms. The use of sums of weighted terms is provided to account for this approach to be applied with the MPO334 and MPO434 STCs. An algorithmic approach for construction of these combining matrices given the channel tap vector and STC matrix is provided in Algorithm S8. For added generality, $\mathbf{H}_C$ and $\mathbf{h}$ are used in place of $\mathbf{H}_{CBU}$ and $\mathbf{h}_{BU}$, respectively.

Example 3.2.1. Given a single RX antenna, channel tap vector $\mathbf{h} = [h_1 \ h_2 \ h_3]^T$, and MPO334 base code originally given in (2.4) as

$$\mathbf{G} = \begin{bmatrix} s_1 & s_2 & \frac{s_3}{\sqrt{2}} \\ -s_2^* & s_1^* & \frac{s_3}{\sqrt{2}} \\ \frac{s_3^*}{\sqrt{2}} & \frac{s_3^*}{\sqrt{2}} & \frac{(-s_1 - s_1^* + s_2 - s_2^*)}{2} \\ \frac{s_3^*}{\sqrt{2}} & \frac{-s_3^*}{\sqrt{2}} & \frac{(s_2 + s_2^* + s_1 - s_1^*)}{2} \end{bmatrix},$$

Algorithm S8 - Combining Matrix Generation

Input: $\mathbf{G}$ {STC Matrix}, $\mathbf{h}$ {Channel tap vector}**Output:** $\mathbf{H}_C$ {Combining Matrix}

```
1:  $\mathbf{H}_C$  = 2-dimensional array of 0's of size  $r$ -by- $2k$ 
2: for  $di = 1$  to  $d$  do
3:   for  $ri = 1$  to  $r$  do
4:     for  $ci = 1$  to  $c$  do
5:       if  $\mathbf{G}(ri, ci, di) \neq 0$  then
6:          $ki$  = symbol index of  $\mathbf{G}(ri, ci, di)$  {e.g.,  $ki = 1$  for  $\mathbf{G}(ri, ci, di) == \frac{-s_1^*}{\sqrt{2}}$ }
7:          $w$  = positive weight applied to  $\mathbf{G}(ri, ci, di)$ 
           {e.g.,  $w = \frac{1}{\sqrt{2}}$  for  $\mathbf{G}(ri, ci, di) == \frac{-s_1^*}{\sqrt{2}}$ }
8:          $\phi$  = PR angle applied to  $\mathbf{G}(ri, ci, di)$  {e.g.,  $\phi = \pi$  for  $\mathbf{G}(ri, ci, di) == \frac{-s_1^*}{\sqrt{2}}$ }
9:         if  $\mathbf{G}(ri, ci, di)$  is conjugated then
10:           $g = 1$ 
11:        else
12:           $g = 0$ 
13:        end if
14:         $\mathbf{H}_C(ri, ki + g \cdot k) = \mathbf{H}_C(ri, ki + g \cdot k) + w \cdot e^{j\phi} \cdot \mathbf{h}(ci)$ 
15:      end if
16:    end for
17:  end for
18: end for
19: return  $\mathbf{H}_C$ 
```

the corresponding $\mathbf{H}_C$ matrix generated using Algorithm S8 is given as

$$\mathbf{H}_C = \begin{bmatrix} h_1 & h_2 & \frac{h_3}{\sqrt{2}} & 0 & 0 & 0 \\ 0 & 0 & \frac{h_3}{\sqrt{2}} & h_2 & -h_1 & 0 \\ \frac{-h_3}{2} & \frac{h_3}{2} & 0 & \frac{-h_3}{2} & \frac{-h_3}{2} & \frac{(h_1+h_2)}{\sqrt{2}} \\ \frac{h_3}{2} & \frac{h_3}{2} & 0 & \frac{-h_3}{2} & \frac{h_3}{2} & \frac{(h_1-h_2)}{\sqrt{2}} \end{bmatrix}. \quad (3.17)$$

Example 3.2.2. A variant of the MPO334 STC is generated using Algorithms S1, S2, and S3 with $a = 3$. Within Algorithm S1, the first row of the base STC in (2.4) is rotated with a PR angle index of $x = 1$. Within Algorithm S2, the second column is rotated with a

PR angle index of $x = 2$. Within Algorithm S3, symbol s_1 is rotated with a PR angle index of $x = 3$. Thus, the resulting variant of the MPO334 STC is represented as

$$\mathbf{G} = \begin{bmatrix} -s_1 & s_2 \cdot e^{j3\pi/4} & \frac{s_3 \cdot e^{j\pi/4}}{\sqrt{2}} \\ -s_2^* & s_1^* \cdot e^{j7\pi/4} & \frac{s_3}{\sqrt{2}} \\ \frac{s_3^*}{\sqrt{2}} & \frac{s_3^* \cdot e^{j\pi/2}}{\sqrt{2}} & \frac{(s_1 \cdot e^{j7\pi/4} + s_1^* \cdot e^{j\pi/4} + s_2 - s_2^*)}{2} \\ \frac{s_3^*}{\sqrt{2}} & \frac{s_3^* \cdot e^{j3\pi/2}}{\sqrt{2}} & \frac{(s_2 + s_2^* + s_1 \cdot e^{j3\pi/4} + s_1^* \cdot e^{j\pi/4})}{2} \end{bmatrix}. \quad (3.18)$$

Given a single RX antenna with this STC and channel tap vector $\mathbf{h} = [h_1 \ h_2 \ h_3]^T$, the corresponding $\mathbf{H}_C$ matrix generated using Algorithm S8 is given as

$$\mathbf{H}_C = \begin{bmatrix} -h_1 & h_2 \cdot e^{j3\pi/4} & \frac{h_3 \cdot e^{j\pi/4}}{\sqrt{2}} & 0 & 0 & 0 \\ 0 & 0 & \frac{h_3}{\sqrt{2}} & h_2 \cdot e^{j7\pi/4} & -h_1 & 0 \\ \frac{h_3 \cdot e^{j7\pi/4}}{2} & \frac{h_3}{2} & 0 & \frac{h_3 \cdot e^{j\pi/4}}{2} & \frac{-h_3}{2} & \frac{(h_1 + h_2 \cdot e^{j\pi/2})}{\sqrt{2}} \\ \frac{h_3 \cdot e^{j3\pi/4}}{2} & \frac{h_3}{2} & 0 & \frac{h_3 \cdot e^{j\pi/4}}{2} & \frac{h_3}{2} & \frac{(h_1 + h_2 \cdot e^{j3\pi/2})}{\sqrt{2}} \end{bmatrix}. \quad (3.19)$$

Example 3.2.3. Given a single RX antenna, channel tap vector $\mathbf{h} = [h_1 \ h_2 \ h_3 \ h_4]^T$, and MPO468 base code originally given in (2.3) as

$$\mathbf{G} = \begin{bmatrix} s_1 & s_2 & s_3 & 0 \\ -s_2^* & s_1^* & 0 & s_6 \\ -s_3^* & 0 & s_1^* & -s_5 \\ 0 & -s_3^* & s_2^* & s_4 \\ s_4 & s_5 & s_6 & 0 \\ -s_5^* & s_4^* & 0 & s_3 \\ -s_6^* & 0 & s_4^* & -s_2 \\ 0 & -s_6^* & s_5^* & s_1 \end{bmatrix},$$

the corresponding $\mathbf{H}_C$ matrix generated using Algorithm S8 is given as

$$\mathbf{H}_C = \begin{bmatrix} h_1 & h_2 & h_3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & h_4 & h_2 & -h_1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -h_4 & 0 & h_3 & 0 & -h_1 & 0 & 0 & 0 \\ 0 & 0 & 0 & h_4 & 0 & 0 & 0 & h_3 & -h_2 & 0 & 0 & 0 \\ 0 & 0 & 0 & h_1 & h_2 & h_3 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & h_4 & 0 & 0 & 0 & 0 & 0 & 0 & h_2 & -h_1 & 0 \\ 0 & -h_4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & h_3 & 0 & -h_1 \\ h_4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & h_3 & -h_2 \end{bmatrix}. \quad (3.20)$$

3.2.4 Non-Orthogonal and Spatially-Multiplexed Space-Time Codes

Receivers of SM or non-orthogonal STCs typically employ sorted QR decomposition (SQRD) along with a successive interference cancellation (SIC) decoder and/or sphere decoding (SD) [60]–[64]. We use these decoding techniques and adapt them as necessary to work with an alternating STC.

To explain this adaptation, we begin with a MIMO system using spatial multiplexing with two TX antennas and two RX antennas. The SM221 STC is used as the base STC along with the set building algorithms with $a = 6$ to build a set of unique alternative STCs. Algorithm S8 is also used with this system to obtain an initial matrix for use with SQRD and follow-on SD techniques.

The received sample array, $\mathbf{Z}$, given in (3.8) still applies to these types of STCs. For this discussion, the next STC is represented as

$$\mathbf{G} = \begin{bmatrix} s_2 \cdot e^{j\pi/8} & s_1^* \cdot e^{j5\pi/32} \end{bmatrix}, \quad (3.21)$$

where it can be seen that multiple operations have been applied to the base code in (2.11).

To begin the decoding process, Algorithm S8 is repeated using CSI for each RX antenna with resulting $\mathbf{H}_C$ matrices concatenated vertically to obtain a final $\mathbf{H}_{CBU}$ matrix of size rN_R -by- $2k$. The channel tap vector input for the i th RX antenna is represented

as $\mathbf{h}_{BU}(:, i) = [h_{1i} \ \dots \ h_{ci}]^T$. Thus, performing these steps for the STC in (3.21) for the described MIMO system employing two RX antennas, the final $\mathbf{H}_{CBU}$ matrix is given as

$$\mathbf{H}_{CBU} = \begin{bmatrix} 0 & h_{11} \cdot e^{j\pi/8} & h_{21} \cdot e^{j5\pi/32} & 0 \\ 0 & h_{12} \cdot e^{j\pi/8} & h_{22} \cdot e^{j5\pi/32} & 0 \end{bmatrix}. \quad (3.22)$$

The remaining approach differs depending on whether or not the STC in use employs transmit diversity. We now present how the remaining portion of this decoding approach differs for these two different types of STCs. Block diagrams are shown in Fig. 3.1 to illustrate decoding steps for both types of STCs. The dashed arrows in each block diagram represent two optional paths of which only one must be taken.

Decoding STCs Without Transmit Diversity

We start by discussing the remaining decoding steps required for SMSTCs.

First, we form a restructured $\mathbf{H}_{CBU}$ matrix, denoted as $\mathbf{H}_{RC}$, while noting which symbols are conjugated. We then solve for the symbols, or conjugate symbols, using SQRD and SIC to obtain an estimate of the transmitted symbol vector. SD techniques may then be employed to obtain a near maximum-likelihood (ML) result. Any symbols noted as being conjugated initially must be conjugated again at the end of this process to solve for the original data symbol vector.

The process of forming $\mathbf{H}_{RC}$ while noting which symbols are conjugated starts by identifying the columns within the left-most k columns of $\mathbf{H}_{CBU}$ that are filled with zeros. These column indices represent symbols that are conjugated from the data symbol vector, $\mathbf{s}$, in order to create the transmitted symbol vector, denoted as $\mathbf{s}_T$. From the two left-most columns of (3.22), it can be seen that only the first column is filled with zeros indicating that s_1 is conjugated and s_2 is not. Upon noting these symbols, $\mathbf{H}_{RC}$ is formed by summing together the left and right halves of $\mathbf{H}_{CBU}$ as

$$\begin{aligned} \mathbf{H}_{RC} &= \mathbf{H}_{CBU}(:, 1 : k) + \mathbf{H}_{CBU}(:, k + 1 : 2k) \\ &= \begin{bmatrix} h_{21} \cdot e^{j5\pi/32} & h_{11} \cdot e^{j\pi/8} \\ h_{22} \cdot e^{j5\pi/32} & h_{12} \cdot e^{j\pi/8} \end{bmatrix}. \end{aligned} \quad (3.23)$$

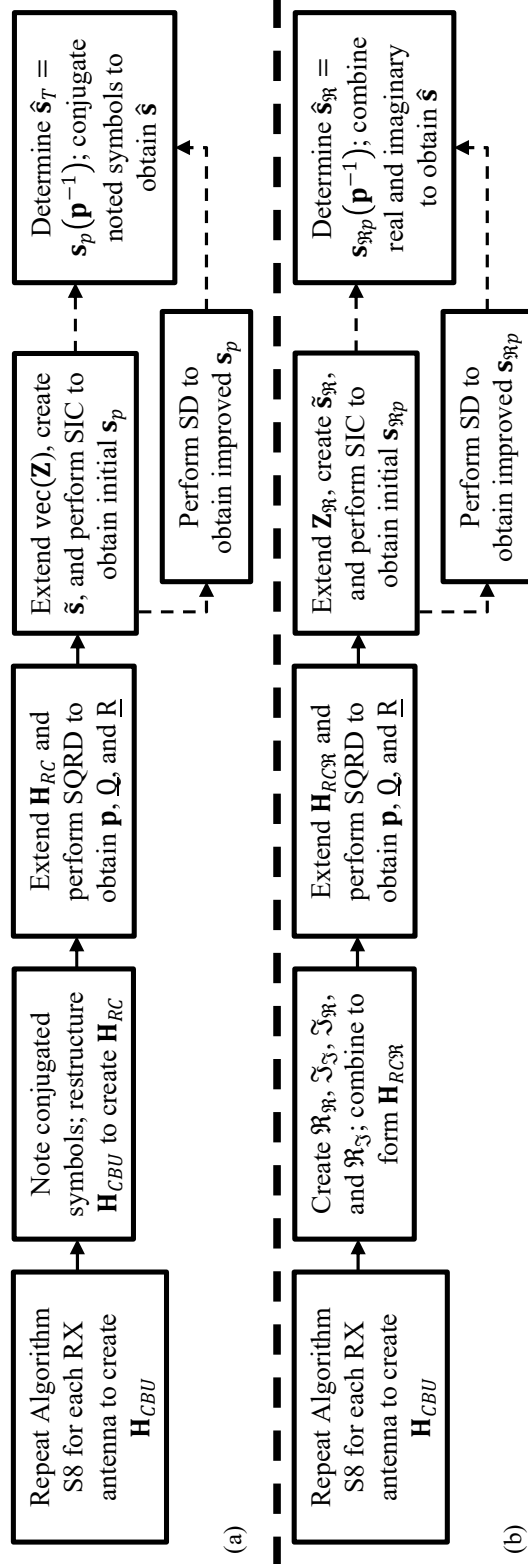


Figure 3.1. Block diagrams illustrating decoding steps for non-orthogonal and SM STCs. (a) STCs without transmit diversity. (b) STCs with transmit diversity.

The received sample array shown in (3.8) may now be rewritten as

$$\begin{aligned}\text{vec}(\mathbf{Z}) &= \mathbf{H}_{RC} \times \mathbf{s}_T + \text{vec}(\mathbf{n}) \\ &= \begin{bmatrix} h_{21} \cdot e^{j5\pi/32} & h_{11} \cdot e^{j\pi/8} \\ h_{22} \cdot e^{j5\pi/32} & h_{12} \cdot e^{j\pi/8} \end{bmatrix} \times \begin{bmatrix} s_1^* \\ s_2 \end{bmatrix} + \begin{bmatrix} n_1 \\ n_2 \end{bmatrix},\end{aligned}\quad (3.24)$$

where $\text{vec}(\cdot)$ represents the vec operator used to “vectorize” or vertically concatenate the columns of the argument to convert the argument into a column vector.

The minimum mean square error (MMSE)-SQRD and SIC processes given in [65] are now completed to obtain an estimate of $\mathbf{s}_T$, denoted as $\hat{\mathbf{s}}_T$. For clarity of discussion, portions of these processes from [65] are repeated here with symbols modified to stay consistent with this dissertation.

The MMSE-SQRD process begins by performing QR decomposition of the extended $\mathbf{H}_{RC}$ matrix,

$$\underline{\mathbf{H}} = \begin{bmatrix} \mathbf{H}_{RC} \\ \sigma_n \mathbf{I}_k \end{bmatrix}, \quad (3.25)$$

according to Tab. 1 of [65], where σ_n represents standard deviation of AWGN, and $\mathbf{I}_k$ is a k -by- k identity matrix. QR decomposition yields

$$\underline{\mathbf{H}}(:, \mathbf{p}) = \underline{\mathbf{Q}} \times \underline{\mathbf{R}}, \quad (3.26)$$

where $\mathbf{p}$ represents a 1-by- k permutation vector that indicates the sorted column order of $\underline{\mathbf{H}}$ obtained while performing the MMSE-SQRD algorithm allowing for a more optimal detection sequence [65]. The post-sorting algorithm shown in Tab. 2 of [65] may be subsequently performed to find the most optimal sequence, but from analysis provided in [65], it is generally not necessary and adds additional complexity. Thus, it is not further discussed here.

The next step is to use the determined $\underline{\mathbf{Q}}$ and $\underline{\mathbf{R}}$ matrices while applying SIC to obtain an initial estimate of $\mathbf{s}_p = \hat{\mathbf{s}}_T(\mathbf{p})$. Thus, $\mathbf{s}_p$ is a permuted version of the transmitted symbol vector estimate utilizing $\mathbf{p}$ for improved sequence of detection. To begin this step, the

left-hand side of (3.24) is extended with a k -by-1 vector of zeros, $\mathbf{0}_k$,

$$\underline{\mathbf{Z}} = \begin{bmatrix} \text{vec}(\mathbf{Z}) \\ \mathbf{0}_k \end{bmatrix}, \quad (3.27)$$

and multiplied by $\underline{\mathbf{Q}}^{-1} = \underline{\mathbf{Q}}^\dagger$ such that

$$\tilde{\mathbf{s}} = \underline{\mathbf{Q}}^\dagger \times \underline{\mathbf{Z}}. \quad (3.28)$$

SIC is then applied to obtain initial estimates of the elements of $\mathbf{s}_p$ in reverse order (i.e., $i = k, \dots, 1$) due to the upper triangular nature of $\underline{\mathbf{R}}$. This SIC process is shown as

$$\mathbf{s}_p(i) = \underset{\forall x \in \mathbb{M}}{\text{argmin}} d\left(\tilde{\mathbf{s}}(i), \underline{\mathbf{R}}_{ii} \cdot x + \sum_{j=i+1}^k \underline{\mathbf{R}}_{ij} \cdot \mathbf{s}_p(j)\right), \quad (3.29)$$

where $d(y, z)$ represents Euclidean distance between y and z , $\underline{\mathbf{R}}_{ij}$ represents the element in the i th row and j th column of $\underline{\mathbf{R}}$, and $\mathbb{M}$ is the set of all symbols in the M -ary modulation scheme [65], [66]. Although somewhat obvious, it should be noted in (3.29) that summation is not performed for $i = k$ as the summation bounds are invalid.

Upon obtaining this estimate of $\mathbf{s}_p$, it must be decided whether or not to continue with SD techniques such as in [61], [64] to obtain near ML results. For conciseness, we do not discuss these techniques further than to say that by using the initial estimate as the starting point for SD, these techniques may be used to yield a more accurate estimate of $\mathbf{s}_p$.

If BER performance loss at this stage is acceptable, and a reduced-complexity decoding process is desired, the initial estimate of $\mathbf{s}_p$ may be used during the next decoding step. The amount of performance loss is shown in Chapter 6. Regardless of whether initial or final estimate of $\mathbf{s}_p$ is employed, an estimate of the data symbol vector, $\hat{\mathbf{s}}$, may be obtained from $\mathbf{s}_p$. This is achieved by first finding the inverse permutation, denoted as $\mathbf{p}^{-1}$, of $\mathbf{p}$ such that $\mathbf{p}(\mathbf{p}^{-1}) = [1 \ \dots \ k]$. The inverse permutation of $\mathbf{s}_p$ is then performed to obtain an estimate of the transmitted symbol vector, $\hat{\mathbf{s}}_T = \mathbf{s}_p(\mathbf{p}^{-1})$. The symbols noted as being conjugated initially must now be conjugated again within $\hat{\mathbf{s}}_T$ to obtain $\hat{\mathbf{s}}$. Recall it was noted that s_1 is conjugated within $\mathbf{s}_T$ for the STC given in (3.21).

This completes the decoding process for SMSTCs. We now discuss the remaining steps involved to decode STCs with transmit diversity.

Decoding STCs With Transmit Diversity

If these same techniques are applied to STCs employing transmit diversity, additional steps are required to maximally combine the multiple copies of each data symbol.

The base STC for this section is the NO442 STC given in (2.10). Due to the transmit diversity employed by this STC, only two RX antennas are required; thus, $N_R = 2$.

The received sample array, $\mathbf{Z}$, for this system is still represented in (3.8) but with array dimensions according to this STC. For this discussion, the next STC is represented as

$$\mathbf{G} = \begin{bmatrix} s_4^* \cdot e^{j21\pi/16} & s_3^* \cdot e^{j15\pi/16} & s_2 \cdot e^{j49\pi/32} & s_1^* \cdot e^{j57\pi/32} \\ s_2^* \cdot e^{j3\pi/2} & s_1 \cdot e^{j39\pi/32} & s_4 \cdot e^{j23\pi/32} & s_3 \cdot e^{j17\pi/16} \end{bmatrix}. \quad (3.30)$$

As in the previous section, Algorithm S8 is repeated using CSI for each RX antenna with resulting $\mathbf{H}_C$ matrices concatenated vertically to obtain a final $\mathbf{H}_{CBU}$ matrix of size rN_R -by- $2k$. As before, the channel tap vector input for the i th RX antenna is represented as $\mathbf{h}_{BU}(:, i) = [h_{1i} \ \dots \ h_{ci}]^T$. Thus, performing these steps for the STC in (3.30) for the described MIMO system employing two RX antennas, the final $\mathbf{H}_{CBU}$ matrix is given as

$$\mathbf{H}_{CBU} = \begin{bmatrix} 0 & h_{21} \cdot e^{j39\pi/32} & 0 & h_{22} \cdot e^{j39\pi/32} \\ h_{31} \cdot e^{j49\pi/32} & 0 & h_{32} \cdot e^{j49\pi/32} & 0 \\ 0 & h_{41} \cdot e^{j17\pi/16} & 0 & h_{42} \cdot e^{j17\pi/16} \\ 0 & h_{31} \cdot e^{j23\pi/32} & 0 & h_{32} \cdot e^{j23\pi/32} \\ h_{41} \cdot e^{j57\pi/32} & 0 & h_{42} \cdot e^{j57\pi/32} & 0 \\ 0 & h_{11} \cdot e^{j3\pi/2} & 0 & h_{12} \cdot e^{j3\pi/2} \\ h_{21} \cdot e^{j15\pi/16} & 0 & h_{22} \cdot e^{j15\pi/16} & 0 \\ h_{11} \cdot e^{j21\pi/16} & 0 & h_{12} \cdot e^{j21\pi/16} & 0 \end{bmatrix}^T. \quad (3.31)$$

Note that (3.31) is given in transposed form to better fit on the page.

As in the previous section, the next step is to form the restructured $\mathbf{H}_{CBU}$ matrix; however,

to properly combine the multiple copies of each transmitted symbol for the STC given in (3.30), this restructured matrix must also be real-valued. This allows for real and imaginary components of each data symbol to be solved for individually. This process first generates

$$\mathbf{R}_{\mathfrak{R}} = \sum_{i=0}^1 \Re \left(\mathbf{H}_{CBU}(:, ik + 1 : ik + k) \right), \quad (3.32)$$

$$\mathfrak{I}_{\mathfrak{I}} = \sum_{i=0}^1 \Im \left(\mathbf{H}_{CBU}(:, ik + 1 : ik + k) \right) \cdot (-1)^{i+1}, \quad (3.33)$$

$$\mathfrak{I}_{\mathfrak{R}} = \sum_{i=0}^1 \Im \left(\mathbf{H}_{CBU}(:, ik + 1 : ik + k) \right), \text{ and} \quad (3.34)$$

$$\mathbf{R}_{\mathfrak{I}} = \sum_{i=0}^1 \Re \left(\mathbf{H}_{CBU}(:, ik + 1 : ik + k) \right) \cdot (-1)^i, \quad (3.35)$$

where $\Re(\cdot)$ represents the real component(s) of the argument and $\Im(\cdot)$ represents the imaginary component(s) of the argument. These four real-valued matrices are then combined to form the restructured, real-valued $\mathbf{H}_{RC\mathfrak{R}}$ matrix of size $2rN_R$ -by- $2k$ represented as

$$\mathbf{H}_{RC\mathfrak{R}} = \begin{bmatrix} \mathbf{R}_{\mathfrak{R}} & \mathfrak{I}_{\mathfrak{I}} \\ \mathfrak{I}_{\mathfrak{R}} & \mathbf{R}_{\mathfrak{I}} \end{bmatrix}. \quad (3.36)$$

The received sample array from (3.8) may now be split into real-valued components and rewritten as

$$\begin{aligned} \mathbf{Z}_{\mathfrak{R}} &= \mathbf{H}_{RC\mathfrak{R}} \times \mathbf{s}_{\mathfrak{R}} + \mathbf{n}_{\mathfrak{R}}, \text{ i.e.,} \\ \begin{bmatrix} \Re(\text{vec}(\mathbf{Z})) \\ \Im(\text{vec}(\mathbf{Z})) \end{bmatrix} &= \begin{bmatrix} \mathbf{R}_{\mathfrak{R}} & \mathfrak{I}_{\mathfrak{I}} \\ \mathfrak{I}_{\mathfrak{R}} & \mathbf{R}_{\mathfrak{I}} \end{bmatrix} \times \begin{bmatrix} \Re(\mathbf{s}) \\ \Im(\mathbf{s}) \end{bmatrix} + \begin{bmatrix} \Re(\text{vec}(\mathbf{n})) \\ \Im(\text{vec}(\mathbf{n})) \end{bmatrix}, \end{aligned} \quad (3.37)$$

where $\mathbf{s}_{\mathfrak{R}}$ is the real-valued data symbol vector such that $\mathbf{s} = \mathbf{s}_{\mathfrak{R}}(1 : k) + j\mathbf{s}_{\mathfrak{R}}(k + 1 : 2k)$ [66], [67]. The MMSE-SQRD and SIC processes given in [65] are now completed to obtain an estimate of $\mathbf{s}_{\mathfrak{R}}$, denoted as $\hat{\mathbf{s}}_{\mathfrak{R}}$.

The MMSE-SQRD process begins by performing QR decomposition of the extended $\mathbf{H}_{RC\mathfrak{R}}$ matrix,

$$\underline{\mathbf{H}} = \begin{bmatrix} \mathbf{H}_{RC\mathfrak{R}} \\ \frac{\sigma_n}{\sqrt{2}} \mathbf{I}_{2k} \end{bmatrix}, \quad (3.38)$$

according to Tab. 1 of [65], where $\mathbf{I}_{2k}$ is a $2k$ -by- $2k$ identity matrix. QR decomposition yields

$$\underline{\mathbf{H}}(:, \mathbf{p}) = \underline{\mathbf{Q}} \times \underline{\mathbf{R}}, \quad (3.39)$$

where $\mathbf{p}$ is a 1-by- $2k$ permutation vector in this instance [65].

The next step is to use the determined $\underline{\mathbf{Q}}$ and $\underline{\mathbf{R}}$ matrices while applying SIC to obtain an initial estimate of $\mathbf{s}_{\mathcal{X}_p} = \hat{\mathbf{s}}_{\mathcal{X}}(\mathbf{p})$. Thus, $\mathbf{s}_{\mathcal{X}_p}$ is a permuted version of the real-valued data symbol vector estimate utilizing $\mathbf{p}$ for improved sequence of detection. To begin this step, the left-hand side of (3.37) is extended with a $2k$ -by-1 vector of zeros, $\mathbf{0}_{2k}$,

$$\underline{\mathbf{Z}}_{\mathcal{X}} = \begin{bmatrix} \mathbf{Z}_{\mathcal{X}} \\ \mathbf{0}_{2k} \end{bmatrix}, \quad (3.40)$$

and multiplied by $\underline{\mathbf{Q}}^{-1} = \underline{\mathbf{Q}}^T$ such that

$$\tilde{\mathbf{s}}_{\mathcal{X}} = \underline{\mathbf{Q}}^T \times \underline{\mathbf{Z}}_{\mathcal{X}}. \quad (3.41)$$

SIC is then applied to obtain initial estimates of the elements of $\mathbf{s}_{\mathcal{X}_p}$ in reverse order (i.e., $i = 2k, \dots, 1$) due to the upper triangular nature of $\underline{\mathbf{R}}$. This SIC process is shown as

$$\mathbf{s}_{\mathcal{X}_p}(i) = \underset{\forall x \in \mathbb{M}_{\mathcal{X}}}{\operatorname{argmin}} d\left(\tilde{\mathbf{s}}_{\mathcal{X}}(i), \underline{\mathbf{R}}_{ii} \cdot x + \sum_{j=i+1}^{2k} \underline{\mathbf{R}}_{ij} \cdot \mathbf{s}_{\mathcal{X}_p}(j)\right), \quad (3.42)$$

where $\mathbb{M}_{\mathcal{X}}$ is the set of all unique real values for symbols in the M -ary modulation scheme [65], [66]. This assumes that the set of all unique imaginary values for these symbols is the same as $\mathbb{M}_{\mathcal{X}}$, which is true for all common digital modulation schemes including those specified in [56] for the 5G NR. As with (3.29), it should be noted in (3.42) that summation is not performed for $i = 2k$ as the summation bounds are invalid.

As in the previous section, it must be decided whether or not to continue with SD techniques to obtain a more accurate estimate of $\mathbf{s}_{\mathcal{X}_p}$. Similar to before, regardless of whether initial or final estimate is employed, an estimate of the data symbol vector, $\hat{\mathbf{s}}$, may be obtained from $\mathbf{s}_{\mathcal{X}_p}$. This is achieved by first performing the inverse permutation of $\mathbf{s}_{\mathcal{X}_p}$ to obtain an estimate of the real-valued data symbol vector, $\hat{\mathbf{s}}_{\mathcal{X}} = \mathbf{s}_{\mathcal{X}_p}(\mathbf{p}^{-1})$. The final estimate of

the data symbol vector is obtained by combining the real and imaginary components as $\hat{\mathbf{s}} = \hat{\mathbf{s}}_{\Re}(1 : k) + j\hat{\mathbf{s}}_{\Re}(k + 1 : 2k)$.

This completes the decoding process for non-orthogonal STCs employing transmit diversity.

3.3 Individual Mapping Building

Similar to individual STC building, it may be desirable to build specific selected mappings from the set, *aka* codebook, for use as needed without storing the entire codebook. Although this is likely unnecessary when b is small, for $b = 8$ as with 256QAM, the full set consists of $2^8 \cdot 8! = 10,321,920$ mappings each consisting of M byte-sized labels; thus, the 256QAM set of mappings requires 2,520 megabytes of storage. This may not be an issue for some systems; however, maintaining the entire set of mappings available for rapid access is likely problematic.

In order to build an individual mapping from the set, three pieces of information are needed:

1. the base mapping and
2. order of operations/algorithms used to build the mapping set and
3. the index value of the desired mapping.

As with STC sets, the base mapping must be known as it is the first indexed mapping in the set. Recall, the order of operations conducted must also be known as this reindexes the set. Again, the set is indexed from 1 to $|\mathbf{S}_m|$. The index value must be known as it specifies the iteration of each operation to be performed in order to build the selected mapping, where the iteration of an operation refers to the value of the it variable in each one of the algorithms provided.

With these pieces of information as with STC building, a specific mapping can be built by starting with the base mapping, determining the specified iteration of each operation to perform based on the given index value, and performing each operation in the given order. Recall, the process for determining the specified iteration of each operation starts by defining a 1-by- l divisor vector, $\mathbf{v}_{div}$, with elements calculated using (3.1).

Similar to calculation of (3.1) for STC building, $\mathbf{v}_{ol}$ is a row vector containing the operation limits for Algorithms M1 - M2 in that order, $\mathbf{v}_{oo}$ is an order of operations row vector

containing the number of each algorithm in the order in which they are performed to build a mapping, and l is the length of each row vector, which equals two when all of the provided algorithms are performed. Upon definition of row vector $\mathbf{v}_{div}$, the iteration of each operation can be contained within a 1-by- l iterations vector, $\mathbf{v}_{it}$, with elements given in (3.2), where IV represents the index value of the desired mapping.

Once the iterations have been determined, the desired mapping is built by performing Algorithm M0 first with the $|\mathbf{S}_m|$ input argument set to 1 since only 1 mapping will be built. Algorithms M1 - M2 are performed next in the order specified by $\mathbf{v}_{oo}$. To modify each of these algorithms for individual mapping building instead of set building, any reference to $\mathbf{S}_m$ is replaced with $\mathbf{B}_m$, the values of PC_i and PC_o are set to 1, the it variable is set equal to the value contained in $\mathbf{v}_{it}$ for that algorithm, and all index values of $PC_i \cdot it + mi$ are replaced with 1. For any element of $\mathbf{v}_{it}$ equal to zero, the corresponding algorithm is not performed.

Example 3.3.1. With a QPSK base map corresponding to BRGC mapping index 1 from Table 2.3, $\mathbf{v}_{oo} = [1 \ 2]$, $IV = 3$, and operation limits as specified in Algorithms M1 - M2, i.e.,

$$\begin{aligned} \mathbf{v}_{ol} &= \begin{bmatrix} 2^b & b! \end{bmatrix} \\ &= \begin{bmatrix} 4 & 2 \end{bmatrix}, \end{aligned} \tag{3.43}$$

$$\mathbf{v}_{div} = \begin{bmatrix} 1 \\ 4 \end{bmatrix}^T, \tag{3.44}$$

$$\mathbf{v}_{it} = \begin{bmatrix} 2 & 0 \end{bmatrix}, \tag{3.45}$$

and the built mapping is

$$\mathbf{B}_m = \begin{bmatrix} 0 & 1 \\ 0 & 0 \\ 1 & 1 \\ 1 & 0 \end{bmatrix} \tag{3.46}$$

corresponding to BRGC mapping index 3 in Table 2.3.

3.4 Gray Code Hopping Modulation and Demodulation

Unlike with alternating the STC in use, the entire mapping is not required to be known in order to perform GCH. Only the specific symbol within the chosen mapping corresponding to the next set of b bits must be known to perform bit to symbol translation. This allows for more flexibility in implementation of GCH. We now present two equivalent implementations of a GCH modulator and demodulator.

Fig. 3.2(a) shows the basic flow through one possible implementation that employs internal mapping functions or LUTs associated with each of the possible BRGC mappings. Fig. 3.2(b) shows the flow through an equivalent two-step method implementation. In each block diagram, the “Next IV Generator” is the element controlling the hopping sequence. In the next chapter, a proposed hopping sequence is presented, but for now, only the next IV must be known to determine the chosen mapping index and corresponding iteration of each operation to be performed. The order of operations for both implementations shown in Fig. 3.2 is $\mathbf{v}_{oo} = [1 \ 2]$.

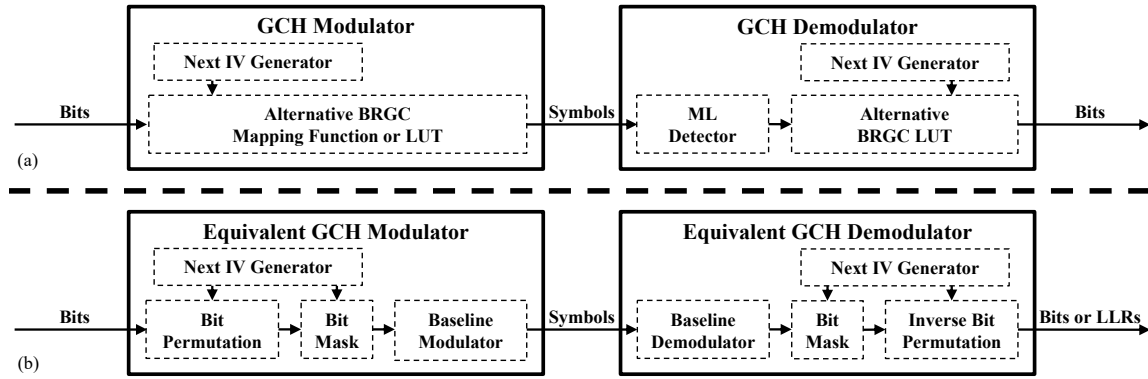


Figure 3.2. Example implementations of a GCH modulator and demodulator. (a) Internal mapping function or LUT implementation. (b) Equivalent two-step method implementation.

3.4.1 Bit to Symbol Translation

Once a mapping has been chosen for use with the next modulation symbol, BS must perform bit to symbol translation. One approach to accomplish this translation involves the use of a mapping function, such as (2.20), given in [56, Section 5.1] or establishment of the chosen

mapping as a LUT. This approach is represented in Fig. 3.2(a) and requires a mapping function or LUT for each alternative BRGC mapping. Whereas this method is valid and possibly well-suited for the case when b is small (e.g., $b < 8$), implementation for 256QAM becomes unduly complex due to the large number of alternative mappings.

An equivalent method of performing this translation involves performing the bit mask and permutation operations on the data bits themselves instead of the binary constellation labels as initially discussed. This equivalent method consists of two steps.

First, the bit mask and bit permutation operations associated with the chosen BRGC mapping are performed on the data bits in reverse order; thus, if the specified order of operations is $\mathbf{v}_{oo} = [1 \ 2]$, then the data bits are permuted and then masked accordingly as shown in Fig. 3.2(b). The associated bit mask and bit permutation operations are determined within Algorithms M1 and M2, according to $\mathbf{v}_{it}$, for the chosen BRGC mapping index, as discussed in Section 3.3.

Example 3.4.1. From Example 3.3.1 for the QPSK mapping set, $\mathbf{v}_{oo} = [1 \ 2]$, $IV = 3$, and $\mathbf{v}_{it} = [2 \ 0]$; thus, within Algorithm M1, $it = 2$, and the associated bit mask vector is $\mathbf{v}_{bm} = de2bi(2, 2, 2) = [0 \ 1]$. With $it = 0$ for Algorithm M2, no operation is required; thus, the original bit permutation of $[1 \ 2]$ is associated with this mapping.

Second, the permuted and masked data bits are translated to symbols using the baseline modulation mapping function or equivalent LUT.

Example 3.4.2. BRGC mapping index 1 in Table 2.3 represents a LUT implementation of the 5G NR modulation mapping function for QPSK shown in (2.20). For this example, BRGC mapping index 1 is selected as the baseline mapping with $\mathbf{v}_{oo} = [1 \ 2]$. The chosen mapping for the next symbol is BRGC mapping index 6. Given data bits of 10, employing Table 2.3 as a set of LUTs indicates that symbol s_4 should be produced as these data bits correspond to the s_4 label for that mapping. Equivalently, performing the bit permutation associated with BRGC mapping index 6 (i.e., $[2 \ 1]$) on the given data bits yields permuted bits of 01. Subsequently performing the bit mask operation associated with BRGC mapping index 6 (i.e., $[1 \ 0]$) on the permuted bits yields 11. From Table 2.3, employing BRGC mapping index 1 with permuted and masked data bits of 11 indicates that symbol s_4 is produced. Thus, both methods produce the same result.

3.4.2 Symbol to Bit Demodulation

Upon reception and downconversion to baseband symbols, UE must perform demodulation of the symbols to obtain an estimate of the transmitted data bits. UE could perform this function by first using a ML detector to determine the most likely transmitted constellation symbol. An alternative BRGC LUT is then used to translate the determined symbol to the corresponding bits based on the chosen alternative BRGC mapping as shown in Fig. 3.2(a).

Similar to the equivalent two-step method employed for bit to symbol translation, the reverse of this method may be used for symbol to bit demodulation. With this method, UE uses a baseline demodulator employing ML detection and demapping function with hard decision decoding to first obtain the permuted and masked data bits. For this system with $\mathbf{v}_{oo} = [1 \ 2]$, the appropriate bit mask for the chosen BRGC mapping index is then applied followed by the inverse of the bit permutation performed by BS to obtain an estimate of the transmitted data bits as shown in Fig. 3.2(b). Although inconsequential for the QPSK scheme with only two data bits, it is important to note the inverse bit permutation is required by UE for higher-order modulation schemes when using this equivalent method.

Example 3.4.3. In the case of 16QAM, if BS permutes the data bits according to permutation [2 3 1 4], UE must perform the inverse permutation [3 1 2 4].

Although hard decision decoding has been assumed until this point, soft decision decoding with the use of log-likelihood ratios (LLRs) for each bit is permitted with the equivalent demodulator. When soft decision decoding is performed by the baseline demodulator, the bit mask operation is replaced with LLR negation as this represents switching the probabilities of the bit being a 0 and a 1; thus, for each bit in the bit mask with a value of 1, the corresponding LLR is negated. The inverse bit permutation operation is maintained as it permutes the LLRs associated with the bits to be permuted. For this case, LLRs exit to the right of the equivalent GCH demodulator shown in Fig. 3.2(b) instead of bits.

THIS PAGE INTENTIONALLY LEFT BLANK

CHAPTER 4:

Proposed Physical Layer Security Framework

This chapter includes adaptations from previous work already published or in publication by the author. Specifically, portions of the introduction and Sections 4.1 and 4.2 are revised from [9].

As stated in Section 1.2, the primary objective for our research is to develop a tailorable PLS framework including optional techniques designed to alternate the STC and/or modulation mapping in use over a wireless communications link. Thus, we now present each of these optional techniques that may be implemented individually or concurrently as desired. As these schemes are predominantly located at the physical layer, details regarding medium access, flow, and error control are not discussed and are left as a topic of future research. Downlink transmission (e.g., BS to UE) is the sole focus of this research; however, both techniques could readily be extended to include uplink transmission.

4.1 Alternating Space-Time Coding

This technique involves three phases: steady state, initialization, and data transmission. The basic elements of each phase are illustrated in Fig. 4.1.

4.1.1 Steady State

During this phase, communication has not yet begun between the BS and UE; however, it is assumed that each entity has certain a priori knowledge. This includes an original base STC vector ($\mathbf{v}_{obs}$) that is a data-type agnostic, or categorical type, vector containing one or more of the base codes provided in Section 2.1.1. Operation limits for each base code as discussed in Sections 2.1.3 and 3.1 for a pre-determined value of a and the unique contents contained on UE's subscriber identity module (SIM) are also considered a priori knowledge. These unique contents include a base STC index vector ($\mathbf{v}_{bsi}$) and a base order of STC operations vector ($\mathbf{v}_{boso}$). The elements of $\mathbf{v}_{bsi}$ are indices for each base code in $\mathbf{v}_{obs}$. With this a priori information, both entities are able to build a unique base STC vector ($\mathbf{v}_{ubs}$) using the process detailed in Section 3.1. The elements in $\mathbf{v}_{ubs}$ represent the STCs built by applying

operations in the order specified by $\mathbf{v}_{bosso}$ to the base codes from $\mathbf{v}_{obs}$ with corresponding index values from $\mathbf{v}_{bsi}$.

Example 4.1.1. One possible system consists of a

$$\mathbf{v}_{obs} = \begin{bmatrix} \text{Alamouti} & \text{SPO334} & \text{MPO468} \end{bmatrix}, \quad (4.1)$$

where Alamouti, SPO334, and MPO468 are given in (2.1), (2.2), and (2.3), respectively. The UE SIM contains unique contents of

$$\mathbf{v}_{bsi} = \begin{bmatrix} 22, 619 & 8, 027, 223, 604 & 67, 554, 014, 461, 997, 720 \end{bmatrix}, \text{ and} \quad (4.2)$$

$$\mathbf{v}_{bosso} = \begin{bmatrix} 3 & 6 & 2 & 1 & 7 & 4 & 5 \end{bmatrix}. \quad (4.3)$$

Thus, with this information and corresponding operation limits for $a = 4$, the elements in $\mathbf{v}_{ubs}$ are

$$\mathbf{v}_{ubs}(1) = \begin{bmatrix} s_2 \cdot e^{j5\pi/8} & s_1^* \cdot e^{j15\pi/8} \\ -s_1 & s_2^* \cdot e^{j5\pi/4} \end{bmatrix}, \quad (4.4)$$

$$\mathbf{v}_{ubs}(2) = \begin{bmatrix} s_1^* \cdot e^{j7\pi/4} & s_2 \cdot e^{j3\pi/4} & 0 \\ s_3 \cdot e^{j\pi/4} & 0 & s_2 \cdot e^{j7\pi/8} \\ 0 & s_3 & s_1^* \cdot e^{j5\pi/8} \\ s_2^* \cdot e^{j9\pi/8} & s_1 \cdot e^{j9\pi/8} & s_3^* \cdot e^{j3\pi/4} \end{bmatrix}, \text{ and} \quad (4.5)$$

$$\mathbf{v}_{ubs}(3) = \begin{bmatrix} s_1^* \cdot e^{j3\pi/8} & s_2 \cdot e^{j9\pi/8} & s_3 \cdot e^{j7\pi/8} & 0 \\ s_2^* \cdot e^{j15\pi/8} & s_1 \cdot e^{j13\pi/8} & 0 & s_4 \cdot e^{j15\pi/8} \\ s_3^* \cdot e^{j13\pi/8} & 0 & s_1 \cdot e^{j9\pi/8} & s_6 \cdot e^{j3\pi/2} \\ 0 & s_3^* \cdot e^{j7\pi/4} & s_2^* \cdot e^{j\pi/2} & s_5 \\ s_5 \cdot e^{j\pi/8} & s_6 \cdot e^{j3\pi/4} & s_4 \cdot e^{j13\pi/8} & 0 \\ s_6^* \cdot e^{j9\pi/8} & s_5^* \cdot e^{j3\pi/4} & 0 & s_3 \\ s_4^* \cdot e^{j9\pi/8} & 0 & s_5^* \cdot e^{j13\pi/8} & s_2 \cdot e^{j\pi/8} \\ 0 & s_4^* \cdot e^{j7\pi/4} & s_6^* \cdot e^{j13\pi/8} & -s_1^* \end{bmatrix}. \quad (4.6)$$

The SIM also includes a STC seed obfuscation key (SOK_s) used during the next phase to obfuscate the received seed to minimize any impact of the seed being intercepted. For a given $\mathbf{v}_{obs}$, the number of unique combinations of $\mathbf{v}_{bsi}$, $\mathbf{v}_{bosso}$, and SOK_s with length of

$|\text{SOK}_s|$ bits is used to determine the number of unique SIMs which is

$$2^{|\text{SOK}_s|} \cdot |\mathbf{v}_{bosso}|! \cdot \prod_{i=1}^{|\mathbf{v}_{obs}|} |\mathbf{S}_s(i)|, \quad (4.7)$$

where $|\mathbf{S}_s(i)|$ is the set cardinality for the i th base code in $\mathbf{v}_{obs}$.

4.1.2 Initialization

During this phase, it is desired for communication to begin. A random seed is transmitted from BS to UE using a secret sharing method such as the post-quantum cross-layer key agreement scheme presented in [2]. Each entity performs modulo-2 addition, denoted by the $\oplus$ symbol in Fig. 4.1, between the bits of this random seed and the SOK_s to obtain an obfuscated seed. Each entity then uses this seed to synchronize PRNGs. Next, pseudorandom numbers (PRNs) are generated to create a temporary $\mathbf{v}_{bsi}$ ($\mathbf{v}_{tbsi}$) and temporary order of STC operations vector ($\mathbf{v}_{toso}$) for use during this session. The elements of $\mathbf{v}_{tbsi}$ are set as

$$\mathbf{v}_{tbsi}(i) = (\text{PRN} \bmod |\mathbf{S}_s(i)|) + 1. \quad (4.8)$$

The $\mathbf{v}_{toso}$ is set by indexing a row from a permutations array containing the elements from $\mathbf{v}_{bosso}$ in lexicographic order, where the row index is set as

$$\text{row index} = (\text{PRN} \bmod 7!) + 1. \quad (4.9)$$

Example 4.1.2. Set cardinalities for the base codes in (4.1), provided in Table 2.2, for $a = 4$ are used in (4.8) with consecutive PRNs of 7, 338, 378, 580, 900, 475; 8, 158, 648, 460, 577, 917; and 1, 143, 795, 557, 080, 799 to obtain

$$\mathbf{v}_{tbsi} = \begin{bmatrix} 14,972 & 12,336,958,590 & 1,143,795,557,080,800 \end{bmatrix}. \quad (4.10)$$

With $\mathbf{v}_{bosso}$ given in (4.3) and $\text{PRN} = 278,570,369,041,398$, the row index = 1639 using (4.9); therefore, $\mathbf{v}_{toso} = [2 \ 6 \ 4 \ 1 \ 3 \ 7 \ 5]$ as shown in Fig. 4.2.

With these session variables, $\mathbf{v}_{ubs}$, and operation limits, both entities build a temporary base STC vector ($\mathbf{v}_{tbs}$). The BS and UE are now ready to begin data transmission.

row indices	
1	3 6 2 1 7 4 5
2	3 6 2 1 7 5 4
$\vdots$	$\vdots \vdots \vdots \vdots \vdots \vdots \vdots$
1638	2 6 4 3 5 7 1
1639	2 6 4 1 3 7 5
1640	2 6 4 1 3 5 7
$\vdots$	$\vdots \vdots \vdots \vdots \vdots \vdots \vdots$
5039	5 4 7 1 2 3 6
5040	5 4 7 1 2 6 3

Figure 4.2. Illustration of indexing into permutations array containing elements of $\mathbf{v}_{bos\phi}$ in lexicographic order with row index = 1639.

Example 4.1.3. Using $\mathbf{v}_{ubs}$ from Example 4.1.1 and the determined session variables from Example 4.1.2, the elements in $\mathbf{v}_{tbs}$ are

$$\mathbf{v}_{tbs}(1) = \begin{bmatrix} s_1^* \cdot e^{j3\pi/4} & s_2^* \cdot e^{j13\pi/8} \\ s_2 \cdot e^{j13\pi/8} & s_1 \cdot e^{j3\pi/2} \end{bmatrix}, \quad (4.11)$$

$$\mathbf{v}_{tbs}(2) = \begin{bmatrix} s_3^* \cdot e^{j7\pi/8} & s_1^* \cdot e^{j5\pi/8} & 0 \\ s_2^* \cdot e^{j7\pi/8} & 0 & s_1^* \cdot e^{j5\pi/8} \\ 0 & s_2^* \cdot e^{j5\pi/4} & s_3^* \cdot e^{j\pi/4} \\ s_1 \cdot e^{j5\pi/4} & s_3 & s_2 \end{bmatrix}, \text{ and} \quad (4.12)$$

$$\mathbf{v}_{tbs}(3) = \begin{bmatrix} s_5^* \cdot e^{j13\pi/8} & 0 & s_4^* & s_2 \cdot e^{j3\pi/4} \\ 0 & s_5^* \cdot e^{j\pi/2} & s_6^* \cdot e^{j\pi/2} & s_1^* \cdot e^{j\pi/8} \\ s_2^* \cdot e^{j9\pi/8} & s_1 \cdot e^{j5\pi/8} & 0 & s_5 \cdot e^{j5\pi/4} \\ s_3^* \cdot e^{j3\pi/2} & 0 & s_1 \cdot e^{j7\pi/8} & s_6 \cdot e^{j3\pi/2} \\ s_1^* \cdot e^{j9\pi/8} & s_2 \cdot e^{j13\pi/8} & s_3 \cdot e^{j3\pi/2} & 0 \\ 0 & s_3^* \cdot e^{j11\pi/8} & s_2^* \cdot e^{j\pi/4} & s_4 \\ -s_6^* & s_4^* \cdot e^{j3\pi/8} & 0 & s_3 \\ s_4 & s_6 \cdot e^{j3\pi/8} & s_5 \cdot e^{j11\pi/8} & 0 \end{bmatrix}. \quad (4.13)$$

4.1.3 Data Transmission

During this phase, the synchronized PRNGs from the initialization phase are used to generate PRNs in order to determine the next STC to be used during transmission from BS to UE. If there is more than one base code in $\mathbf{v}_{obs}$, then the code set must be determined first. The next STC set (NSS) is calculated as

$$NSS = (\text{PRN} \bmod |\mathbf{v}_{obs}|) + 1. \quad (4.14)$$

From the set of STCs built using the temporary base code specified by NSS, the next STC index (NSI) is calculated as

$$NSI = (\text{PRN} \bmod |\mathbf{S}_s(NSS)|) + 1. \quad (4.15)$$

Using these set and index values, $\mathbf{v}_{toso}$, $\mathbf{v}_{tbs}$, and operation limits, both entities build the next STC using the process detailed in Section 3.1.

Example 4.1.4. For the system in Examples 4.1.1, 4.1.2, and 4.1.3 with $\text{PRN} = 3,841,906,826,998,665$, (4.14) is used to calculate $NSS = 1$; thus, $\mathbf{v}_{tbs}(1)$ is the base code to build the next STC. With the next $\text{PRN} = 3,177,477,576,842,494$, (4.15) is used to calculate $NSI = 23,807$. Thus, the next STC is

$$\mathbf{G} = \begin{bmatrix} s_1 \cdot e^{j3\pi/8} & s_2 \cdot e^{j\pi/4} \\ s_2^* \cdot e^{j11\pi/8} & s_1^* \cdot e^{j\pi/4} \end{bmatrix}. \quad (4.16)$$

As each STC consists of k data symbols, the steps described here for determining the NSS, NSI, and building the next STC are repeated for each set of k data symbols to be transmitted. The value of k is also variable depending on the base codes contained within $\mathbf{v}_{obs}$. UE performs the decoding steps described in Section 3.2 for each STC as applicable.

4.1.4 Pseudorandom Number Size Considerations

An assumption made with this technique thus far is that PRNs are capable of being generated at least as large as the largest set cardinality of any STC in $\mathbf{v}_{obs}$ for the pre-determined value of a ; however, from Table 2.2 it can be seen that set cardinalities are capable of being very large. For instance, MPO468 with $a = 9$ produces a set of $1.54 \cdot 10^{44}$ unique STCs; thus,

PRNs would be required to be at least 147 bits.

In situations where set cardinalities exceed the length of the PRNG, multiple PRNs may be generated to create an alternative iterations vector ($\mathbf{v}_{ait}$) to be used in place of $\mathbf{v}_{it}$ when building individual STCs using the Section 3.1 process. The $\mathbf{v}_{ait}$ is a 1-by- l vector similar to $\mathbf{v}_{it}$ in (3.2) containing the iteration of each operation to be performed with elements

$$\mathbf{v}_{ait}(i) = (\text{PRN mod } \mathbf{v}_{ol}(\mathbf{v}_{oo}(i))) + 1, \quad i \in \{1, \dots, l\}, \quad (4.17)$$

where l is the length of each row vector.

For the initialization phase when this is the case, $\mathbf{v}_{ait}$ is determined for each code in $\mathbf{v}_{ubs}$ by using $\mathbf{v}_{boso}$ in place of $\mathbf{v}_{oo}$. Then the elements of $\mathbf{v}_{tbsi}$ are set using (3.3) by replacing $\mathbf{v}_{it}$ with $\mathbf{v}_{ait}$ as

$$\mathbf{v}_{tbsi}(i) = \mathbf{v}_{ait} \times \mathbf{v}_{div}^T + 1. \quad (4.18)$$

For the data transmission phase when this is the case, $\mathbf{v}_{ait}$ is determined in place of the NSI and subsequent $\mathbf{v}_{it}$ by using $\mathbf{v}_{toso}$ in place of $\mathbf{v}_{oo}$.

The workaround provided in this section still relies on the assumption that PRNs are capable of being generated at least as large as the largest operation limit contained in $\mathbf{v}_{ol}$; however, for MPO468 with $a = 9$, the row PRs operation limit is $2^{ar} = 2^{72}$. Thus, in situations where operation limits exceed the length of the PRNG, it is assumed that multiple PRNs are generated to overcome this limitation. Details involved with this particular workaround are left as an area of future research.

4.1.5 Antenna Mapping

When multiple STCs are contained within $\mathbf{v}_{obs}$ with different number of TX antenna requirements, a decision must be made as to which TX antennas to use when the next STC to be employed does not require all TX antennas. Whereas a new vector could be defined for this purpose and added as a SIM unique content, it is chosen at this time to simply use another PRN to select the next antenna permutation. This method is chosen to provide load balancing across TX antenna radio frequency (RF) chains.

4.1.6 Base Code Permutations

When multiple STCs are contained within $\mathbf{v}_{obs}$, a permutation could be applied to the STCs within the $\mathbf{v}_{tbs}$. This could increase the number of possible SIMs by allowing for a base code order vector and add additional complexity to the exhaustive key search to be discussed in Section 5.2.3. For now, this is left as an area of future research.

4.2 Gray Code Hopping

This technique involves the same three phases as the previous scheme. The basic elements of each phase are illustrated in Fig. 4.3. The GCH modulator and demodulator shown in Fig. 4.3 are the same as those shown in Fig. 3.2. Either implementation may be utilized.

4.2.1 Steady State

During this phase, communication has not yet begun between the BS and UE; however, it is assumed that each entity has certain a priori knowledge. This includes an original base mapping vector ($\mathbf{v}_{obm}$) containing baseline mappings as discussed in Section 2.2.1, operation limits for each mapping as discussed in Sections 2.2.3 and 3.3, and the unique contents contained on the UE's SIM. These unique contents include a base mapping index vector ($\mathbf{v}_{bmi}$) and a base order of mapping operations vector ($\mathbf{v}_{bomo}$). The elements of $\mathbf{v}_{bmi}$ are indices for each mapping in $\mathbf{v}_{obm}$. With this a priori information, both entities are able to build a unique base mapping vector ($\mathbf{v}_{ubm}$) using the process detailed in Section 3.3. The elements in $\mathbf{v}_{ubm}$ represent the mappings built by applying operations in the order specified by $\mathbf{v}_{bomo}$ to the mappings from $\mathbf{v}_{obm}$ with corresponding index values from $\mathbf{v}_{bmi}$.

Example 4.2.1. One possible system consists of a

$$\mathbf{v}_{obm} = \left[\text{QPSK} \right], \quad (4.19)$$

where QPSK corresponds to BRGC mapping index 1 in Table 2.3. The UE SIM contains unique contents of

$$\mathbf{v}_{bmi} = \left[7 \right], \text{ and} \quad (4.20)$$

$$\mathbf{v}_{bomo} = \left[\begin{matrix} 2 & 1 \end{matrix} \right]. \quad (4.21)$$

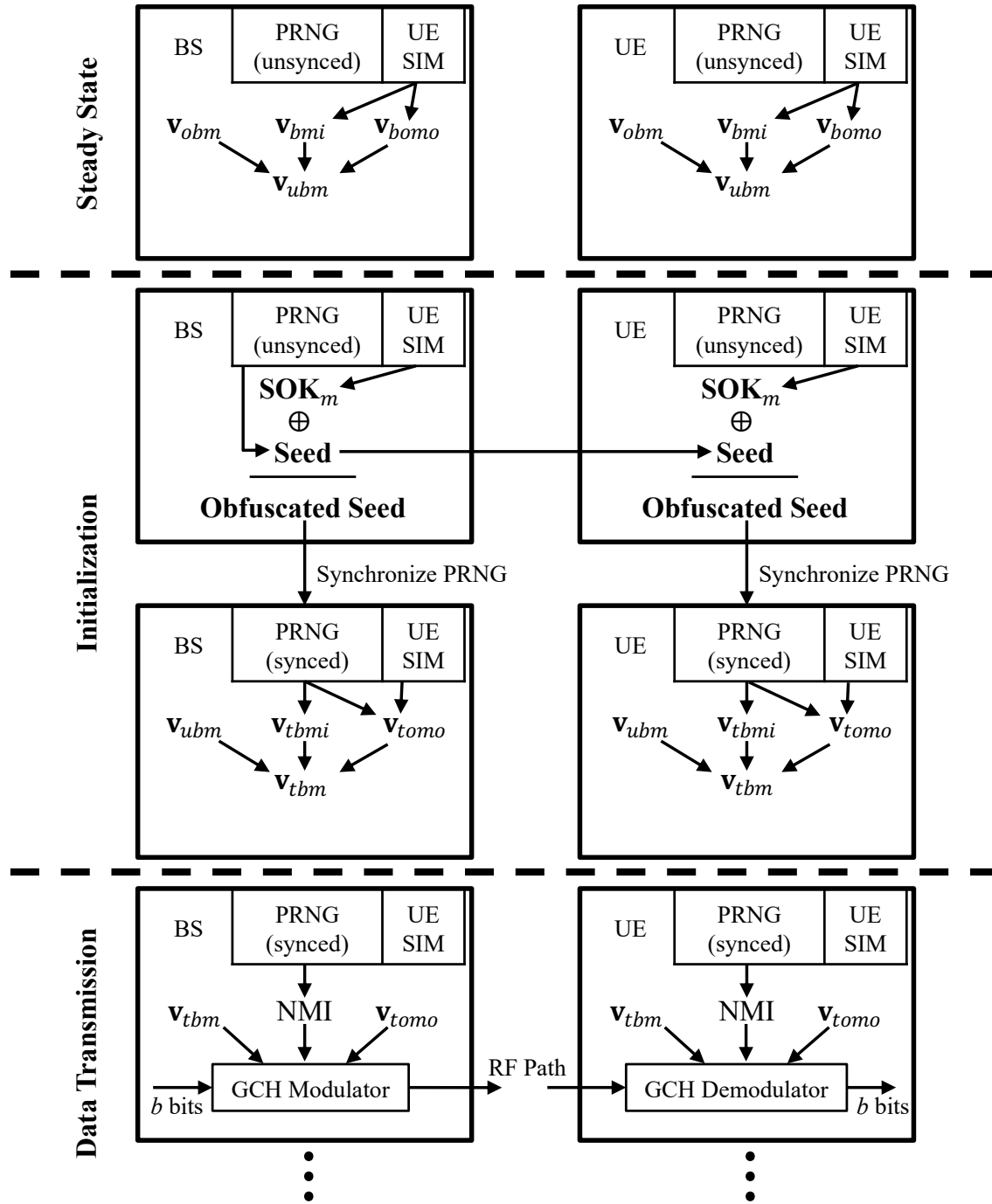


Figure 4.3. Basic elements for each phase of the GCH PLS scheme.

With this information and corresponding operation limits for $b = 2$, (3.1) is calculated using $\mathbf{v}_{bomo}$ in place of $\mathbf{v}_{oo}$ to obtain $\mathbf{v}_{div} = [1 \ 2]$, and (3.2) is calculated using $\mathbf{v}_{bomo}$ in place of $\mathbf{v}_{oo}$ and $\mathbf{v}_{bmi}$ in place of IV to obtain $\mathbf{v}_{it} = [0 \ 3]$. Thus, the associated bit permutation is $[1 \ 2]$, and associated bit mask vector is $\mathbf{v}_{bm} = de2bi(3, 2, 2) = [1 \ 1]$. This results in

$$\mathbf{v}_{ubm}(1) = \begin{bmatrix} 1 & 1 \\ 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix}. \quad (4.22)$$

Note that $\mathbf{v}_{ubm}(1)$ is not equal to BRGC mapping index 7 from Table 2.3 as the two mappings are constructed with different $\mathbf{v}_{oo}$ despite having the same base mapping and index value.

The SIM also includes a mapping seed obfuscation key (SOK_m) used during the next phase to obfuscate the received seed to minimize any impact of the seed being intercepted. For a given $\mathbf{v}_{obm}$, the number of unique combinations of $\mathbf{v}_{bmi}$, $\mathbf{v}_{bomo}$, and SOK_m with length of $|\text{SOK}_m|$ bits is used to determine the number of unique SIMs which is

$$2^{|\text{SOK}_m|} \cdot |\mathbf{v}_{bomo}|! \cdot \prod_{i=1}^{|\mathbf{v}_{obm}|} |\mathbf{S}_m(i)|, \quad (4.23)$$

where $|\mathbf{S}_m(i)|$ is the set cardinality for the i th base mapping in $\mathbf{v}_{obm}$.

4.2.2 Initialization

During this phase, it is desired for communication to begin. A random seed is transmitted from BS to UE using a secret sharing method. Each entity performs modulo-2 addition between the bits of this random seed and the SOK_m to obtain an obfuscated seed. Each entity then uses this seed to synchronize PRNGs. Next, PRNs are generated to create a temporary $\mathbf{v}_{bmi}$ ($\mathbf{v}_{tbmi}$) and temporary order of mapping operations vector ($\mathbf{v}_{tomo}$) for use during this session. The elements of $\mathbf{v}_{tbmi}$ are set as

$$\mathbf{v}_{tbmi}(i) = (\text{PRN} \bmod |\mathbf{S}_m(i)|) + 1. \quad (4.24)$$

The $\mathbf{v}_{tomo}$ is set by indexing a row from a permutations array containing the elements from $\mathbf{v}_{bomo}$ in lexicographic order, where the row index is set as

$$\text{row index} = (\text{PRN} \bmod 2!) + 1. \quad (4.25)$$

Example 4.2.2. Set cardinality for the base mapping in (4.19), provided in Table 2.4, is used in (4.24) with $\text{PRN} = 8, 226, 958, 330, 713, 789$ to obtain

$$\mathbf{v}_{tomi} = \begin{bmatrix} 6 \end{bmatrix}. \quad (4.26)$$

With $\mathbf{v}_{bomo}$ given in (4.21) and $\text{PRN} = 5, 695, 785, 731, 330, 084$, the row index = 1 using (4.25); therefore, $\mathbf{v}_{tomo} = \begin{bmatrix} 2 & 1 \end{bmatrix}$ as shown in Fig. 4.4.

row indices		
1	2	1
2	1	2

Figure 4.4. Illustration of indexing into permutations array containing elements of $\mathbf{v}_{bomo}$ in lexicographic order with row index = 1.

With these session variables, $\mathbf{v}_{ubm}$, and operation limits, both entities build a temporary base mapping vector ($\mathbf{v}_{tbi}$). The BS and UE are now ready to begin data transmission.

Example 4.2.3. Using $\mathbf{v}_{ubm}$ from Example 4.2.1 and the determined session variables from Example 4.2.2, $\mathbf{v}_{div} = \begin{bmatrix} 1 & 2 \end{bmatrix}$ and $\mathbf{v}_{it} = \begin{bmatrix} 1 & 2 \end{bmatrix}$; thus, the associated bit permutation is $\begin{bmatrix} 2 & 1 \end{bmatrix}$, and associated bit mask vector is $\mathbf{v}_{bm} = de2bi(2, 2, 2) = \begin{bmatrix} 0 & 1 \end{bmatrix}$. This results in

$$\mathbf{v}_{tbi}(1) = \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 1 & 1 \\ 0 & 1 \end{bmatrix}. \quad (4.27)$$

4.2.3 Data Transmission

During this phase, the synchronized PRNGs from the initialization phase are used to generate PRNs in order to determine the next mapping to be used to produce the next modulation symbol during transmission from BS to UE. If there is more than one base mapping in $\mathbf{v}_{obm}$, then it is assumed that the next mapping set (NMS) to be used is determined by some other means such as the RSSI. This is based on the assumption that the particular modulation order to be employed is selected as part of a common adaptive modulation coding scheme.

From the determined NMS, the next mapping index (NMI) is calculated as

$$\text{NMI} = (\text{PRN} \bmod |\mathbf{S}_m(\text{NMS})|) + 1. \quad (4.28)$$

Using these set and index values, $\mathbf{v}_{tomo}$, $\mathbf{v}_{tbm}$, and operation limits, BS performs GCH modulation as discussed in Section 3.4.

Example 4.2.4. For the system in Examples 4.2.1, 4.2.2, and 4.2.3, $\text{NMS} = 1$; thus, $\mathbf{v}_{tbm}(1)$ is the base mapping to be employed. With the next $\text{PRN} = 878, 565, 863, 217, 818$, (4.28) is used to calculate $\text{NMI} = 3$. Thus, with $\mathbf{v}_{div} = [1 \ 2]$, $\mathbf{v}_{it} = [0 \ 1]$, the associated bit permutation is $[1 \ 2]$, and associated bit mask vector is $\mathbf{v}_{bm} = \text{de2bi}(1, 2, 2) = [1 \ 0]$. The next mapping to be used is

$$\begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 1 \\ 1 & 1 \end{bmatrix}. \quad (4.29)$$

The steps described here are repeated for each symbol to be transmitted. If the GCH modulator given in Fig. 3.2(a) is employed, the next mapping function or LUT is selected by NMI from a set of mappings built using $\mathbf{v}_{tbm}$ and $\mathbf{v}_{tomo}$. If the GCH modulator given in Fig. 3.2(b) is employed, NMI is used in conjunction with $\mathbf{v}_{tomo}$ to determine the order of operations, associated bit permutation, and bit mask, and the baseline modulator corresponds to $\mathbf{v}_{tbm}$. UE performs GCH demodulation of each symbol as discussed in Section 3.4.

4.3 Combining Techniques

When combining alternating STC and GCH PLS techniques, it must be determined whether or not they will use the same PRNG. For the purposes of security to be discussed in Chapter 5, it is better if the two techniques employ separate PRNGs with different seeds. Additionally, it is recommended that $\text{SOK}_s \neq \text{SOK}_m$ to increase the number of unique SIMs.

For given $\mathbf{v}_{obs}$ and $\mathbf{v}_{obm}$ when combining techniques, the SIM unique contents include $\mathbf{v}_{bsi}$, $\mathbf{v}_{bosso}$, SOK_s , $\mathbf{v}_{bmi}$, $\mathbf{v}_{bomo}$, and SOK_m . Thus, the number of unique SIMs is

$$2^{(|\text{SOK}_s|+|\text{SOK}_m|)} \cdot |\mathbf{v}_{bosso}|! \cdot |\mathbf{v}_{bomo}|! \cdot \prod_{i=1}^{|\mathbf{v}_{obs}|} |\mathbf{S}_s(i)| \cdot \prod_{i=1}^{|\mathbf{v}_{obm}|} |\mathbf{S}_m(i)|. \quad (4.30)$$

THIS PAGE INTENTIONALLY LEFT BLANK

CHAPTER 5:

Security Analysis

This chapter includes adaptations from previous work already published or in publication by the author. Specifically, portions of Section 5.1.4 and 5.2.2 are revised from [8]. Portions of the introduction and Sections 5.1, 5.2, and 5.3 are revised from [9]. Portions of Sections 5.1 and 5.2 are revised from [10]. Portions of Section 5.3 are revised from [11].

Before analyzing the security of the proposed PLS framework, we first define some common nomenclature for use in characterization of ST coding-based PLS schemes. We then discuss security offered by the proposed alternating STC technique against two possible passive eavesdropper attack models. Next, we take a look at the security offered by the proposed GCH scheme, and finally, we discuss the security benefit of operating the two techniques in unison.

5.1 Security System Nomenclature

In [13], definitions were provided for a cryptogram, message residue class, and cryptogram residue class pertaining to secrecy systems in general. These terms are now related to ST coding-based PLS schemes. Additionally, the notion and definitions for a key residue (KR) and key residue class (KRC) are provided.

5.1.1 Cryptogram Definition

The cryptogram, $\mathbf{E}$, is defined as the transmitted STC matrix, $\mathbf{G}$, with embedded data symbol vector, $\mathbf{s}$, containing k specific data symbols of the chosen digital modulation scheme.

Example 5.1.1. Given the MPO468 base code shown in (2.3) employing 16QAM with $\mathbf{s} = [+1 + 1j, -3 + 3j, -3 - 1j, -1 - 1j, +3 - 3j, +3 + 1j]^T$, the cryptogram is

represented as

$$\mathbf{E} = \begin{bmatrix} +1+1j & -3+3j & -3-1j & 0 \\ +3+3j & +1-1j & 0 & +3+1j \\ +3-1j & 0 & +1-1j & -3+3j \\ 0 & +3-1j & -3-3j & -1-1j \\ -1-1j & +3-3j & +3+1j & 0 \\ -3-3j & -1+1j & 0 & -3-1j \\ -3+1j & 0 & -1+1j & +3-3j \\ 0 & -3+1j & +3+3j & +1+1j \end{bmatrix}. \quad (5.1)$$

By incorporating PR algorithms with $a \geq 3$, cryptogram elements are not guaranteed to be $\in \mathbb{M}$ of the chosen M -ary modulation scheme. This is also true for the MPO334 and MPO434 STCs even without PR algorithms due to weights applied to matrix elements.

Example 5.1.2. A variant of the Alamouti STC built using PR algorithms and $a = 3$ is represented as

$$\mathbf{G} = \begin{bmatrix} s_1 \cdot e^{j3\pi/4} & -s_2^* \\ s_2 \cdot e^{j\pi/4} & s_1^* \cdot e^{j3\pi/2} \end{bmatrix}. \quad (5.2)$$

Employing 16QAM with $\mathbf{s} = [+1+1j, -3-1j]^T$, the cryptogram is represented as

$$\mathbf{E} = \begin{bmatrix} -\sqrt{2} & +3-1j \\ -\sqrt{2}-2\sqrt{2}j & -1-1j \end{bmatrix}. \quad (5.3)$$

Example 5.1.3. Given the MPO334 variant code shown in (3.18) employing 16QAM with $\mathbf{s} = [+1+1j, -3+3j, -3-1j]^T$, the cryptogram is represented as

$$\mathbf{E} = \begin{bmatrix} -1-1j & -3\sqrt{2}j & -1-2j \\ +3+3j & -\sqrt{2}j & \frac{(-3-1j)}{\sqrt{2}} \\ \frac{(-3+1j)}{\sqrt{2}} & \frac{(-1-3j)}{\sqrt{2}} & \sqrt{2}+3j \\ \frac{(-3+1j)}{\sqrt{2}} & \frac{(+1+3j)}{\sqrt{2}} & -3 \end{bmatrix}. \quad (5.4)$$

5.1.2 Key Residue Definition

We herein define the KR as the pattern or set of symbol relationships and their matrix locations within a given STC. For the KR, the particular symbol (e.g., s_1 versus s_2) in each relationship is irrelevant. The pattern is defined only by the relationship and location of each symbol element.

We define symbol relationships first as those which exist between the two variants in any present symbol pair. When both variants are similarly-conjugated, the relationship is determined by dividing the second variant by the first. When the variants are dissimilarly-conjugated, the relationship is determined by dividing the second variant by the conjugate of the entire first variant, including any PR that exists. Note that this is in contrast to the symbol conjugation algorithm provided in Section 2.1.3 where the conjugation operation does not get applied to any PRs.

Example 5.1.4. If the first variant is $s_1 \cdot e^{j\pi/4}$, and the second variant is $s_1^* \cdot e^{j\pi/2}/2$, the variants are dissimilarly-conjugated. Thus, the relationship is determined by performing

$$\frac{s_1^* \cdot e^{j\pi/2}}{2} \bigg/ (s_1 \cdot e^{j\pi/4}) = \frac{e^{j3\pi/4}}{2}. \quad (5.5)$$

The symbol relationship between these two variants is referred to as conjugate $e^{j3\pi/4}/2$.

If all cryptogram elements are guaranteed to be $\in \mathbb{M}$ of the chosen M -ary modulation scheme, symbol relationships could be limited to only those which exist between symbols; however, as this guarantee does not always hold, symbol relationships must be further defined to include the PRs, with applicable weighting factors, applied to data symbols within the reference column(s) of the STC matrix. These PRs are recorded as minimal equivalent, non-negative PRs obtained by subtracting the maximum number of $\pi/2$ multiples without obtaining a negative result; thus, all recorded PRs are $\in [0, \pi/2)$ radians.

For all STCs discussed in this work, the reference column(s) are the leftmost column(s) of the STC matrix that do not contain summations of terms but collectively contain a variation of every data symbol within the data symbol vector. For all SMSTCs, every column is a reference column in order to contain a variation of every data symbol.

Two KR forms are now presented. Several examples of these forms are provided in the

subsection that follows, and additional KRs are included in Appendix B.

Tabular Form Key Residue

The KR may be represented in tabular form by grouping these relationships together. Symbol groups that produce a zero inner product are formed first. These groups consist of one or more symbol relationships for one or more data symbols. Symbol relationships for the reference column PRs are grouped together last and listed as corresponding complex exponentials with applicable weighting factors. As these relationships consist of only a single symbol variant, the symbol location is listed for both variant location 1 and 2. The tabular form KR is produced by listing the relationship and variant locations of each symbol relationship, of each data symbol, for all symbol groups within the STC.

Within the tabular form KR, a symbol group index (SGI) is used to abstractly identify the count of symbol relationships for each data symbol within each symbol group. For instance, SGI 3.2.1 refers to the first symbol relationship for the second data symbol in the third symbol group. Whereas the symbol relationship count is unnecessary for many STCs, it is important for the MPO334 and MPO434 STCs.

Matrix Form Key Residue

A KR may also be represented in matrix form. Given a STC, the matrix form KR may be determined by performing symbol permutations, symbol PRs, and symbol conjugations. Each operation is performed as necessary in order to place the unconjugated and minimally-rotated symbols from the data symbol vector, s , in order from top-to-bottom, left-to-right within the STC reference column(s). A minimally-rotated symbol is one with a minimal equivalent, non-negative PR $\in [0, \pi/2)$ radians as previously described; thus, symbol PRs performed to achieve this result must be integer multiples of $\pi/2$ radians. Recall within Algorithm S7 that the symbol conjugation operation does not impact applied PRs.

For consistency with the MPO334 and MPO434 STCs, the symbol in the highest row of the reference column that does not have a weight of $1/\sqrt{2}$ is considered s_1 , the next like term is considered s_2 , and the terms weighted by $1/\sqrt{2}$ are considered $s_3/\sqrt{2}$.

Example Key Residues

Several examples follow to provide further clarification about the defined KR forms. Additional KR examples are included in Appendix B.

Example 5.1.5. Recall the Alamouti variant code originally given in (5.2) as

$$\mathbf{G} = \begin{bmatrix} s_1 \cdot e^{j3\pi/4} & -s_2^* \\ s_2 \cdot e^{j\pi/4} & s_1^* \cdot e^{j3\pi/2} \end{bmatrix}.$$

The tabular form KR of (5.2) is provided in Table 5.1. The matrix form KR of (5.2) is determined by phase rotating symbol s_1 and is represented as

$$\mathbf{G}_{KR} = \begin{bmatrix} s_1 \cdot e^{j\pi/4} & -s_2^* \\ s_2 \cdot e^{j\pi/4} & s_1^* \end{bmatrix}. \quad (5.6)$$

Table 5.1. Tabular form KR for the Alamouti variant code shown in (5.2).

SGI	Relationship	Location of Variant 1	Location of Variant 2
1.1.1	Conj. $e^{j\pi/4}$	(1, 1)	(2, 2)
1.2.1	Conj. $e^{j5\pi/4}$	(2, 1)	(1, 2)
2.1.1	$e^{j\pi/4}$	(1, 1)	(1, 1)
2.2.1	$e^{j\pi/4}$	(2, 1)	(2, 1)

Example 5.1.6. Recall the NO442 variant code originally given in (3.30) as

$$\mathbf{G} = \begin{bmatrix} s_4^* \cdot e^{j21\pi/16} & s_3^* \cdot e^{j15\pi/16} & s_2 \cdot e^{j49\pi/32} & s_1^* \cdot e^{j57\pi/32} \\ s_2^* \cdot e^{j3\pi/2} & s_1 \cdot e^{j39\pi/32} & s_4 \cdot e^{j23\pi/32} & s_3 \cdot e^{j17\pi/16} \end{bmatrix}.$$

The tabular form KR of (3.30) is provided in Table 5.2. Note the two leftmost columns are the reference columns for this code. The matrix form KR of (3.30) is determined by phase

rotating all symbols, conjugating all symbols except for s_1 , and permuting symbols s_1 and s_4 . The matrix form KR is represented as

$$\mathbf{G}_{KR} = \begin{bmatrix} s_1 \cdot e^{j5\pi/16} & s_3 \cdot e^{j7\pi/16} & s_2^* \cdot e^{j33\pi/32} & s_4^* \cdot e^{j25\pi/32} \\ s_2 & s_4 \cdot e^{j7\pi/32} & s_1^* \cdot e^{j55\pi/32} & s_3^* \cdot e^{j50\pi/32} \end{bmatrix}. \quad (5.7)$$

Table 5.2. Tabular form KR for the NO442 variant code shown in (3.30).

SGI	Relationship	Location of Variant 1	Location of Variant 2
1.1.1	Conj. $e^{j\pi/32}$	(1, 1)	(2, 3)
1.2.1	Conj. $e^{j33\pi/32}$	(2, 1)	(1, 3)
2.1.1	Conj.	(1, 2)	(2, 4)
2.2.1	Conj. -1	(2, 2)	(1, 4)
3.1.1	$e^{j5\pi/16}$	(1, 1)	(1, 1)
3.2.1	1	(2, 1)	(2, 1)
3.3.1	$e^{j7\pi/16}$	(1, 2)	(1, 2)
3.4.1	$e^{j7\pi/32}$	(2, 2)	(2, 2)

Example 5.1.7. Recall the SM221 variant code originally given in (3.21) as

$$\mathbf{G} = \begin{bmatrix} s_2 \cdot e^{j\pi/8} & s_1^* \cdot e^{j5\pi/32} \end{bmatrix}.$$

The tabular form KR of (3.21) is provided in Table 5.3. The matrix form KR of (3.21) is determined by conjugating s_1 and permuting symbols s_1 and s_2 and is represented as

$$\mathbf{G}_{KR} = \begin{bmatrix} s_1 \cdot e^{j\pi/8} & s_2 \cdot e^{j5\pi/32} \end{bmatrix}. \quad (5.8)$$

Table 5.3. Tabular form KR for the SM221 variant code shown in (3.21).

SGI	Relationship	Location of Variant 1	Location of Variant 2
1.1.1	$e^{j\pi/8}$	(1, 1)	(1, 1)
1.2.1	$e^{j5\pi/32}$	(1, 2)	(1, 2)

5.1.3 Key Residue Class Definition

In [68], the term KRC was used to refer to keys which act similarly and may be indistinguishable under certain conditions. Whereas this usage of the term is compatible with STC PLS schemes, [68] adds that when the KRC is known, the message is also known. As this latter statement is not true for our purposes, we provide a more complete definition of KRC as it pertains to ST coding-based PLS schemes.

The KRC is a subset of the full code set consisting of all “keys,” or STCs, sharing the same KR. All KRCs are of equal size and collectively partition the full code set for any given STC. The KRC size when $a = 1$ is given as

$$|KRC| = 4^k \cdot k!. \quad (5.9)$$

The KRC size for any given STC depends on the number of data symbols per codeword and the number of allowed symbol PRs that may be performed without changing the associated KR. From the definition given in the previous subsection, all $k!$ symbol permutations and 2^k symbol conjugations may be performed without affecting the KR. The number of symbol PRs permitted depends on the value of a . When $a = 1$, only 0 and π radian PRs are allowed for each symbol; however, when $a \geq 2$, all symbols may be phase-rotated by 0, $\pi/2$, π , or $3\pi/2$ radians without changing the associated KR.

Thus, for any value of a , KRC size is given as

$$|KRC| = 2^{k(\min(a, 2)+1)} \cdot k!, \quad (5.10)$$

where $\min(a, 2)$ represents the standard minimum operation between a and 2.

For all base STCs given in Section 2.1.1, it can be seen that the number of KRCs increases as the value of a increases by dividing the corresponding full set cardinality given in (2.17) or (2.18) as applicable by the KRC size in (5.10). Table 5.4 provides the number of KRCs for each base STC when incorporating PR algorithms with various values of a .

5.1.4 Message and Cryptogram Residue Class Definition

From [13], a message residue class is defined as the set of messages that might have produced a given cryptogram. Also, from [13], a cryptogram residue class is defined as the set of cryptograms that may be produced from a given message. We supplement these definitions by adding our cryptogram definition, clarifying that the message is the embedded data symbol vector for a STC, and adding a KRC caveat to the cryptogram residue class definition.

A message residue class is defined herein as the set of data symbol vectors that might have produced a given cryptogram. Similarly, a cryptogram residue class is the set of cryptograms that may be produced from a given data symbol vector by employing only those STCs from a single KRC. This caveat regarding the KRC is important as each KRC has different cryptogram residue classes matching the KR for that class.

When $a = 1$, the size of these classes is given as

$$\phi_i = \frac{4^k \cdot k!}{f}. \quad (5.11)$$

The repetition metric, f , in this case is calculated as

$$f = \prod_{l=1}^u n_{s_l}!, \quad (5.12)$$

where u is the number of unique symbol sets (USSs) represented in each data symbol vector of the message residue class, and $\mathbf{n}_s = [n_{s_1} \ \dots \ n_{s_u}]^T$ is a u -by-1 column vector containing the number of times a symbol appears in each data symbol vector from each USS represented.

Table 5.4. Number of KRCs for base STCs when incorporating PR algorithms with various values of a .

STC	Alamouti	SPO334	MPO468	MPO334	MPO434	NO442	SM221	SM441
# of KRCs ($a = 1$)	2	192	2,580,480	576	2,304	12	1	1
# of KRCs ($a = 2$)	4	1,536	$3.30 \cdot 10^8$	9,216	73,728	48	1	1
# of KRCs ($a = 3$)	32	98,304	$2.71 \cdot 10^{12}$	1,179,648	$1.89 \cdot 10^7$	3,072	4	16
# of KRCs ($a = 4$)	256	6,291,456	$2.22 \cdot 10^{16}$	$1.51 \cdot 10^8$	$4.83 \cdot 10^9$	196,608	16	256
# of KRCs ($a = 5$)	2,048	$4.03 \cdot 10^8$	$1.82 \cdot 10^{20}$	$1.93 \cdot 10^{10}$	$1.24 \cdot 10^{12}$	$1.26 \cdot 10^7$	64	4,096
# of KRCs ($a = 6$)	16,384	$2.58 \cdot 10^{10}$	$1.49 \cdot 10^{24}$	$2.47 \cdot 10^{12}$	$3.17 \cdot 10^{14}$	$8.05 \cdot 10^8$	256	65,536
# of KRCs ($a = 9$)	8,388,608	$6.76 \cdot 10^{15}$	$8.18 \cdot 10^{35}$	$5.19 \cdot 10^{18}$	$5.31 \cdot 10^{21}$	$2.11 \cdot 10^{14}$	16,384	$2.68 \cdot 10^8$

When $a = 1$, a USS is a set of four symbols within the chosen modulation scheme equal to the original, negative, conjugate, or negative-conjugate of one another. It is this set of equalities that results in the 4^k term in (5.11) [9]. For any modulation scheme of order M with constellation exhibiting bilateral symmetry across both the real and imaginary axes, the number of USSs when $a = 1$ is equal to $M/4$. As a point of reference, the USSs for the 16QAM scheme when $a = 1$ are shown with differing colors and shapes in Fig. 5.1.

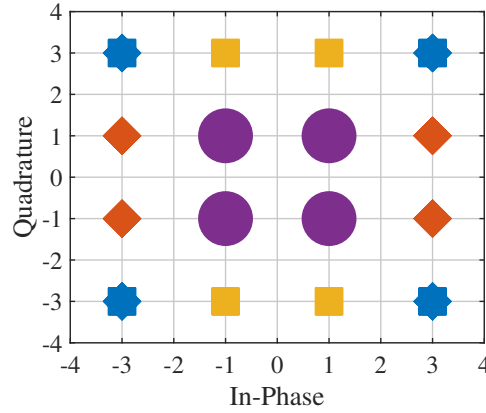


Figure 5.1. Illustration of USSs within the 16QAM scheme when $a = 1$.

Employing PR algorithms with $a \geq 2$ introduces additional equalities requiring the USS definition to be revised. Thus, when $a \geq 2$, a USS is herein defined as a set of symbols within the chosen modulation scheme equal to the original or conjugate of one another phase-rotated by an integer multiple of $\pi/2$ radians. This affects the 4^k term in (5.11) and the u variable and elements of $\mathbf{n}_s$ within the f metric.

As a point of reference, symbols belonging to each USS for the 64QAM scheme when $a \geq 2$ are shown in Fig. 5.2 with colors representing the number of symbols within each set. Constant energy rings are added as an aid to the reader. The two USSs on the red and blue dashed energy ring are separated due to this revised USS definition.

Thus, when $a \geq 2$, the 4^k term in (5.11) must be replaced by the product of the number of symbols within the USS of each symbol in the data symbol vector. Performing this

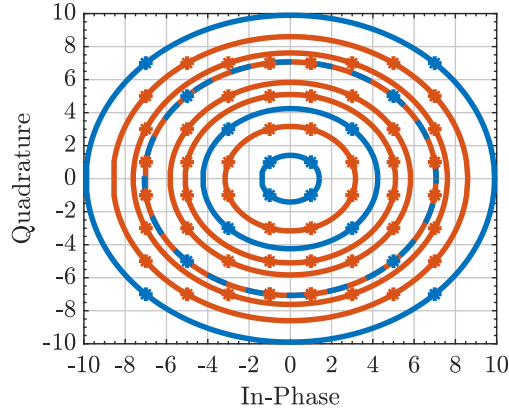


Figure 5.2. Illustration of USSs within the 64QAM scheme when $a \geq 2$.

replacement along with substituting (5.12) into (5.11) provides

$$\phi_i = \prod_{l=1}^u \frac{|USS_l|^{n_{s_l}}}{n_{s_l}!} \cdot k!, \quad (5.13)$$

where $|USS_l|$ represents the number of symbols contained within USS_l , and u and $\mathbf{n}_s$ are as previously defined in (5.12).

For convenience, the terms in (5.13) that change for different residue classes are grouped together in a new metric, Q_f , defined as

$$Q_f = \prod_{l=1}^u \frac{|USS_l|^{n_{s_l}}}{n_{s_l}!} \quad (5.14)$$

such that

$$\phi_i = Q_f \cdot k!. \quad (5.15)$$

To understand how the repetition metric, f , is affected by incorporation of PR algorithms with $a \geq 2$ and its relation to Q_f , we review the properties of message and cryptogram residue classes within a pure cipher system [13]. When employing only the STCs from a single KRC of a chosen STC, it can be shown that the message and cryptogram residue classes meet the properties for those of a pure cipher system in [13, Theorem 3]. A summary and adaptation of these properties is provided here:

- Property 1) Message residue classes $\mathbf{C}_1, \mathbf{C}_2$, etc. collectively form a partition of the data symbol vector space for a given digital modulation scheme. Similarly for the cryptogram residue classes $\mathbf{C}'_1, \mathbf{C}'_2$, etc.
- Property 2) Embedding any data symbol vector in $\mathbf{C}_i$ within any STC of the chosen KRC produces a cryptogram in $\mathbf{C}'_i$. Decoding any cryptogram in $\mathbf{C}'_i$ with any STC of the chosen KRC leads to a data symbol vector in $\mathbf{C}_i$.
- Property 3) The number of data symbol vectors in $\mathbf{C}_i$, say ϕ_i , is equal to the number of cryptograms in $\mathbf{C}'_i$ and is a divisor of the number of STCs in the chosen KRC.
- Property 4) Each data symbol vector in $\mathbf{C}_i$ can be embedded within exactly f different STCs of the chosen KRC to produce each cryptogram in $\mathbf{C}'_i$, where f is defined such that

$$f = \frac{|KRC|}{\phi_i}. \quad (5.16)$$

Substituting (5.10) for $|KRC|$ and (5.15) for ϕ_i in (5.16) yields

$$\begin{aligned} f &= \frac{2^{k(\min(a,2)+1)} \cdot k!}{Q_f \cdot k!} = \frac{2^{k(\min(a,2)+1)}}{Q_f} \\ &= 2^{k(\min(a,2)+1)} \cdot \prod_{l=1}^u \frac{n_{sl}!}{|USS_l|^{n_{sl}}}. \end{aligned} \quad (5.17)$$

When $a = 1$, each USS contains four symbols. For this case, it can be seen how (5.17) is equal to (5.12).

5.1.5 Residue Classes for the Alamouti STC

To better understand the message and cryptogram residue classes pertaining to STC PLS schemes, we now examine these classes for the Alamouti STC employing 16QAM data symbols and only the STCs from one of its two KRCs with $a = 1$. For each message and cryptogram residue class, with $k = 2$ and $M = 16$, either $u = 1$ or $u = 2$ leading to $\mathbf{n}_s = [2]$ or $\mathbf{n}_s = [1 \ 1]^T$, respectively; thus, under these conditions, the repetition metric for all message and cryptogram residue classes is either $f = 2$ or $f = 1$. Note that $2 = k!$ in this case.

For any chosen base code with $a = 1$, the number of data symbol vectors collectively contained within all message residue classes with a repetition metric of $f = k!$ can easily

be understood by determining the number of potential data symbol vectors that may be produced by selecting all k data symbols from within a single USS. This selection process can be further explained as selecting any of the M modulation symbols to be the first data symbol followed by selecting any of the 4 variations (e.g., original, negative, conjugate, or negative-conjugate) of the first modulation symbol for the remaining $k - 1$ data symbols. By this process, there are

$$M \cdot 4^{k-1} \quad (5.18)$$

data symbol vectors within message residue classes with $f = k!$. By substituting $k!$ for f in (5.11), it can be seen that there are exactly

$$\phi_i = \frac{4^k \cdot k!}{k!} = 4^k \quad (5.19)$$

data symbol vectors in each of these classes; therefore, dividing (5.18) by (5.19), there are

$$\frac{M \cdot 4^{k-1}}{4^k} = \frac{M}{4} \quad (5.20)$$

message residue classes and associated cryptogram residue classes with a repetition metric of $f = k!$. This is an intuitive solution as each of these classes represents the set of all data symbol vectors that can be produced by selecting all k data symbols from within a single USS. As there are $M/4$ USSs when $a = 1$, there are $M/4$ message and cryptogram residue classes with $f = k!$.

For the Alamouti STC employing 16QAM data symbols with $a = 1$, there are $16/4 = 4$ classes with a repetition metric of $f = 2$. Fig. 5.3 shows a template for these classes with the following variable definitions. Since there are 4 classes based on this template with $\phi_i = 4^k = 16$ data symbol vectors and cryptograms in each class, $i \in \{1, 2, 3, 4\}$, $l = 16 \cdot (i - 1) + 1$, $o = l + 15$, and $l < n < o$. The multi-colored lines connecting each data symbol vector on the left-hand side to a cryptogram on the right-hand side represent 6 of the 32 STCs within the chosen KRC. The particular colors serve to aid in distinguishing between different STCs, to illustrate properties 2 and 4 for the message and cryptogram residue classes of a pure cipher system given in Section 5.1.4, and to illustrate that a given STC cannot produce a cryptogram in C'_i from more than one data symbol vector in C_i . Any other pattern or relation that can be found within the chosen colors is unintended and does

not represent a known property of these classes.

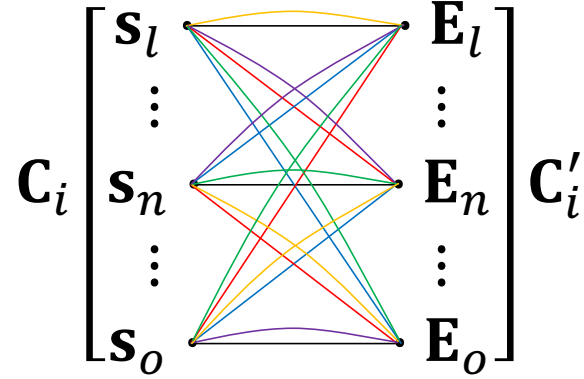


Figure 5.3. Template for Alamouti message and cryptogram residue classes with $f = 2$ employing 16QAM data symbols and STCs from a single KRC with $a = 1$. Adapted from [13].

For any chosen base code with $a = 1$, the number of data symbol vectors collectively contained within all message residue classes with a repetition metric of $f = 1$ can be understood as the set of all data symbol vectors produced by selecting all k data symbols from different USSs. For $a = 1$, this is only possible when $k < M/4$. Given these conditions are met, there are

$$\prod_{l=0}^{k-1} (M - 4 \cdot l) \quad (5.21)$$

data symbol vectors within message residue classes with a repetition metric of $f = 1$. By substituting 1 for f in (5.11), it can be seen that there are exactly

$$\phi_i = \frac{4^k \cdot k!}{1} = 4^k \cdot k! \quad (5.22)$$

data symbol vectors in each of these classes.

For the Alamouti STC employing 16QAM data symbols with $a = 1$, dividing (5.21) by (5.22) provides that there are

$$\frac{M \cdot (M - 4)}{4^k \cdot k!} = \frac{16 \cdot 12}{4^2 \cdot 2!} = 6 \quad (5.23)$$

message residue classes and associated cryptogram residue classes with a repetition metric of $f = 1$. Fig. 5.4 shows a template for these classes with the following variable definitions. Since there are 6 classes based on this template with $\phi_i = 4^k \cdot k! = 32$ data symbol vectors and cryptograms in each class, and 4 previously defined classes with $f = 2$ with a total of $4 \cdot 16 = 64$ data symbol vectors and cryptograms combined, $i \in \{5, 6, 7, 8, 9, 10\}$, $l = 32 \cdot (i - 5) + 65$, $o = l + 31$, and $l < n < o$. The multi-colored lines serve the same purpose as those in Fig. 5.3.

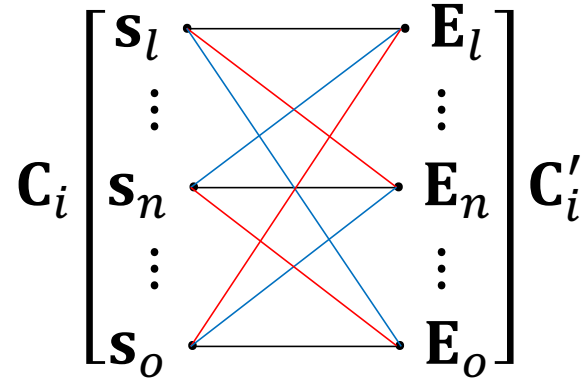


Figure 5.4. Template for Alamouti message and cryptogram residue classes with $f = 1$ employing 16QAM data symbols and STCs from a single KRC with $a = 1$. Adapted from [13].

It can be seen that property 1 in Section 5.1.4 is confirmed as these ten message residue classes collectively partition the data symbol vector space for the Alamouti STC employing 16QAM with a total of $M^k = 16^2 = 256$ data symbol vectors. Likewise for the cryptogram space matching the KR of the chosen KRC.

5.2 Alternating Space-Time Code Security

We now use established nomenclature to analyze the security offered by the proposed alternating STC PLS scheme from Section 4.1 against two possible attack models. The communications link described in Section 3.2.1 is used as the basis for this discussion in conjunction with the proposed scheme with $a = 9$ and $\mathbf{v}_{obs} = [\text{Alamouti}]$. GCH is not employed at this time. After presentation of security offered by only a single-STC

implementation, we will expound on differences by moving to a multi-STC implementation, but first we list the assumptions made about the eavesdropper.

5.2.1 Eavesdropper Assumptions

It is assumed that Eve:

- has a priori knowledge of the chosen $\mathbf{v}_{obs}$, value of a indicating allowed number of PRs, and digital modulation scheme employed,
- may have a priori knowledge of some PT data,
- employs N_{RE} RX antennas,
- has perfect CSI of the channel between BS and Eve denoted by the c -by- N_{RE} channel tap array $\mathbf{h}_{BE}$,
- has perfect timing synchronization,
- passively records all received data samples for an entire communications session,
- and has the ability to generate the matrix form KR, $\mathbf{G}_{KR}$, associated with all KRCs for the chosen $\mathbf{v}_{obs}$.

5.2.2 Key Residue Class Decoding

The first attack model is referred to as KRC decoding which attempts to determine the transmitted data frame using similarities between the STCs within a KRC. This involves gaining information about the embedded data symbol vector and STC employed for each codeword from the recorded sample array, $\mathbf{Z}$, and then using this information to search for the correct data frame. We first describe the process Eve may take to gain this information, and then we discuss the data frame search. Finally, we discuss the expected eavesdropper BER that may be achieved when employing this attack.

Given the previous non-trivial assumptions, Eve performs the following steps [9]:

- Step 1) For each KRC of the $\mathbf{v}_{obs}$, use $\mathbf{G}_{KR}$ along with channel tap array, $\mathbf{h}_{BE}$, to determine the corresponding matrix, $\mathbf{H}_{CBE}$, necessary for the applicable decoding method to be employed from Section 3.2.
- Step 2) Perform decoding method of the received sample array using the $\mathbf{H}_{CBE}$ matrix associated with each KRC to obtain results, $\hat{s}_i$ for $i \in \{1, \dots, l\}$ where l is the

number of KRCs.

Step 3) Determine the KRC to which the STC used for transmission of the cryptogram belongs. This is performed by finding the index value, i , of the result, $\hat{\mathbf{s}}_i$, with the minimum Euclidean distance to any one of the potential data symbol vectors, $\mathbf{s} \in \mathbb{M}^k$. This can be represented as

$$i = \underset{\forall i \in \{1, \dots, I\}}{\operatorname{argmin}} \left(\min_{\forall \mathbf{s} \in \mathbb{M}^k} d(\mathbf{s}, \hat{\mathbf{s}}_i) \right). \quad (5.24)$$

Step 4) Obtain a coded symbol vector, $\mathbf{s}_c$, by selecting the data symbol vector, $\mathbf{s}$, with the minimum Euclidean distance to the determined result, $\hat{\mathbf{s}}_i$, from step 3 as represented by

$$\mathbf{s}_c = \underset{\forall \mathbf{s} \in \mathbb{M}^k}{\operatorname{argmin}} d(\mathbf{s}, \hat{\mathbf{s}}_i). \quad (5.25)$$

From this point forward, the worst case is analyzed by assuming that Eve has been able to successfully obtain a coded symbol vector within the correct message residue class for each codeword of the entire recorded communications session. Likewise, it is assumed that Eve has successfully determined the correct KRC for each codeword. Although this requires a substantial amount of work, this case is possible with the assumptions provided in Section 5.2.1 validated with sufficient SNR, substantial amount of time and computing resources. The second assumption is evaluated in Chapter 6.

Message Equivocation

Once the coded symbol vector, $\mathbf{s}_c$, within the correct message residue class has been obtained, the amount of uncertainty remaining about the transmitted data symbol vector within each codeword is referred to as the equivocation of the message [13]. This is represented as

$$H(\mathbf{s}|\mathbf{s}_c) = \log_2(Q_f \cdot k!), \quad (5.26)$$

where $Q_f \cdot k!$ is the size of the message residue class to which the coded symbol vector belongs [9]. Given a particular coded symbol vector, Q_f may be calculated using (5.14) and inserted into (5.26) to determine message equivocation.

To calculate the mean and upper bounds (UBs) on message equivocation, the probability mass function of Q_f is required for all modulations evaluated. These functions are empir-

ically derived by first calculating Q_f , according to (5.14), for all M^k possible data symbol vectors for a given M -order modulation scheme and value of k . Assuming equally probable data symbol vectors, the probability of each Q_f value is calculated by counting the number of vectors resulting in each value and dividing that count by the total number of vectors, M^k . These mass functions and mean value of Q_f for 16QAM, 64QAM, and 256QAM with $k = 2$ are provided in Figs. 5.5 - 5.7.

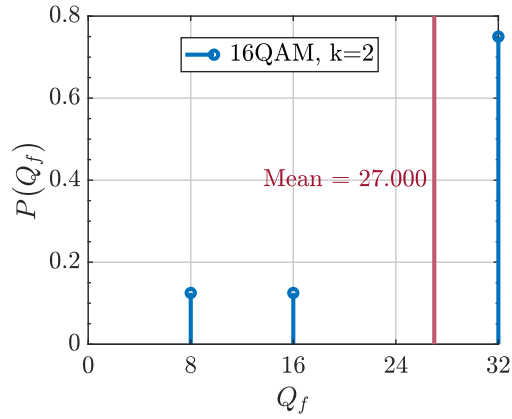


Figure 5.5. Probability mass function of the Q_f metric for 16QAM with $k = 2$.

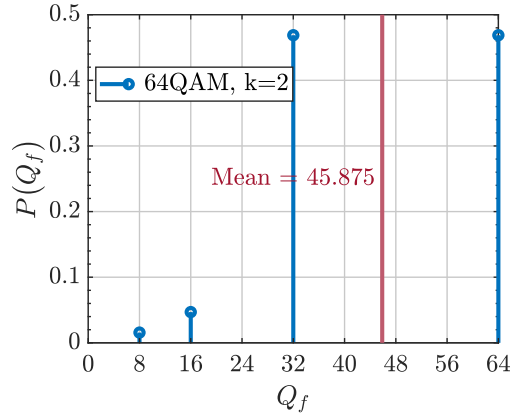


Figure 5.6. Probability mass function of the Q_f metric for 64QAM with $k = 2$.

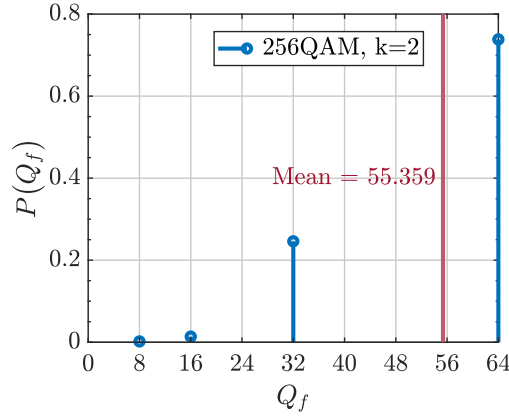


Figure 5.7. Probability mass function of the Q_f metric for 256QAM with $k = 2$.

For all modulations evaluated, all USSs contain at least four symbols. Thus, the lower bound (LB) on message equivocation occurs when $Q_f = 4^k/k!$. This is the case when a coded symbol vector contains k symbols from a single USS with $|USS_l| = 4$. This is always true for QPSK. Mean, UB, and LB values of message equivocation for various STCs with k symbols per codeword employing various modulations can be seen in Fig. 5.8.

The amount of information Eve gains about the message by obtaining $\mathbf{s}_c$ is the mutual information between $\mathbf{s}$ and $\mathbf{s}_c$, represented as

$$\begin{aligned} I(\mathbf{s}; \mathbf{s}_c) &= H(\mathbf{s}) - H(\mathbf{s}|\mathbf{s}_c) \\ &= k \cdot \log_2(M) - \log_2(Q_f \cdot k!), \end{aligned} \quad (5.27)$$

where $H(\mathbf{s})$ is the entropy in $\mathbf{s}$ before obtaining $\mathbf{s}_c$ which is equal to the number of bits per codeword [9], [13]. When QPSK is used, $M = 4$, $Q_f = 4^k/k!$, $I(\mathbf{s}; \mathbf{s}_c) = 0$, and Eve gains no information about the message by obtaining the coded symbol vector. Mean, UB, and LB values of this mutual information for various STCs with k symbols per codeword employing various modulations can be seen in Fig. 5.9.

Key Equivocation

Once the correct KRC is determined for a given codeword, the KR associated with that KRC (i.e., $\mathbf{G}_{KR}$) is readily available. The amount of uncertainty remaining about the STC used

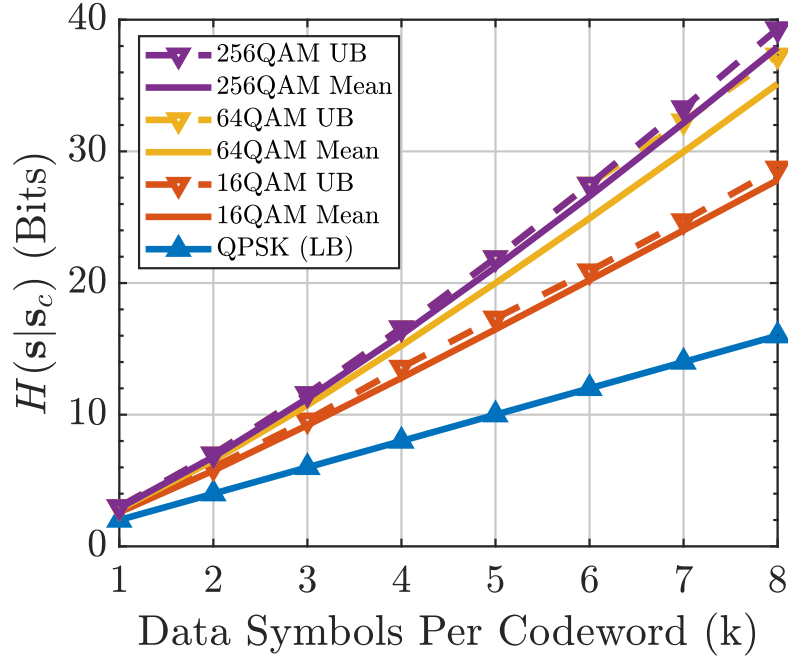


Figure 5.8. Message equivocation given STCs with various number of data symbols per codeword and modulation schemes.

for transmission of each cryptogram is called the equivocation of the key [13]. In this case, key equivocation depends only on the KRC size as given in (5.10). Thus, key equivocation is represented as

$$H(\mathbf{G}|\mathbf{G}_{KR}) = \log_2(2^{k(\min(a, 2)+1)} \cdot k!). \quad (5.28)$$

The amount of information Eve gains about the key by obtaining the KR is the mutual information between $\mathbf{G}$ and $\mathbf{G}_{KR}$, represented as

$$I(\mathbf{G}; \mathbf{G}_{KR}) = H(\mathbf{G}) - H(\mathbf{G}|\mathbf{G}_{KR}), \quad (5.29)$$

where

$$H(\mathbf{G}) = \log_2(|\mathbf{S}_s|) \quad (5.30)$$

is the entropy in $\mathbf{G}$ before obtaining the KR in bits. $H(\mathbf{G})$ depends on the set cardinality

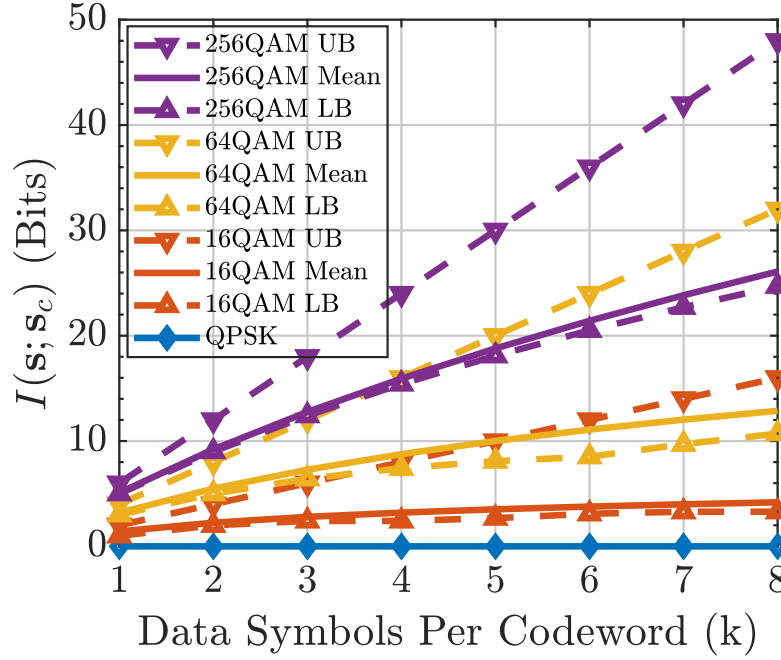


Figure 5.9. Mutual information between $\mathbf{s}$ and $\mathbf{s}_c$ given STCs with various number of data symbols per codeword and modulation schemes.

which depends on the STC chosen and the value of a . For various base STCs given in Section 2.1.1 with $a = 9$, the values of $H(\mathbf{G})$, $H(\mathbf{G}|\mathbf{G}_{KR})$, and $I(\mathbf{G}; \mathbf{G}_{KR})$ are provided in Table 5.5.

Attack Analysis

After gathering information for every recorded codeword, Eve may use the key and message equivocation to search for the correct transmitted data frame.

If a static STC from the set is employed throughout the entire recorded communications session rather than alternating the STC as proposed, Eve could relatively easily obtain the correct data frame using the key equivocation. After performing the initial four step approach of this attack for only a few codewords, Eve could determine the KRC of the employed STC with confidence. Upon determination of the KRC, the correct data frame is one of $|\text{KRC}|$ possibilities corresponding to the outputs produced by performing the applicable decoding method of all received codewords for all STCs within the determined KRC.

Table 5.5. Entropy in $\mathbf{G}$, key equivocation, and mutual information between $\mathbf{G}$ and $\mathbf{G}_{KR}$ in bits for various STCs with $a = 9$

STC	$H(\mathbf{G})$	$H(\mathbf{G} \mathbf{G}_{KR})$	$I(\mathbf{G}; \mathbf{G}_{KR})$
Alamouti	30.0	7.0	23.0
SPO334	64.2	11.6	52.6
MPO468	146.8	27.5	119.3
MPO334	73.8	11.6	62.2
MPO434	83.8	11.6	72.2
SM221	21.0	7.0	14.0
NO442	64.2	16.6	47.6

In this case with a static STC employed, having known PT could further reduce the number of possible data frames down to the repetition metric, f , as shown in (5.17). The value of the repetition metric depends on the data symbol vectors contained within the known PT portion of the transmission. This would require solving for the transmitted cryptogram from the received sample array and determining the f STCs within the KRC capable of producing the transmitted cryptogram from the known data symbol vector.

The complexity of the KRC decoding attack increases exponentially by employing the proposed alternating STC scheme. Without knowledge of the SIM unique contents and a synchronized PRNG, selection of the STC used for transmission of each codeword appears to be independent from the perspective of the eavesdropper. Thus, Eve could search for the correct data frame using the message equivocation instead of the key equivocation. This search involves all possible combinations of the ϕ_i unique data symbol vectors in the determined message residue class of each received codeword. For a frame of n codewords when using this approach, the correct data frame is one of $(E[\phi_i])^n$ possible combinations, where $E[\phi_i] = E[Q_f] \cdot k!$ and $E[\]$ is the expectation operator.

Example 5.2.1. For a single data frame of length corresponding to a common ethernet maximum frame length of 1,518 bytes [69], the number of codewords and possible output frame combinations when attempting to search for the correct data frame using the message

equivocation of each codeword is shown in Table 5.6 for various modulations employing the proposed scheme with $a \geq 2$ and $\mathbf{v}_{obs} = [\text{Alamouti}]$. The values of $E[Q_f]$ for 16QAM, 64QAM, and 256QAM can be seen in Figs. 5.5 - 5.7. When QPSK is employed, $E[Q_f]$ is deterministic as QPSK has only a single USS containing all four symbols; thus, $Q_f = 4^k/k!$.

Table 5.6. Number of codewords and possible output frame combinations for a given frame length when performing a search using the message equivocation of each codeword.

Modulation	$E[Q_f]$	$E[\phi_i]$	# Codewords	Possible Combinations
QPSK	8	16	3036	$5.11 \cdot 10^{3655} = 2^{1518 \cdot 8}$
16QAM	27	54	1518	$5.94 \cdot 10^{2629}$
64QAM	45.875	91.75	1012	$1.44 \cdot 10^{1986}$
256QAM	55.359	110.718	759	$3.64 \cdot 10^{1551}$

In this case with an alternating STC, having known PT yields little benefit as Eve is unable to determine the NSIs corresponding to the STCs used to transmit the known data. Without knowledge of the NSIs, $\mathbf{v}_{tbs}$, and $\mathbf{v}_{toso}$, Eve is unable to improve upon the search complexity for the codewords received either before or after the known PT portion of the transmission.

Expected Eavesdropper Bit Error Rate

For this attack, if Eve chooses to forgo the frame search process and instead attempts to obtain an estimate of the data symbol vector, $\hat{\mathbf{s}}$, from the coded symbol vector, $\mathbf{s}_c$, for each codeword, we can determine the expected BER Eve will achieve. Again, this employs the worst-case assumption that Eve has obtained a coded symbol vector from the correct message residue class for each codeword.

In order for Eve to obtain an estimate of the data symbol vector, Eve has two equivalent techniques available to decode $\mathbf{s}_c$. First, Eve could create the message residue class to which the coded symbol vector belongs and randomly select one of the data symbol vectors from the class to be $\hat{\mathbf{s}}$ as they are all equally probable. Alternatively, $\hat{\mathbf{s}}$ may be obtained

by performing *symbol assignment* and *variation decision* operations to decode $\mathbf{s}_c$. The amount of uncertainty in either decoding process is equivalent and represented in (5.26). To determine the expected BER, we further describe the latter decoding technique.

Symbol assignment is performed by assigning each of the symbols in $\mathbf{s}_c$ to a given data symbol in $\hat{\mathbf{s}}$. There are $k!$ permutations of performing symbol assignment. The number of permutations that will assign l symbols correctly and the remaining $k-l$ symbols incorrectly is

$$\text{Perms}(l, k) = \begin{cases} 1 & k = l \\ \sum_{i=2}^{k-l} \left(\frac{(-1)^i \cdot k!}{i! \cdot l!} \right) & k - l \geq 2. \\ 0 & \text{otherwise} \end{cases} \quad \begin{matrix} (5.31a) \\ (5.31b) \\ (5.31c) \end{matrix}$$

Variation decision is performed for each symbol in $\mathbf{s}_c$ by deciding which of the symbols contained within the corresponding USS should be inserted into $\hat{\mathbf{s}}$ for the assigned symbol. As each symbol within the USS is equally probable to have produced the corresponding symbol in $\mathbf{s}_c$, Eve decides on a symbol at random. When $a = 1$, there are four symbols within each USS, and thus there are four possible variation decisions for each symbol. When $a \geq 2$, the number of possible variation decisions for a given symbol from $\mathbf{s}_c$ is dependent on the number of symbols within the particular USS to which that symbol belongs.

Applying this decoding technique, we calculate Eve's expected BER as

$$E[\text{BER}] = \frac{E[l|k] \cdot E[B_{ca}] + (k - E[l|k]) \cdot E[B_{ia}]}{k \cdot b}, \quad (5.32)$$

where $E[\]$ is the expectation operator, l is the number of correctly assigned symbols per codeword, k is the number of symbols per codeword, B_{ca} is the number of bit errors per correctly assigned symbol, B_{ia} is the number of bit errors per incorrectly assigned symbol, $b = \log_2(M)$ is the number of bits per symbol, and M is the order of the digital modulation scheme employed.

The expected value of l given k , can be calculated as

$$E[l|k] = \sum_{l=0}^k l \cdot P(l|k), \quad (5.33)$$

where

$$P(l|k) = \frac{\text{Perms}(l, k)}{k!} \quad (5.34)$$

is the probability of l given k , assuming each permutation is equally probable. Substituting (5.31) into (5.34), (5.34) into (5.33), and removing the $l = 0$ and $l = k - 1$ cases as they always evaluate to 0 results in

$$E[l|k] = \begin{cases} 1 & k \leq 2 \\ \frac{k}{k!} + \sum_{l=1}^{k-2} l \cdot \sum_{i=2}^{k-l} \left(\frac{(-1)^i}{i! \cdot l!} \right) & k > 2. \end{cases} \quad (5.35a)$$

$$(5.35b)$$

It can be shown that evaluating (5.35) for any value of k yields $E[l|k] = 1$.

Assuming codeword symbols are independent of one another, we know that $E[B_{ia}] = b/2$. When $a = 1$, $E[B_{ca}] = 1$ based on the average of the four possible variation decisions for any common BRGC modulation mapping. When $a \geq 2$, $E[B_{ca}]$ is empirically determined by calculating the expected number of bit errors when demodulating each symbol within each USS by every other symbol within the same USS and averaging results across all symbols within the chosen modulation scheme. This assumes modulation symbols are equally probable. These empirical results for various 5G NR modulations employing bit mappings specified in [56] are provided in Table 5.7.

Inserting stated values for $E[l|k]$ and $E[B_{ia}]$ into (5.32) yields

$$E[\text{BER}] = \frac{1 \cdot E[B_{ca}] + (k - 1) \cdot \frac{b}{2}}{k \cdot b}. \quad (5.36)$$

This represents the average expected eavesdropper BER for all message residue classes once the eavesdropper has obtained a coded symbol vector within the correct message residue class. A plot of (5.36) for various STCs with k symbols per codeword employing various

Table 5.7. Expected number of bit errors per correctly assigned symbol for various modulations when $a \geq 2$.

Modulation	$E[B_{ca}]$
QPSK	1
16QAM	1.5
64QAM	2
256QAM	2.5

modulation schemes with corresponding values of $E[B_{ca}]$ from Table 5.7 can be seen in Fig. 5.10.

Additionally, 100,000 Monte Carlo simulations are performed for the Alamouti, SPO334, and MPO468 STCs employing QPSK, 16QAM, 64QAM, and 256QAM with energy per bit to noise power spectral density ratio (E_b/N_0) values of 5, 10, 15, and 20 dB, respectively. For each simulation, the worst-case assumption that Eve has obtained a coded symbol vector within the correct message residue class is enforced. The results of these simulations are included in Fig. 5.10.

5.2.3 Exhaustive Key Search

Whereas the KRC decoding attack complexity increases exponentially with the data frame length and is dependent on the employed modulation scheme, the complexity of the exhaustive key search attack is independent of these factors [70]. For a transmission longer than only a few codewords, the least complex attack on the alternating STC PLS scheme, to our knowledge, is for Eve to perform the exhaustive key search. This attack involves performing a brute force search for the $\mathbf{v}_{tbs}$, $\mathbf{v}_{toso}$, and PRNG state at the beginning of the data transmission phase. This is based on a reasonable assumption that every unique combination of these variables yields a unique sequence of alternative STCs for a communications session of a given length. Exhaustive validation of this assumption depends upon the PRNG implementation and length of communications session and is left as an area of future research.

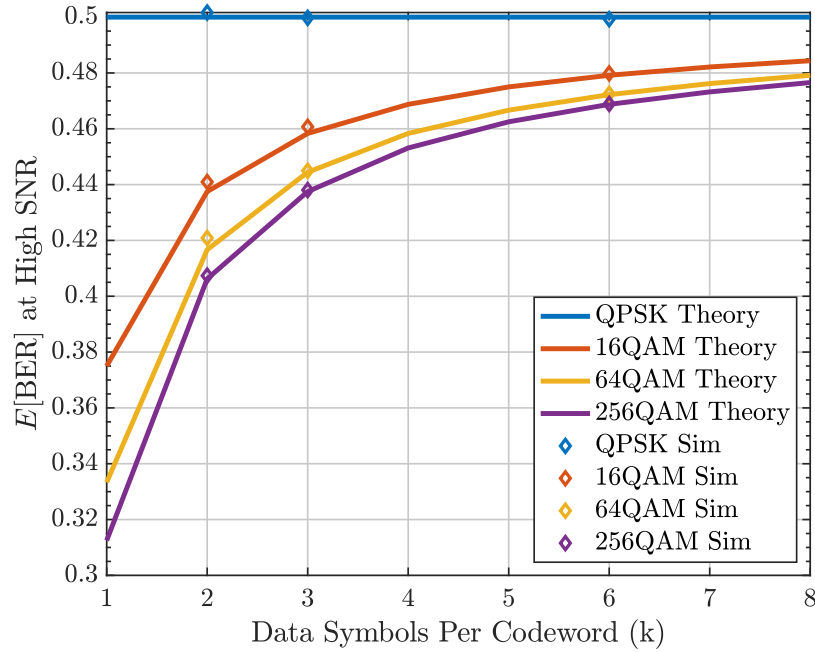


Figure 5.10. Theoretical plot of (5.36) with various number of data symbols per codeword and modulation schemes including high SNR Monte Carlo simulations for the Alamouti, SPO334, and MPO468 STCs.

The complexity of this attack depends on the number of possibilities for each of these variables. Assuming that the seed space of the employed PRNG in bits is equal to $|\text{SOK}_s|$ and that Eve knows the chosen system $\mathbf{v}_{obs}$, the number of possible combinations that must be searched is equal to the number of unique SIMs given in (4.7).

Using this attack method, the correct data frame is the decoding result of one of these possible combinations. The number of combinations for a few example systems with PRNG seed space equal to $|\text{SOK}_s| = 64$ bits, variable values of a and $\mathbf{v}_{obs}$ are provided in Table 5.8 for comparison. For Systems A and B when Alamouti is the only employed STC, $|\mathbf{v}_{bos}| = |\mathbf{v}_{toso}| = 5$ since symbol PRs and symbol permutations are not allowed for that STC; however, $|\mathbf{v}_{bos}| = 7$ for all remaining systems. Attack complexity in bits is calculated by taking the base two logarithm of the possible combinations and rounding up to the nearest bit.

Table 5.8. Exhaustive key search attack complexity for alternating STC PLS based on number of possible combinations for a few example systems.

System	a	$\mathbf{v}_{obs}$	Possible Combinations	Attack Complexity (bits)
A	1	[Alamouti]	$1.42 \cdot 10^{23}$	77
B	9	[Alamouti]	$2.38 \cdot 10^{30}$	101
C	1	[MPO468]	$7.08 \cdot 10^{35}$	120
D	9	[MPO468]	$1.44 \cdot 10^{67}$	224
E	1	[Alamouti SPO334 MPO468]	$3.34 \cdot 10^{42}$	142
F	9	[Alamouti SPO334 MPO468]	$3.20 \cdot 10^{95}$	318
G	1	[MPO468 NO442 SM441]	$3.20 \cdot 10^{44}$	148
H	9	[MPO468 NO442 SM441]	$7.86 \cdot 10^{99}$	332

5.2.4 Single versus Multi-Space-Time Code Security

Although the KRC decoding attack was previously only discussed for a single-STC implementation, it can be seen by comparing Tables 5.6 and 5.8 that the exhaustive key search attack is significantly less complex. The exhaustive key search attack for a multi-STC implementation, such as with system F, is even less complex than the KRC decoding attack on the single-STC system evaluated. Performing the KRC decoding attack on system F would be substantially more difficult as Eve would first have to determine the specific STC set being used for each codeword. Thus, the security offered by single- and multi-STC systems is considered to be based on the exhaustive key search attack complexity.

5.3 Gray Code Hopping Security

In order to discuss the security provided by the proposed GCH technique, we start by making assumptions about the eavesdropper's attack methodology. It is assumed that Eve:

- has a priori knowledge of the modulation order in use,

- may have a priori knowledge of some PT data,
- has perfect CSI of the channel between BS and Eve,
- passively records all received data samples for an entire communications session,
- and is aware that GCH is being employed.

As a worst-case scenario, it is further assumed that Eve is able to obtain the correct transmitted constellation symbol for every symbol received throughout the entire recorded communications session.

5.3.1 Information-Theoretic Analysis

Given these assumptions, the only uncertainty remaining are the data bits that correspond to each of the received constellation symbols. As stated in Section 2.2.2, it can be shown that there are $b!$ BRGC mappings that pair each of the constellation symbols with each of the binary labels. Thus, assuming that each BRGC mapping is equally probable when employing the GCH scheme, Eve gains no information about the data bits by determining the correct transmitted symbol. The amount of uncertainty remaining about the data bits given the constellation symbol, referred to as the equivocation of the message [13], is represented as

$$H(m|s) = \log_2(M) = b. \quad (5.37)$$

Since this is equal to the uncertainty of the data bits prior to receiving the constellation symbol, $H(m)$, perfect secrecy is attained by the GCH scheme [13]. Likewise, since every mapping contains every symbol, the amount of uncertainty remaining about the mapping in use given the constellation symbol, referred to as the equivocation of the key [13], is represented as

$$H(\mathbf{B}_m|s) = \log_2(2^b \cdot b!). \quad (5.38)$$

Thus, no information about the mapping is available given the constellation symbol alone.

If Eve has a priori knowledge of some PT data, then the mapping employed to create each corresponding symbol is known to be one of the $b!$ mappings that pairs the received symbol with the known PT data. As these mappings are equally probable, uncertainty of the mapping in use for each of these symbols is reduced, but not eliminated, and can be

represented as

$$H(\mathbf{B}_m | s \text{ and known PT}) = \log_2(b!). \quad (5.39)$$

Depending on the amount of known PT and specific implementation, Eve may be able to find an attack on the hopping sequence. Assuming this is not possible as assumed in [21], this reduced uncertainty for the mapping used with known PT data provides no information about the mapping in use before or after the known PT portion of the attack.

Comparison to Other Constellation Remapping Schemes

Perfect secrecy can also be achieved by the constellation remapping schemes presented in [16]–[18]; however, as discussed in Section 1.1.2, each of these schemes is more susceptible to a known PT attack than GCH due to the use of a static mapping throughout the communications session.

When only $2^b = M$ alternative mappings are employed in a static fashion as discussed in [16] and [17], only a single symbol of known PT is required for Eve to determine the static mapping in use and decode the entire recorded session. Alternatively, when all $M!$ possible mappings are employed in a static fashion as discussed in [18], $M - 1$ unique symbols from the constellation corresponding to known PT data must be received before the static mapping is determined and the entire recorded session may be decoded.

If these other schemes are modified to alternate the mapping in use as with GCH, techniques [16] and [17] are more susceptible to combined known PT and hopping sequence attacks than GCH due to the lesser number of alternative mappings. Alternating CD-PHY from [18] would yield greater uncertainty than GCH about the mapping in use for each of the known PT symbols; however, as discussed in Section 1.1.2, CD-PHY comes at a cost of reduced BER performance between the transmitter and intended receiver.

5.3.2 Exhaustive Key Search

Similar to the exhaustive key search attack for the alternating STC PLS scheme, Eve may also attempt this attack on the GCH scheme. This attack involves performing a brute force search for the $\mathbf{v}_{tbn}$, $\mathbf{v}_{tomo}$, and PRNG state at the beginning of the data transmission phase. This is based on a reasonable assumption that every unique combination of these variables yields a unique sequence of mappings for a communications session of a given length.

Again, exhaustive validation of this assumption depends upon the PRNG implementation and length of communications session and is left as an area of future research.

Unlike the alternating STC scheme, it is assumed that only a single modulation order is in use throughout a communications session for the purpose of this discussion. This is not a limitation of the proposed GCH scheme but an imposed assumption for evaluation of attack complexity. If an adaptive modulation coding scheme is employed, Eve would first have to determine the modulation order in use for each symbol before completing this attack. Evaluation of the efficacy of such a determination is beyond the scope of this dissertation and is left as an area of future research.

Using this exhaustive key search attack method, the correct data frame is the decoding result of one of the possible combinations of the brute force search variables. The number of possible combinations is given as

$$2^b \cdot b! \cdot 2! \cdot 2^{|SOK_m|}. \quad (5.40)$$

The number of combinations for a few example systems with PRNG seed space equal to $|SOK_m| = 64$ bits and various modulations are provided in Table 5.9 for comparison. Attack complexity in bits is calculated by taking the base two logarithm of the possible combinations and rounding up to the nearest bit.

Table 5.9. Exhaustive key search attack complexity for GCH PLS based on number of possible combinations for a few example systems.

System	Modulation	Possible Combinations	Attack Complexity (bits)
I	QPSK	$2.95 \cdot 10^{20}$	68
J	16QAM	$1.42 \cdot 10^{22}$	74
K	64QAM	$1.70 \cdot 10^{24}$	81
L	256QAM	$3.81 \cdot 10^{26}$	89

5.4 Combined Security

When both proposed PLS techniques are combined, the exhaustive key search attack complexity of both schemes in bits is additive. This is a result of all brute force search variables being independent of one another; thus, the number of possible combinations that must be attempted in order to perform the exhaustive key search attack on the combined system is the product of possible combinations required to attack the individual systems.

The number of combinations and attack complexity in bits for a few example systems with PRNG seed space for each technique equal to $|\text{SOK}_s| = |\text{SOK}_m| = 64$ bits incorporating various modulations and values of $\mathbf{v}_{obs}$ with $a = 9$ are provided in Table 5.10 for comparison.

Table 5.10. Exhaustive key search attack complexity for both proposed PLS techniques combined based on number of possible combinations for a few example systems with $a = 9$.

System	Modulation	$\mathbf{v}_{obs}$	Possible Combinations	Attack Complexity (bits)
M	QPSK	[Alamouti]	$7.02 \cdot 10^{50}$	169
N	256QAM	[Alamouti]	$9.05 \cdot 10^{56}$	190
O	QPSK	[MPO468]	$4.24 \cdot 10^{87}$	292
P	256QAM	[MPO468]	$5.46 \cdot 10^{93}$	312
Q	QPSK	[Alamouti SPO334 MPO468]	$9.44 \cdot 10^{115}$	386
R	256QAM	[Alamouti SPO334 MPO468]	$1.22 \cdot 10^{122}$	406
S	QPSK	[MPO468 NO442 SM441]	$2.32 \cdot 10^{120}$	400
T	256QAM	[MPO468 NO442 SM441]	$2.99 \cdot 10^{126}$	421

CHAPTER 6: Results

In this chapter we present various results including primary channel BER curves for Monte Carlo simulations and OTA testing of the alternating STC PLS scheme, evaluation of KRC decoding as presented in Section 5.2.2 based on simulation and OTA testing, comparison of GCH to alternating CD-PHY from the literature, and confirmation of compatibility of GCH to hard or soft decision decoding methods.

This chapter includes adaptations from previous work already published or in publication by the author. Specifically, portions of Section 6.1.1 are revised from [10]. Portions of Sections 6.3 and 6.4 are revised from [11].

6.1 Alternating Space-Time Coding Bit Error Rate

BER performance over the primary channel when employing the alternating STC PLS scheme is now provided for simulations first followed by OTA testing.

6.1.1 Simulation Results

Monte Carlo simulations are performed for three MIMO systems employing the alternating STC PLS scheme with $a = 4$ over a Rayleigh fading channel with AWGN. The three systems include:

1. $\mathbf{v}_{obs} = [\text{SM441}]$ with $N_R = 4$,
2. $\mathbf{v}_{obs} = [\text{NO442}]$ with $N_R = 2$, and
3. $\mathbf{v}_{obs} = [\text{Alamouti}]$ with $N_R = 2$.

The decoding techniques presented in Section 3.2.4 are compared for the first two systems, and MRC as described in Section 3.2.2 is employed for the third system. A baseline performance curve is plotted corresponding to theoretical expected performance of the third system. This curve represents the baseline as all three MIMO systems under test exhibit a combined diversity order (DO) of four; thus, expected ML performance for the first two systems should exhibit the same slope as that of the baseline curve but with

horizontal separation due to non-orthogonality of the employed STCs [66]. The decoding techniques tested for the first two systems include the combined MMSE-SQRD and SIC algorithms, denoted as SIC for short, along with the MATLAB communications toolbox Schnorr-Euchner sphere decoder (SESD) [61], [64], [71]. Results for various modulations are shown in Fig. 6.1.

It can be seen from Fig. 6.1 that BER performance degradation is more significant between the two compared decoding techniques when employing SM441 rather than NO442. This indicates that the significant BER performance boost is likely worth the additional decoding complexity of performing follow-on SD techniques when using a SMSTC; however, this may not be true for non-orthogonal STCs with transmit diversity depending on the application's required error rate. The BER performance of the third system matches theoretical expected performance as shown by the baseline curve.

The same Monte Carlo simulations are performed again with twice the number of RX antennas for each system to better understand decoding performance differences with additional antenna resources. Thus, for the first system using the SM441 STC, eight RX antennas are now employed. For the second and third systems using the NO442 and Alamouti STCs, respectively, four RX antennas are now employed. Baseline performance curves are plotted once again but with DO of eight. Results of these simulations with additional RX antennas are shown in Fig. 6.2.

It can be seen by comparing Figs. 6.1 and 6.2 that there is much less degradation in BER performance between the two compared decoding techniques when employing additional RX antennas with either of the first two MIMO systems under test. This indicates that SD techniques become less beneficial as additional RX antennas are employed. The BER performance of the third system matches theoretical expected performance as shown by the baseline curve.

6.1.2 Over-The-Air Testing Results

Portions of the alternating STC PLS scheme with $\mathbf{v}_{obs} = [\text{Alamouti}]$ were implemented as an out-of-tree module in open source software called GNU Radio [72]. For simplicity, the first two phases were assumed to be completed already; thus, with synchronized PRNGs and session variables established, the data transmission phase commenced upon execution

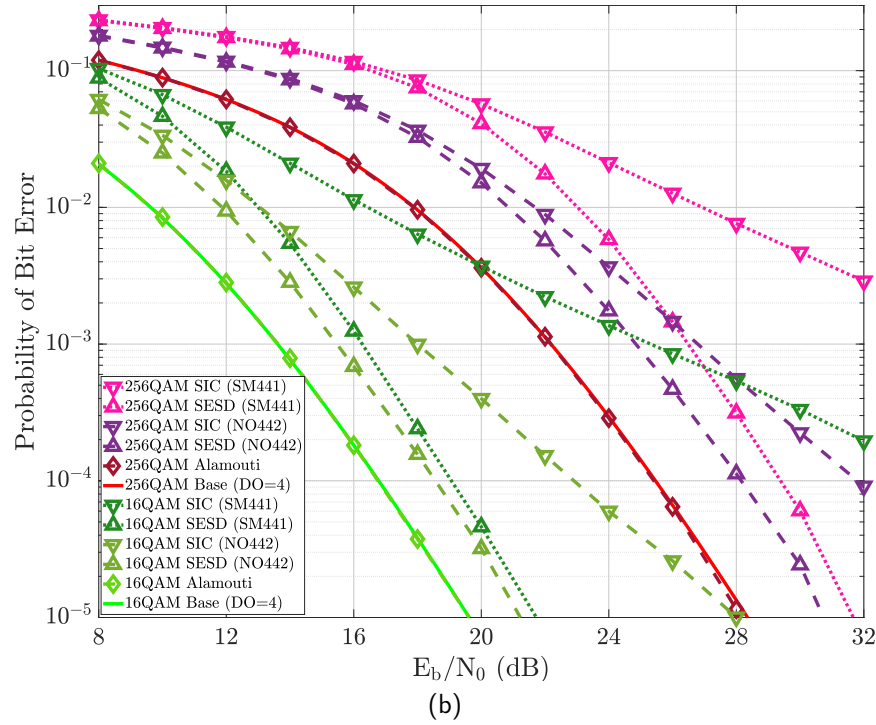
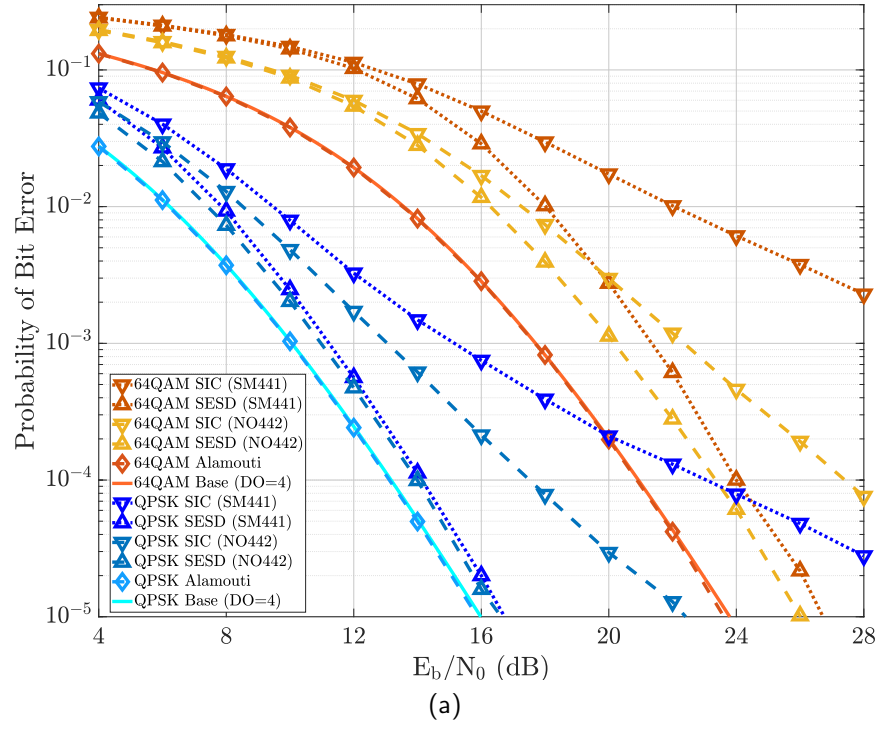


Figure 6.1. Comparison of decoding techniques for SM441, NO442, and Alamouti STCs using various modulations and DO of four. (a) QPSK and 64QAM. (b) 16QAM and 256QAM.

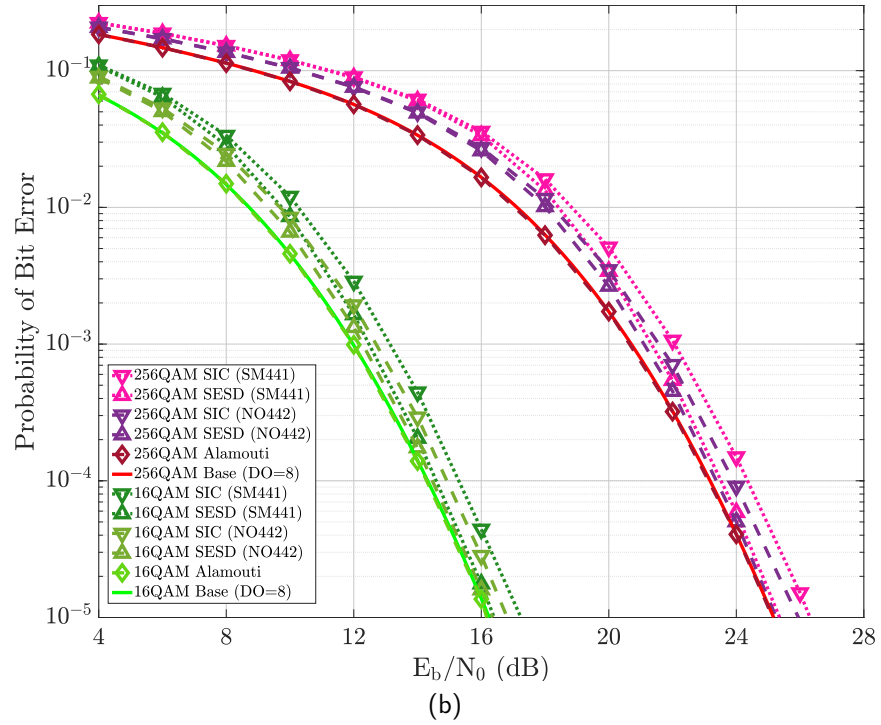
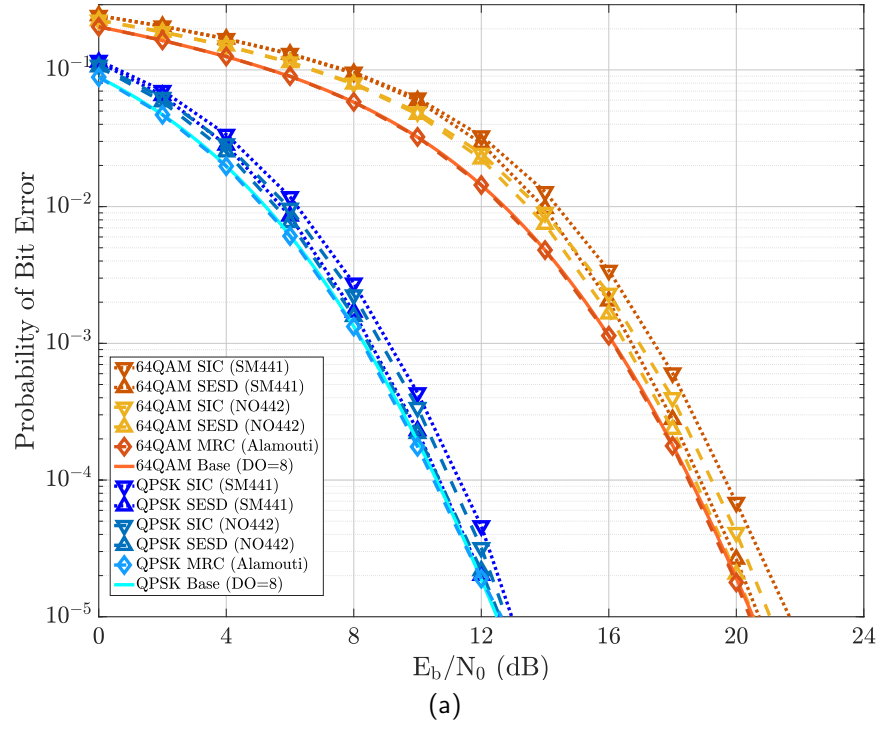


Figure 6.2. Comparison of decoding techniques for SM441, NO442, and Alamouti STCs using various modulations and DO of eight. (a) QPSK and 64QAM. (b) 16QAM and 256QAM.

of the GNU Radio flowgraph. The frame structure employed for OTA testing is shown in Fig. 6.3, where each subframe contains sixty data symbols and four pilot symbols, denoted as “P,” used for frequency synchronization. The training sequence at the beginning of each frame is utilized for CSI estimation.

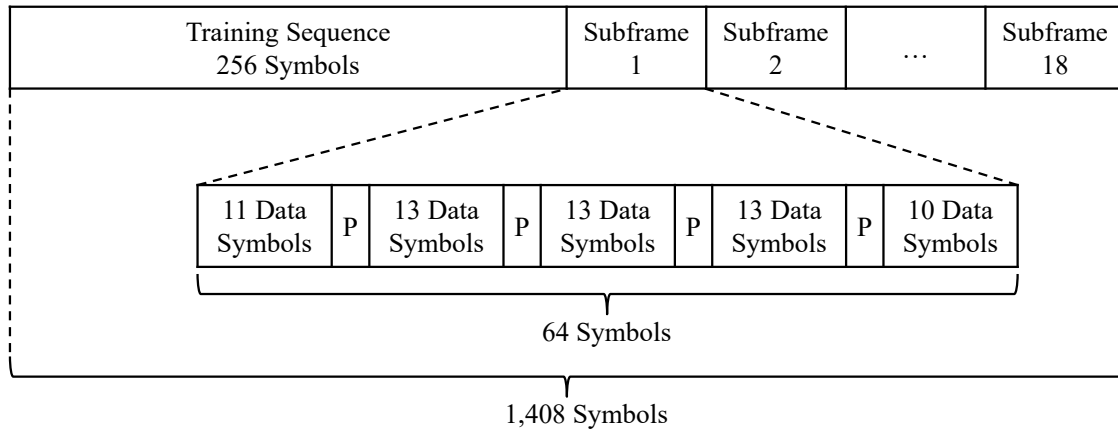


Figure 6.3. Illustration of OTA frame structure.

This setup allowed for initial OTA validation using Universal Software Radio Peripherals (USRPs) at a lower center frequency and sample rate; however, it was desired to perform final OTA testing within the 5G NR FR2 n258 band with a center frequency of 26.5 GHz [73]. To meet these requirements, in-phase and quadrature (IQ) samples from the GNU Radio flowgraph were captured prior to being sent to the USRP RF frontend as shown in Fig. 6.4.

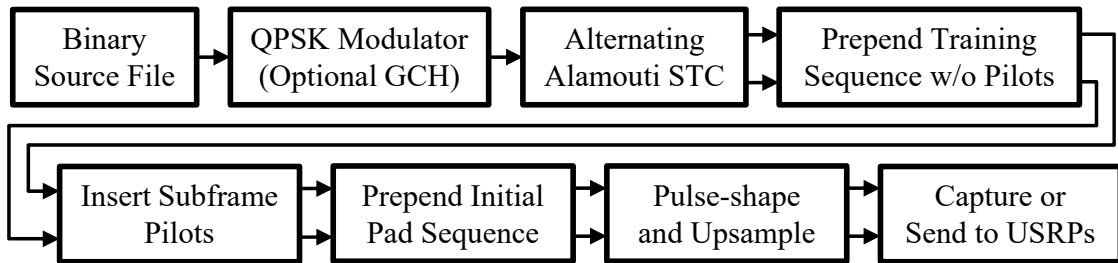


Figure 6.4. Block diagram illustration of GNU Radio flowgraph for transmission.

These samples represent multiple frames of data plus an initial pad sequence that have been pulse-shaped and upsampled using a root-raised cosine pulse with rolloff factor of

0.2 and upsampling factor of 2. The IQ data was then converted into arbitrary waveform files (ARBs) and loaded into two vector signal generators (VSGs) with horn antennas that could support the desired center frequency and sample rate. Together, these VSGs served as the BS. Each VSG contained an ARB with the IQ data corresponding to a single TX antenna data stream. The VSGs were operated in a master/slave configuration and set to continuously repeat the loaded ARBs when triggered by the master VSG.

A single spectrum analyzer with horn antenna was employed to serve as the UE in a multiple-input single-output (MISO) system. For simplicity, the spectrum analyzer was set to record IQ data when triggered by the master VSG; thus, coarse timing synchronization was ensured. The MISO test setup is shown in Fig. 6.5.

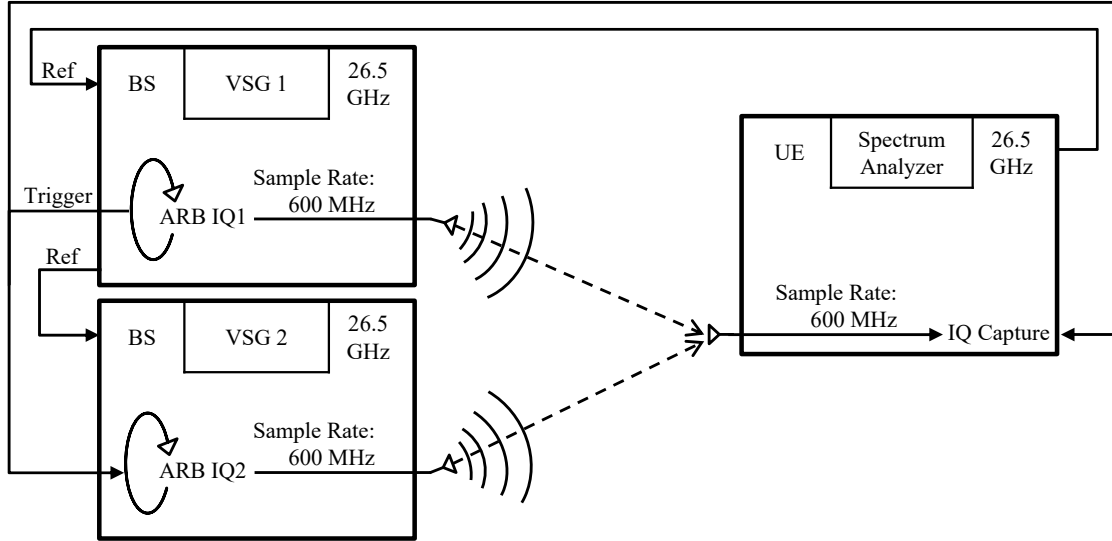


Figure 6.5. Illustration of MISO OTA test setup.

After IQ data was captured by the spectrum analyzer, it was exported for post-processing within GNU Radio and MATLAB. The GNU Radio flowgraph depicted in Fig. 6.6 was used to perform all synchronization and CSI estimation processes before exporting the received sample vector, $\mathbf{Z}$, and channel tap vector, $\mathbf{h}_{BU}$, from (3.8) to MATLAB for further analysis. MRC as described in Section 3.2.2 was performed in MATLAB to determine BER results.

In order to anchor BER to E_b/N_0 , observed SNR was recorded from the spectrum analyzer. A screen capture of one SNR measurement of 17.02 dB is shown in Fig. 6.7.

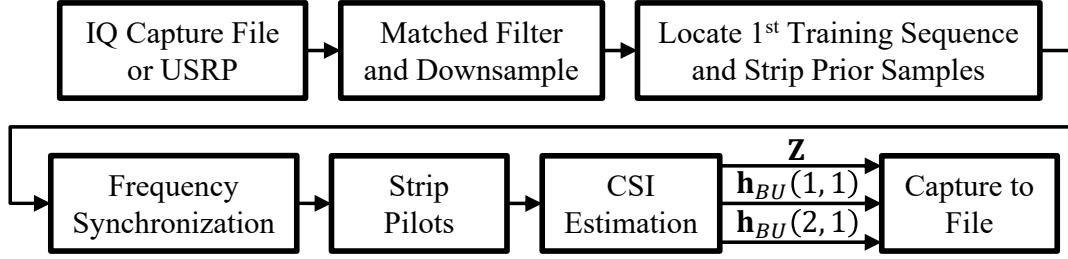


Figure 6.6. Block diagram illustration of GNU Radio flowgraph for reception and data export.

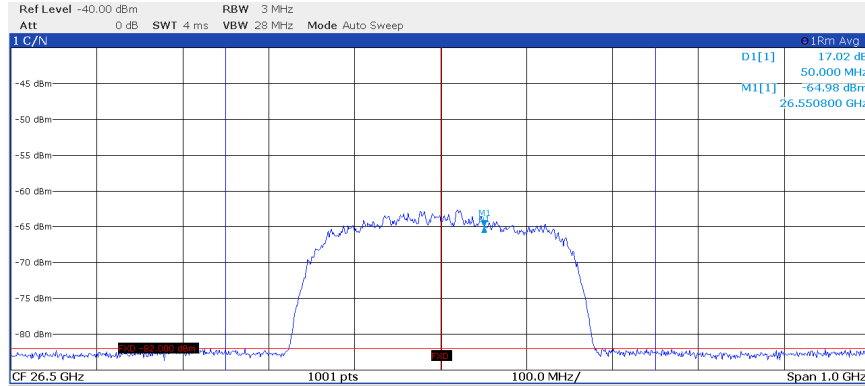


Figure 6.7. OTA testing SNR measurement.

Due to the choice of modulation, STC, and upsampling factor, the SNR is equal to E_b/N_0 in this case. The observed BER results during OTA testing are given in Fig. 6.8. A baseline performance curve showing theoretical results over a Rayleigh fading channel with DO of two is provided as a reference. A theoretical AWGN curve is also included to serve as a LB. In Fig. 6.8, it can be seen that the BER curve obtained through OTA testing lies between these two theoretical curves as expected.

All of the final OTA test parameters are given in Table 6.1 along with calculation of data rate.

6.2 Evaluation of Key Residue Class Decoding

In Section 5.2.2, after performing the initial four steps of the KRC decoding attack, analysis continues by making the worst-case assumption that Eve is able to successfully determine

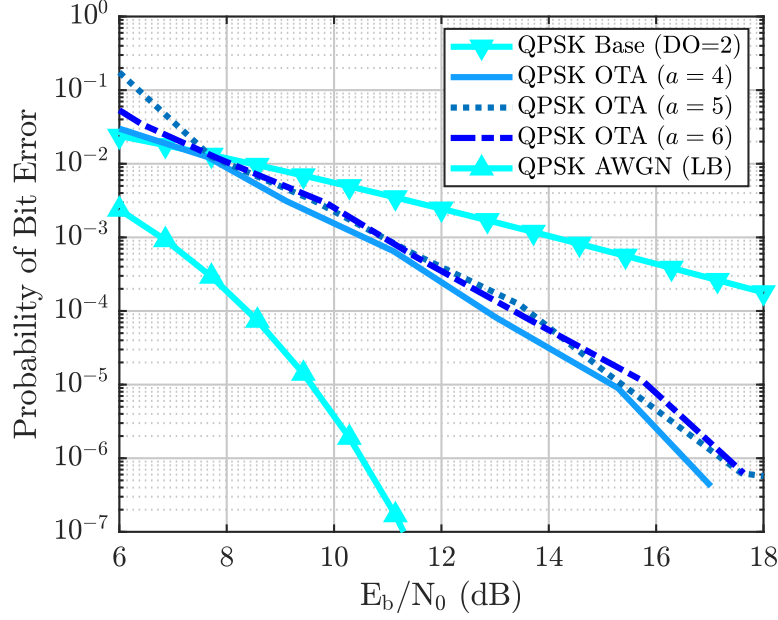


Figure 6.8. OTA primary channel BER results for various values of a with baseline and LB theoretical curves.

the correct KRC for each codeword. We now look at both simulated and OTA results for QPSK with various values of a to better evaluate if this worst case is truly achievable.

We determine the probability of correct KRC determination, under specified conditions, by measuring the percentage of codewords for which Eve could successfully determine the correct KRC index, denoted as i_c , during step 3 of the KRC decoding attack. Eve is considered to be successful if $i = i_c$. This measurement is possible by applying eavesdropper assumptions from Section 5.2.1 along with knowledge of the correct KRC based on the STC employed for each codeword.

This probability is calculated for Monte Carlo simulations of a MISO system employing the alternating STC PLS scheme with $\mathbf{v}_{obs} = [\text{Alamouti}]$, QPSK modulation, and $a = 4, 5$, and 6 for various values of E_b/N_0 . The results of these simulations are provided in Fig. 6.9.

As the value of a increases and more PR angles are permitted, the number of KRCs increases as shown in Table 5.4. Whereas the possible data symbol vectors remains constant for a fixed STC and modulation, this increase in the number of KRCs allows for noise to have a

Table 6.1. OTA test parameters.

Parameter	Value
a	{4, 5, 6}
$\mathbf{v}_{obs}$	[Alamouti]
$\mathbf{v}_{bsi}$	[1]
$\mathbf{v}_{tbsi}$	[1]
$\mathbf{v}_{toso}$	[2 1 7 4 5]
PRNG Seed	300
Center Frequency	26.5 GHz
Sample Rate	600 MHz
Capture Time (milliseconds)	12.1
Captured Samples	7,259,968
Pulse Shape	Root-Raised Cosine
Upsampling Factor	2
Captured Symbols	3,629,984
Initial Pad Overhead (Symbols)	500,000
Frame Length (Symbols)	1,408
Captured Frames	2,223
Data per Frame (Symbols)	1,080
Captured Data (Symbols)	2,400,840
Captured STC Codewords	1,200,420
Modulation	QPSK
Captured Data (bits)	4,801,680
Data Rate (bits per second)	396,833,058
Data Rate (megabytes per second)	47.3

greater impact on KRC determination. Thus, an increase in E_b/N_0 is expected to be required to obtain the same probability of correct KRC determination as a increases. It can be seen from the simulated results in Fig. 6.9, that indeed approximately 6 dB of additional E_b/N_0

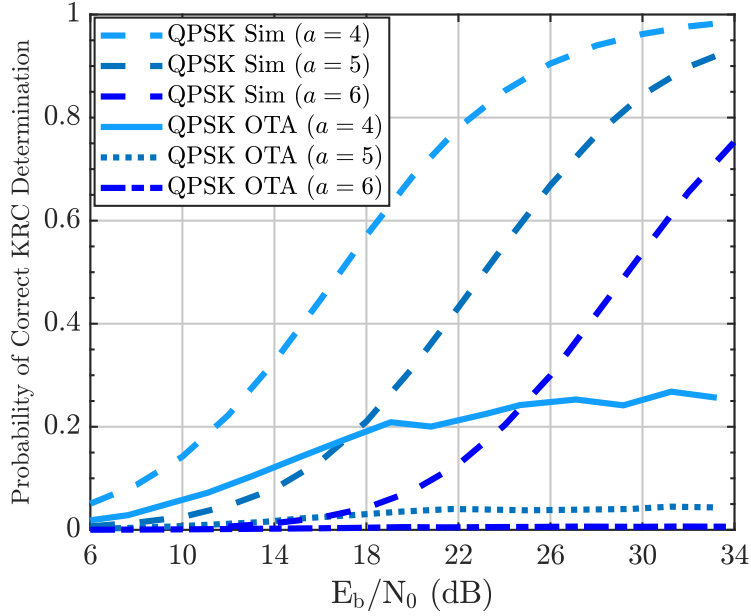


Figure 6.9. Probability of correct KRC determination when employing the Alamouti STC with QPSK and various values of a based on Monte Carlo simulations and OTA testing of a MISO system.

is required with each unit increase in a to achieve the same probability of determining the correct KRC. From these simulations, the likelihood of achieving the assumption of correct KRC determination for every codeword is very low unless a is small and/or the E_b/N_0 observed by Eve is very high.

The same measurement is repeated for OTA tests discussed in the previous section. The results of these tests are also provided in Fig. 6.9. It can be seen that the likelihood of achieving the assumption of correct KRC determination is even lower for these OTA conditions than with the Monte Carlo simulations.

It was desired to understand whether or not application of additional RX antennas by Eve could improve this likelihood; thus, Monte Carlo simulations were repeated for the $a = 4$ and $a = 6$ cases with $N_R = 2, 3$, and 4. Results of these simulations are provided in Fig. 6.10. It can be seen that likelihood improves by approximately 2-3 dB by utilizing two RX antennas, but application of more than two does not significantly improve results with diminishing returns with each additional antenna.

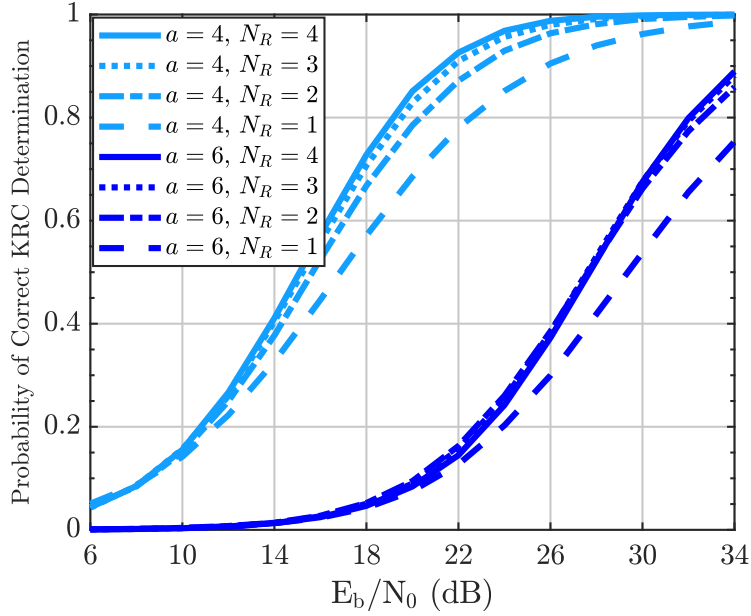


Figure 6.10. Probability of correct KRC determination when employing the Alamouti STC with QPSK for various values of a and N_R based on Monte Carlo simulations.

6.2.1 Extended Analysis

A natural extension of this analysis is to determine how many KRC decoding results have lesser Euclidean distances to a potential data symbol vector than that of the correct KRC on average. To facilitate this determination, we start by creating a set of Euclidean distances between each KRC decoding result and corresponding nearest data symbol vector. This set is then sorted in ascending order. A random variable, $x_c \in \{1, \dots, l\}$, is defined as the rank of the distance for the correct KRC within the set, where l is the set size corresponding to the number of KRCs. Thus, the distance of the correct KRC decoding result to nearest symbol vector has rank 1 (i.e., $x_c = 1$) if it has the minimum distance to nearest symbol vector of all KRC decoding results.

To obtain samples of this random variable for the completed Monte Carlo simulations and OTA tests with $N_R = 1$, an alternative step 3 of the KRC decoding attack is performed as follows:

Step 3a) Sort the index values of the l KRCs in ascending order by Euclidean distance of

the corresponding decoding result, $\hat{\mathbf{s}}_i$ for $i \in \{1, \dots, l\}$, to nearest potential data symbol vector, $\mathbf{s} \in \mathbb{M}^k$. Start by finding these distances, denoted as $\mathbf{D}(i)$, for each KRC decoding result represented as

$$\mathbf{D}(i) = \min_{\mathbf{s} \in \mathbb{M}^k} d(\mathbf{s}, \hat{\mathbf{s}}_i). \quad (6.1)$$

Then sort the elements of $\mathbf{D}$ using the sort function, as described in Section 2.1.3, to obtain the index values in ascending order of distance to nearest symbol vector [55]. This is represented as $[\sim, \mathbf{I}] = \text{sort}(\mathbf{D})$.

For each codeword, the index value of $\mathbf{I}$ corresponding to the correct KRC index, i_c , is a sample of x_c (i.e., $\mathbf{I}(x_c) = i_c$). The expected value of x_c minus one provides the number of KRC decoding results that have lesser distances than that of the correct KRC on average. Thus, $E[x_c]$ is calculated empirically for the completed simulations and OTA tests with Alamouti STC and $N_R = 1$. Results are shown in Fig. 6.11, where it can be seen that $E[x_c]$ is significantly greater than one at low E_b/N_0 . Whereas all simulations converge to an expected value of one as E_b/N_0 increases, OTA tests with $a = 5$ and $a = 6$ never approach this value.

The impact of this finding is a modification to key equivocation upon completion of the KRC decoding attack. In Section 5.2.2, this uncertainty was given based on the assumption that the correct KRC is obtained; however, this evaluation invalidates that assumption. The revised key equivocation is based on completion of the four steps given in Section 5.2.2 with modified step 3a to obtain $\mathbf{I}$. The expected value of key equivocation is dependent on the KRC size as given in (5.10) and $E[x_c]$ and is represented as

$$E[H(\mathbf{G}|\mathbf{I})] = \log_2(E[x_c]) + \log_2(2^{k(\min(a, 2)+1)} \cdot k!). \quad (6.2)$$

Empirically determined values for $E[x_c]$ are inserted into (6.2) with resulting values of $E[H(\mathbf{G}|\mathbf{I})]$ shown in Fig. 6.12. It can be seen from this figure that the initial value of key equivocation from (5.28) (i.e., $H(\mathbf{G}|\mathbf{G}_{KR}) = 7$ bits) serves as a LB that is approached as E_b/N_0 increases for the simulated cases; whereas the OTA test cases never reach this LB.

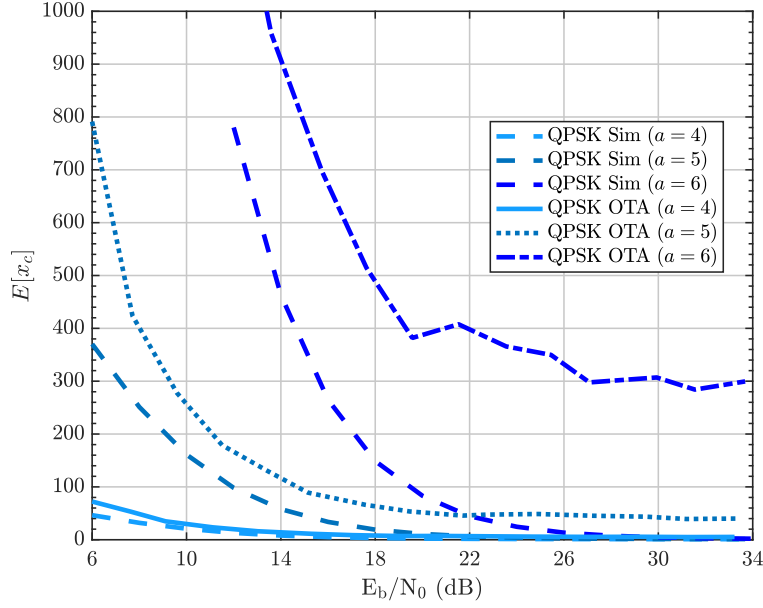


Figure 6.11. Expectation of x_c based on simulations and OTA testing.

6.2.2 Alternative Perspective

Another way of looking at this extension from Eve's perspective is to determine how many KRCs would need to be searched, in minimum distance order, to have a certain probability that the correct KRC is within the set of searched KRCs. To facilitate this determination, we look at the cumulative distribution function of x_c , defined as

$$F_{x_c}(x) = P(x_c \leq x). \quad (6.3)$$

For the Monte Carlo simulations and OTA tests at E_b/N_0 values nearest 32 dB, the cumulative distribution functions of x_c are provided in Figs. 6.13 and 6.14, respectively.

With matching parameters and channel conditions, these functions can be used to determine how many KRCs would need to be searched by Eve to have a certain probability that the correct KRC is within the set of searched KRCs. Under specified conditions for the Monte Carlo simulations with $a = 4, 5$, and 6 , Eve would need to search the first 2, 5, and 25 KRC indices from $\mathbf{I}$, respectively, to have a 99 percent probability that the correct KRC is within the searched KRCs. On the other hand, from the conditions observed during the OTA tests

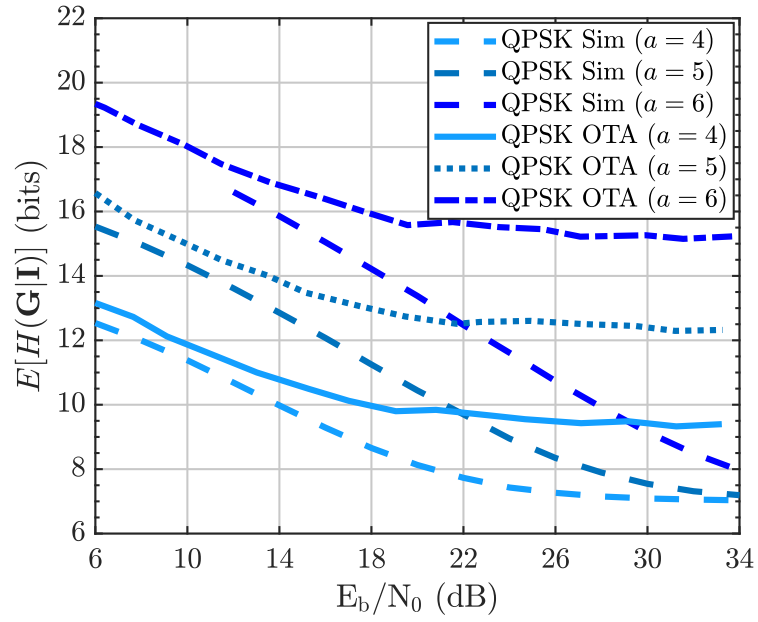


Figure 6.12. Expected value of key equivocation based on simulations and OTA testing.

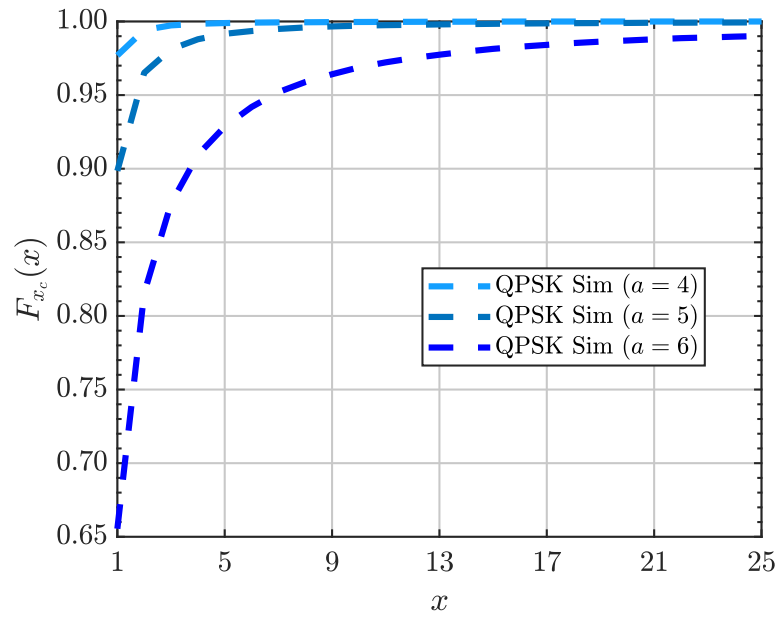


Figure 6.13. Cumulative distribution function of x_c based on Monte Carlo simulations at E_b/N_0 of 32 dB.

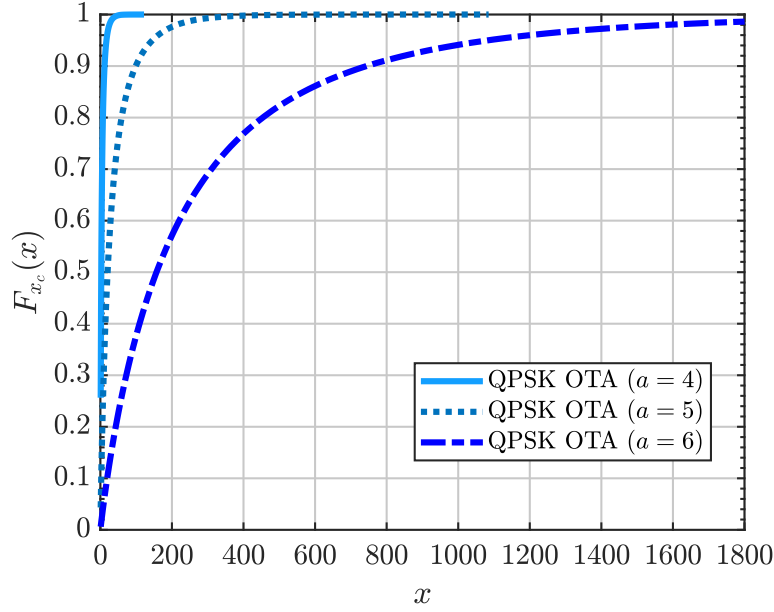


Figure 6.14. Cumulative distribution function of x_c based on OTA testing at E_b/N_0 values nearest 32 dB.

with $a = 4, 5$, and 6 , Eve would need to search the first 32, 270, and 1987 KRC indices from $\mathbf{I}$, respectively, to have the same probability.

6.3 Gray Code Hopping Bit Error Rate Comparison

As the proposed GCH PLS scheme is similar to other constellation remapping techniques presented in Section 1.1.2, we now provide a comparison between GCH and CD-PHY [18].

To illustrate the benefit of using GCH over CD-PHY, Monte Carlo simulations are performed employing each of the baseline mappings from Section 2.2.1 with results provided in Fig. 6.15. For all simulations, a Rayleigh fading channel with AWGN is represented. Perfect CSI is assumed between BS and UE, and all assumptions about Eve provided in Section 5.3 are enforced. For the purposes of comparison, the mapping employed by CD-PHY is alternated as with GCH but with the larger set of all $M!$ alternative mappings available to CD-PHY [18]. A baseline performance curve showing theoretical results for each modulation is provided as a reference.

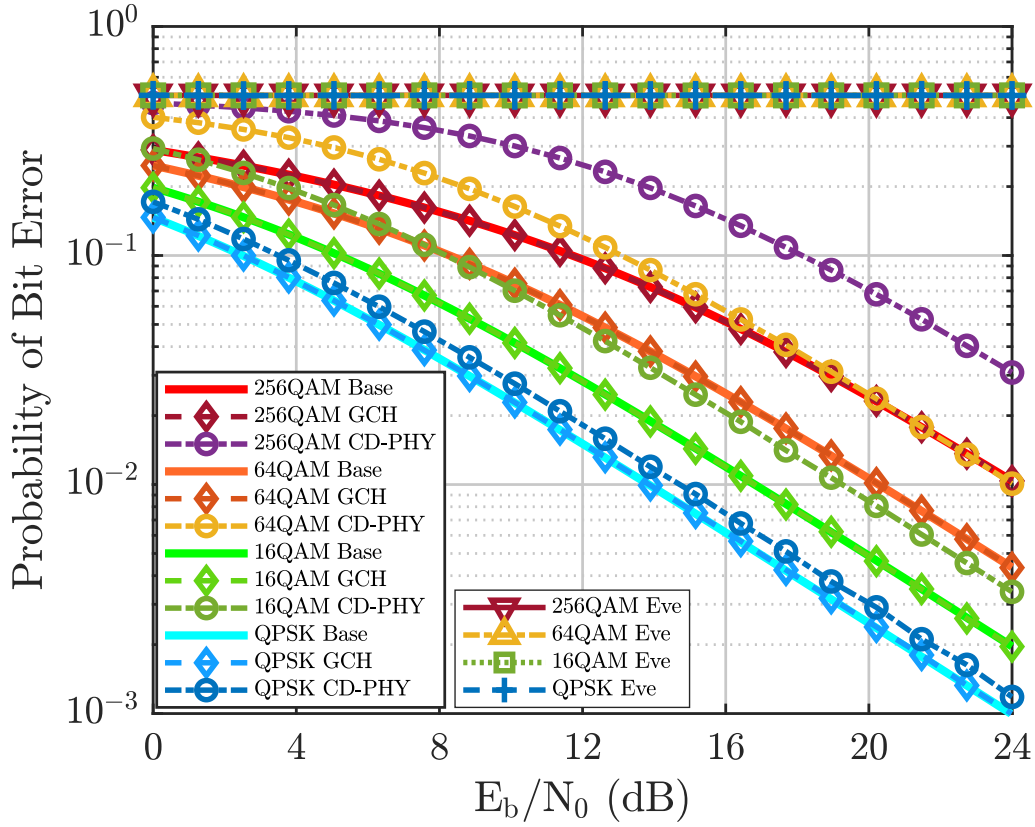


Figure 6.15. Theoretical baseline, GCH, alternating CD-PHY, and eavesdropper BER comparison.

In Fig. 6.15, it can be seen that no BER degradation occurs between BS and UE when employing the proposed GCH scheme as only BRGC mappings are employed. Conversely, performance loss increases with modulation order when employing an alternating version of CD-PHY. This behavior is expected as there are more non-BRGC mappings available with higher order modulations, and CD-PHY does not restrict the set of mappings for use as does GCH. For any incorrect ML decision, more bit errors are expected to occur when a non-BRGC mapping is employed by the CD-PHY scheme as opposed to a BRGC mapping used by GCH.

When employing either of these two PLS schemes, Eve makes a random guess at the mapping in use and obtains an average BER of 0.5 regardless of the modulation type.

6.4 Gray Code Hopping Hard versus Soft Decision Decoding

As stated in Section 3.4.2, the equivalent GCH demodulator may be used regardless of whether hard or soft decision decoding is employed. To confirm this statement, Monte Carlo simulations are performed over an AWGN channel using the MATLAB 5G toolbox implementation of LDPC encoding and decoding with 64QAM symbols [71]. Simulations are repeated using hard and soft decision decoding, with and without incorporation of the equivalent two-step method representation of the proposed GCH modulator and demodulator. Results are provided in Fig. 6.16. It can be seen that GCH results in no BER performance degradation regardless of employed decoding method as expected.

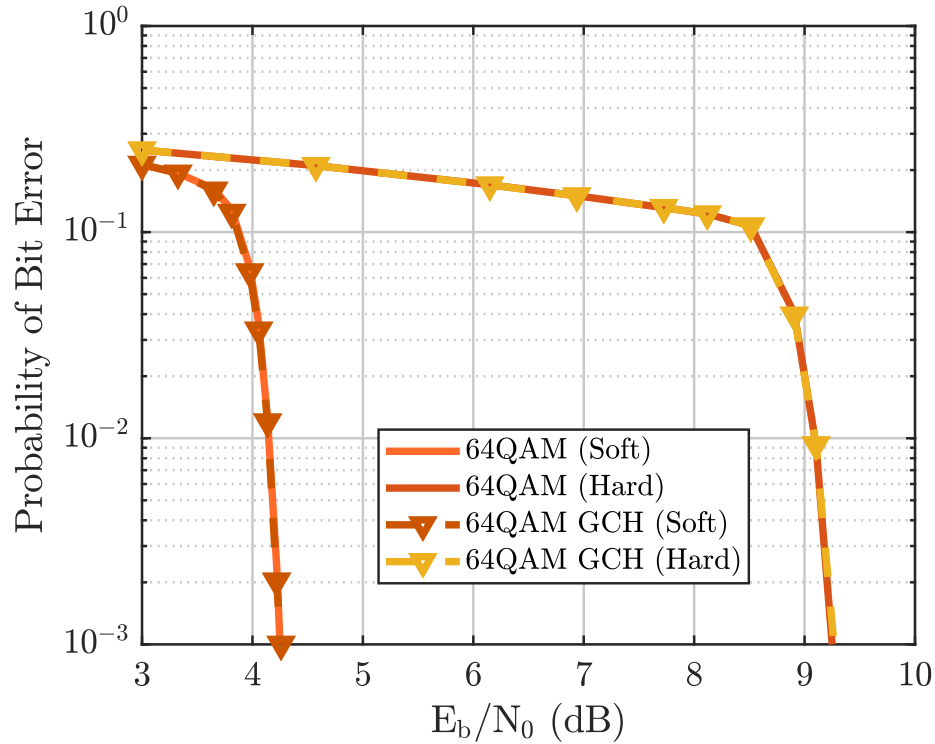


Figure 6.16. BER comparison of LDPC encoded 64QAM using hard and soft decision decoding with and without GCH.

THIS PAGE INTENTIONALLY LEFT BLANK

CHAPTER 7:

Conclusion

In this dissertation, we started by reviewing several different types of PLS from the literature including those with and without elements of computational security. Based on our literature review, we identified that no existing constellation remapping approach utilizes all alternative BRGC mappings to maximize the uncertainty of an eavesdropper while maintaining BER performance of the intended receiver. Additionally, we found that none of the ST coding-based PLS schemes make use of all available degrees of freedom within the STC structure to maximize security. Within this solution space, we developed a tailorable PLS framework including optional techniques to alternate the STC and/or BRGC modulation mapping in use over a wireless communications link.

In support of this framework, we presented implementation details including set building algorithms, transmission and reception support methods, and single- versus multi-STC alternation methods. The proposed solution was evaluated through information-theoretic and computational security analysis, Monte Carlo simulations, and OTA testing. Analysis showed that exhaustive key search is the least complex attack method for this PLS framework based on findings that the alternative KRC decoding attack is unlikely to succeed and requires considerably more computational time and effort when the STC in use alternates as proposed.

This chapter includes adaptations from previous work already published or in publication by the author. Specifically, portions of Section 7.1 are revised from [9]–[11].

7.1 Significant Contributions

Here we present a brief summary of the significant contributions of this dissertation.

7.1.1 Space-Time Code Sets

Whereas other ST coding-based PLS schemes have applied PRs across the TX antenna data streams or columns of the STC matrix [25], [26], no existing approach has utilized

these column PRs in conjunction with row and symbol PRs; row, column, and symbol permutations; and symbol conjugations. This dissertation provided a set of algorithms to systematically implement these operations to build extremely large sets of unique alternative STCs.

7.1.2 Individual Space-Time Code Building

Traditional schemes often perform beamforming, precoding, or encoding by selecting a code for use from a known codebook stored as a LUT [41], [58]; however, this practice becomes prohibitive as set cardinalities exceed device storage capacity. To avoid this problem, a method was presented to build specific selected STCs from the codebook for use as needed without storing the entire codebook.

7.1.3 Decoding Methods for Alternating Space-Time Codes

Orthogonal, non-orthogonal, and SM STCs employ different decoding methodologies, but each relies on a known code structure to properly perform channel equalization, co-phase symbol copies as necessary, and sort data symbols into the correct location before demodulation of a received sample array. We presented adaptive techniques to perform these functions based on a STC that changes with each codeword. Specifically, an algorithm was given allowing for the intended receiver to update the combining matrix associated with the alternating STC. A sufficient statistic for MRC of all orthogonal STC sets was presented with proof shown in Appendix A. Monte Carlo simulations of these techniques provided insight that SQRD and SIC yield adequate results with non-orthogonal STCs at low combined DOs and with SMSTCs at higher combined DOs. SD provides ML results but with increased complexity.

7.1.4 Novel Gray Code Hopping Technique

Whereas many PLS schemes modify the bit-to-symbol mapping of common modulation constellations, no existing approach utilizes all alternative BRGC mappings to maximize the uncertainty of an eavesdropper while maintaining BER performance of the intended receiver. This dissertation presented a modulation-based PLS scheme called GCH that meets these requirements. Two equivalent implementations of a GCH modulator and demodulator

were presented offering flexibility and simplicity. This allows for GCH to be readily implemented in adaptive 5G NR modulation coding schemes. Monte Carlo simulations confirmed compatibility of GCH with either hard or soft decision decoding methods.

7.1.5 Proposed Tailorable Physical Layer Security Framework

A framework was established for incorporation of one or both optional three phase PLS techniques presented in this dissertation. Alternating STC may be used in any MIMO system with choice of orthogonal, non-orthogonal, or SM STCs to achieve the desired diversity-multiplexing balance in addition to demonstrated PLS benefits. GCH may be used in any system employing common digital modulation schemes to provide PLS without sacrificing BER performance. The two techniques may be combined to provide excellent information-theoretic and computational security.

7.1.6 Adaptation of Security System Nomenclature

Nomenclature common to secrecy systems had not previously been applied to ST coding-based PLS schemes. We adapted definitions of cryptograms, KR_s, KRC_s, and message and cryptogram residue classes for this purpose. These definitions were utilized to provide information-theoretic security analysis of the proposed alternating STC scheme. It was shown that our message and cryptogram residue classes created when employing only a single KRC meet the properties for those of a pure cipher system in [13, Theorem 3].

7.1.7 Over-The-Air Testing of Alternating Space-Time Code Physical Layer Security

The data transmission phase of the proposed alternating STC PLS scheme was implemented first in open source software called GNU Radio and finally tested OTA in a MISO system using two VSGs and a single spectrum analyzer to serve as the BS and UE, respectively. The Alamouti STC was employed with QPSK modulation at a center frequency of 26.5 GHz utilizing 300 MHz bandwidth and achieving a data rate of 47.3 megabytes per second. A BER performance curve was established demonstrating an observed channel within the range of AWGN and Rayleigh fading.

7.1.8 Comparison of Alternative Attack Methods

KRC decoding and exhaustive key search attack methods were compared for the alternating STC scheme. KRC decoding was performed against Monte Carlo simulations and OTA test results proving that determination of the correct KRC for every codeword of a recorded communications session is highly unlikely. Exhaustive key search was shown to be the least complex attack against both PLS schemes presented.

7.2 Future Work

In our discussion of base code permutations in Section 4.1.6, it was mentioned that complexity of the exhaustive key search attack for a multi-STC alternation scheme could potentially be increased by introducing a base code order vector as an additional SIM unique content. In addition to this modification specific to multi-STC implementations, our discussion of the exhaustive key search attack complexity of all systems employing the alternating STC PLS technique was based on a reasonable assumption that could be more extensively validated. Specifically, it was assumed that every unique combination of session variables and PRNG state at the beginning of the data transmission phase yields a unique sequence of alternative STCs for a communications session of a given length.

Evaluation of the exhaustive key search attack complexity when incorporating GCH within an adaptive modulation coding scheme where the modulation order changes throughout a communications session based on channel conditions was not completed. Similar to the alternating STC attack, our discussion of the exhaustive key search attack complexity of all GCH implementations was based on a reasonable assumption that could be more extensively validated. Specifically, it was assumed that every unique combination of session variables and PRNG state at the beginning of the data transmission phase yields a unique sequence of mappings for a communications session of a given length.

General purpose processors were used to implement the data transmission phase of alternating STC. For the Alamouti STC with $a = 4$, an average of nearly 100 nanoseconds was required to build each individual alternative STC. If this technique were executed in realtime, then data rate for QPSK would be limited to 4.8 megabytes per second at this rate of Alamouti STC generation. A field programmable gate array implementation would likely be able to improve the speed at which individual STCs are built for use with this technique.

Alternatively, algorithm optimization may be able to provide necessary speed-ups. One possible optimization would be to apply all symbol algorithms to the data symbol vector instead of the STC.

Although multi-STC alternation schemes were discussed in general, it may be desirable to devise a system to balance TX diversity versus bandwidth efficiency demands through a selection bias for each of the STCs provided in a multi-STC implementation of this scheme. This system could be adaptive in order to update the bias within a session; thus, if frame error rates begin to rise, the bias towards an orthogonal STC could be increased to compensate. Likewise, if frame error rates begin to drop, the bias towards an SMSTC could be increased to improve bandwidth efficiency. This system would obviously require a method for both the BS and UE to adjust biases in the same manner with or without feedback.

Finally, additional modes of operation for the alternating STC scheme could be researched. At present, the data symbol vector contains the transmitted message; however, a pre-determined data symbol vector could be employed instead, creating a known message residue class. The next set of data bits to be transmitted could then be used to index into this message residue class to retrieve a data symbol vector to be embedded within the next codeword. Thus, to decode the message, UE would use the next alternating STC to retrieve the embedded symbol vector. The index of this symbol vector within the known message residue class would be the received message.

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX A:

Proof of Maximal Ratio Combining Sequence

In this appendix, we prove the MPO334 STC to be compatible with the two-step MRC sequence given in Section 3.2.2 along with Algorithm S8 given in Section 3.2.3.

This appendix includes adaptations from previous work already published or in publication by the author. Specifically, portions are revised from [9] and [10].

For a MISO system employing the MPO334 STC represented as

$$\mathbf{G} = \begin{bmatrix} s_1 & s_2 & \frac{s_3}{\sqrt{2}} \\ -s_2^* & s_1^* & \frac{s_3}{\sqrt{2}} \\ \frac{s_3^*}{\sqrt{2}} & \frac{s_3^*}{\sqrt{2}} & \frac{(-s_1 - s_1^* + s_2 - s_2^*)}{2} \\ \frac{s_3^*}{\sqrt{2}} & \frac{-s_3^*}{\sqrt{2}} & \frac{(s_2 + s_2^* + s_1 - s_1^*)}{2} \end{bmatrix}, \quad (\text{A.1})$$

with channel tap vector, $\mathbf{h}_{BU} = [h_1 \ h_2 \ h_3]^T$, Algorithm S8 is performed to construct

$$\mathbf{H}_{CBU} = \begin{bmatrix} h_1 & h_2 & \frac{h_3}{\sqrt{2}} & 0 & 0 & 0 \\ 0 & 0 & \frac{h_3}{\sqrt{2}} & h_2 & -h_1 & 0 \\ \frac{-h_3}{2} & \frac{h_3}{2} & 0 & \frac{-h_3}{2} & \frac{-h_3}{2} & \frac{(h_1 + h_2)}{\sqrt{2}} \\ \frac{h_3}{2} & \frac{h_3}{2} & 0 & \frac{-h_3}{2} & \frac{h_3}{2} & \frac{(h_1 - h_2)}{\sqrt{2}} \end{bmatrix}. \quad (\text{A.2})$$

To confirm the sufficient statistic, (3.14) is calculated first to obtain

$$\begin{aligned}
\mathbf{H}_{int} &= \mathbf{H}_{CBU}^\dagger \times \mathbf{H}_{CBU} \\
&= \begin{bmatrix} h_1^* & 0 & \frac{-h_3^*}{2} & \frac{h_3^*}{2} \\ h_2^* & 0 & \frac{h_3^*}{2} & \frac{h_3^*}{2} \\ \frac{h_3^*}{\sqrt{2}} & \frac{h_3^*}{\sqrt{2}} & 0 & 0 \\ 0 & h_2^* & \frac{-h_3^*}{2} & \frac{-h_3^*}{2} \\ 0 & -h_1^* & \frac{-h_3^*}{2} & \frac{h_3^*}{2} \\ 0 & 0 & \frac{(h_1^*+h_2^*)}{\sqrt{2}} & \frac{(h_1^*-h_2^*)}{\sqrt{2}} \end{bmatrix} \times \begin{bmatrix} h_1 & h_2 & \frac{h_3}{\sqrt{2}} & 0 & 0 & 0 \\ 0 & 0 & \frac{h_3}{\sqrt{2}} & h_2 & -h_1 & 0 \\ \frac{-h_3}{2} & \frac{h_3}{2} & 0 & \frac{-h_3}{2} & \frac{-h_3}{2} & \frac{(h_1+h_2)}{\sqrt{2}} \\ \frac{h_3}{2} & \frac{h_3}{2} & 0 & \frac{-h_3}{2} & \frac{h_3}{2} & \frac{(h_1-h_2)}{\sqrt{2}} \end{bmatrix} \\
\therefore \mathbf{H}_{int} &= \begin{bmatrix} h_1 h_1^* + \frac{h_3 h_3^*}{2} & h_1^* h_2 & \frac{h_1^* h_3}{\sqrt{2}} & 0 & \frac{h_3 h_3^*}{2} & -\frac{h_2 h_3^*}{\sqrt{2}} \\ h_1 h_2^* & h_2 h_2^* + \frac{h_3 h_3^*}{2} & \frac{h_2^* h_3}{\sqrt{2}} & -\frac{h_3 h_3^*}{2} & 0 & \frac{h_1 h_3^*}{\sqrt{2}} \\ \frac{h_1 h_3^*}{\sqrt{2}} & \frac{h_2 h_3^*}{\sqrt{2}} & h_3 h_3^* & \frac{h_2 h_3^*}{\sqrt{2}} & -\frac{h_1 h_3^*}{\sqrt{2}} & 0 \\ 0 & -\frac{h_3 h_3^*}{2} & \frac{h_2^* h_3}{\sqrt{2}} & h_2 h_2^* + \frac{h_3 h_3^*}{2} & -h_1 h_2^* & -\frac{h_1 h_3^*}{\sqrt{2}} \\ \frac{h_3 h_3^*}{2} & 0 & -\frac{h_1^* h_3}{\sqrt{2}} & -h_1^* h_2 & h_1 h_1^* + \frac{h_3 h_3^*}{2} & -\frac{h_2 h_3^*}{\sqrt{2}} \\ -\frac{h_2^* h_3}{\sqrt{2}} & \frac{h_1^* h_3}{\sqrt{2}} & 0 & -\frac{h_1^* h_3}{\sqrt{2}} & -\frac{h_2^* h_3}{\sqrt{2}} & h_1 h_1^* + h_2 h_2^* \end{bmatrix}.
\end{aligned} \tag{A.3}$$

Then, (3.13) is used iteratively for $x, y \in \{1, \dots, k\}$ to confirm (3.12).

With $k = 3, x = 1$, and $y = 1$,

$$\begin{aligned}
\hat{\mathbf{H}}(1, 1) &= \mathbf{H}_{int}(1, 1) + \mathbf{H}_{int}^*(4, 4) \\
&\quad + \mathbf{H}_{int}(1, 4) + \mathbf{H}_{int}^*(4, 1) \\
&= h_1 h_1^* + \frac{h_3 h_3^*}{2} + h_2 h_2^* + \frac{h_3 h_3^*}{2} + 0 + 0 \\
&= \|\mathbf{h}_{BU}\|^2.
\end{aligned} \tag{A.4}$$

With $k = 3, x = 2$, and $y = 1$,

$$\begin{aligned}
\hat{\mathbf{H}}(2, 1) &= \mathbf{H}_{int}(2, 1) + \mathbf{H}_{int}^*(5, 4) \\
&\quad + \mathbf{H}_{int}(2, 4) + \mathbf{H}_{int}^*(5, 1) \\
&= h_1 h_2^* - h_1 h_2^* - \frac{h_3 h_3^*}{2} + \frac{h_3 h_3^*}{2} = 0.
\end{aligned} \tag{A.5}$$

With $k = 3$, $x = 3$, and $y = 1$,

$$\begin{aligned}
\hat{\mathbf{H}}(3, 1) &= \mathbf{H}_{int}(3, 1) + \mathbf{H}_{int}^*(6, 4) \\
&\quad + \mathbf{H}_{int}(3, 4) + \mathbf{H}_{int}^*(6, 1) \\
&= \frac{h_1 h_3^*}{\sqrt{2}} - \frac{h_1 h_3^*}{\sqrt{2}} + \frac{h_2 h_3^*}{\sqrt{2}} - \frac{h_2 h_3^*}{\sqrt{2}} = 0.
\end{aligned} \tag{A.6}$$

With $k = 3$, $x = 1$, and $y = 2$,

$$\begin{aligned}
\hat{\mathbf{H}}(1, 2) &= \mathbf{H}_{int}(1, 2) + \mathbf{H}_{int}^*(4, 5) \\
&\quad + \mathbf{H}_{int}(1, 5) + \mathbf{H}_{int}^*(4, 2) \\
&= h_1^* h_2 - h_1^* h_2 + \frac{h_3 h_3^*}{2} - \frac{h_3 h_3^*}{2} = 0.
\end{aligned} \tag{A.7}$$

With $k = 3$, $x = 2$, and $y = 2$,

$$\begin{aligned}
\hat{\mathbf{H}}(2, 2) &= \mathbf{H}_{int}(2, 2) + \mathbf{H}_{int}^*(5, 5) \\
&\quad + \mathbf{H}_{int}(2, 5) + \mathbf{H}_{int}^*(5, 2) \\
&= h_2 h_2^* + \frac{h_3 h_3^*}{2} + h_1 h_1^* + \frac{h_3 h_3^*}{2} + 0 + 0 \\
&= ||\mathbf{h}_{BU}||^2.
\end{aligned} \tag{A.8}$$

With $k = 3$, $x = 3$, and $y = 2$,

$$\begin{aligned}
\hat{\mathbf{H}}(3, 2) &= \mathbf{H}_{int}(3, 2) + \mathbf{H}_{int}^*(6, 5) \\
&\quad + \mathbf{H}_{int}(3, 5) + \mathbf{H}_{int}^*(6, 2) \\
&= \frac{h_2 h_3^*}{\sqrt{2}} - \frac{h_2 h_3^*}{\sqrt{2}} - \frac{h_1 h_3^*}{\sqrt{2}} + \frac{h_1 h_3^*}{\sqrt{2}} = 0.
\end{aligned} \tag{A.9}$$

With $k = 3$, $x = 1$, and $y = 3$,

$$\begin{aligned}
\hat{\mathbf{H}}(1, 3) &= \mathbf{H}_{int}(1, 3) + \mathbf{H}_{int}^*(4, 6) \\
&\quad + \mathbf{H}_{int}(1, 6) + \mathbf{H}_{int}^*(4, 3) \\
&= \frac{h_1^* h_3}{\sqrt{2}} - \frac{h_1^* h_3}{\sqrt{2}} - \frac{h_2 h_3^*}{\sqrt{2}} + \frac{h_2 h_3^*}{\sqrt{2}} = 0.
\end{aligned} \tag{A.10}$$

With $k = 3$, $x = 2$, and $y = 3$,

$$\begin{aligned}
\hat{\mathbf{H}}(2, 3) &= \mathbf{H}_{int}(2, 3) + \mathbf{H}_{int}^*(5, 6) \\
&\quad + \mathbf{H}_{int}(2, 6) + \mathbf{H}_{int}^*(5, 3) \\
&= \frac{h_2^* h_3}{\sqrt{2}} - \frac{h_2^* h_3}{\sqrt{2}} + \frac{h_1 h_3^*}{\sqrt{2}} - \frac{h_1 h_3^*}{\sqrt{2}} = 0.
\end{aligned} \tag{A.11}$$

With $k = 3$, $x = 3$, and $y = 3$,

$$\begin{aligned}
\hat{\mathbf{H}}(3, 3) &= \mathbf{H}_{int}(3, 3) + \mathbf{H}_{int}^*(6, 6) \\
&\quad + \mathbf{H}_{int}(3, 6) + \mathbf{H}_{int}^*(6, 3) \\
&= h_3 h_3^* + h_1 h_1^* + h_2 h_2^* + 0 + 0 = \|\mathbf{h}_{BU}\|^2.
\end{aligned} \tag{A.12}$$

Thus, (3.12) is validated, and the sufficient statistic is met for MRC.

For the communications link given in Section 3.2.1 with $N_R = 1$, the r -by-1 received sample vector is represented as

$$\mathbf{Z} = \mathbf{G} \times \mathbf{h}_{BU} + \mathbf{n} = \mathbf{H}_{CBU} \times \mathbf{s}_{ext} + \mathbf{n} \tag{A.13}$$

$$= \begin{bmatrix} s_1 & s_2 & \frac{s_3}{\sqrt{2}} \\ -s_2^* & s_1^* & \frac{s_3}{\sqrt{2}} \\ \frac{s_3^*}{\sqrt{2}} & \frac{s_3^*}{\sqrt{2}} & \frac{(-s_1 - s_1^* + s_2 - s_2^*)}{2} \\ \frac{s_3^*}{\sqrt{2}} & \frac{-s_3^*}{\sqrt{2}} & \frac{(s_2 + s_2^* + s_1 - s_1^*)}{2} \end{bmatrix} \times \begin{bmatrix} h_1 \\ h_2 \\ h_3 \end{bmatrix} + \begin{bmatrix} n_1 \\ n_2 \\ n_3 \\ n_4 \end{bmatrix}$$

$$\begin{aligned}
&= \begin{bmatrix} h_1 & h_2 & \frac{h_3}{\sqrt{2}} & 0 & 0 & 0 \\ 0 & 0 & \frac{h_3}{\sqrt{2}} & h_2 & -h_1 & 0 \\ \frac{-h_3}{2} & \frac{h_3}{2} & 0 & \frac{-h_3}{2} & \frac{-h_3}{2} & \frac{(h_1+h_2)}{\sqrt{2}} \\ \frac{h_3}{2} & \frac{h_3}{2} & 0 & \frac{-h_3}{2} & \frac{h_3}{2} & \frac{(h_1-h_2)}{\sqrt{2}} \end{bmatrix} \times \begin{bmatrix} s_1 \\ s_2 \\ s_3 \\ s_1^* \\ s_2^* \\ s_3^* \end{bmatrix} + \begin{bmatrix} n_1 \\ n_2 \\ n_3 \\ n_4 \end{bmatrix} \\
\therefore \mathbf{Z} &= \begin{bmatrix} s_1 h_1 + s_2 h_2 + \frac{s_3 h_3}{\sqrt{2}} + n_1 \\ \frac{s_3 h_3}{\sqrt{2}} + s_1^* h_2 - s_2^* h_1 + n_2 \\ -\frac{s_1 h_3}{2} + \frac{s_2 h_3}{2} - \frac{s_1^* h_3}{2} - \frac{s_2^* h_3}{2} + \frac{s_3^* (h_1+h_2)}{\sqrt{2}} + n_3 \\ \frac{s_1 h_3}{2} + \frac{s_2 h_3}{2} - \frac{s_1^* h_3}{2} + \frac{s_2^* h_3}{2} + \frac{s_3^* (h_1-h_2)}{\sqrt{2}} + n_4 \end{bmatrix}.
\end{aligned}$$

With this received vector, the first step of the MRC sequence given in (3.9) is calculated to obtain

$$\begin{aligned}
\hat{\mathbf{s}}_{int} &= \mathbf{H}_{CBU}^\dagger \times \mathbf{Z} \\
&= \begin{bmatrix} h_1^* & 0 & \frac{-h_3^*}{2} & \frac{h_3^*}{2} \\ h_2^* & 0 & \frac{h_3^*}{2} & \frac{h_3^*}{2} \\ \frac{h_3^*}{\sqrt{2}} & \frac{h_3^*}{\sqrt{2}} & 0 & 0 \\ 0 & h_2^* & \frac{-h_3^*}{2} & \frac{-h_3^*}{2} \\ 0 & -h_1^* & \frac{-h_3^*}{2} & \frac{h_3^*}{2} \\ 0 & 0 & \frac{(h_1^*+h_2^*)}{\sqrt{2}} & \frac{(h_1^*-h_2^*)}{\sqrt{2}} \end{bmatrix} \times \begin{bmatrix} s_1 h_1 + s_2 h_2 + \frac{s_3 h_3}{\sqrt{2}} + n_1 \\ \frac{s_3 h_3}{\sqrt{2}} + s_1^* h_2 - s_2^* h_1 + n_2 \\ -\frac{s_1 h_3}{2} + \frac{s_2 h_3}{2} - \frac{s_1^* h_3}{2} - \frac{s_2^* h_3}{2} + \frac{s_3^* (h_1+h_2)}{\sqrt{2}} + n_3 \\ \frac{s_1 h_3}{2} + \frac{s_2 h_3}{2} - \frac{s_1^* h_3}{2} + \frac{s_2^* h_3}{2} + \frac{s_3^* (h_1-h_2)}{\sqrt{2}} + n_4 \end{bmatrix}
\end{aligned} \tag{A.14}$$

$$\therefore \hat{\mathbf{s}}_{int} = \begin{bmatrix} s_1 h_1 h_1^* + s_2 h_1^* h_2 + \frac{s_3 h_1^* h_3}{\sqrt{2}} + n_1 h_1^* + \frac{s_1 h_3 h_3^*}{2} + \frac{s_2^* h_3 h_3^*}{2} - \frac{s_3^* h_2 h_3^*}{\sqrt{2}} - \frac{n_3 h_3^*}{2} + \frac{n_4 h_3^*}{2} \\ s_1 h_1 h_2^* + s_2 h_2 h_2^* + \frac{s_3 h_2^* h_3}{\sqrt{2}} + n_1 h_2^* + \frac{s_2 h_3 h_3^*}{2} - \frac{s_1^* h_3 h_3^*}{2} + \frac{s_3^* h_1 h_3^*}{\sqrt{2}} + \frac{n_3 h_3^*}{2} + \frac{n_4 h_3^*}{2} \\ \frac{s_1 h_1 h_3^*}{\sqrt{2}} + \frac{s_2 h_2 h_3^*}{\sqrt{2}} + s_3 h_3 h_3^* + \frac{n_1 h_3^*}{\sqrt{2}} + \frac{s_1^* h_2 h_3^*}{\sqrt{2}} - \frac{s_2^* h_1 h_3^*}{\sqrt{2}} + \frac{n_2 h_3^*}{\sqrt{2}} \\ \frac{s_3 h_2^* h_3}{\sqrt{2}} + s_1^* h_2 h_2^* - s_2^* h_1 h_2^* + n_2 h_2^* - \frac{s_2 h_3 h_3^*}{2} + \frac{s_1^* h_3 h_3^*}{2} - \frac{s_3^* h_1 h_3^*}{\sqrt{2}} - \frac{n_3 h_3^*}{2} - \frac{n_4 h_3^*}{2} \\ - \frac{s_3 h_1^* h_3}{\sqrt{2}} - s_1^* h_1^* h_2 + s_2^* h_1 h_1^* - n_2 h_1^* + \frac{s_1 h_3 h_3^*}{2} + \frac{s_2^* h_3 h_3^*}{2} - \frac{s_3^* h_2 h_3^*}{\sqrt{2}} - \frac{n_3 h_3^*}{2} + \frac{n_4 h_3^*}{2} \\ - \frac{s_1 h_2^* h_3}{\sqrt{2}} + \frac{s_2 h_1^* h_3}{\sqrt{2}} - \frac{s_1^* h_1^* h_3}{\sqrt{2}} - \frac{s_2^* h_2^* h_3}{\sqrt{2}} + s_3^* h_1 h_1^* + s_3^* h_2 h_2^* + \frac{n_3 (h_1^* + h_2^*)}{\sqrt{2}} + \frac{n_4 (h_1^* - h_2^*)}{\sqrt{2}} \end{bmatrix}.$$

The second step given in (3.10) is then used iteratively for $x \in \{1, \dots, k\}$ to obtain the estimated data symbol vector, $\hat{\mathbf{s}}$.

With $k = 3$ and $x = 1$,

$$\begin{aligned} \hat{\mathbf{s}}(1) &= \frac{\hat{\mathbf{s}}_{int}(1) + \hat{\mathbf{s}}_{int}^*(4)}{\|\mathbf{h}_{BU}\|^2} \\ &= s_1 + \frac{n_1 h_1^* + n_2^* h_2 - \frac{h_3^* (n_3 - n_4)}{2} - \frac{h_3 (n_3^* + n_4^*)}{2}}{\|\mathbf{h}_{BU}\|^2}. \end{aligned} \quad (\text{A.15})$$

With $k = 3$ and $x = 2$,

$$\begin{aligned} \hat{\mathbf{s}}(2) &= \frac{\hat{\mathbf{s}}_{int}(2) + \hat{\mathbf{s}}_{int}^*(5)}{\|\mathbf{h}_{BU}\|^2} \\ &= s_2 + \frac{n_1 h_2^* - n_2^* h_1 + \frac{h_3^* (n_3 + n_4)}{2} - \frac{h_3 (n_3^* - n_4^*)}{2}}{\|\mathbf{h}_{BU}\|^2}. \end{aligned} \quad (\text{A.16})$$

With $k = 3$ and $x = 3$,

$$\begin{aligned} \hat{\mathbf{s}}(3) &= \frac{\hat{\mathbf{s}}_{int}(3) + \hat{\mathbf{s}}_{int}^*(6)}{\|\mathbf{h}_{BU}\|^2} \\ &= s_3 + \frac{\frac{h_3^* (n_1 + n_2)}{\sqrt{2}} + \frac{h_1 (n_3^* + n_4^*)}{\sqrt{2}} + \frac{h_2 (n_3^* - n_4^*)}{\sqrt{2}}}{\|\mathbf{h}_{BU}\|^2}. \end{aligned} \quad (\text{A.17})$$

APPENDIX B: Additional Key Residues

Additional KR examples are included in this appendix to provide further clarification of the tabular and matrix form KRs with different base STCs.

This appendix includes adaptations from previous work already published or in publication by the author. Specifically, portions are revised from [9] and [10].

Recall the MPO334 variant code originally given in (3.18) as

$$\mathbf{G} = \begin{bmatrix} -s_1 & s_2 \cdot e^{j3\pi/4} & \frac{s_3 \cdot e^{j\pi/4}}{\sqrt{2}} \\ -s_2^* & s_1^* \cdot e^{j7\pi/4} & \frac{s_3}{\sqrt{2}} \\ \frac{s_3^*}{\sqrt{2}} & \frac{s_3^* \cdot e^{j\pi/2}}{\sqrt{2}} & \frac{(s_1 \cdot e^{j7\pi/4} + s_1^* \cdot e^{j\pi/4} + s_2 - s_2^*)}{2} \\ \frac{s_3}{\sqrt{2}} & \frac{s_3^* \cdot e^{j3\pi/2}}{\sqrt{2}} & \frac{(s_2 + s_2^* + s_1 \cdot e^{j3\pi/4} + s_1^* \cdot e^{j\pi/4})}{2} \end{bmatrix}.$$

The tabular form KR of (3.18) is provided in Table B.1. The matrix form KR of (3.18) is determined by phase rotating symbols s_1 and s_2 and conjugating symbols s_2 and s_3 . The matrix form KR is represented as

$$\mathbf{G}_{KR} = \begin{bmatrix} s_1 & s_2^* \cdot e^{j7\pi/4} & \frac{s_3^* \cdot e^{j\pi/4}}{\sqrt{2}} \\ s_2 & s_1^* \cdot e^{j3\pi/4} & \frac{s_3^*}{\sqrt{2}} \\ \frac{s_3}{\sqrt{2}} & \frac{s_3 \cdot e^{j\pi/2}}{\sqrt{2}} & \frac{(s_1 \cdot e^{j3\pi/4} + s_1^* \cdot e^{j5\pi/4} - s_2^* + s_2)}{2} \\ \frac{s_3}{\sqrt{2}} & \frac{s_3 \cdot e^{j3\pi/2}}{\sqrt{2}} & \frac{(-s_2^* - s_2 + s_1 \cdot e^{j7\pi/4} + s_1^* \cdot e^{j5\pi/4})}{2} \end{bmatrix}. \quad (\text{B.1})$$

Table B.1. Tabular form KR for the MPO334 variant code shown in (3.18).

SGI	Relationship	Location of Variant 1	Location of Variant 2
1.1.1	Conj. $e^{j3\pi/4}$	(1, 1)	(2, 2)
1.2.1	Conj. $e^{j7\pi/4}$	(2, 1)	(1, 2)
2.1.1	1	(3, 1)	(4, 1)
2.1.2	-1	(3, 2)	(4, 2)
3.1.1	Conj. $\frac{e^{j5\pi/4}}{2}$	(1, 1)	(3, 3)
3.2.1	Conj. $e^{j\pi/4}$	(3, 1)	(1, 3)
3.1.2	Conj. $\frac{e^{j5\pi/4}}{2}$	(1, 1)	(4, 3)
3.2.2	Conj. $e^{j\pi/4}$	(4, 1)	(1, 3)
4.1.1	Conj. $-1/2$	(2, 1)	(3, 3)
4.2.1	Conj.	(3, 1)	(2, 3)
4.1.2	Conj. $-1/2$	(2, 1)	(4, 3)
4.2.2	Conj.	(4, 1)	(2, 3)
5.1.1	1	(3, 1)	(4, 1)
5.2.1	-1	(3, 3)	(4, 3)
5.3.1	-1	(3, 3)	(4, 3)
6.1.1	Conj. $\frac{e^{j7\pi/4}}{2}$	(1, 2)	(3, 3)
6.2.1	Conj. $e^{j3\pi/4}$	(3, 2)	(1, 3)
6.1.2	Conj. $\frac{e^{j3\pi/4}}{2}$	(1, 2)	(4, 3)
6.2.2	Conj. $e^{j7\pi/4}$	(4, 2)	(1, 3)

Table B.1. (continued)

SGI	Relationship	Location of Variant 1	Location of Variant 2
7.1.1	Conj. $\frac{e^{j3\pi/2}}{2}$	(2, 2)	(3, 3)
7.2.1	Conj. $e^{j\pi/2}$	(3, 2)	(2, 3)
7.1.2	Conj. $\frac{e^{j\pi/2}}{2}$	(2, 2)	(4, 3)
7.2.2	Conj. $e^{j3\pi/2}$	(4, 2)	(2, 3)
8.1.1	-1	(3, 2)	(4, 2)
8.2.1	1	(3, 3)	(4, 3)
8.3.1	1	(3, 3)	(4, 3)
9.1.1	1	(1, 1)	(1, 1)
9.2.1	1	(2, 1)	(2, 1)
9.3.1	$\frac{1}{\sqrt{2}}$	(3, 1)	(3, 1)
9.3.2	$\frac{1}{\sqrt{2}}$	(4, 1)	(4, 1)

A variant of the MPO334 STC generated using row and column permutations alone is represented as

$$\mathbf{G} = \begin{bmatrix} \frac{(-s_1 - s_1^* + s_2 - s_2^*)}{2} & \frac{s_3^*}{\sqrt{2}} & \frac{s_3^*}{\sqrt{2}} \\ \frac{s_3}{\sqrt{2}} & -s_2^* & s_1^* \\ \frac{s_3}{\sqrt{2}} & s_1 & s_2 \\ \frac{(s_2 + s_2^* + s_1 - s_1^*)}{2} & \frac{s_3^*}{\sqrt{2}} & \frac{-s_3^*}{\sqrt{2}} \end{bmatrix}. \quad (\text{B.2})$$

The tabular form KR of (B.2) is provided in Table B.2. The matrix form KR of (B.2) is determined by phase rotating symbol s_2 , conjugating symbols s_2 and s_3 , and permuting symbols s_1 and s_2 . Note that since the left-hand column contains summation terms, the second column is used as the reference column. Also, the top and bottom terms in the reference column are weighted by $1/\sqrt{2}$; therefore, these terms are considered $s_3/\sqrt{2}$. The

matrix form KR is represented as

$$\mathbf{G}_{KR} = \begin{bmatrix} \frac{(-s_2 - s_2^* + s_1 - s_1^*)}{2} & \frac{s_3}{\sqrt{2}} & \frac{s_3}{\sqrt{2}} \\ \frac{s_3^*}{\sqrt{2}} & s_1 & s_2^* \\ \frac{s_3^*}{\sqrt{2}} & s_2 & -s_1^* \\ \frac{(-s_1 - s_1^* + s_2 - s_2^*)}{2} & \frac{s_3}{\sqrt{2}} & \frac{-s_3}{\sqrt{2}} \end{bmatrix}. \quad (\text{B.3})$$

Table B.2. Tabular form KR for the MPO334 variant code shown in (B.2).

SGI	Relationship	Location of Variant 1	Location of Variant 2
1.1.1	Conj. -1	(1, 1)	(2, 2)
1.2.1	Conj.	(2, 1)	(1, 2)
1.1.2	Conj. -1	(4, 1)	(2, 2)
1.2.2	Conj.	(2, 1)	(4, 2)
2.1.1	Conj. -1	(1, 1)	(3, 2)
2.2.1	Conj.	(3, 1)	(1, 2)
2.1.2	Conj. -1	(4, 1)	(3, 2)
2.2.2	Conj.	(3, 1)	(4, 2)
3.1.1	-1	(1, 1)	(4, 1)
3.2.1	-1	(1, 1)	(4, 1)
3.3.1	1	(1, 2)	(4, 2)
4.1.1	Conj. -1	(1, 1)	(2, 3)
4.2.1	Conj.	(2, 1)	(1, 3)
4.1.2	Conj.	(4, 1)	(2, 3)
4.2.2	Conj. -1	(2, 1)	(4, 3)

Table B.2. (continued)

SGI	Relationship	Location of Variant 1	Location of Variant 2
5.1.1	Conj. -1	(1, 1)	(3, 3)
5.2.1	Conj.	(3, 1)	(1, 3)
5.1.2	Conj.	(4, 1)	(3, 3)
5.2.2	Conj. -1	(3, 1)	(4, 3)
6.1.1	1	(1, 1)	(4, 1)
6.2.1	1	(1, 1)	(4, 1)
6.3.1	-1	(1, 3)	(4, 3)
7.1.1	1	(1, 2)	(4, 2)
7.1.2	-1	(1, 3)	(4, 3)
8.1.1	Conj. -1	(2, 2)	(3, 3)
8.2.1	Conj.	(3, 2)	(2, 3)
9.1.1	$\frac{1}{\sqrt{2}}$	(1, 2)	(1, 2)
9.2.1	1	(2, 2)	(2, 2)
9.3.1	1	(3, 2)	(3, 2)
9.1.2	$\frac{1}{\sqrt{2}}$	(4, 2)	(4, 2)

Recall the NO442 base code originally given in (2.10) as

$$\mathbf{G} = \begin{bmatrix} s_1 & s_2 & s_3 & s_4 \\ -s_2^* & s_1^* & -s_4^* & s_3^* \end{bmatrix}.$$

The tabular form KR of (2.10) is provided in Table B.3. The matrix form KR of (2.10) is determined by phase rotating and conjugating s_2 and s_4 . Note that the first and third columns are the reference columns as they are the leftmost columns that collectively contain all data symbols within the data symbol vector. Column two is not a reference column as it only

contains data symbols that are also contained within the first reference column. The matrix form KR is represented as

$$\mathbf{G}_{KR} = \begin{bmatrix} s_1 & -s_2^* & s_3 & -s_4^* \\ s_2 & s_1^* & s_4 & s_3^* \end{bmatrix}. \quad (\text{B.4})$$

Table B.3. Tabular form KR for the NO442 base code shown in (2.10).

SGI	Relationship	Location of Variant 1	Location of Variant 2
1.1.1	Conj.	(1, 1)	(2, 2)
1.2.1	Conj. -1	(2, 1)	(1, 2)
2.1.1	Conj.	(1, 3)	(2, 4)
2.2.1	Conj. -1	(2, 3)	(1, 4)
3.1.1	1	(1, 1)	(1, 1)
3.2.1	1	(2, 1)	(2, 1)
3.3.1	1	(1, 3)	(1, 3)
3.4.1	1	(2, 3)	(2, 3)

List of References

- [1] M. Bloch, J. Barros, M. R. D. Rodrigues, and S. W. McLaughlin, “Wireless information-theoretic security,” *IEEE Transactions on Information Theory*, vol. 54, no. 6, pp. 2515–2534, Jun 2008.
- [2] Y. Qassim, M. E. Magaña, and A. Yavuz, “Post-quantum hybrid security mechanism for MIMO systems,” in *2017 Int. Conf. Comput. Netw. Commun.*, Jan 2017, pp. 684–689.
- [3] J. Hua, S. Jiang, W. Lu, Z. Xu, and F. Li, “A novel physical layer encryption algorithm based on statistical characteristics of time-selective channels,” *IEEE Access*, vol. 6, pp. 38 225–38 233, 2018.
- [4] B. Chen, C. Zhu, W. Li, J. Wei, V. C. M. Leung, and L. T. Yang, “Original symbol phase rotated secure transmission against powerful massive MIMO eavesdropper,” *IEEE Access*, vol. 4, pp. 3016–3025, 2016.
- [5] L. Sun and Q. Du, “Physical layer security with its applications in 5G networks: A review,” *China Commun.*, vol. 14, no. 12, pp. 1–14, Dec 2017.
- [6] Y. Wu, A. Khisti, C. Xiao, G. Caire, K. Wong, and X. Gao, “A survey of physical layer security techniques for 5G wireless networks and challenges ahead,” *IEEE J. Sel. Areas Commun.*, vol. 36, no. 4, pp. 679–695, Apr 2018.
- [7] R. Melki, H. N. Noura, M. M. Mansour, and A. Chehab, “Physical layer security schemes for MIMO systems: An overview,” *Wireless Networks*, vol. 26, no. 3, pp. 2089–2111, Jun 2019.
- [8] M. R. Cribbs, R. A. Romero, and T. T. Ha, “Orthogonal STBC set building and physical layer security application,” in *2020 IEEE 21st International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*, May 2020.
- [9] M. R. Cribbs, R. A. Romero, and T. T. Ha, “Physical layer security for multiple-input multiple-output systems by alternating orthogonal space-time block codes,” *IEEE Open Journal of the Communications Society*, vol. 1, pp. 1503–1523, 2020.
- [10] M. R. Cribbs, R. A. Romero, and T. T. Ha, “Alternative codes and phase rotation extensions for alternating space-time coding-based physical layer security,” *Accepted to IEEE Open Journal of the Communications Society*, 2021.

- [11] M. R. Cribbs, R. A. Romero, and T. T. Ha, "Modulation-based physical layer security via Gray code hopping," in *Accepted to 35th IEEE International Workshop Technical Committee on Communications Quality and Reliability*, May 2021.
- [12] Y. Shiu, S. Y. Chang, H. Wu, S. C. . Huang, and H. Chen, "Physical layer security in wireless networks: A tutorial," *IEEE Wireless Communications*, vol. 18, no. 2, pp. 66–74, 2011.
- [13] C. E. Shannon, "Communication theory of secrecy systems," *Bell Syst. Tech. J.*, vol. 28, no. 4, pp. 656–715, Oct 1949.
- [14] J. L. Massey, "Cryptography-a selective survey," *Alta Frequenza*, vol. 55, no. 1, pp. 4–11, 1986.
- [15] X. Zhou, L. Song, and Y. Zhang, Eds., *Physical Layer Security in Wireless Communications* (Wireless networks and mobile communications; 20). Boca Raton, Florida: Taylor & Francis/CRC Press, 2014.
- [16] G. T. Rendon, W. K. Harrison, M. A. C. Gomes, and J. P. Vilela, "Nested QPSK encoding for information theoretic security," in *2018 IEEE International Conference on Communications (ICC)*, 2018, pp. 1–6.
- [17] S. V. Pechetti, A. Jindal, and R. Bose, "Exploiting mapping diversity for enhancing security at physical layer in the internet of things," *IEEE Internet of Things Journal*, vol. 6, no. 1, pp. 532–544, 2019.
- [18] M. I. Husain, S. Mahant, and R. Sridhar, "CD-PHY: Physical layer security in wireless networks through constellation diversity," in *MILCOM 2012 - 2012 IEEE Military Communications Conference*, 2012, pp. 1–9.
- [19] I. Bang and T. Kim, "Secure modulation based on constellation mapping obfuscation in OFDM based TDD systems," *IEEE Access*, vol. 8, pp. 197 644–197 653, 2020.
- [20] D. Xu, P. Ren, Q. Du, L. Sun, and Y. Wang, "Physical layer security improvement by constellation selection and artificial interference," in *2017 IEEE Wireless Communications and Networking Conference (WCNC)*, 2017, pp. 1–6.
- [21] R. Ma, L. Dai, Z. Wang, and J. Wang, "Secure communication in TDS-OFDM system using constellation rotation and noise insertion," *IEEE Transactions on Consumer Electronics*, vol. 56, no. 3, pp. 1328–1332, 2010.
- [22] M. Xiang-ning, L. Kai-jia, and L. Hao, "A physical layer security algorithm based on constellation," in *2017 IEEE 17th International Conference on Communication Technology (ICCT)*, 2017, pp. 50–53.

- [23] T. Xiong, W. Lou, J. Zhang, and H. Tan, "MIO: Enhancing wireless communications security through physical layer multiple inter-symbol obfuscation," *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 8, pp. 1678–1691, 2015.
- [24] C. Xi, Y. Gao, S. Nan, and P. Lei, "Constellation symbol obfuscation design approach for physical layer security," in *2018 10th International Conference on Communication Software and Networks (ICCSN)*, 2018, pp. 264–269.
- [25] N. Sanandaji and A. Falahati, "A hidden OSTBC scheme to enhance physical layer security by employing a pseudorandom precoder," *IEEE Trans. Commun.*, vol. 68, no. 1, pp. 363–375, Jan 2020.
- [26] T. Allen, J. Cheng, and N. Al-Dhahir, "Secure space-time block coding without transmitter CSI," *IEEE Wireless Communications Letters*, vol. 3, no. 6, pp. 573–576, 2014.
- [27] X. Zhang, Y. Wang, Juan Zeng, and Y. Wang, "A secure OFDM transmission scheme based on chaos mapping," in *2015 IEEE 34th International Performance Computing and Communications Conference (IPCCC)*, 2015, pp. 1–6.
- [28] Y. Wang and L. Zhang, "High security orthogonal factorized channel scrambling scheme with location information embedded for MIMO-based VLC system," in *2017 IEEE 85th Vehicular Technology Conference (VTC Spring)*, 2017, pp. 1–5.
- [29] A. D. Wyner, "The wire-tap channel," *The Bell System Technical Journal*, vol. 54, no. 8, pp. 1355–1387, 1975.
- [30] I. Csiszar and J. Korner, "Broadcast channels with confidential messages," *IEEE Transactions on Information Theory*, vol. 24, no. 3, pp. 339–348, 1978.
- [31] A. Thangaraj, S. Dihidar, A. R. Calderbank, S. W. McLaughlin, and J. Merolla, "Applications of LDPC codes to the wiretap channel," *IEEE Transactions on Information Theory*, vol. 53, no. 8, pp. 2933–2945, 2007.
- [32] D. Kline, J. Ha, S. W. McLaughlin, J. Barros, and B. Kwak, "LDPC codes for the Gaussian wiretap channel," *IEEE Transactions on Information Forensics and Security*, vol. 6, no. 3, pp. 532–540, 2011.
- [33] A. Subramanian, A. Thangaraj, M. Bloch, and S. W. McLaughlin, "Strong secrecy on the binary erasure wiretap channel using large-girth LDPC codes," *IEEE Transactions on Information Forensics and Security*, vol. 6, no. 3, pp. 585–594, 2011.
- [34] S. Goel and R. Negi, "Guaranteeing secrecy using artificial noise," *IEEE Transactions on Wireless Communications*, vol. 7, no. 6, pp. 2180–2189, 2008.

- [35] S. A. A. Fakoorian, H. Jafarkhani, and A. L. Swindlehurst, "Secure space-time block coding via artificial noise alignment," in *2011 Conference Record of the Forty Fifth Asilomar Conference on Signals, Systems and Computers (ASILOMAR)*, 2011, pp. 651–655.
- [36] J. M. Hamamreh, E. Guvenkaya, T. Baykas, and H. Arslan, "A practical physical-layer security method for precoded OSTBC-based systems," in *2016 IEEE Wireless Communications and Networking Conference*, 2016, pp. 1–6.
- [37] M. Ahmed and Lin Bai, "Space time block coding aided physical layer security in Gaussian MIMO channels," in *2017 14th Int. Bhurban Conf. Appl. Sci. Technol.*, Jan 2017, pp. 805–808.
- [38] T. Allen, A. Tajer, and N. Al-Dhahir, "Secure Alamouti multiple access channel transmissions: Multiuser transmission and multi-antenna eavesdroppers," *IEEE Wireless Communications Letters*, vol. 8, no. 5, pp. 1510–1513, 2019.
- [39] O. Souihli and T. Ohtsuki, "The two-way MIMO wire-tap channel," in *2009 IEEE International Conference on Communications*, 2009, pp. 1–6.
- [40] E. Tekin and A. Yener, "The general Gaussian multiple-access and two-way wiretap channels: Achievable rates and cooperative jamming," *IEEE Transactions on Information Theory*, vol. 54, no. 6, pp. 2735–2751, 2008.
- [41] D. J. Love, R. W. Heath, and T. Strohmer, "Grassmannian beamforming for multiple-input multiple-output wireless systems," *IEEE Transactions on Information Theory*, vol. 49, no. 10, pp. 2735–2747, 2003.
- [42] S. Shafiee and S. Ulukus, "Achievable rates in Gaussian MISO channels with secrecy constraints," in *2007 IEEE International Symposium on Information Theory*, 2007, pp. 2466–2470.
- [43] E. Agrell, J. Lassing, E. G. Strom, and T. Ottosson, "Gray coding for multilevel constellations in Gaussian noise," *IEEE Transactions on Information Theory*, vol. 53, no. 1, pp. 224–235, 2007.
- [44] Y. Zhao, Y. tao Xu, H. Jiang, Y. jie Luo, and Z. wen Wang, "Recognition of digital modulation signals based on high-order cumulants," in *2015 International Conference on Wireless Communications & Signal Processing (WCSP)*. IEEE, Oct 2015.
- [45] X. Zhang, S. Zhang, X. Shan, and L. Ma, "A physical layer encryption scheme based on chaotic maps in OFDM systems," in *2020 27th International Conference on Telecommunications (ICT)*, 2020, pp. 1–6.

- [46] S. M. Alamouti, "A simple transmit diversity technique for wireless communications," *IEEE Journal. Sel. Areas Commun.*, vol. 16, no. 8, pp. 1451–1458, Oct 1998.
- [47] S. N. Diggavi, N. Al-Dhahir, A. Stamoulis, and A. R. Calderbank, "Great expectations: The value of spatial diversity in wireless networks," *Proceedings of the IEEE*, vol. 92, no. 2, pp. 219–270, 2004.
- [48] Xue-Bin Liang, "Orthogonal designs with maximal rates," *IEEE Transactions on Information Theory*, vol. 49, no. 10, pp. 2468–2503, 2003.
- [49] A. V. Geramita and J. Seberry, *Orthogonal Designs, Quadratic Forms and Hadamard Matrices, Lecture Notes in Pure and Applied Mathematics*. New York and Basel: Marcel Dekker, 1979, vol. 43.
- [50] V. Tarokh, H. Jafarkhani, and A. R. Calderbank, "Space-time block codes from orthogonal designs," *IEEE Trans. Inf. Theory*, vol. 45, no. 5, pp. 1456–1467, 1999.
- [51] S. Baro, G. Bauch, A. Pavlic, and A. Semmler, "Improving BLAST performance using space-time block codes and turbo decoding," in *Globecom '00 - IEEE. Global Telecommunications Conference. Conference Record (Cat. No.00CH37137)*, 2000, vol. 2, pp. 1067–1071.
- [52] The MathWorks, Inc., "Convert decimal numbers to binary vectors - MATLAB de2bi." Available: <https://www.mathworks.com/help/comm/ref/de2bi.html>
- [53] The MathWorks, Inc., "Create diagonal matrix or get diagonal elements of matrix - MATLAB diag." Available: <https://www.mathworks.com/help/matlab/ref/diag.html>
- [54] The MathWorks, Inc., "All possible permutations - MATLAB perms." Available: <https://www.mathworks.com/help/matlab/ref/perms.html>
- [55] The MathWorks, Inc., "Sort array elements - MATLAB sort." Available: <https://www.mathworks.com/help/matlab/ref/sort.html>
- [56] 3GPP, "Technical Specification Group Radio Access Network; NR; Physical channels and modulation," 3rd Generation Partnership Project (3GPP), Technical Specification (TS) 38.211, 2020, version 16.2.0. Available: <https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=3213>
- [57] E. Agrell, J. Lassing, E. G. Strom, and T. Ottosson, "On the optimality of the binary reflected Gray code," *IEEE Transactions on Information Theory*, vol. 50, no. 12, pp. 3170–3182, 2004.
- [58] J. Zhu, X. She, and L. Chen, "Efficient CQI update scheme for codebook based MU-MIMO with single CQI feedback in E-UTRA," in *2008 IEEE 19th International Symposium on Personal, Indoor and Mobile Radio Communications*, 2008, pp. 1–6.

- [59] T. T. Ha, *Theory and Design of Digital Communication Systems*. New York: Cambridge University Press, 2011.
- [60] D. Wubben, R. Bohnke, J. Rinas, V. Kuhn, and K. D. Kammeyer, "Efficient algorithm for decoding layered space-time codes," *Electronics Letters*, vol. 37, no. 22, pp. 1348–1350, 2001.
- [61] E. Agrell, T. Eriksson, A. Vardy, and K. Zeger, "Closest point search in lattices," *IEEE Transactions on Information Theory*, vol. 48, no. 8, pp. 2201–2214, 2002.
- [62] L. G. Barbero and J. S. Thompson, "A fixed-complexity MIMO detector based on the complex sphere decoder," in *2006 IEEE 7th Workshop on Signal Processing Advances in Wireless Communications*, 2006, pp. 1–5.
- [63] Zhan Guo and P. Nilsson, "Algorithm and implementation of the K-best sphere decoding for MIMO detection," *IEEE Journal on Selected Areas in Communications*, vol. 24, no. 3, pp. 491–503, 2006.
- [64] C. Studer, A. Burg, and H. Bolcskei, "Soft-output sphere decoding: Algorithms and VLSI implementation," *IEEE Journal on Selected Areas in Communications*, vol. 26, no. 2, pp. 290–300, 2008.
- [65] D. Wubben, R. Bohnke, V. Kuhn, and K. . Kammeyer, "MMSE extension of V-BLAST based on sorted QR decomposition," in *2003 IEEE 58th Vehicular Technology Conference. VTC 2003-Fall (IEEE Cat. No.03CH37484)*, 2003, vol. 1, pp. 508–512.
- [66] M. da Silva and F. Monteiro, Eds., *MIMO Processing for 4G and Beyond: Fundamentals and Evolution*. Boca Raton, Florida: Taylor & Francis/CRC Press, 2014.
- [67] C. Hung and T. Sang, "A sphere decoding algorithm for MIMO channels," in *2006 IEEE International Symposium on Signal Processing and Information Technology*, 2006, pp. 502–506.
- [68] J. van Tilburg and D. E. Boeke, "Divergence bounds on key equivocation and error probability in cryptanalysis," in *Adv. Cryptol. Proc. CRYPTO '85*, H. Williams, Ed. Berlin, Heidelberg: Springer-Verlag, 1986, pp. 489–513.
- [69] IEEE Standards Association, "IEEE std 802.3-2018, IEEE standard for ethernet." Available: <https://doi.org/10.1109/ieeestd.2018.8457469>
- [70] M. J. Wiener, "Exhaustive key search," in *Encyclopedia of Cryptography and Security*, H. C. A. van Tilborg, Ed. Boston, MA: Springer US, 2005, pp. 206–209. Available: https://doi.org/10.1007/0-387-23483-7_147

- [71] The MathWorks, Inc., “MATLAB Documentation.” Available: <https://www.mathworks.com/help/index.html>
- [72] GNU Radio, “OutOfTreeModules - GNU Radio.” Available: <https://wiki.gnuradio.org/index.php/OutOfTreeModules>
- [73] 3GPP, “Technical Specification Group Radio Access Network; NR; User Equipment (UE) radio transmission and reception; Part 2: Range 2 Standalone,” 3rd Generation Partnership Project (3GPP), Technical Specification (TS) 38.101-2, Dec 2019, version 16.2.0. Available: <https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=3284>

THIS PAGE INTENTIONALLY LEFT BLANK

Initial Distribution List

1. Defense Technical Information Center
Ft. Belvoir, Virginia
2. Dudley Knox Library
Naval Postgraduate School
Monterey, California