UNIVERSITY OF CALIFORNIA
RIVERSIDE

Novel Methods for Wireless Network Security from Continuous Encryption to Information-Theoretic Secret-Key Generation and Beyond

A Dissertation submitted in partial satisfaction
of the requirements for the degree of

Doctor of Philosophy

in

Electrical Engineering

by

Ahmed Maksud

September 2024

Dissertation Committee:
- Dr. Yingbo Hua, Chairperson
- Dr. Shaolei Ren
- Dr. Basak Guler

The Dissertation of Ahmed Maksud is approved:

Committee Chairperson

University of California, Riverside

Acknowledgments

First and foremost, I would like to express my deepest gratitude to my PhD advisor, Dr. Yingbo Hua, for his unwavering support and invaluable guidance throughout my research journey. I am profoundly grateful for the countless hours we have spent engaging in insightful discussions about my work, career choices, and life in general. Yingbo, your mentorship has been instrumental in shaping my approach to research. Through your guidance, I have learned to tackle problems at a high level and identify novel and relevant research questions, which is often the most challenging aspect of research. Moreover, I have gained invaluable skills in efficiently presenting my work, ensuring clarity and accessibility to readers from diverse backgrounds. Your trust and encouragement have empowered me with the freedom to explore and think at a fundamental level, ultimately contributing to my personal and academic growth. Additionally, I am grateful for your unwavering motivation during moments of doubt and for lifting my spirits when I needed it the most. Thank you, Yingbo, for your outstanding mentoring and constant support; I am truly fortunate to have had you as my advisor.

I extend my heartfelt gratitude to my PhD committee members, Dr. Shaolei Ren and Dr. Basak Guler, for their invaluable feedback and guidance throughout the process of completing this dissertation.

I am also grateful to Dr. Ertem Tuncel and Dr. Jiasi Chen for their invaluable feedback and guidance during my oral qualifying exam.

I would like to acknowledge Dr. Ananthram Swami, Dr. Eric Graves, and Dr. Jake Perazzone from the U.S. Army Research Laboratory for their insightful suggestions,

which have been instrumental in refining my research. I am also deeply thankful to the U.S. Department of Defense and Army Research Office for their generous support.

I have had the privilege of being surrounded by an exceptional group of fellow students at UCR. They are not only intelligent but also incredibly caring individuals who have made my PhD experience truly memorable. I would like to extend my heartfelt thanks to those with whom I have worked closely: Qiping, Ishmam, Shuo, Saydur, Moontasir. Although we did not publish together, I am grateful for the many insightful discussions I had. Thank you all for enriching my journey.

Lastly, I want to thank my better half, Tayafa. You have been a constant support for me over the past two years. In this foreign country, you alleviated my loneliness and were always there when I needed you most. Thank you for always uplifting my spirits. Be there always as you are.

Acknowledgment of previously published materials: The text of this dissertation, in part or in full, is a reprint of the material as appeared in following previously published papers for which I am the lead author or co-author. Dr. Yingbo Hua directed and supervised the research which forms the basis for this dissertation.

1. Yingbo Hua and Ahmed Maksud, "Unconditional Secrecy and Computational Complexity against Wireless Eavesdropping," 2020 IEEE International Workshop on Signal Processing Advances in Wireless Communications (SPAWC), Atlanta, GA, USA, 2020, pp. 1-5.

2. Ahmed Maksud and Yingbo Hua, "Physical Layer Encryption for UAV-to-Ground Communications," 2022 IEEE International Conference on Communications Workshops (ICC Workshops), Seoul, Republic of Korea, 2022, pp. 1077-1082.

3. Ahmed Maksud and Yingbo Hua, "Secret Key Generation by Continuous Encryption Before Quantization," in IEEE Signal Processing Letters, vol. 29, pp. 1497-1501, 2022.

4. Yingbo Hua, Ahmed Maksud, "Continuous Encryption Functions for Security Over Networks," in Signal Processing, Volume 203, 2023, 108807, ISSN 0165-1684.

5. Yingbo Hua and Ahmed Maksud, "Secret Key Generation from MIMO Channel With or Without Reciprocity," 2023 Annual Conference on Information Sciences and Systems (CISS), Baltimore, MD, USA, 2023, pp. 1-6.

6. Ahmed Maksud and Yingbo Hua, "Second-Order Analysis of Secret-Key Capacity From a MIMO Channel," 2023 IEEE Military Communications Conference (MILCOM), Boston, MA, USA, 2023, pp. 833-838.

7. Yingbo Hua and Ahmed Maksud, "Secret-Key Capacity From MIMO Channel Probing," in IEEE Wireless Communications Letters, vol. 13, no. 5, pp. 1434-1438, May 2024.

Additionally, I have co-authored the following previously published papers. The content of these papers is not covered in this dissertation:

1. Ishmam Zabir, Ahmed Maksud, Brian M. Sadler and Yingbo Hua, "Secure Downlink Transmission to Full-Duplex User against Randomly Located Eavesdroppers," 2019 IEEE Global Communications Conference (GLOBECOM), Waikoloa, HI, USA, 2019, pp. 1-6.

2. Ishmam Zabir, Ahmed Maksud, Gaojie Chen, Brian M. Sadler and Yingbo Hua, "Secrecy of Multi-Antenna Transmission With Full-Duplex User in the Presence of Randomly Located Eavesdroppers," in IEEE Transactions on Information Forensics and Security, vol. 16, pp. 2060-2075, 2021.

To Almighty Allah for the countless blessings and to my parents for their support.

ABSTRACT OF THE DISSERTATION

Novel Methods for Wireless Network Security from Continuous Encryption to Information-Theoretic Secret-Key Generation and Beyond

by

Ahmed Maksud

Doctor of Philosophy, Graduate Program in Electrical Engineering
University of California, Riverside, September 2024
Dr. Yingbo Hua, Chairperson

Wireless network security is an increasingly important problem for current and future generations of wireless networks as the computing powers available for attackers to break the traditional encryption methods increase rapidly. This problem is compounded by the need for low-latency required by real-time artificial intelligence and virtual reality applications. This thesis focuses on the issue of information security against eavesdropping and examines the performances of two types of novel methods.

The first type is called continuous encryption which encrypts and decrypts transmitted messages directly using continuous numbers for which good estimates are only available at legitimate nodes. The examples of such continuous numbers include the reciprocal channel parameters between two legitimate wireless nodes in typical scattering rich environment. Continuous encryption does not need the traditional step for two nodes to first agree upon a secret key from their estimates of a reciprocal channel response, and hence reduces the encryption latency. This thesis also examines an application of continuous encryption for UAV communications, the advantages of our proposed continuous encryption

function over prior continuous one-way functions, and a useful role of continuous encryption for secret-key generation.

The second type is based on channel probing for secret-key generation or secret-message transmission. The channel probing methods do not require any reciprocal channel response between two legitimate nodes, and are able to yield a positive secret-key rate and/or secrecy rate in bits per channel use even if the channel coherence time is infinite. This is again useful for low-latency security. This thesis will present an insightful expression of the secret-key capacity for Gaussian probing signals over Gaussian MIMO channels, and also a number of power scheduling policies for a multiple carrier version of Secret-message Transmission by Echoing Encrypted Probes (STEEP).

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Communications and data storage using modern network infrastructure are indispensable in modern life, making information security and privacy paramount concerns. With the advent of fast wireless networks like 5G, the Internet of Things (IoT), and next-gen wireless systems, the volume of transmitted and stored data is growing exponentially. Ensuring this data is securely transmitted and stored has become increasingly critical. Additionally, applications like real-time artificial intelligence and virtual reality require low-latency, secure communication. The inherent broadcast nature of wireless communication exacerbates the challenge of maintaining privacy between parties, leading to significant security issues that impede the development of ultra-fast, reliable networks. Addressing these wireless security challenges is crucial for safeguarding data and supporting the advancement of technologies that benefit society.

Conventionally, the security of wireless communications is established through encryption in the network layer using cryptographic schemes [1]. However, the emergence

of 5G networks with features such as device-to-device and heterogeneous communications, ultra-low latency requirements, and others have made key establishment, management, and distribution processes in wireless networks challenging [2]. Additionally, these schemes depend on the unproven difficulty of solving certain computational problems, commonly referred to as computational complexity security. This approach assumes that an adversary has limited computational power and lacks efficient algorithms to quickly obtain the secret key. However, this assumption is becoming less reliable with the development of more efficient algorithms and the increasing computational power of modern computers. [3].

Physical layer Security (PLS) offers significant advantages over network layer encryption (NLE). Developed on the foundation of Information-theoretic security introduced by Shannon [4], PLS ensures that an eavesdropper cannot obtain any information about a secret message, even with unlimited resources, making it highly attractive for its absolute security and compatibility with existing encryption schemes. Properly implemented PLS schemes are quantum secure, generate on-the-fly secret keys, and are advantageous for IoT and low-latency scenarios due to their lightweight nature [5]. These properties are crucial for applications requiring low-latency secure communication such as real-time artificial intelligence and virtual reality. Once an adversary fails to hack PLS, the secret information generally cannot be hacked later at the network layer, as most physical layer signals do not move up to the network layer. For these reasons, Physical Layer Security (PLS) has drawn significant research attention recently [6–17].

Physical layer security methods can be broadly categorized into Secret Information Transmission (SIT) [6–12] and Secret Key Generation (SKG) [13–17]. The SIT schemes

stem from the wiretap channel (WTC) model introduced by Wyner [18] which relies on the superior channel capacity of legitimate users compared to the wiretap channels to achieve positive secrecy. Methods of SIT include beamforming [9], where the transmit signal is directed towards the receiver using the knowledge of channel state information (CSI), the use of artificial noise [19] to degrade the wiretap channel, and optimized power allocation over different antennas and sub-carriers [20] to achieve optimal secrecy rates. However, it is hard to achieve a positive secrecy rate through this model if the eavesdropper has a large number of antennas and/or better SNR.

In SKG, two or more legitimate parties share secret keys, enabling them to encrypt and decrypt data through similar processing on both sides. This sharing must occur without a prior secret key or physical contact which poses a challenge in the presence of covert adversaries (eavesdroppers). The random variations of the reciprocal wireless channel between legitimate users, independent of the wiretap channel, are exploited for this purpose [14, 21]. Keys can be extracted through methods such as quantization of complex channel coefficients [15] and using measured received signal strength indicators (RSSI) [22]. MIMO systems can significantly increase the shared randomness of the channel, enhancing the secret key.

SKG consists of quantization, information reconciliation, and privacy amplification. During quantization, estimated channels are mapped into a sequence of secret key bits. Due to estimation errors, the mapped sequences may differ, necessitating information reconciliation to correct key mismatches through message exchange. Privacy amplification then eliminates any information leaked to the eavesdropper during reconciliation [23]. How-

ever, in these SKG methods from reciprocal channels, secrecy rate in bits per second per Hz becomes too small in static environment.

In this thesis, two types of novel methods for wireless network security are explored. The focus is on the issue of information security against eavesdropping, i.e., how to transmit secret information from the transmitter (Alice) to the receiver (Bob) in the presence of an eavesdropper (Eve). The first method is continuous encryption which directly encrypts and decrypts transmitted messages using continuous shared secret vectors between legitimate users such as reciprocal channel parameters between two wireless nodes in typical scattering-rich environment. Chapter 2 discusses continuous encryption functions (CEF) and proposes a novel SVD-based CEF. Chapter 3 and 4 examine application of SVD-CEF for secure UAV communications and its useful role in secret key generation. These chapters also discuss contrasts between continuous encryption and traditional encryption based on secret keys. The second method focuses on channel probing for secret-key generation or secret-message transmission. Using secret key capacity bounds, chapter 5 shows that positive secrecy rate is possible between users even if the channel is non-reciprocal and coherence time is infinite. Also, chapter 6 examines various pairing and power scheduling policies in SISO Multi-Carrier setup for a novel scheme, Secret-message Transmission by Echoing Encrypted Probes (STEEP).

In chapter 2, the concept of "Unconditional Secrecy (UNS)," is introduced ensuring secrecy even if the eavesdropper (Eve) has unlimited antennas and zero noise. Achieving positive UNS is possible through either SKG or SIT by exploiting users' reciprocal CSI, provided Eve's receive CSI remains independent of the users. Despite practical constraints

like finite power and antennas within each CSI coherence period, significant virtual UNS can be achieved if Eve cannot overcome the computational complexity of PLE, which offers advantages over NLE by making later hacks impossible due to discarded physical layer data.

Motivated by the limitations and potential of existing methods, new approaches to secure communications were explored. One promising method is Randomized Reciprocal Channel Modulation (RRCM) [24]. The chapter evaluates RRCM for its ability to maintain virtual UNS and shows that different search algorithms fail to break RRCM in feasible time frame. Additionally, the chapter proposes a Singular Value Decomposition (SVD) based continuous encryption function; SVD-CEF. The concept of Continuous Encryption Function (CEF) is widely discussed in the context of biometric template security for cancelable biometrics. However, in this chapter, we observe that existing CEFs, e.g., Random Projection [25], Dynamic Random Projection [26], Index-of-Max Hashing [27] are prone to attack from the adversary.

The proposed SVD-CEF encrypts a shared secret vector $\mathbf{x} \in \mathcal{R}^{N\times 1}$ among legitimate users using publicly known random unitary matrices $\mathbf{Q}_{k,1}, \cdots, \mathbf{Q}_{k,l}$ and generates $\mathbf{y} \in \mathcal{R}^{M\times 1}$ where $M \gg N$. Here, all $\mathbf{x}$, $\mathbf{Q}_{k,l}$, $\mathbf{y}$ are in continuous domain. Unlike traditional discrete one-way/encryption methods requiring 100% reliable secret keys [17, 21, 28], Continuous Encryption Function (CEF) allows encryption using limited, noisy secret vectors. The chapter also proposes qualities of good CEF and evaluates SVD-CEF based on them. Particularly, it is observed through simulation that SVD-CEF is robust against an adversary employing Newton's search algorithm and exhaustive search. The statistical properties of SVD-CEF, e.g., sensitivity, correlation and invariance are also shown and it is observed

that SVD-CEF performs well on these criteria. As the output dimension of the SVD-CEF can be much larger than the input dimension, it can be used to generate long sequence of quasi-continuous pseudo-random numbers, the applications of which are discussed in detail in chapter 3 and 4.

Unmanned Aerial Vehicles (UAVs) represent a compelling application of these security principles. UAVs, increasingly deployed for surveillance, transportation, and mobile communication roles, transmit information that is particularly vulnerable to eavesdropping. Chapter 3 proposes a physical layer encryption (PLE) technique to protect UAV to Ground (U2G) communications. From the reciprocal CSI $\mathbf{x}$ as shared secret between UAV and Ground Station (GS), a stream of output y_k is generated using SVD-CEF. The output y_k is then transformed into a uniform random variable z_k, which is used to encrypt both transmitted symbol and its constellation. The chapter discusses mathematical analysis of the noise propagation through the transformations and comparison with simulation results. The evaluations of the performance of this PLE method in terms of symbol error rate (SER) for different noise levels were performed. Finally, the chapter proposes a discrete version of the PLE method which is easier to implement and provides similar performance.

Secret Key Generation (SKG) is a long standing problem to perform encryption-decryption for secure data transmission and storage. For biometric security, a biometric feature of a person can be collected to generate a secret key for future authentication of this person over any network [25, 29]. In chapter 4, a generalized approach for SKG is presented, referred to as continuous encryption before quantization (CEbQ). Directly quantizing a shared secret vector (SV) of limited dimension poses challenge to obtain secret

key of sufficient length even with well-known methods such as guard-band-quantization [30] and over-quantization [31]. Specifically, extracting multiple bits (as needed for desired key length) by quantizing each element of the SV results in high Key Error Rate (KER). The proposed method first uses SVD-CEF to encrypt the highly correlated SVs into sequences of quasi-continuous pseudorandom numbers (QCPRNs). This expands the shared SV of limited length into a much longer stream of QCPRNs which is then passed through the quantizer. Now the quantizer can extract fewer bits (as low as 1) from each element of the QCPRNs. The chapter also proposes fractional quantization method, where multiple QCPRN samples are used to extract one 1 secret bit. Simulation results show that KER is significantly reduced by the proposed method compared to direct quantization. Finally, it is shown that the secret keys generated by this method are random enough by performing tests using NIST randomness test suit [32].

Chapter 5 explores the Secret Key Capacity (SKC) of MIMO channel subject to Gaussian probing. Using MAC bound [14], the expressions of the upper and lower bound of SKC are derived which reveal useful insights. The expressions of the bounds are divided into First-order-Terms (FoT) and Second-order-Terms (SoT). The FoT is also known as Degree of Freedom (DoF) which scales with the transmit powers of the nodes, whereas, SoT is invariant to the transmit powers. The obtained FoT (which is same as found in [33]) vanishes if Eve has more antenna than both Alice and Bob. However, it is shown that the SoT remains positive even if the channels between users are non-reciprocal and Eavesdropper has more antennas and less noise than both users. Random matrix theory was used to derive the FoT and SoT which relied on the assumption of large transmit power and large number

of antennas. The bounds were also expressed without using random matrix theory and these assumptions. It was also revealed that the gap between the bounds vanishes if the probing is done from the user with more antennas to the user with fewer antennas.

Based on the findings of secrecy key capacity, STEEP [34] introduces a novel scheme of transmitting secret messages in presence of an eavesdropper. Chapter 6 explores different policies to apply STEEP in a Multi-Carrier setup (MC-STEEP) between two single antenna users. STEEP is a 2-way round trip communication scheme for 1-way transmission of secret messages consisting of 2 phases; probing phase and echoing phase. Unlike the classical wiretap channel model, MC-STEEP can benefit from pairing different carriers of probing phase and echoing phase. Through simulations, it is observed that the Average Achievable Secrecy Rate (AASR) significantly increases by pairing strong probing channel with strong echo channel. The chapter also proposes power scheduling for both phases to further improve the AASR. It is observed that power scheduling at echoing phase increases the AASR and power scheduling at probing phase achieves the same AASR for lower probing power budget. Finally, it is observed through simulation that AASR approaches Secret key capacity with high echoing power.

Finally, Chapter 7 summarizes the findings of this thesis and discusses possible extensions and future avenues for further research.

Chapter 2

Continuous Encryption Functions for Security Over Networks

2.1 Introduction

Communications and data storage via the Internet and Clouds are vital to modern life, making information security, particularly privacy, critically important. For optimal privacy in communication, parties must share secret keys, but establishing these keys initially through wireless transmissions without prior contact is challenging. Physical layer security methods, categorized as either secret information transmission (SIT) or secrecy key generation (SKG), address this issue. Though SIT and SKG schemes are considered to combat secrecy issues, only a limited number of SIT schemes can handle the challenge arising from eavesdropper with a large number of antennas [6,7]. Many SIT schemes shown in literature would have zero secrecy if Eve is allowed to have a large number of antennas and zero noise.

We introduce the concept, "Unconditional Secrecy (UNS)" which is the secrecy provided against Eve without any condition on Eve's number of antennae and SNR (Eve having an infinite number of antennas and zero noise). Achieving a positive UNS is possible via either SKG or SIT if the users exploit their own (reciprocal) channel state information (CSI) while Eve's receive CSI is independent of user's CSI. The strict UNS is limited [35] due to practical constraints like finite power and antennas within each CSI coherence period, especially in low-mobility environments. However, significant virtual UNS can be achieved if Eve cannot overcome the computational complexity of physical layer encryption (PLE), offering advantages over network layer encryption (NLE) by making later hacks impossible due to discarded physical layer data.

Section 2.2 of this chapter discusses a new scheme, Randomized Reciprocal Channel Modulation (RRCM) [24], and evaluates the complexity Eve must overcome to breach the virtual UNS provided by RRCM.

Extending the concept of RRCM, later in this chapter in section 2.6, we developed an encryption function that can be used to encrypt in the continuous domain, a shared secret vector $\mathbf{x} \in \mathcal{R}^{N\times 1}$ among the legitimate users with a set of publicly known random unitary matrices $\mathbf{Q}_{k,1}, \cdots, \mathbf{Q}_{k,l}$. The central concept of Continuous Encryption Functions (CEF) is not entirely new. CEFs has been widely discussed in the context of biometric template security to be used for 'cancelable biometrics' [29,36,37]. The central concept in cancelable biometrics is that a biometric template vector such as fingerprint, retinal scan, etc. is first encrypted in the continuous domain through a CEF to obtain a cancelable template. The motivation behind the transformation being if one realization of cancelable template is

somehow compromised, more realization of cancelable templates can be generated without compromising the original permanent biometric template vector. Many prior works have been done on CEFs [25–27, 38]. However, in section 2.4 we will show that these CEFs can be easily attacked by the adversary. The motivation of this chapter is to develop a CEF that has desired characteristics (discussed in section 2.5) which is then used in the context of wireless physical layer security in chapter 3 and 4. However, the proposed CEF is equally applicable for both physical layer security and biometric template security.

In contrast to discrete one-way/encryption which generally requires 100% reliable secret keys [17, 21, 28], Continuous Encryption Function (CEF) allows encryption from limited amount of secrecy available in noisy form. For example, the legitimate parties can obtain the shared secret $\mathbf{x}$ using the estimates of the analog reciprocal channel between them. Both the transmitter and receiver can obtain their own versions of the channel estimate (let, for a MIMO case) $\widehat{\mathbf{H}_{A,B}}$ and $\widehat{\mathbf{H}_{B,A}}$ and use it as $\mathbf{x}$ after a simple manipulation like vectorization. The output of the function $\mathbf{y} \in \mathcal{R}^{M\times 1}$ can be used to encrypt transmit symbols for wireless communication. The noises in the estimated feature vectors in general will degrade the encryption-decryption but only in a soft or controllable way as long as the signal-to-noise ratio (SNR) of one estimated feature vector relative to the other is high and the CEF has a good enough figure-of-merit (FoM). Primary qualities of a good CEF are discussed in section 2.5.

2.2 Randomized Reciprocal Channel Modulation (RRCM)

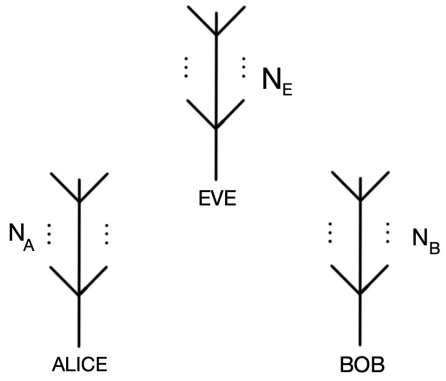


Figure 2.1: Alice Bob and Eve having N_A, N_B and N_E number of antennas

Let, N_A, N_B and N_E the number of antennas of the transmitter (Alice), the receiver (Bob) and the Eavesdropper (Eve) as in figure 2.1 where $N_A = n_A^2 \geq 4$ and $N_B = 1$ (also applicable for any $N_A \geq 1$ and $N_B \geq 1$). Using a pilot from Bob, Alice obtains the channel vector $\mathbf{h} = [h_1, \cdots, h_{N_A}]^T$. Then, Alice computes $\mathbf{D}_s = diag[m_{s,1}, \cdots, m_{s,N_A}]$ for $1 \leq s \leq S$ as follows:

Define $\mathbf{H}_s \in \mathcal{C}^{n_A \times n_A}$ with $(\mathbf{H}_s)_{i,l} = h_{(i-1)n_A+l} m_{s,(i-1)n_A+l}$ where the SVD of $\mathbf{H}_s$ in written as:

$$\mathbf{H}_s = \sum_{i=1}^{n_A} \sigma_{i,s} \mathbf{u}_{i,s} \mathbf{v}_{i,s}^H = \mathbf{U}_s \mathbf{\Sigma}_s \mathbf{V}_s^H \tag{2.1}$$

The first element of the vector $\mathbf{u}_{i,s}$ is normalized to be real. Also let:

$$r_s = \sigma_{1,s} e^{j\mu_{1,s}} \tag{2.2}$$

where $\mu_{1,s}$ is the phase of the first element of $\mathbf{v}_{i,s}$. For each s, Alice chooses a sufficiently random r_s to hide the information of c_s in $r_s c_s$, and also chooses randomly all other components in $\mathbf{U}_s$, $\mathbf{\Sigma}_s$ and $\mathbf{V}_s$ (subject to some bound constraint on each diagonal entry of $\mathbf{\Sigma}_s$

for reliable reception at Bob). Then Alice determines $\mathbf{D}_s = diag[m_{s,1}, \cdots, m_{s,N_A}]$ from $\mathbf{H}_s$. (Any realization of $\mathbf{D}_s$ could be rejected if any of its diagonal entries are too small.) Then, Alice sends a pure and several randomized pilots as follows:

$$\sqrt{P_T}\mathbf{I}_{N_A}, \sqrt{P_T}\mathbf{D}_1\mathbf{I}_{N_A}, \cdots, \sqrt{P_T}\mathbf{D}_s\mathbf{I}_{N_A} \tag{2.3}$$

Then, Bob obtain $\mathbf{h}$ and all entries in $\mathbf{H}_s$ and thus r_s for $1 \leq s \leq S$ using (2.1) and (2.2).

Alice then sends $\sqrt{P_T}r_s c_s$ for $1 \leq s \leq S$ from the antenna corresponding to the strongest channel (h_{max}), and then Bob receives $y_{B,s} = \sqrt{P_T}h_{max}r_s c_s + w_{B,s}$. All channel estimation errors (if not too large) can be lumped into $w_{B,s}$. Since Bob knows $\mathbf{h}$ and r_s, Bob can decode all information in c_s for all s (assuming that the information rate in c_s is so controlled that the probability of detection error is negligible).

On the other hand, Eve with $N_E \geq 2$ and negligible noise can perfectly estimate $\mathbf{G}_A$ and $\mathbf{G}_A\mathbf{D}_s$ for all s from the pilots from Alice and hence, knows $m_{s,i}$ for all s and i. Then, Eve receives from Alice $\mathbf{y}_{E,s} = \mathbf{g}_A r_s c_s$ where $\mathbf{g}_A$ is one of the N_A columns in $\mathbf{G}_A$ and can be estimated by Eve, which means Eve knows $r_s c_s$ for all s. In order to decode the information in c_s, Eve must first determine r_s. Assume that Eve has guessed correctly c_s for $1 \leq s \leq S_0$ and hence knows r_s for $1 \leq s \leq S_0$. In order to determine r_s for $s > S_0$, Eve now must determine $\mathbf{h}$ using r_s for $1 \leq s \leq S_0$ via (2.1) along with the conditions $\mathbf{U}_s^H\mathbf{U}_s = \mathbf{I}_{n_A}$ and $\mathbf{V}_s^H\mathbf{V}_s = \mathbf{I}_{n_A}$. One can verify that the total number of real unknowns (i.e., those in $\mathbf{h}$ and all other unknowns in the SVD equation (2.1)) is $N_{unk} = 2n_A^2 + 2(n_A^2 - 1)S_0$ and the total number of effective real equations is $N_{equ} = 2n_A^2 S_0$. For a finite number of solutions of $\mathbf{h}$, it is necessary (but not sufficient) that $N_{unk} \leq N_{equ}$ or equivalently $S_0 \geq n_A^2$.

Hence, the strict amount of UNS of RRCM is no less than the entropy of n_A^2 symbols from Alice. Here (2.1) is nonlinear and if Eve uses exhaustive search to find $\mathbf{h}$, Eve has to compute the $n_A \times n_A$ SVD for each choice of $\mathbf{h}$. With N_q to be the number of quantization levels for each real element in $\mathbf{h}$, the number of these choices is in the order of $\mathcal{O}(N_q^{2N_A^2})$ Alternatively, Eve may apply the Newton's method to search for $\mathbf{h}$ (Which is to our knowledge the best way to solve these nonlinear equations). The complexity per iteration of the Newton's algorithm is in the order of $\mathcal{O}(N_{unk}^3)$. Unlike many other schemes, RRCM forces Eve to solve a nonlinear inverse problem to obtain user's CSI.

2.3 Simulation of Eve's Complexity to Break (RRCM)

Here we present two approaches Eve can take to break RRCM. First, we present a search method based on Newton's search algorithm. Then we also show the search complexity of exhaustive search.

2.3.1 Using Newton's Search Algorithm

Details about the the applied newton's algorithm are given in Appendix A in [39].

Four Channel Unknowns

Let $N_A = 4$ and $N_B = 1$. It follows that

$$\mathbf{H}_s = \begin{bmatrix} h_1 m_{1,s} & h_2 m_{2,s} \\ h_3 m_{3,s} & h_4 m_{4,s} \end{bmatrix} \tag{2.4}$$

where each element of $\mathbf{h} = [h_1, \cdots, h_4]^T$ was randomly chosen from $\mathcal{CN}(0,1)$, and $\mathbf{m}_s = [m_{s,1}, \cdots, m_{s,4}]$ for each s was so chosen that r_s defined via (2.1) and (2.2) is sufficiently

random (and the singular values have sufficient distances from each other). Assume that Eve has correctly guessed c_s for $s = 1, \cdots, S_0$ and hence, Eve now knows r_s for $s = 1, \cdots, S_0$. We simulated the Newton's method to find $\mathbf{h}$ using r_s for $s = 1, \cdots, S_0$. For $S_0 = 4$, the Newton's method yielded correct solutions of $\mathbf{h}$ from 94 of 100 random initializations of $\mathbf{h}$. (Note that the correct solutions of $\mathbf{h}$ include those that may be different from $\mathbf{h}$ but yield the same r_s via the SVD equation (2.1) for all s including $s > S_0$.) But for $S_0 = 5$, the Newton's method yielded a correct solution of $\mathbf{h}$ from each of 100 random initializations. We also tested a phase-only modulation where $r_s = e^{j_{1,s}}$. In this case, the number of unknowns is no larger than the number of equations if and only if $S_0 \geq 8$. It is somewhat expected that using r_s for $s = 1, \cdots, S_0$ with $S_0 \leq 7$, the Newton's method did not find any correct solution of $\mathbf{h}$. But for $S_0 = 8$, the Newton's method yielded a correct solution of $\mathbf{h}$ (valid for all s) from 1 out of 500 random initializations. Furthermore, we found that the Newton's method has a very poor convergence property for the phase-only modulation.

Nine Channel Unknowns

Now we consider $N_A = 9$ and $N_B = 1$. In this case, $\mathbf{H}_s$ is a 3×3 matrix and $\mathbf{h}$ is a 9×1 vector. The necessary condition on S_0 for a finite number of solutions of $\mathbf{h}$ is now $S_0 \geq 9$. But with $S_0 = 9$, the Newton's method with 1000 random initializations of $\mathbf{h}$ did not even converge to a reasonable solution of $\mathbf{h}$ that is valid for $1 \leq s \leq S_0$. In other words, the Newton's method could not handle this case in our simulation.

This is apparently due to the non-linearity of the problem. In this case, the effective degree of non-linearity for each unknown in (2.1) increases as the dimension of $\mathbf{h}$ increases. (For a set of n arbitrary second-order polynomial equations with n unknowns, for example,

the number of possible solutions from these equations could be up to 2^n.) As Newton's method could not handle the case of even $N_A = 9$ (which is quite small), it is apparent that this method will not be able handle cases regarding larger N_A.

2.3.2 Using Exhaustive Search

Since the Newton's method could not handle the case with 9 channel unknowns, we now consider the exhaustive search over a discrete space $\mathcal{S}_h$ of the 9×1 complex vector $\mathbf{h}$. For each $\mathbf{h} \in \mathcal{S}_h$, we need to compute the SVD (2.1) for $s = 1, \cdots, S_0$ with $S_0 = 9$. (A correct solution of $\mathbf{h}$ might be found with a nonzero probability if the consequent r_s for $s = 1, \cdots, S_0$ from (2.2) match the "known" r_s for $s = 1, \cdots, S_0$.) With N_q quantization levels for each real component in $\mathbf{h}$, we have $|\mathcal{S}_h| = N_q^{18}$. To obtain an estimate of how the required computational time varies with N_q, we used our PC with 11.1 Giga-flops to compute the 3×3 SVD (2.1) for all realizations of h with $N_q = 2$ and for $s = 1, \cdots, 9$. We recorded this time as T_2. Then the required time for N_q can be estimated by $T_{N_q} = \frac{N_q^{18}}{2^{18}} T_2$ which is illustrated in figure 2.2. Also shown in this figure is the time required if a supercomputer with 50 Peta-flops is applied here. For easy reference, we have also marked the times for 1 day, 1 year and 1 decade. We see that with just $N_q = 8$ (or 3 bits for each real component of h), finding h using the exhaustive search could require more than a decade on our PC or more than a day on a supercomputer. With a randomized exhaustive search, the averaged time required to find a correct solution of h can be reduced by a factor, but the order of complexity $\mathcal{O}(N_q^{2N_A})$ as discussed before does not change.

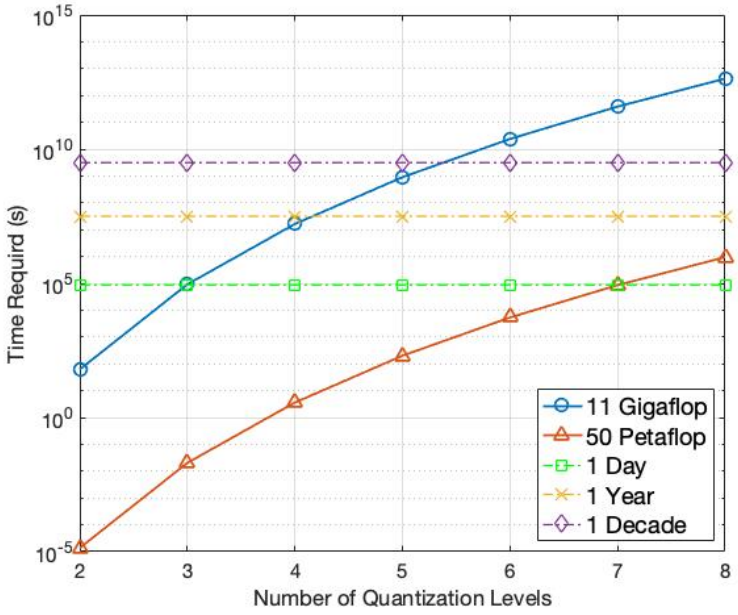


Figure 2.2: Time required for exhaustive search to break RRCM

Subject to Eve's failure to obtain a correct solution of h (and hence rs for any $s \geq S_0$) based on random guesses of c_s for $s = 1, \cdots, S_0$, all information in c_s for all s transmitted from Alice to Bob remains secure from Eve with any number of antennas and any noise level.

2.4 Previously Developed CEFs

In this section, we will show that prior CEFs discussed in the literature are vulnerable to attacks. The CEFs can be categorized into two main families:

2.4.1 Linear family of CEFs

A family of linear CEFs can be expressed as follows:

$$\mathbf{y} = \mathbf{R}_S \mathbf{x} \tag{2.5}$$

where $\mathbf{y} = [y_1, y_2, \cdots, y_M]^T$, M is a large integer, $\mathbf{R}_S$ is a $M \times N$ pseudorandom matrix dependent on a secret key S. Let the ith $M_i \times 1$ sub-vector of $\mathbf{y}$ be $\mathbf{y}_i$, and the ith $M_i \times N$ block matrix of $\mathbf{R}_S$ be $\mathbf{R}_{S,i}$. Then it follows that

$$\mathbf{y}_i = \mathbf{R}_{S,i} \mathbf{x} \tag{2.6}$$

where $i = 1, \cdots, I$ and $\sum_{i=1}^{I} M_i = M$.

Random Projection

The linear family of CEFs includes the random projection (RP) method shown in [25] and applied in [40]. If S is known, so is $\mathbf{R}_{S,i}$ for all i. If $\mathbf{y}_i$ for some i is known/exposed and $\mathbf{R}_{S,i}$ is of the full column rank N, then $\mathbf{x}$ is given by

$$\mathbf{R}_{S,i}^{+} \mathbf{y}_i = (\mathbf{R}_{S,i}^T \mathbf{R}_{S,i})^{-1} \mathbf{R}_{S,i}^T \mathbf{y}_i \tag{2.7}$$

where $+$ denotes pseudo-inverse. If $\mathbf{R}_{S,i}$ is not of full column rank, then $\mathbf{x}$ can be computed from a set of outputs like (for example) $\mathbf{y}_1, \cdots, \mathbf{y}_L$ where L is such that the vertical stack of $\mathbf{R}_{S,1}, \cdots, \mathbf{R}_{S,L}$, denoted by $\mathbf{R}_{S,1:L}$, is of the full column rank N. If S is unknown, then a method to compute $\mathbf{x}$ includes a discrete search for the N_S bits of S as follows

$$\min_S \min_{\mathbf{x}} \|\mathbf{y}_{1:L} - \mathbf{R}_{S,1:L} \mathbf{x}\| = \min_S \|\mathbf{y}_{1:L} - \mathbf{R}_{S,1:L} \mathbf{R}_{S,1:L}^{+} \mathbf{y}_{1:L}\| \tag{2.8}$$

where $\mathbf{y}_{1:L}$ is the vertical stack of $\mathbf{y}_1, \cdots, \mathbf{y}_L$. The total complexity of the above attack algorithm with unknown key S is $P_{N,M}2^{N_S}$ with $P_{N,M}$ being a linear function of $\sum_{i=1}^{L} M_i$ and a cubic function of N. So, RP is not hard to attack (subject to a small N_S).

Dynamic Random Projection

The dynamic random projection (DRP) method proposed in [26] and also discussed in [37] can be described by:

$$\mathbf{y}_i = \mathbf{R}_{S,i,\mathbf{x}}\mathbf{x} \tag{2.9}$$

where $\mathbf{R}_{S,i,\mathbf{x}}$ is the ith realization of a random matrix that depends on both S and $\mathbf{x}$. Since $\mathbf{R}_{S,i,\mathbf{x}}$ is discrete, $\mathbf{y}_i$ in (4) is a locally linear function of $\mathbf{x}$. (There is a nonzero probability that a small perturbation $\mathbf{w}$ in $\mathbf{x}' = \mathbf{x}+\mathbf{w}$ leads to $\mathbf{R}_{S,i,\mathbf{x}'}$ being substantially different from $\mathbf{R}_{S,i,\mathbf{x}}$. This is not a desirable outcome for biometric templates although the probability may be small.) Two methods were proposed in [26] to construct $\mathbf{R}_{S,i,\mathbf{x}}$, which were called "Functions I and II" respectively. For simplicity of notation, we will now suppress i and S in (4) and write it as

$$\mathbf{y} = \mathbf{R}_{\mathbf{x}}\mathbf{x} \tag{2.10}$$

Assuming "Function I" in [26]

In this case, the ith element of $\mathbf{y}$, denoted by v_i, corresponds to the ith slot shown in [26] and can be written as

$$v_i = \mathbf{r}_{\mathbf{x},i}^T\mathbf{x} \tag{2.11}$$

where $\mathbf{r}_{\mathbf{x},i}^T$ is the ith row of $\mathbf{R}_{\mathbf{x}}$. But $\mathbf{r}_{\mathbf{x},i}^T$ is one of L key-dependent pseudorandom vectors $\mathbf{r}_{i,1}^T, \cdots, \mathbf{r}_{i,L}^T$ that are independent of $\mathbf{x}$ and known if S is known. So we can also write

$$v_i = \mathbf{r}_i^T \bar{\mathbf{x}} \tag{2.12}$$

where $\mathbf{r}_i^T = [\mathbf{r}_{i,1}^T, \cdots, \mathbf{r}_{i,L}^T]^T$, and $\bar{\mathbf{x}} \in \mathcal{R}^{LN \times 1}$ is a sparse vector consisting of zeros and $\mathbf{x}$. Before $\mathbf{x}$ is known, the position of $\mathbf{x}$ in $\bar{\mathbf{x}}$ is initially unknown. If an attacker has stolen K realizations of v_i (denoted by $v_{i,1}, \cdots, v_{i,K}$), then it follows that

$$\mathbf{v}_i = \mathbf{R}_i \bar{\mathbf{x}} \tag{2.13}$$

where $\mathbf{v}_i = [v_{i,1}, \cdots, v_{i,K}]^T$, and $\mathbf{R}_i$ is the vertical stack of K key-dependent random realizations of $\mathbf{r}_i^T$. With $K \geq LN$, $\mathbf{R}_i$ is of full column rank LN with probability one, and in this case the above equation (when given the key S) is linearly invertible with a complexity order equal to $\mathcal{O}((LN)^3)$. An even simpler method of attack is as follows. Since $v_{i,k} = \mathbf{r}_{i,k,l}^T \mathbf{x}$ where $l \in \{1, \cdots, L\}$ and $\mathbf{r}_{i,k,l}$ for all i, k and l are known, then we can compute

$$l^* = \arg \min_{l \in \{1, \cdots, L\}} \min_{\mathbf{x}} \|\mathbf{v}_i - \mathbf{R}_{i,l}\mathbf{x}\|^2 = \arg \min_{l \in \{1, \cdots, L\}} \|\mathbf{v}_i - \mathbf{R}_{i,l}\mathbf{R}_{i,l}^{+}\mathbf{v}_i\|^2 \tag{2.14}$$

where $\mathbf{R}_{i,l}$ is the vertical stack of $\mathbf{r}_{i,k,l}^T$ for $k = 1, \cdots, K$. Provided $K \geq N$, $\mathbf{R}_{i,l}$ has the full column rank with probability one. In this case, the correct solution of $\mathbf{x}$ is given by $\mathbf{R}_{i,l^*}^{+}\mathbf{v}_i$. This method has a complexity order equal to $\mathcal{O}(LN^3)$.

Assuming "Function II" in [26]

To attack "Function II" with known S, it is equivalent to consider the following signal model:

$$v_k = \sum_{n=1}^{N} r_{k,l_k,n} x_n \tag{2.15}$$

where v_k is available for $k = 1, \cdots, K$, $r_{k,l,n}$ for $1 \le k \le K$, $1 \le l \le L$ and $1 \le n \le N$ are random but known numbers (when given S), x_n for all n are unknown, and l_k is a k-dependent random/unknown choice from $[1, \cdots, L]$. We can write

$$\mathbf{v} = \mathbf{R}\mathbf{x} \tag{2.16}$$

where $\mathbf{v}$ is a stack of all v_k, $\mathbf{x}$ is a stack of all x_n, and $\mathbf{R}$ is a stack of all $r_{k,l_k,n}$ (i.e., $(\mathbf{R})_{k,n} = r_{k,l_k,n}$). In this case, $\mathbf{R}$ is a random and unknown choice from L^K possible known matrices. An exhaustive search would require the $\mathcal{O}(L^K)$ complexity with $K \ge N + 1$.

Now we consider a different approach of attack. Since $r_{k,l,n}$ for all k, l, n are known, we can compute

$$c_{n,n'} = \frac{1}{KL} \sum_{k=1}^{K} \sum_{l=1}^{L} \sum_{l'=1}^{L} r_{k,l,n} r_{k,l',n'} \tag{2.17}$$

If $r_{k,l,n}$ are pseudo i.i.d. random (but known) numbers of zero mean and variance one, then for large K (e.g., $K \gg L^2$) we have

$$c_{n,n'} \approx \delta_{n,n'} \tag{2.18}$$

Also define

$$y_n = \frac{1}{K} \sum_{k=1}^{K} \sum_{l=1}^{L} v_k r_{k,l,n} = \sum_{n'=1}^{N} \hat{c}_{n,n'} x_{n'} \tag{2.19}$$

where $n = 1, \cdots, N$ and

$$\hat{c}_{n,n'} = \frac{1}{K} \sum_{k=1}^{K} \sum_{l=1}^{L} r_{k,l,n} r_{k,l_k,n'} \tag{2.20}$$

If $r_{k,l,n}$ are i.i.d. of zero mean and unit variance, then for large K we have $\hat{c}_{n,n'} \approx c_{n,n'} \approx \delta_{n,n'}$ and hence

$$y_n \approx x_n \tag{2.21}$$

More generally, if we have $\hat{c}_{n,n'} \approx c_{n,n'}$ with a large K, then

$$\mathbf{y} \approx \mathbf{C}\mathbf{x} \tag{2.22}$$

where $(\mathbf{y})_n = y_n$, and $(\mathbf{C})_{n,n'} = c_{n,n'}$. Hence,

$$\mathbf{x} \approx \mathbf{C}^{-1}\mathbf{y} \tag{2.23}$$

With an initial estimate $\hat{\mathbf{x}}$ of $\mathbf{x}$, we can then do the following to refine the estimate:

1. For each $k = 1, \cdots, K$, compute

$$l_k^* = \arg \min_{l \in [1,\cdots,L]} \left| v_k - \sum_{n=1}^{N} r_{k,l,n} \hat{x}_n \right|. \tag{2.24}$$

2. Recall $\mathbf{v} = \mathbf{R}\mathbf{x}$. But now use $(\mathbf{R})_{k,n} = r_{k,l_k^*,n}$ for all k and n, and replace $\hat{\mathbf{x}}$ by

$$\hat{\mathbf{x}} = (\mathbf{R}^T\mathbf{R})^{-1}\mathbf{R}^T\mathbf{v} \tag{2.25}$$

3. Go to step 1 until convergence.

Note that all entries in $\mathbf{R}$ are discrete. Once the correct $\mathbf{R}$ is found, the exact $\mathbf{x}$ is obtained. The above algorithm converges to either the exact $\mathbf{x}$ or a wrong $\mathbf{x}$. But with a sufficiently large K with respect to a given pair of N and L, our simulation shows that the above attack algorithm yields the exact $\mathbf{x}$ with high probabilities. For example, for $N = 8$, $L = 8$ and $K = 23L$, the success rate is 99%. And for $N = 16$, $L = 48$ and $K = 70L$, the success rate is 98%. In the experiment, for each set of N, L and K, 100 independent realizations of all elements in $\mathbf{x}$ and $\mathbf{R}$ were chosen from an i.i.d. Gaussian distribution with zero mean and unit variance, i.e., $\mathcal{N}(0, 1)$. The success rate was based on the 100 realizations. In [26],

an element-wise quantized version of $\mathbf{v}$ was further suggested to improve the hardness to invert. In this case, the vector potentially exposable to an attacker can be written as:

$$\hat{\mathbf{v}} = \mathbf{R}\mathbf{x} + \mathbf{w} \tag{2.26}$$

where $\mathbf{w}$ can be modelled as a white noise vector uncorrelated with $\mathbf{R}\mathbf{x}$. The above attack algorithm with $\mathbf{v}$ replaced by $\hat{\mathbf{v}}$ also applies although a larger K is needed to achieve the same rate of successful attack. In all of the above cases, the computational complexity for a successful attack is a polynomial function of N, L and/or K when the secret key S is given.

2.4.2 Unitary Random Projection (URP)

None of the RP and DRP methods is homomorphic. To have a homomorphic CEF whose input and output have the same distance measure, we can use

$$\mathbf{y}_k = \mathbf{Q}_k\mathbf{x} \tag{2.27}$$

where $\mathbf{Q}_k \in \mathcal{R}^{N\times N}$ for each realization index k is a pseudorandom unitary matrix governed by a secret key S. One way to generate $\mathbf{Q}_k$ is to compute the QR decomposition [41] of a random matrix $\mathbf{X}_k$ whose entries are pseudorandom numbers (including Gaussian random numbers) from a standard cryptographically secure pseudorandom number generator. It is important to note that if there is a secret key with its length $N_S \geq N$, then URP is also hard to invert strictly speaking. But as stressed earlier, this paper focuses on the case where $N_S \ll N$ or simply $N_S = 0$. Let $\mathbf{x}' = \mathbf{x} + \mathbf{w}$ with $\mathbf{w}$ being a noise. Then

$$\mathbf{y}'_k = \mathbf{Q}_k\mathbf{x}' = \mathbf{Q}_k\mathbf{x} + \mathbf{Q}_k\mathbf{w} \tag{2.28}$$

It follows that the SNR of $\mathbf{y}'_k$ equals the SNR of $\mathbf{x}'$, and hence the FoM of URP equals one. We can view the noise sensitivity of URP as optimal. In fact, if Alice and Bob do share a strong secret key, then the URP would be an ideal CEF as it would meet perfectly all the five criteria. However, like RP and DRP, URP is easy to attack if the secret key is weak or does not exist. Furthermore, as shown later, without a secret key or equivalently with a known set of $\mathbf{Q}_k$ for all k, the output samples of URP are highly correlated with each other. Note that each of the linear CEFs requires a forward per-sample computation complexity equal to $O(N)$. For example, to produce N output samples of URP, we need to generate the $N \times N$ unitary matrix $\mathbf{Q}_k$, which requires a computational complexity equal to $O(N^2)$. We also need to compute the product $\mathbf{Q}_k\mathbf{x}$ which costs another $O(N^2)$. So, the per-sample complexity is $O(N)$. If $\mathbf{x}$ consists of i.i.d. $N(0, \sigma_x^2)$, all entries of $\mathbf{y}_i$ for all i are also $N(0, \sigma_x^2)$, which is a desired invariance of statistical distribution. But the entries of $\mathbf{y}_i$ in general have significant correlations with entries of $\mathbf{y}_j$ for $j \neq i$ (even though the N entries of $\mathbf{y}_i$ for each i have zero correlations among themselves). Simulation results on the correlations of RP, DRP, and URP will be shown later in fig. 2.6.

2.4.3 Nonlinear family of CEFs

If the secret key S available is not large enough, then we will need a CEF that is hard to attack even if S is known. Such a CEF has to be nonlinear.

More recently a method called index-of-max (IoM) hashing was proposed in [27] and applied in [42]. There are algorithms 1 and 2 based on IoM, which will be referred to as IoM-1 and IoM-2. In IoM-1, the feature vector $\mathbf{x} \in \mathcal{R}^{N\times 1}$ is multiplied (from the left) by a sequence of $L\times N$ pseudorandom matrices $\mathbf{R}_1, \cdots, \mathbf{R}_{K_1}$ to produce $\mathbf{v}_1, \cdots, \mathbf{v}_{K_1}$ respectively.

The index of the largest element in each $\mathbf{v}_k$ is used as an output y_k. With $\mathbf{y} = [y_1, \cdots, y_{K_1}]^T$, we see that $\mathbf{y}$ is a nonlinear ("piece-wise" constant and "piece-wise" continuous) function of $\mathbf{x}$. The generation of each of $\mathbf{R}_1, \cdots, \mathbf{R}_{K_1}$ requires $O(N^2)$ complexity, and the computation of each of $\mathbf{v}_1, \cdots, \mathbf{v}_{K_1}$ requires additional $O(N^2)$ complexity. The search for the maximum entry within each $\mathbf{v}_k$ costs $O(N)$. Hence, the per-sample complexity of IoM-1 is $O(N^2)$.

In IoM-2, $\mathbf{R}_1, \cdots, \mathbf{R}_{K_1}$ used in IoM-1 are replaced by $N \times N$ pseudorandom permutation matrices $\mathbf{P}_1, \cdots, \mathbf{P}_{K_1}$ to produce $\mathbf{v}_1, \cdots, \mathbf{v}_{K_1}$, and then a sequence of vectors $\mathbf{h}_1, \cdots, \mathbf{h}_{K_2}$ are produced in such a way that each $\mathbf{h}_k$ is the element-wise product of an exclusive set of p vectors from $\mathbf{v}_1, \cdots, \mathbf{v}_{K_1}$. The index of the largest element in each $\mathbf{h}_k$ is used as an output y_k. With $\mathbf{y} = [y_1, \cdots, y_{K_2}]^T$, we see that $\mathbf{y}$ is another nonlinear continuous function of $\mathbf{x}$.

The complexity of p random permutations of $\mathbf{x}$ to produce p of $\mathbf{v}_k$ is $O(pN^2)$ (even though there is no multiplication required). The complexity to produce each $\mathbf{h}_k$ is $O(pN)$. Then the per-sample complexity of IoM-2 is also $O(N^2)$ provided that p is independent of N. If $p = N$, the per-sample complexity of IoM-2 becomes $O(N^3)$. Next, we show that IoM-1 is not hard to invert if the secret key S or equivalently the random matrices $\mathbf{R}_1, \cdots, \mathbf{R}_{K_1}$ are known. We also show that IoM-2 is not hard to invert up to the sign of each element in $\mathbf{x}$ if the secret key S or equivalently the random permutations $\mathbf{P}_1, \cdots, \mathbf{P}_{K_1}$ are known.

Attack of IoM-1

Assume that each $\mathbf{R}_k$ has L rows and the secret key S is known. Then knowing y_k for $k = 1, \cdots, K_1$ means knowing $\mathbf{r}_{k,a,l}$ and $\mathbf{r}_{k,b,l}$ satisfying

$$\mathbf{r}_{k,a,l}^T \mathbf{x} > \mathbf{r}_{k,b,l}^T \mathbf{x} \tag{2.29}$$

with $l = 1, \cdots, L-1$ and $k = 1, \cdots, K_1$. Here $\mathbf{r}_{k,a,l}^T$ and $\mathbf{r}_{k,b,l}^T$ for all l are rows of $\mathbf{R}_k$. The above is equivalent to

$$\mathbf{d}_{k,l}^T \mathbf{x} > 0 \tag{2.30}$$

with

$$\mathbf{d}_{k,l} = \mathbf{r}_{k,a,l} - \mathbf{r}_{k,b,l} \tag{2.31}$$

or more simply

$$\mathbf{d}_k^T \mathbf{x} > 0 \tag{2.32}$$

where $\mathbf{d}_k$ is known for $k = 1, \cdots, K$ with $K = K_1(L-1)$. Note that any scalar change to $\mathbf{x}$ does not affect the output y. Also note that even though IoM-1 defines a nonlinear function from $\mathbf{x}$ to y, the conditions in (2.32) useful for attack are linear with respect to $\mathbf{x}$. To attack IoM-1, we can simply compute $\hat{\mathbf{x}}$ satisfying $\mathbf{d}_k^T \hat{\mathbf{x}} > 0$ for all k. One such algorithm of attack is as follows:

1. **Initialization/averaging:** Let $\hat{\mathbf{x}} = \bar{\mathbf{d}} \coloneqq \frac{1}{K} \sum_{k=1}^{K} \mathbf{d}_k$.

2. **Refinement:** Until $\mathbf{d}_k^T \hat{\mathbf{x}} > 0$ for all k, choose $k^* = \arg\min_k \mathbf{d}_k^T \hat{\mathbf{x}}$, and compute

$$\hat{\mathbf{x}} \leftarrow \hat{\mathbf{x}} - \eta(\mathbf{d}_{k^*}^T \hat{\mathbf{x}}) \mathbf{d}_{k^*} \tag{2.33}$$

   where η is a step size.

Our simulation (using $\eta = \frac{1}{\|\mathbf{d}_{k^*}\|^2}$) shows that using the initialization alone can yield a good estimate of $\mathbf{x}$ as K increases. More specifically, the normalized projection $\frac{\bar{\mathbf{d}}^T\mathbf{x}}{\|\mathbf{d}\|\cdot\|\mathbf{x}\|}$ converges to one as K increases. Our simulation also shows that the second step in the above algorithm improves the convergence slightly. Examples of the attack results are shown in Tables 2.1 and 2.2 where $L = N$. We see that IoM-1 (with its key S exposed) can be inverted with a complexity order no larger than a linear function of N and K_1 respectively.

Table 2.1: Normalized projection of $\mathbf{x}$ onto its estimate using only averaging for attack of IoM-1.

	K_1			
N	8	16	32	64
8	0.8546	0.9171	0.9562	0.9772
16	0.8022	0.8842	0.9365	0.9666
32	0.7328	0.8351	0.9060	0.9494

Table 2.2: Normalized projection of $\mathbf{x}$ onto its estimate after convergence of refinement for attack of IoM-1.

N	K_1			
	8	16	32	64
8	0.8807	0.9467	0.9804	0.9937
16	0.8174	0.9080	0.9612	0.9861
32	0.7390	0.8497	0.9268	0.9699

Attack of IoM-2

To attack IoM-2, we need to know the sign of each element of $\mathbf{x}$, which is assumed below. Given the output of IoM-2 and all the permutation matrices $\mathbf{P}_1, \cdots, \mathbf{P}_{K_1}$, we know which of the elements in each $\mathbf{h}_k$ is the largest and which of these elements are negative. If the largest element in $\mathbf{h}_k$ is positive, we will ignore all the negative elements in $\mathbf{h}_k$. If the largest element in $\mathbf{h}_k$ is negative, we know which of the elements in $\mathbf{h}_k$ has the smallest absolute value. Let $|\mathbf{h}_k|$ be the vector consisting of the corresponding absolute values of the elements in $\mathbf{h}_k$. Also let $\log|\mathbf{h}_k|$ be the vector of element-wise logarithm of $|\mathbf{h}_k|$. It follows that

$$\log|\mathbf{h}_k| = \mathbf{T}_k \log|\mathbf{x}| \tag{2.34}$$

where $\mathbf{T}_k$ is the sum of the permutation matrices used for $\mathbf{h}_k$. The knowledge of an output y_k of IoM-2 implies the knowledge of $\mathbf{t}_{k,a,l}^T$ and $\mathbf{t}_{k,b,l}^T$ (i.e., row vectors of $\mathbf{T}_k$) such that either

$$\mathbf{t}_{k,a,l}^T \log|\mathbf{x}| > \mathbf{t}_{k,b,l} \log|\mathbf{x}| \tag{2.35}$$

with $l = 1, \cdots, L_k - 1$ if $\mathbf{h}_k$ has $L_k \geq 2$ positive elements, or

$$\mathbf{t}_{k,a,l}^T \log|\mathbf{x}| < \mathbf{t}_{k,b,l} \log|\mathbf{x}| \tag{2.36}$$

with $l = 1, \cdots, N - 1$ if $\mathbf{h}_k$ has no positive element.

If $\mathbf{h}_k$ has only one positive element, corresponding y_k can be ignored as it yields no useful constraint on $\log|\mathbf{x}|$. We assume that no element in $\mathbf{x}$ is zero. Equivalently, knowledge of y_k implies $\mathbf{c}_{k,l}^T \log|\mathbf{x}| > 0$ where $\mathbf{c}_{k,l} = \mathbf{t}_{k,a,l} - \mathbf{t}_{k,b,l}$ for $l = 1, \cdots, L_k - 1$ if $\mathbf{h}_k$ has $L_k \geq 2$ positive elements, or $\mathbf{c}_{k,l} = -\mathbf{t}_{k,a,l} + \mathbf{t}_{k,b,l}$ for $l = 1, \cdots, N - 1$ if $\mathbf{h}_k$ has no positive element. Simpler form of the constraints on $\log|\mathbf{x}|$:

$$\mathbf{c}_k^T \log|\mathbf{x}| > 0 \tag{2.37}$$

where $\mathbf{c}_k$ is known for $k = 1, \cdots, K$ with $K = \sum_{k=1}^{K_2}(\bar{L}_k - 1)$. Here $\bar{L}_k = L_k$ if $\mathbf{h}_k$ has a positive element, and $\bar{L}_k = N$ if $\mathbf{h}_k$ has no positive element.

The algorithm to find $\log|\mathbf{x}|$ satisfying (2.37) for all k is similar to that for (2.32), which consists of "initialization/averaging" and "refinement". Knowing $\log|\mathbf{x}|$, we also know $|\mathbf{x}|$. Examples of the attack results are shown in Tables 2.3 and 2.4 where $p = N$ and all entries of $\mathbf{x}$ are assumed to be positive. The above analysis shows that IoM-2 effectively extracts out a binary (sign) secret from each element of $\mathbf{x}$ and utilizes that secret to construct its output. Other than that secret, IoM-2 is not a hard-to-invert function. In other words, IoM-2 can be inverted with a complexity order no larger than $L_{N,K_2}2^N$ where L_{N,K_2} is a linear function of N and K_2, respectively, and 2^N is due to an exhaustive search of the sign of each element in $\mathbf{x}$. Note that if an additional key S_x of N bits is first extracted with 100% reliability from the signs of the elements in $\mathbf{x}$, then a linear CEF could be used while maintaining an attack complexity order equal to $O(N^3 2^N)$.

Table 2.3: Normalized projection of $|\mathbf{x}|$ onto its estimate using only averaging for attack of IoM-2.

K_2	8	16	32	64
$N = 8$	0.9244	0.9540	0.9698	0.9783
$N = 16$	0.9068	0.9418	0.9603	0.9694
$N = 32$	0.8844	0.9206	0.9379	0.9466

Table 2.4: Normalized projection of $|\mathbf{x}|$ onto its estimate after convergence of refinement for attack of IoM-2.

K_2	8	16	32	64
$N = 8$	0.9432	0.9711	0.9802	0.9816
$N = 16$	0.9182	0.9525	0.9649	0.9653
$N = 32$	0.8887	0.9258	0.9403	0.9432

2.5 Qualities of a Good CEF

We propose to measure the primary qualities of a CEF $y_k = f_k(\mathbf{x})$ by the following criteria:

1. **(Hardness to invert)** If $\mathbf{x}$ can be computed (up to a desired precision) from $\{y_k, k \geq 1\}$ with a complexity order that is a polynomial function of N, the CEF is said to be easy (or not hard) to invert. Otherwise, the CEF is said to be hard to invert, which is desired for a good CEF.

2. **(Hardness to substitute)** If there are such functions g_k that $f_k(\mathbf{x}) = g_k(s(\mathbf{x}))$ for all $k \geq 1$ where $s(\mathbf{x})$ is a function of $\mathbf{x}$ and invariant to k, then $s(\mathbf{x})$ is said to be a substitute input of the CEF. If $s(\mathbf{x})$ is easy to compute from $\{y_k, k \geq 1\}$, then the CEF is said to be easy to substitute. Otherwise, the CEF is said to be hard to substitute, which is desired for a good CEF.

3. **(Sensitivity)** A good CEF should be sufficiently responsive to its input but not overly sensitive to small perturbations or noise in its input. The optimal benchmark of the

sensitivity to a small perturbation is the sensitivity of a unitary random projection of $\mathbf{x}$. The "noise" referred to in this chapter is the difference between two input vectors of interest.

4. **(Correlation)** Every pair of the output samples of a good CEF should have zero or near-zero correlation if $\mathbf{x}$ has the white Gaussian distribution $\mathcal{N}(0, \sigma_x^2 \mathbf{I}_N)$. If there are strong correlations among the output samples of a CEF, then the CEF is vulnerable to attacks by linear prediction (i.e., y_{k_0} could be estimated by a linear combination of y_k with $k < k_0$).

5. **(Invariance)** The statistical distribution of y_k for a good CEF should be invariant or nearly invariant to k if $\mathbf{x}$ is of $\mathcal{N}(0, \sigma_x^2 \mathbf{I}_N)$. One benefit from the invariance is that it makes quantization of y_k for all k easier (i.e., a good quantizer for y_{k_0} would be equally good for y_k for all $k \neq k_0$).

If a CEF meets all of the above criteria, the CEF is said to be a good CEF. A good CEF can be viewed as a generator of quasi-continuous pseudorandom numbers (QPRNs). These QPRNs are based on a continuous feature vector $\mathbf{x}$ as its "seed", which is different from the traditional PRN generators that rely on a discrete seed. It seems not possible to prove whether a CEF is hard to invert or hard to substitute although one can try to prove that a CEF is not hard to invert or not hard to substitute. This is an open problem similar to that of discrete one-way functions [1, 43, 44] even though the use of discrete one-way functions in practice is indispensable. We will say that a CEF is empirically hard to attack if there is a strong empirical evidence suggesting that the CEF is hard to invert and hard to substitute.

2.6 Proposed Continuous Encryption Function

We propose an encryption function $\mathbf{y} = \mathcal{F}(\mathbf{Q}, \mathbf{x})$ where all $\mathbf{y}$ and $\mathbf{x}$ are continuous in nature. The $\mathbf{Q}$ is a publicly known set of unitary matrices and $\mathbf{x} \in \mathcal{R}^{N\times 1}$ is the shared secret vector between two legitimate parties. The objective of the function is such that obtaining $\mathbf{y}$ from $\mathbf{Q}$ and $\mathbf{x}$ should be straight forward but obtaining $\mathbf{x}$ using $\mathbf{y}$ and $\mathbf{Q}$ (to predict future values of $\mathbf{y}$ from other given set of $\mathbf{Q}$) is not. The construction of the proposed CEF is given below:

step 1:

For each pair of k and l, let $\mathbf{Q}_{k,l}$ be a random $N \times N$ unitary matrix. Now, we define:

$$\mathbf{M}_{k,x} = [\mathbf{Q}_{k,1}\mathbf{x}, \cdots, \mathbf{Q}_{k,N}\mathbf{x}] \tag{2.38}$$

Here each column of $\mathbf{M}_{k,x}$ is a random rotation of $\mathbf{x}$.

step 2:

Let $\mathbf{u}_{k,x,1}$ be the principal left singular vector of $\mathbf{M}_{k,x}$, i.e.,

$$\mathbf{u}_{k,x,1} = \arg \max_{\mathbf{u},\|\mathbf{u}\|=1} \mathbf{u}^T \mathbf{M}_{k,x}\mathbf{M}_{k,x}^T\mathbf{u} \tag{2.39}$$

Then for each k, choose N_y $(1 \leq N_y < N)$ elements in $\mathbf{u}_{k,x,1}$ to be N_y elements in $\mathbf{y} = [y_1, y_2, \cdots]^T$. If we choose $N_y = 1$, then y_k for each k is an entry (such as the 1st entry) of $\mathbf{u}_{k,x,1}$. We will refer to the above function (from $\mathbf{x}$ to $\mathbf{y}$) as SVD-CEF. Note that there are efficient ways to perform the forward computation needed for (2.39) given $\mathbf{M}_{k,x}\mathbf{M}_{k,x}^T$. One of them is the power method [41], which has complexity equal to $\mathcal{O}(N^2)$. But the construction

of $\mathbf{M}_{k,x}\mathbf{M}_{k,x}^T$ (starting from the generation of $\mathbf{Q}_{k,1}, \cdots, \mathbf{Q}_{k,N}$) for each k requires $\mathcal{O}(N^3)$ complexity.

We can see that for each random realization of $\mathbf{Q}_{k,l}$ for all k and l and a random realization $\mathbf{x}_0$ of $\mathbf{x}$, with probability one there is a neighborhood around $\mathbf{x}_0$ within which $\mathbf{y}$ is a continuous function of $\mathbf{x}$. It is also clear that for any fixed $\mathbf{x}$ the elements in $\mathbf{y}$ appear random to anyone who does not have access to the secret key used to produce the pseudorandom $\mathbf{Q}_{k,l}$.

More importantly, we will show in Section 2.7 that SVD-CEF is empirically hard to attack even with $\mathbf{Q}_{k,l}$ known for all k and l; and in Section 2.8 that if $\mathbf{x}$ consists of i.i.d. $\mathcal{N}(0, \sigma_x^2)$, then all entries of $\mathbf{y} = [y_1, y_2, \cdots]^T$ have nearly zero correlations and the same distribution even with $\mathbf{Q}_{k,l}$ being fixed for all k and l. The noise sensitivity of SVD-CEF is also discussed in Section 2.8.

2.7 Attack on SVD-CEF

We now consider how to compute $\mathbf{x} \in \mathcal{R}^{N\times 1}$ from a given $\mathbf{y} \in \mathcal{R}^{M\times 1}$ with $M \geq N$ for SVD-CEF based on (2.38) and (2.39) assuming that $\mathbf{Q}_{k,l}$ for all k and l are given. A universal method for inverting a function is via exhaustive search, i.e., searching for an $\mathbf{x}$ that produces the known $\mathbf{y}$ via the forward function up to a desired precision. This method has a complexity order no less than $O(2^{NN_B})$ with N_B being an effective number of bits needed to represent each of the N elements in $\mathbf{x}$. The value of N_B depends on an expected noise level in $\mathbf{x}$. It is not uncommon in practice that N_B ranges from 3 to 8 or even higher.

The only other *known* method that we know to invert SVD-CEF is the Newton's method, which is considered next. To prepare for the application of the Newton's method, we need to formulate a set of equations which must be satisfied by all unknown variables.

2.7.1 Preparation

We now assume that for each of $k = 1, \cdots, K$, N_y elements of $\mathbf{u}_{k,x,1}$ are used to construct $\mathbf{y} \in \mathcal{R}^{M\times 1}$ with $M = KN_y$. Computing $\mathbf{x}$ from $\mathbf{y}$ and $\mathbf{Q}_{k,l}$ for all k and l is equivalent to solving the following eigenvalue-decomposition (EVD) equations:

$$\mathbf{M}_{k,x}\mathbf{M}_{k,x}^T\mathbf{u}_{k,x,1} = \sigma_{k,x,1}^2\mathbf{u}_{k,x,1} \tag{2.40}$$

with $k = 1, \cdots, K$. Here $\sigma_{k,x,1}^2$ is the principal eigenvalue of $\mathbf{M}_{k,x}\mathbf{M}_{k,x}^T$. But this is not a conventional EVD problem because the vector $\mathbf{x}$ inside $\mathbf{M}_{k,x}$ is unknown along with $\sigma_{k,x,1}^2$ and $N - N_y$ elements in $\mathbf{u}_{k,x,1}$ for each k. We will refer to (2.40) as the EVD equilibrium conditions for $\mathbf{x}$.

If the unknown $\mathbf{x}$ is multiplied by α, so should be the corresponding unknowns $\sigma_{k,x,1}$ for all k but $\mathbf{u}_{k,x,1}$ for any k is not affected. So, we will only need to consider the solution satisfying $\|\mathbf{x}\|^2 = 1$, i.e., $\mathbf{x} \in S^{N-1}(1)$.

The number of unknowns in the system of nonlinear equations (2.40) is

$$N_{\text{unk,EVD,1}} = N + (N - N_y)K + K, \tag{2.41}$$

which consists of all N elements of $\mathbf{x}$, $N - N_y$ elements of $\mathbf{u}_{k,x,1}$ for each k, and $\sigma_{k,x,1}^2$ for all k. The number of the nonlinear equations is

$$N_{\text{equ,EVD,1}} = NK + K + 1, \tag{2.42}$$

which consists of (2.40) for all k, $\|\mathbf{u}_{k,x,1}\| = 1$ for all k, and $\|\mathbf{x}\|^2 = 1$. Then, the necessary condition for a finite set of solutions is $N_{\text{equ,EVD,1}} \geq N_{\text{unk,EVD,1}}$ or equivalently $N_y K \geq N-1$.

If $N_y < N$, there are $N - N_y$ unknowns in $\mathbf{u}_{k,x,1}$ for each k and hence the left side of(2.40) is a third-order function of unknowns. To reduce the nonlinearity, we can expand the space of unknowns as follows. Since

$$\mathbf{M}_{k,x}\mathbf{M}_{k,x}^T = \sum_{l=1}^{N} \mathbf{Q}_{k,l}\mathbf{X}\mathbf{Q}_{k,l}^T \tag{2.43}$$

with $\mathbf{X} = \mathbf{x}\mathbf{x}^T$ (a substitute input), we can treat $\mathbf{X}$ as an $N \times N$ symmetric unknown matrix (without the rank-1 constraint), and rewrite (2.40) as

$$\left(\sum_{l=1}^{N} \mathbf{Q}_{k,l}\mathbf{X}\mathbf{Q}_{k,l}^T\right) \mathbf{u}_{k,x,1} = \sigma_{k,x,1}^2 \mathbf{u}_{k,x,1} \tag{2.44}$$

with $\text{Tr}(\mathbf{X}) = 1$, $\|\mathbf{u}_{k,x,1}\| = 1$, and $k = 1, \cdots, K$. In this case, both sides of (2.44) are of the 2nd order of all unknowns. But the number of unknowns is now

$$N_{\text{unk,EVD,2}} = \frac{1}{2}N(N+1) + (N - N_y)K + K > N_{\text{unk,EVD,1}} \tag{2.45}$$

while the number of equations is not changed, i.e.,

$$N_{\text{equ,EVD,2}} = N_{\text{equ,EVD,1}} = NK + K + 1. \tag{2.46}$$

In this case, the necessary condition for a finite set of solutions for $\mathbf{X}$ is $N_{\text{equ,EVD,2}} \geq N_{\text{unk,EVD,2}}$ or equivalently $N_y K \geq \frac{1}{2}N(N+1) - 1$.

Note that $\mathbf{X}$ seems the only useful substitute for $\mathbf{x}$. But this substitute still seems hard to compute from $\mathbf{y}$ as shown later. Alternatively, we know that $\mathbf{x}$ satisfies the following SVD equations:

$$\mathbf{M}_{k,x}\mathbf{V}_{k,x} = \mathbf{U}_{k,x}\mathbf{\Sigma}_{k,x} \tag{2.47}$$

with $\mathbf{U}_{k,x}^T\mathbf{U}_{k,x} = \mathbf{I}_N$ and $\mathbf{V}_{k,x}^T\mathbf{V}_{k,x} = \mathbf{I}_N$. Here $\mathbf{U}_{k,x}$ is the matrix of all left singular vectors, $\mathbf{V}_{k,x}$ is the matrix of all right singular vectors, and $\mathbf{\Sigma}_{k,x}$ is the diagonal matrix of all singular values. The above equations are referred to as the SVD equilibrium conditions on $\mathbf{x}$.

With N_y elements of the first column of $\mathbf{U}_{k,x}$ for each k to be known, the unknowns are the vector $\mathbf{x}$, $N^2 - N_y$ elements in $\mathbf{U}_{k,x}$ for each k, all N^2 elements in $\mathbf{V}_{k,x}$ for each k, and all diagonal elements in $\mathbf{\Sigma}_{k,x}$ for each k. Then, the number of unknowns is now

$$N_{\text{unk,SVD}} = N + (N^2 - N_y)K + N^2K + NK, \tag{2.48}$$

and the number of equations is

$$N_{\text{equ,SVD}} = N^2K + N(N+1)K + 1. \tag{2.49}$$

In this case, $N_{\text{equ,SVD}} \geq N_{\text{unk,SVD}}$ iff $N_yK \geq N - 1$. This is the same condition as that for EVD equilibrium. But the SVD equilibrium equations in (2.47) are all of the second order. Note that for the EVD equilibrium, there is no coupling between different eigen-components. But for the SVD equilibrium, there are couplings among all singular-components. Hence the latter involves a much larger number of unknowns than the former. Specifically,

$$N_{\text{unk,SVD}} > N_{\text{unk,EVD,2}} > N_{\text{unk,EVD,1}}. \tag{2.50}$$

Every set of equations that $\mathbf{x}$ must fully satisfy (given $\mathbf{y}$) is a set of nonlinear equations, regardless of how the parameterization is chosen. This seems to be the fundamental reason why SVD-CEF is hard to invert. SVD is a three-factor decomposition of a real-valued matrix, for which there are efficient ways for forward computations but no easy way for backward computation. If a two-factor decomposition of a real-valued matrix (such

as QR decomposition) is used, the hard-to-invert property does not seem achievable. In section 2.10.1, the details of an attack algorithm based on Newton's method are given.

2.7.2 Performance of Attack Algorithm

Since the conditions useful for attack of SVD-CEF are always nonlinear, any attack algorithm with a random initialization $\mathbf{x}'$ can converge to the true vector $\mathbf{x}$ (or its equivalent which produces the same $\mathbf{y}$) only if $\mathbf{x}'$ is close enough to $\mathbf{x}$. To translate the local convergence into a computational complexity needed to successfully obtain $\mathbf{x}$ from $\mathbf{y}$, we now consider the following.

Let $\mathbf{x}$ be an N-dimensional unit-norm vector of interest. Any unit-norm initialization of $\mathbf{x}'$ can be written as

$$\mathbf{x}' = \pm\sqrt{1-r^2}\mathbf{x} + r\mathbf{w} \tag{2.51}$$

where $0 < r \leq 1$ and $\mathbf{w}$ is a unit-norm vector orthogonal to $\mathbf{x}$. For any $\mathbf{x}$, $r\mathbf{w}$ is a vector (or "point") on the sphere of dimension $N-2$ and radius r, denoted by $S^{N-2}(r)$. The total area of $S^{N-2}(r)$ is known to be

$$|S^{N-2}(r)| = \frac{2\pi^{\frac{N-1}{2}}}{\Gamma\left(\frac{N-1}{2}\right)} r^{N-2}. \tag{2.52}$$

Then the probability for a uniformly random $\mathbf{x}'$ from $S_{N-1}(1)$ to fall onto $S_{N-2}(r_0)$ orthogonal to $\sqrt{1-r_0^2}\mathbf{x}$ with $r \leq r_0 \leq r+dr$ is

$$2\frac{|S^{N-2}(r)|}{|S^{N-1}(1)|}dr \tag{2.53}$$

where the factor 2 accounts for $\pm$ in (2.51). Therefore, the probability of convergence from $\mathbf{x}'$ to $\mathbf{x}$ is:

$$P_{\text{conv}} = E_{\mathbf{x}} \left\{ \int_0^1 2P_{x,r} \frac{|S^{N-2}(r)|}{|S^{N-1}(1)|} dr \right\} \tag{2.54}$$

$$= \frac{\Gamma(\frac{N}{2})}{2\sqrt{\pi}\Gamma(\frac{N-1}{2})} \int_0^1 P_r r^{N-2} dr \tag{2.55}$$

where $E_{\mathbf{x}}$ is the expectation over $\mathbf{x}$, $P_{x,r}$ is the probability of convergence from $\mathbf{x}'$ to $\mathbf{x}$ when $\mathbf{x}'$ is chosen randomly from $S^{N-2}(r)$ orthogonal to a given $\sqrt{1-r^2}\mathbf{x}$, and $E_{\mathbf{x}}\{P_{x,r}\} = P_r$.

We see that P_r is the probability that the algorithm converges from $\mathbf{x}'$ to $\mathbf{x}$ (including its equivalent) subject to a fixed r, uniformly random unit-norm $\mathbf{x}$, and uniformly random unit-norm $\mathbf{w}$ satisfying $\mathbf{w}^T\mathbf{x} = 0$. And P_r can be estimated via simulation. Let $r_{\max} < 1$ be such that $P_r = 0$ for $r \geq r_{\max}$. Then

$$P_{\text{conv}} = \frac{\Gamma(\frac{N}{2})}{2\sqrt{\pi}\Gamma(\frac{N-1}{2})} \int_0^{r_{max}} P_r r^{N-2} dr \tag{2.56}$$

$$< \frac{2\Gamma(N/2)}{(N-1)\sqrt{\pi}\Gamma(\frac{N-1}{2})} r_{\max}^{N-1} < r_{\max}^{N-1}. \tag{2.57}$$

which converges to zero exponentially as N increases. In other words, for such an algorithm to find $\mathbf{x}$ or its equivalent from random initializations has a complexity order equal to $\mathcal{O}\left\{\frac{1}{P_{conv}}\right\} > \mathcal{O}\left\{(\frac{1}{r_{max}})^{N-1}\right\}$.

Table 2.5: $P_{r,N}$ and $P^*_{r,N}$ versus r and N.

r	0.001	0.01	0.1	0.3	0.5	0.7	0.9	1
$P_{r,4}$	0.46	0.24	0.06	0	0.01	0.01	0.01	0
$P^*_{r,4}$	0.45	0.17	0.04	0	0.01	0	0.01	0
$P_{r,8}$	0.29	0.07	0.01	0	0	0	0	0
$P^*_{r,8}$	0.25	0.05	0	0	0	0	0	0

In our simulation, we have found that $r_{\max}$ decreases rapidly as N increases. Let $P_{r,N}$ be P_r as a function of N. Also let $P^*_{r,N}$ be the probability of convergence to $\hat{\mathbf{x}}$ which via SVD-CEF not only yields the correct y_k for $k = 1, \cdots, K$ but also the correct y_k for $k > K$ (up to maximum absolute element-wise error no larger than 0.02). Here K is the number of output elements used to compute the input vector $\mathbf{x}$. In the simulation, we chose $N_y = 1$ and $N_{\text{equ,EVD,2}} = N_{\text{unk,EVD,2}} + 1$, which is equivalent to $K = \frac{1}{2}N(N+1)$.

Shown in Table 2.5 are the percentage values of $P_{r,N}$ versus r and N, which are based on 100 random choices of $\mathbf{x}$. For each choice of $\mathbf{x}$ and each value of r, we used one random initialization of $\mathbf{x}'$. (For $N = 8$ and the values of r in this table, it took two days on a PC with CPU 3.4 GHz Dual Core to complete the 100 runs.)

The above discussions have explained why SVD-CEF is empirically hard to attack. Next, we will discuss the sensitivity, correlation, and invariance of SVD-CEF.

2.8 Statistical Properties of SVD-CEF

In this section, we show a statistical study of SVD-CEF to understand some of the statistical properties of its output. Since each entry of the output $\mathbf{y} = [y_1, y_2, \cdots, y_M]^T$ of SVD-CEF is an element in the principal eigenvector $\mathbf{u}_{k,x,1}$ of the matrix $\mathbf{M}_{k,x}\mathbf{M}_{k,x}^T$, we can mostly focus on the statistics of $\mathbf{u}_{k,x,1}$.

2.8.1 Sensitivity

Unlike unitary random projections, relationship between the normalized distance at the input $\frac{1}{\sqrt{N}}||\Delta\mathbf{x}||$ and the normalized distance at the output $\frac{1}{\sqrt{M}}||\Delta\mathbf{y}||$ is not trivial.

Sensitivity to small perturbation

We now consider the sensitivity of SVD-CEF to a small perturbation, i.e., the relationship between the differential $\partial \mathbf{u}_{k,x,1}$ (or a corresponding ∂y_k) and the differential $\partial \mathbf{x}$. It follows from [45] that

$$\partial \mathbf{u}_{k,x,1} = \sum_{j=2}^{N} \frac{1}{\lambda_1 - \lambda_j} \mathbf{u}_{k,x,j} \mathbf{u}_{k,x,j}^T \partial(\mathbf{M}_{k,x} \mathbf{M}_{k,x}^T) \mathbf{u}_{k,x,1} \tag{2.58}$$

where λ_j is the j-th eigenvalue of $\mathbf{M}_{k,x}\mathbf{M}_{k,x}^T$, and $\mathbf{u}_{k,x,j}$ is the corresponding j-th eigenvector. Since

$$\mathbf{M}_{k,x}\mathbf{M}_{k,x}^T = \sum_{l=1}^{N} \mathbf{Q}_{k,l} \mathbf{x}\mathbf{x}^T \mathbf{Q}_{k,l}^T, \tag{2.59}$$

$$\partial(\mathbf{M}_{k,x}\mathbf{M}_{k,x}^T) = \sum_{l} \mathbf{Q}_{k,l} \partial(\mathbf{x}\mathbf{x}^T) \mathbf{Q}_{k,l}^T + \sum_{l} \mathbf{Q}_{k,l} \mathbf{x} \partial \mathbf{x}^T \mathbf{Q}_{k,l}^T \tag{2.60}$$

It follows that $\partial \mathbf{u}_{k,x,1} = \mathbf{T} \partial \mathbf{x}$ where $\mathbf{T} = \mathbf{A} + \mathbf{B}$ with

$$\mathbf{A} = \sum_{j=2}^{N} \frac{1}{\lambda_1 - \lambda_j} \mathbf{u}_{k,x,j} \mathbf{u}_{k,x,j}^T \sum_{l=1}^{N} \mathbf{Q}_{k,l} \mathbf{x}^T \mathbf{Q}_{k,l}^T \mathbf{u}_{k,x,1} \tag{2.61}$$

$$\mathbf{B} = \sum_{j=2}^{N} \frac{1}{\lambda_1 - \lambda_j} \mathbf{u}_{k,x,j} \mathbf{u}_{k,x,j}^T \sum_{l=1}^{N} \mathbf{Q}_{k,l} \mathbf{x} \mathbf{u}_{k,x,1}^T \mathbf{Q}_{k,l} \tag{2.62}$$

We can also write

$$\mathbf{T} = \left(\sum_{j=2}^{N} \frac{1}{\lambda_1 - \lambda_j} \mathbf{u}_{k,x,j} \mathbf{u}_{k,x,j}^T \right) \cdot \left(\sum_{l=1}^{N} \mathbf{Q}_{k,l} \left[\left(\mathbf{x}^T \mathbf{Q}_{k,l}^T \mathbf{u}_{k,x,1} \right) \mathbf{I}_N + \mathbf{x} \mathbf{u}_{k,x,1}^T \mathbf{Q}_{k,l} \right] \right) \tag{2.63}$$

where the first matrix component has the rank $N-1$ and hence so does $\mathbf{T}$. Let $\partial \mathbf{x} = \mathbf{w}$ which consists of i.i.d. elements with zero mean and variance $\sigma_w^2 \ll 1$. It then follows that

$$E_{\mathbf{w}}\{\|\partial \mathbf{u}_{k,x,1}\|^2\} = \mathrm{Tr}\{\mathbf{T} \sigma_w^2 \mathbf{T}^T\} = \sigma_w^2 \sum_{j=1}^{N-1} \sigma_j^2 \tag{2.64}$$

where σ_j for $j = 1, \cdots, N-1$ are the nonzero singular values of $\mathbf{T}$ and $E_{\mathbf{w}}\{\cdot\}$ is the Expectation taken over $\mathbf{w}$. Since $E_{\mathbf{w}}\{\|\partial\mathbf{x}\|^2\} = N\sigma_w^2$, we have

$$\eta_{k,x}^2 = \frac{E_{\mathbf{w}}\{\|\partial\mathbf{u}_{k,x,1}\|^2\}}{E_{\mathbf{w}}\{\|\partial\mathbf{x}\|^2\}} = \frac{1}{N}\sum_{j=1}^{N-1}\sigma_j^2 \tag{2.65}$$

which measures the sensitivity of $\mathbf{u}_{k,x,1}$ to a small perturbation in $\mathbf{x}$. Since each of the N entries in $\partial\mathbf{u}_{k,x,1}$ has the same variance due to symmetry, then the corresponding ∂y_k satisfies

$$E_{\mathbf{w}}\{\|\partial y_k\|^2\} = \frac{1}{N}E_{\mathbf{w}}\{\|\partial\mathbf{u}_{k,x,1}\|^2\}. \tag{2.66}$$

Since both $\mathbf{x}$ and $\mathbf{u}_{k,x,1}$ have unit norm, the input SNR of SVD-CEF is

$$\mathrm{SNR}_x = \frac{1}{E_{\mathbf{w}}\{\|\partial\mathbf{x}\|^2\}} = \frac{1}{N\sigma_w^2}, \tag{2.67}$$

and the output SNR of SVD-CEF for y_k is

$$\mathrm{SNR}_{y,k} = \mathcal{O}\left(\frac{1}{NE_{\mathbf{w}}\{\|\partial y_k\|^2\}}\right) = \mathcal{O}\left(\frac{1}{E_{\mathbf{w}}\{\|\partial\mathbf{u}_{k,x,1}\|^2\}}\right) \tag{2.68}$$

Therefore, the FoM of SVD-CEF for y_k is

$$\sqrt{\frac{\mathrm{SNR}_x}{\mathrm{SNR}_{y,k}}} = \mathcal{O}(\eta_{k,x}). \tag{2.69}$$

Here $\mathcal{O}$ denotes the order as $\sigma_w^2 \to 0$.

For each given $\mathbf{x}$, there is a small percentage of realizations of $\{\mathbf{Q}_{k,l}, l = 1, \cdots, N\}$ that make $\eta_{k,x}$ relatively large. To reduce $\eta_{k,x}$, we can prune away such bad realizations.

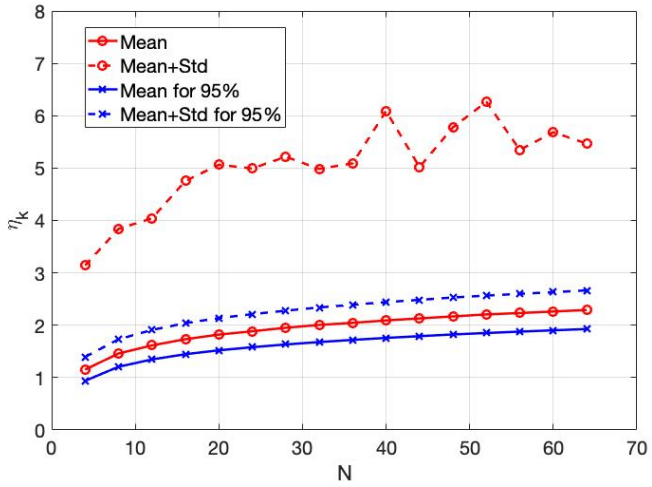


Figure 2.3: The mean and mean-plus-deviation of $\eta_{k,x}$ versus N. The red plots correspond to un-pruned $\eta_{k,x}$ and the blue plots correspond to 5% pruned $\eta_{k,x}$

Shown in Fig. 2.3 are the means and means-plus-deviations of $\eta_{k,x}$ (over choices of k and $\mathbf{x}$) versus N, with and without pruning respectively. Here "std" stands for standard deviation. We see that 5% pruning (or equivalently 95% inclusion shown in the figure) results in a substantial reduction of $\eta_{k,x}$. We used 1000×1000 realizations of $\mathbf{x}$ and $\{\mathbf{Q}_{k,l}, l = 1, \cdots, N\}$. Shown in Table 2.6 are statistics of $\eta_{k,x}$ subject to $\eta_{k,x} < 2.5$ where P_{good} is the probability of $\eta_{k,x} < 2.5$. We see that P_{good} is relatively large at around or above 80% and the mean of $\eta_{k,x}$ ranges roughly from 1.3 to 1.6 for $N = 16, 32, 64$.

Table 2.6: Statistics of $\eta_{k,x}$ subject to $\eta_{k,x} < 2.5$ and P_{good}

N	16	32	64
Mean	1.325	1.489	1.645
Std	0.414	0.397	0.371
P_{good}	0.88	0.84	0.78

This noise sensitivity is far from perfect when compared to the unitary random projection. But SVD-CEF has the hard-to-attack property as empirically established earlier.

Sensitivity to large perturbation

Any unit-norm vector $\mathbf{x}'$ can be written as $\mathbf{x}' = \pm\sqrt{1-\alpha}\mathbf{x}+\sqrt{\alpha}\mathbf{w}$ where $0 \leq \alpha \leq 1$, and $\mathbf{w}$ is of unit norm and satisfies $\mathbf{w}^T\mathbf{x} = 0$. Then

$$\|\Delta\mathbf{x}\| = \|\mathbf{x} - \mathbf{x}'\| = \sqrt{2 - 2\sqrt{1-\alpha}} \tag{2.70}$$

It follows that

$$\|\Delta\mathbf{x}\| \leq \sqrt{2} \quad \text{and} \quad \|\Delta\mathbf{u}_{k,x,1}\| \leq \sqrt{2} \tag{2.71}$$

For given α in $\mathbf{x}' = \pm\sqrt{1-\alpha}\mathbf{x} + \sqrt{\alpha}\mathbf{w}$, $\|\Delta\mathbf{x}\|$ is given while $\|\Delta\mathbf{u}_{k,x,1}\|$ still depends on $\mathbf{w}$. We can call $\|\Delta\mathbf{u}_{k,x,1}^T\|/\|\Delta\mathbf{x}\|$ a deviation gain of SVD-CEF, which is dependent on $\mathbf{x}$, k, and $\|\Delta\mathbf{x}\|$. Here a different k means a different set of $\{\mathbf{Q}_{k,l}, l = 1, \cdots, N\}$. Shown in Fig. 2.4 are the means and means-plus-deviations of the deviation gain versus $\|\Delta\mathbf{x}\|$ subject to $\eta_{k,x} < 2.5$. This figure is based on 1000×1000 realizations of $\mathbf{x}$ and $\{\mathbf{Q}_{k,l}, l = 1, \cdots, N\}$. We see that the mean of the deviation gain is somewhat constant and comparable to the mean of $\eta_{k,x}$ for $\|\Delta\mathbf{x}\| < 0.1$.

2.8.2 Correlation

We show below via simulation that the correlation between the input and output of SVD-CEF as well as the correlation among the output samples of SVD-CEF are practically zero.

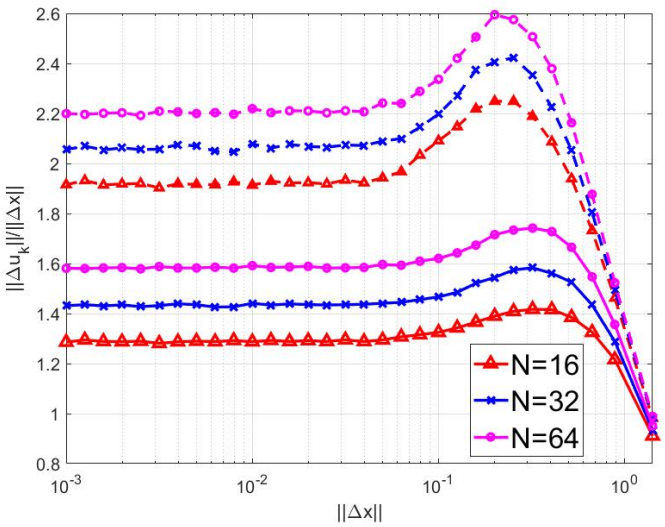


Figure 2.4: The means (lower three curves) and means-plus-deviations (upper three curves) of $\frac{||\Delta \mathbf{u}_{k,x,1}||}{||\Delta \mathbf{x}||}$ subject to $\eta_{k,x} < 2.5$.

Correlation between input and output

Recall $\mathbf{M}_{k,x} = [\mathbf{Q}_{k,1}\mathbf{x}, \cdots, \mathbf{Q}_{k,N}\mathbf{x}]$. If there is a secret key, then $\mathbf{Q}_{k,l}$ for all k and l are uniformly random unitary matrices (from the adversary's perspective). Then $\mathbf{u}_{k,x,1}$ for all k and any $\mathbf{x}$ are uniformly random on $S^{N-1}(1)$. It follows that $E_{\mathbf{Q}}\{\mathbf{u}_{k,x,1}\mathbf{u}_{m,x,1}^T\} = 0$ for $k \neq m$, and $E_{\mathbf{Q}}\{\mathbf{u}_{k,x,1}\mathbf{x}^T\} = 0$. Furthermore, it can be shown that $E_{\mathbf{Q}}\{\mathbf{u}_{k,x,1}\mathbf{u}_{k,x,1}^T\} = \frac{1}{N}\mathbf{I}_N$, i.e., the entries of $\mathbf{u}_{k,x,1}$ are uncorrelated with each other. Here $E_{\mathbf{Q}}$ denotes the expectation over the distributions of $\mathbf{Q}_{k,l}$.

If there is no secret key, then $\mathbf{Q}_{k,l}$ for all k and l must be treated as known. We will consider typical random realizations of $\mathbf{Q}_{k,l}$ for all k and l, which exclude those (such as $\mathbf{Q}_{k,l} = \mathbf{Q}_{k',l'}$ for some $k' \neq k$ or $l' \neq l$) that would occur with extremely small probability. To understand the correlation between $\mathbf{x} \in S^{N-1}(1)$ and $\mathbf{u}_{k,x,1} \in S^{N-1}(1)$ subject to a fixed set of $\mathbf{Q}_{k,l}$, we consider the following measure:

$$\rho_k = N \max_{i,j} \left| [E_{\mathbf{x}}\{\mathbf{x}\mathbf{u}_{k,x,1}^T\}]_{i,j} \right| \tag{2.72}$$

where $E_{\mathbf{x}}$ denotes the expectation over the distribution of $\mathbf{x}$. If $\mathbf{u}_{k,x,1} = \mathbf{x}$, then $\rho_k = 1$. So, if the correlation between $\mathbf{x}$ and $\mathbf{u}_{k,x,1}$ is small, so should be ρ_k. For comparison, we define ρ_k^* as ρ_k with $\mathbf{u}_{k,x,1}$ replaced by a random unit-norm vector (independent of $\mathbf{x}$).

For a different k, there is a different realization of $\mathbf{Q}_{k,1}, \cdots, \mathbf{Q}_{k,N}$. Hence, ρ_k changes with k. Shown in Fig. 2.5 are the mean and mean $\pm$ deviation of ρ_k and ρ_k^* versus N subject to $\eta_{k,x} < 2.5$. We used 10000×100 realizations of $\mathbf{x}$ and $\{\mathbf{Q}_{k,1}, \cdots, \mathbf{Q}_{k,N}\}$. We see that ρ_k and ρ_k^* have virtually the same mean and deviation. (Without the constraint $\eta_{k,x} < 2.5$, ρ_k and ρ_k^* match even better with each other.) In other words, the correlation between the input and output of SVD-CEF is virtually the same as the correlation between the (unit-norm) input of SVD-CEF and a (unit-norm) random vector.

Correlation among output samples

We now consider the correlation among $y_k = f_k(\mathbf{x})$ for $k = 1, \cdots, K$ of SVD-CEF subject to $\mathbf{x}$ being $\mathcal{N}(0, \mathbf{I}_N)$ and a typical realization of $\mathbf{Q}_{k,l}$ for $k = 1, \cdots, K$ and $l = 1, \cdots, N$. We define the following normalized sample covariance/correlation matrix:

$$\mathbf{C}_{\text{SVD-CEF},R} = N E_{\mathbf{x},R}\{\mathbf{y}_{\text{SVD-CEF}} \mathbf{y}_{\text{SVD-CEF}}^T\} \tag{2.73}$$

where $\mathbf{y}_{\text{SVD-CEF}} = [y_1, \cdots, y_K]^T$ with its k-th entry y_k being the first entry of $\mathbf{u}_{k,x,1}$, and $E_{\mathbf{x},R}$ denotes the sample average over R realizations of $\mathbf{x}$ (which treats all other quantities such as key-dependent matrices as fixed). We also define $\mathbf{C}_{\text{URP},R} = E_{x,R}\left\{\mathbf{y}_{\text{URP}} \mathbf{y}_{\text{URP}}^T\right\}$ with $\mathbf{y}_{\text{URP}}$ being a vertical stack of $\mathbf{y}_k$ in (2.27) for $k = 1, \cdots, K_0$ with $NK_0 = K$. Similarly, we

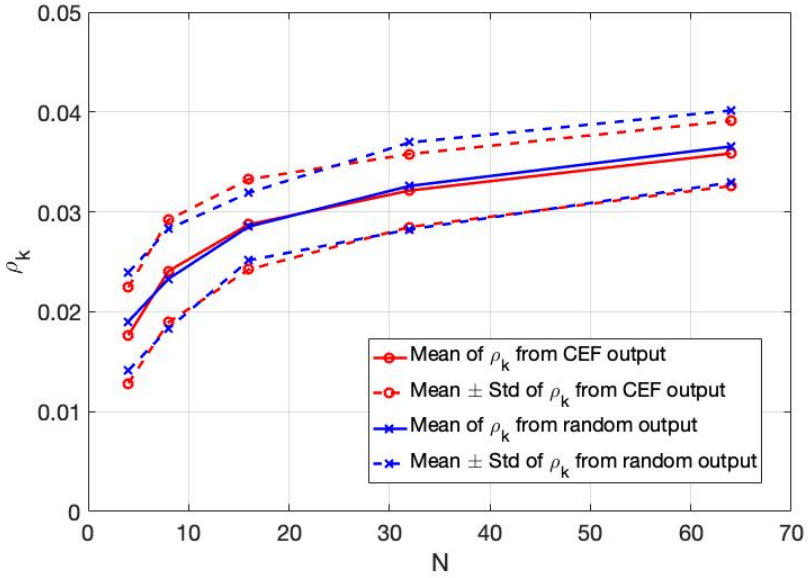


Figure 2.5: The means and means ± deviations of ρ_k (using SVD-CEF output) and ρ_k^* (using random output) versus N subject to $\eta_{k,x} < 2.5$.

let $\mathbf{C}_{\mathrm{DRP},R} = c_{\mathrm{DRP}} E_{x,R}\left\{\mathbf{y}_{\mathrm{DRP}}\mathbf{y}_{\mathrm{DRP}}^T\right\}$ and $\mathbf{C}_{\mathrm{IoM},R} = c_{\mathrm{IoM}} E_{x,R}\left\{\mathbf{y}_{\mathrm{IoM}}\mathbf{y}_{\mathrm{IoM}}^T\right\}$ where c_{DRP} and c_{IoM} are such that the diagonal elements of each of $\mathbf{C}_{\mathrm{DRP},R}$ and $\mathbf{C}_{\mathrm{IoM}}$ have their averaged value equal to one. For IoM, each entry of $\mathbf{y}_{\mathrm{IoM}}$ is an integer "index-of-max" (ranging from 0 to $N-1$) minus $\frac{N-1}{2}$, which ensures that each entry of $\mathbf{y}_{\mathrm{IoM}}$ has the zero mean.

Shown in Table 2.7 are the maximum values of the absolute off-diagonal elements of each of the above defined sample covariance matrices with $N = 16$, $K = 128$ and $R = 10^5$. The first column in Table 2.7 is for $\mathbf{C}_{\mathbf{x},R} = \mathbb{E}_{\mathbf{x},R}\{\mathbf{x}\mathbf{x}^T\}$ of $\mathbf{x} \sim \mathcal{N}(0, \mathbf{I}_N)$, which serves as a reference. We know that as $R \to \infty$, the peak sample correlation of the elements in $\mathbf{x}$ goes to zero. (The mean and deviation of each off-diagonal element of $\mathbf{C}_{\mathbf{x},R}$ are zero and $\frac{1}{\sqrt{R}}$, respectively. At $R = 10^5$, $\frac{1}{\sqrt{R}} = 0.0032$.) We see that the peak sample correlation of SVD-CEF is very small and comparable to (about 1.4 times) that of $\mathbf{x}$. On the other hand, the peak sample correlations of IoM, DRP, and URP are about 17 to 67 times larger than that

of SVD-CEF. We should stress that the values in this table will change, but only slightly with high probability, if different realizations of the random matrices and/or operations in the CEFs are used.

Illustrated in Fig. 2.6 are the "heatmaps" of the absolute values of the entries of the sample covariance matrices of SVD-CEF, IoM-2, DRP and URP, where all parameters are the same as those for Table 2.7. Each of these heatmaps is based on a random realization of their embedded pseudorandom transformations. However, the overall patterns of the heatmaps in general do not change much as these pseudorandom transformations are chosen differently. We see that the output samples of SVD-CEF have virtually zero correlations, which in fact do not differ much from the sample correlations of the entries in $\mathbf{x}$. This is because of the unique relationship between the principal eigenvector $\mathbf{u}_{k,x,1}$ of $\mathbf{M}_{k,x}\mathbf{M}_{k,x}^T$ and the input vector $\mathbf{x}$. We also see that most of the correlations of IoM-2 are also small but not as small as those of SVD-CEF and there are still a lot of scattered "peaks" in the heatmap of IoM-2, which are quite significant. The heatmaps of DRP and URP show overwhelmingly large correlation values. For URP, the sample correlations among samples within each subvector $\mathbf{y}_k$ are small in the order of $\frac{1}{\sqrt{R}}$, which is due to the unitary transformation. But the correlation between $\mathbf{y}_k$ and $\mathbf{y}_l$ for $k \neq l$ is rather large as shown in this figure, which is because of the linear nature of URP and the non-orthogonality among any set of L N-dimensional vectors with $L > N$. For the same reason, RP proposed in [25] also has a very poor property in correlation.

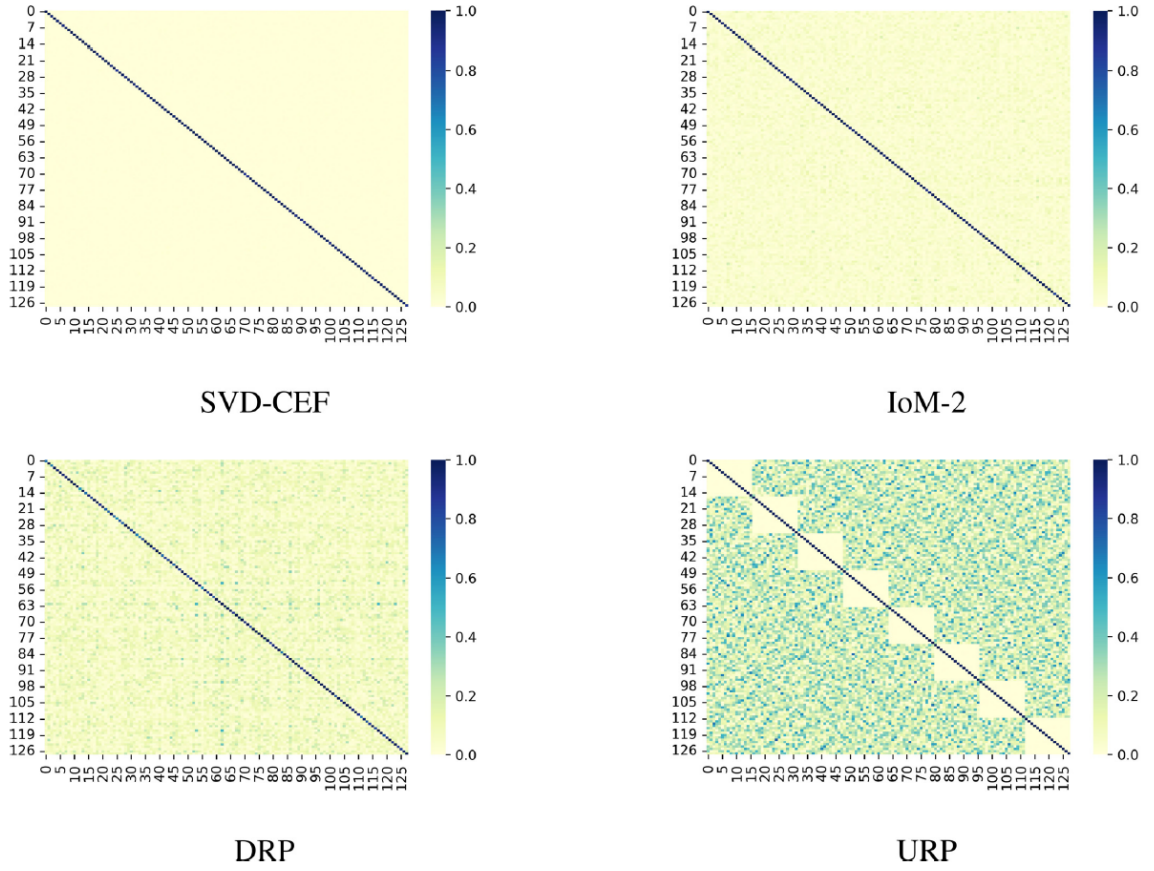


Figure 2.6: Correlation "heatmaps" of the output samples of SVD-CEF, IoM-2, DRP and URP. There is no secret key used in any of these CEFs, i.e., same set of random matrices used for different realizations of $\mathbf{x}$

Table 2.7: Maximums of absolute normalized correlations among the outputs of CEFs.

	$\mathbf{x}$	**SVD-CEF**	**IoM-2**	**IoM-1**	**DRP**	**URP**
Value	0.0085	0.012	0.21	0.25	0.49	0.81

Invariance

We show next via simulation that $\mathbf{u}_{k,x,1}$ for each k is nearly uniformly distributed on $S^{N-1}(1)$ when $\mathbf{x}$ is uniformly distributed on $S^{N-1}(1)$, which implies that y_k of SVD-CEF for each k has the same distribution (i.e., invariant to k).

To show that the distribution of $\mathbf{u}_{k,x,1}$ for each k is nearly uniform on $S^{N-1}(1)$, we show that for any k and any unit-norm vector $\mathbf{v}$, the probability density function (PDF) $p_{k,v}(x)$ of $\mathbf{v}^T\mathbf{u}_{k,x,1}$ subject to a fixed set of $\{\mathbf{Q}_{k,1}, \cdots, \mathbf{Q}_{k,N}\}$ and a uniform random $\mathbf{x}$ on $S^{N-1}(1)$ is nearly the same as the PDF $p(x)$ of an element in $\mathbf{x}$. The expression of $p(x)$ is derived in Appendix B in [46] and given by:

$$f_{x_1}(x_1) = \frac{\Gamma\frac{N}{2}}{\sqrt{\pi}\Gamma\frac{N-1}{2}}(1-x_1^2)^{\frac{N-3}{2}} \tag{2.74}$$

The distance between $p(x)$ and $p_{k,v}(x)$ can be measured by

$$D_{k,v} = \int p(x)\ln\frac{p(x)}{p_{k,v}(x)}dx \geq 0 \tag{2.75}$$

Clearly, $D_{k,v}$ changes as k and $\mathbf{v}$ change. Shown in Fig. 2.7 are the mean and mean $\pm$ deviation of $D_{k,v}$ versus N subject to $\eta_{k,x} < 2.5$. We used $50 \times 1000 \times 500$ realizations of $\mathbf{v}$, $\mathbf{x}$, and $\{\mathbf{Q}_{k,1}, \cdots, \mathbf{Q}_{k,N}\}$. We see that $D_{k,v}$ becomes very small as N increases beyond 15. This means that for a moderate or large N, $\mathbf{u}_{k,x,1}$ is (at least approximately) uniformly distributed on $S^{N-1}(1)$ when $\mathbf{x}$ is uniformly distributed on $S^{N-1}(1)$. (Without the constraint $\eta_{k,x} < 2.5$, $D_{k,\mathbf{v}}$ versus N has a similar pattern and is even slightly smaller.) In other words, for a moderate or large N, the output sample y_k of SVD-CEF for each k has a PDF approximately given by (2.74) which is invariant to k.

2.9 Conclusion

In this chapter we examined a new scheme called randomized reciprocal channel modulation (RRCM). For a MISO user channel with $N_A \geq N_B = 1$, the UNS of RRCM can be up to the entropy of N_A transmitted symbols from Alice. Furthermore, we found via

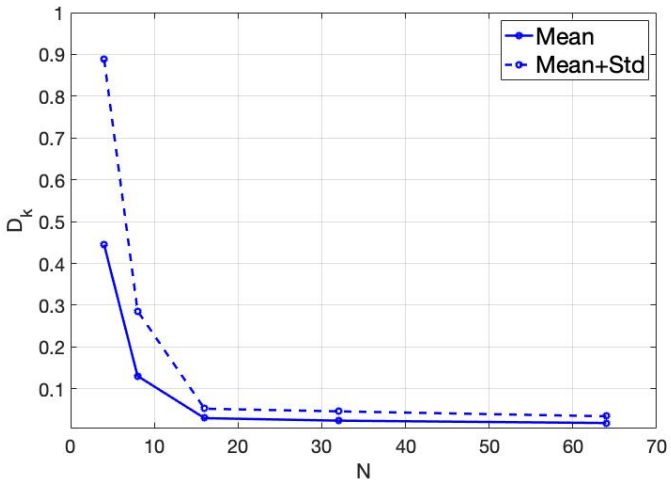


Figure 2.7: The mean and mean ±deviation of $D_{k,v}$ versus N subject to $\eta_{k,x} < 2.5$

simulation that for $N_A \geq 9$ the computational complexity for Eve to break the secrecy beyond the UNS of RRCM is infeasible on a PC with 11 Gigaflops or even on a supercomputer depending on the delay requirement.

Then, a systematic development of continuous encryption functions (CEFs) that transcend the boundaries of wireless network science and biometric data science is presented. We proposed a list of criteria for a good CEF desirable in applications, which are the hardness to invert, the hardness to substitute, the sensitivity to noise, the correlation among the output samples and the invariance of the output distributions. We have introduced a singular value decomposition (SVD) based CEF, which is shown empirically to be hard to attack. Our statistical analyses and simulation results also verified that SVD-CEF has relatively good properties in its noise sensitivity, its output correlation and the invariance of its output distribution.

2.10 Appendix

2.10.1 Newton's search algorithm to attack SVD-CEF

It is easy to verify that $\mathbf{X} = \alpha\mathbf{I}_N + (1-\alpha)\mathbf{x}\mathbf{x}^T$ with any $-\infty < \alpha < \infty$ is a solution to the following:

$$\left(\sum_{l=1}^{N} \mathbf{Q}_{k,l}\mathbf{X}\mathbf{Q}_{k,l}^T\right)\mathbf{u}_{k,x,1} = c_{k,x,1}\mathbf{u}_{k,x,1} \tag{2.76}$$

Where, $c_{k,x,1} = \alpha + (1-\alpha)\sigma_{k,x,1}^2$. To ensure that $\mathbf{u}_{k,x,1}$ from (2.76) is unique, it is sufficient and necessary to find an $\mathbf{X}$ with the above structure and $1-\alpha \neq 0$. To ensure $1-\alpha \neq 0$, we assume that $x_1 x_2 \neq 0$ where x_1 and x_2 are the first two elements of $\mathbf{x}$. Then we add the following constraints:

$$(\mathbf{X})_{1,2} = (\mathbf{X})_{2,1} = 1 \tag{2.77}$$

Which is in addition to the previous condition $Tr(\mathbf{X}) = 1$. Now for the expected solution structure $\mathbf{X} = \alpha\mathbf{I}_N + (1-\alpha)\mathbf{x}\mathbf{x}^T$, we have $1-\alpha = \frac{1}{x_1 x_2} \neq 0$

Note that $c_{k,x,1}$ in (2.76) is either the largest or the smallest eigenvalue of $\sum_{l=1}^{N} \mathbf{Q}_{k,l}\mathbf{X}\mathbf{Q}_{k,l}^T$ corresponding to whether $1-\alpha$ is positive or negative.

To develop the Newton's algorithm, we now take the differentiation of (2.76):

$$\left(\sum_{l=1}^{N} \mathbf{Q}_{k,l}\partial\mathbf{X}\mathbf{Q}_{k,l}^T\right)\mathbf{u}_{k,x,1} + \left(\sum_{l=1}^{N} \mathbf{Q}_{k,l}\mathbf{X}\mathbf{Q}_{k,l}^T\right)\partial\mathbf{u}_{k,x,1} = \partial c_k \mathbf{u}_{k,x,1} + c_k\partial\mathbf{u}_{k,x,1} \tag{2.78}$$

where we have used $\mathbf{u}_{k,x,1} = \mathbf{u}_{k,x,1}$ and $c_k = c_{k,x,1}$ for convenience. The first term is equivalent to $\tilde{\mathbf{Q}}_k\partial\tilde{\mathbf{x}}$ with $\tilde{\mathbf{Q}}_k = \left(\sum_{l=1}^{N} \mathbf{u}_{k,x,1}^T\mathbf{Q}_{k,l} \otimes \mathbf{Q}_{k,l}\right)$ and $\tilde{\mathbf{x}} = \text{vec}(\mathbf{X})$. (For basics of matrix differentiation, see [47].) Since $\mathbf{X} = \mathbf{X}^T$, there are repeated entries in $\tilde{\mathbf{x}}$. We can write $\tilde{\mathbf{x}} = [\tilde{\mathbf{x}}_1^T, \cdots, \tilde{\mathbf{x}}_N^T]^T$ with $\tilde{\mathbf{x}}_n = [\tilde{x}_{n,1}, \cdots, \tilde{x}_{n,N}]^T$ and $\tilde{x}_{i,j} = \tilde{x}_{j,i}$ for all $i \neq j$.

Let $\hat{\mathbf{x}}$ be the vectorized form of the lower triangular part of $\mathbf{X}$. Then it follows that

$$\tilde{\mathbf{Q}}_k \partial\tilde{\mathbf{x}} = \hat{\mathbf{Q}}_k \partial\hat{\mathbf{x}} \tag{2.79}$$

where $\hat{\mathbf{Q}}_k$ is a compressed form of $\tilde{\mathbf{Q}}_k$ as follows. Let $\tilde{\mathbf{Q}}_k = [\tilde{\mathbf{Q}}_{k,1}, \cdots, \tilde{\mathbf{Q}}_{k,N}]$ with $\tilde{\mathbf{Q}}_{k,n} = [\tilde{\mathbf{q}}_{k,n,1}, \cdots, \tilde{\mathbf{q}}_{k,n,N}]$. For all $1 \leq i < j \leq N$, replace $\tilde{\mathbf{q}}_{k,i,j}$ by $\tilde{\mathbf{q}}_{k,i,j} + \tilde{\mathbf{q}}_{k,j,i}$, and then drop $\tilde{\mathbf{q}}_{k,j,i}$. The resulting matrix is $\hat{\mathbf{Q}}_k$.

The differential of $\mathrm{Tr}(\mathbf{X}) = 1$ is $\mathrm{Tr}(\partial\mathbf{X}) = 0$ or equivalently $\mathbf{t}^T \partial\hat{\mathbf{x}} = 0$ where $\mathbf{t}^T = [\mathbf{t}_1^T, \cdots, \mathbf{t}_N^T]$ and $\mathbf{t}_n^T = [1, \mathbf{0}_{1\times(N-n)}]^T$.

Combining the above for all k along with $\mathbf{u}_{k,x,1}^T \partial\mathbf{u}_{k,x,1} = 0$ (due to the norm constraint $\|\mathbf{u}_{k,x,1}\|^2 = 1$) for all k, we have $\mathbf{A}_x \partial\hat{\mathbf{x}} + \mathbf{A}_u \partial\mathbf{u} + \mathbf{A}_z \partial\mathbf{z} = 0$ where:

$$\mathbf{A}_u = \begin{bmatrix} \mathbf{0}_{1\times NK} \\ \mathrm{diag}(\mathbf{G}_{1,x}, \cdots, \mathbf{G}_{K,x}) \\ \mathrm{diag}(\mathbf{u}_1^T, \cdots, \mathbf{u}_K^T) \end{bmatrix} \tag{2.80}$$

$$\mathbf{A}_x = \begin{bmatrix} \mathbf{t}^T \\ \hat{\mathbf{Q}}_1 \\ \vdots \\ \hat{\mathbf{Q}}_K \\ \mathbf{0}_{K\times\frac{1}{2}N(N+1)} \end{bmatrix} \tag{2.81}$$

$$\mathbf{A}_z = \begin{bmatrix} \mathbf{0}_{1\times K} \\ -\text{diag}(\mathbf{u}_1, \cdots, \mathbf{u}_K) \\ \mathbf{0}_{K\times K} \end{bmatrix} \tag{2.82}$$

with $\mathbf{G}_{k,x} = \mathbf{M}_{k,x}\mathbf{M}_{k,x}^T - c_k\mathbf{I}_M$.

Now we partition $\mathbf{u}$ into two parts: $\mathbf{u}_a$ (known) and $\mathbf{u}_b$ (unknown). Also partition $\mathbf{A}_u$ into $\mathbf{A}_{u,a}$ and $\mathbf{A}_{u,b}$ such that $\mathbf{A}_u\partial\mathbf{u} = \mathbf{A}_{u,a}\partial\mathbf{u}_a + \mathbf{A}_{u,b}\partial\mathbf{u}_b$. Since $(\mathbf{X})_{1,2} = (\mathbf{X})_{2,1} = 1$, we also let $\hat{\mathbf{x}}_0$ be $\hat{\mathbf{x}}$ with its second element removed, and $\mathbf{A}_{x,0}$ be $\mathbf{A}_x$ with its second column removed. It follows from (2.80)-(2.82) that

$$\mathbf{A}\partial\mathbf{a} + \mathbf{B}\partial\mathbf{b} = 0 \tag{2.83}$$

where $\mathbf{a} = \mathbf{u}_a$, $\mathbf{b} = [\hat{\mathbf{x}}_0^T, \mathbf{u}_b^T, \mathbf{z}^T]^T$, $\mathbf{A} = \mathbf{A}_{u,a}$, $\mathbf{B} = [\mathbf{A}_{x,0}, \mathbf{A}_{u,b}, \mathbf{A}_z]$. Based on (2.83), the Newton's algorithm is

$$\hat{\mathbf{x}}_0^{(i+1)} = \hat{\mathbf{x}}_0^{(i)} - \eta(\mathbf{B}^T\mathbf{B})^{-1}\mathbf{B}^T\mathbf{A}(\mathbf{u}_a - \mathbf{u}_a^{(i)}) \tag{2.84}$$

$\mathbf{u}_a^{(i)}$ is the i-th step "estimate" of the known vector $\mathbf{u}_a$ (through forward computation) based on the i-th step estimate $\hat{\mathbf{x}}_0^{(i)}$ of the unknown vector $\hat{\mathbf{x}}_0$. This algorithm requires $N_yK \geq \frac{1}{2}N(N+1) - 1$ in order for $\mathbf{B}$ to have full column rank.

For a random initialization around $\mathbf{X}$, we can let $\mathbf{X}' = (1-\beta)\mathbf{X} + \beta\mathbf{W}$ where $\mathbf{W}$ is a symmetric random matrix with $Tr(\mathbf{W}) = 1$. Furthermore, $(\mathbf{W})_{1,2} = (\mathbf{W})_{2,1}$ is such that $(\mathbf{X}')_{1,2} = (\mathbf{X}')_{2,1} = 1$. Keep in mind that at every step of iteration, we keep $(\mathbf{X}^{(i)})_{1,2} = (\mathbf{X}^{(i)})_{2,1} = 1$.

Upon convergence of $\mathbf{X}$, we can also update $\mathbf{x}$ as follows. Let the eigenvalue decomposition of $\mathbf{X}$ be $\mathbf{X} = \sum_{i=1}^{N} \lambda_i \mathbf{e}_i \mathbf{e}_i^T$ where $\lambda_1 > \lambda_2 > \cdots > \lambda_N$. Then the update of $\mathbf{x}$ is given by $\mathbf{e}_1$ if $1-\alpha > 0$ or by $\mathbf{e}_N$ if $1-\alpha < 0$. With each renewed $\mathbf{x}$, there are a renewed α and hence a renewed $\mathbf{X}$ (i.e., by setting $\mathbf{X} = \alpha\mathbf{I} + (1-\alpha)\mathbf{x}\mathbf{x}^T$ with $1-\alpha = \frac{1}{x_1 x_2}$). Using the new $\mathbf{X}$ as the initialization, we can continue the search using (2.84).

Chapter 3

Physical Layer Encryption for UAV-to-Ground Communications

3.1 Introduction

Unmanned Aerial Vehicles (UAVs) are expected to be widely deployed in near future for applications such as surveillance, transportation, mobile base stations and mobile relays [48]. UAV is often exposed in air, and in this case the information transmitted from UAV is particularly vulnerable to eavesdropping. To protect the information with information-theoretic secrecy against eavesdroppers (Eve), there are two fundamental approaches. One is network layer security where a secret key must be pre-established between two legitimate nodes (Alice and Bob) and this secret key can be then used to encrypt and decrypt a large volume of information to gain a computation-based secrecy (in addition to the information-theoretic secrecy from the secret key). The other approach is physical layer

security where either a secret key is generated from correlated observations at Alice and Bob or secret information is directly transmitted from Alice to Bob. The direct transmission by the wiretap channel model requires schemes to make the channel from Alice to Bob stronger than that from Alice to Eve, e.g., see [49] and [50] for UAV trajectory and/or resource allocation, [51] for beamforming, and [7] for using full-duplex radios. This requirement is often not possible especially for UAV-to-Ground (U2G) communications where Eves are often hidden and their capabilities are often unknown. And the range of current in-band full-duplex communication may not be sufficiently large. So, without a pre-established secret key between a legitimate pair of UAV (Alice) and GS (Bob), and without the knowledge of Eve's capability, achieving an information-theoretic secrecy for U2G communications should exploit correlated observations that are available to the pair of UAV and GS but are independent of the observations by any Eve. In this chapter, we will focus on the use of a pair of estimated reciprocal-channel vectors (ERCVs) obtained by the legitimate pair of UAV and GS respectively for secure U2G communications.

Given two ERCVs at UAV and GS respectively, a central task of the traditional physical layer security approach would be to extract a pair of digital keys at UAV and GS respectively [14, 17, 26]. But in practice, much of the statistics of ERCVs is unknown, and hence reliable key generation directly from the two ERCVs remains a challenge.

However, without an explicit key generation from ERCVs, we can still exploit the secrecy inherent in ERCVs via what is called physical layer encryption as first explored in [24] and [39]. A crucial tool for physical layer encryption is called continuous encryption function (CEF) which is discussed in detail in chapter 2. This chapter shows how to apply

a CEF developed in chapter 2 to hide transmitted symbols and/or constellations, which consequently achieves a secure U2G communication.

There have been many works on constellation detection, e.g., see [52, 53]. More recently, many machine learning based methods have been developed for constellation detection, e.g., see [54, 55]. Furthermore, various methods have also been developed to degrade the performances of constellation detection methods [56–60]. The constellation hiding method shown in this chapter exploits an information-theoretic secrecy in ERCVs, which in principle prevents *any* method from successfully detecting the hidden constellation. Hence, our work also differs from [61] where a hidden constellation can be detected by adversary, and differs from [49] where the hidden information is detectable by adversary.

Contrast between Direct Quantization (DQ) for Secret Key Generation in order to perform digital encryption and proposed Physical Layer Encryption (PLE) is illustrated in fig. 3.1. In the key generation approach, the shared Secret Vector (SV) is first quantized and converted into secret bits which then go through reconciliation and privacy amplification process. Then a secret key is obtained which is subsequently used to generate pseudo random bit stream for data encryption. In the PLE approach, the SV is first encrypted to generate stream of continuous pseudo random numbers which is then used in the proposed symbol and constellation hiding method to secure transmitted bits. Information reconciliation and privacy amplification are crucial for DQ approach as any bit mismatch in the shared keys will result in almost all the bits being different in the later stage. Unlike the DQ approach, the PLE approach does not require these steps reducing the latency. However, the generated stream of continuous pseudo random numbers from the PLE approach is not error free and

should paired with error correction coding for transmission (which is also necessary for DQ approach due to channel noise).

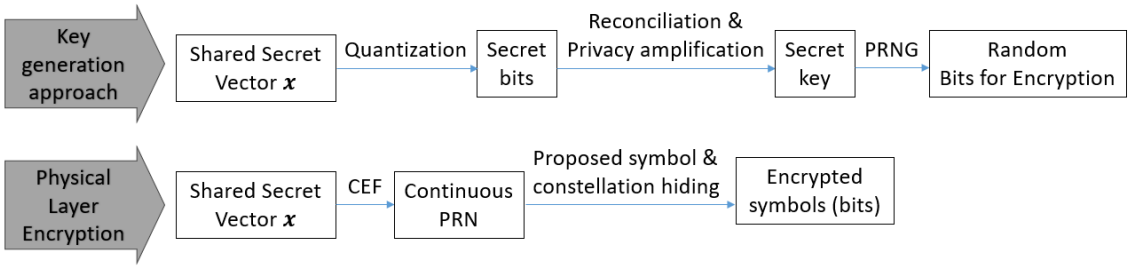


Figure 3.1: Contrast between Direct Quantization (DQ) for Secret Key Generation and use of Physical Layer Encryption (PLE)

The remainder of the chapter is organized as follows. Section 3.2 describes the wireless channel model used in this chapter. See Fig. 3.2. Section 3.3 provides a brief discussion of the SVD-CEF described in chapter 2 which is relevant to the proposed method in this chapter. Section 3.4.1 describes the proposed method for symbol and/or constellation hiding, and highlights the main issues to be discussed. In Section 3.4.2, we discuss how to generate uniformly distributed quasi-continuous pseudo-random numbers (UD-QCPRNs) from the output of a CEF, and evaluate the noise propagation in the encryption/decryption process. In section 3.5, we evaluate the impact of the encryption noise on the performance of the legitimate receiver. A quantized scheme is shown in section 3.6, which is an efficient form of the proposed method. Finally, section 3.7 concludes the chapter.

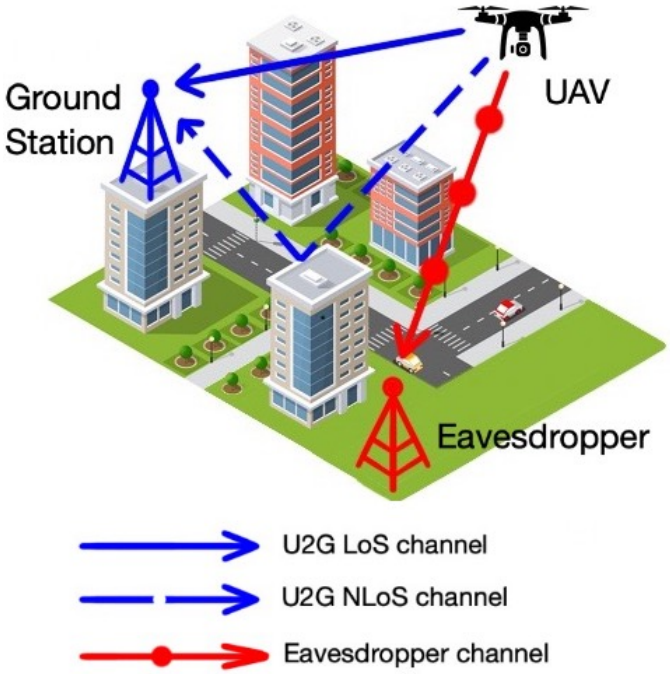


Figure 3.2: Illustration of wireless channel model for U2G communication with eavesdropper present.

3.2 Wireless Channel Model

Following [62], we model the reciprocal complex channel gain at time n (within a time window in the order of milliseconds) between UAV and GS as:

$$g_n = \sqrt{\beta_0 {d_n}^{-\alpha_n}} h_n \tag{3.1}$$

where β_0 is the large-scale average channel power gain at unit distance, d_n is the U2G distance, α_n is the path loss exponent, and h_n is the small-scale fading coefficient. We assume that Alice and Bob can each get an estimate of g_n by a standard channel estimation technique. Furthermore, we assume that d_n and α_n do not change significantly within the time window of interest and hence Alice and Bob can also each get an estimate of

h_n by scaling the estimate of g_n. For U2G communication, h_n in general consists of two components: line-of-sight (LoS) [63] and non-line-of-sight (NLoS), which is often called Rician fading. In this case, h_n can be modelled as:

$$h_n = \sqrt{\frac{K_n}{K_n+1}}e^{j\theta_n} + \sqrt{\frac{1}{K_n+1}}\xi_n \tag{3.2}$$

where θ_n for all n are i.i.d. and uniformly distributed over $[0, 2\pi]$, denoted by $\mathcal{U}(0, 2\pi)$; and ξ_n for all n are i.i.d. complex Gaussian random variables with zero mean and unit variance, denoted by $\mathcal{CN}(0, 1)$. It follows that $\mathcal{E}\{|h_n|^2\} = 1$ and $\angle h_n \sim \mathcal{U}(0, 2\pi)$ ($\mathcal{E}\{\cdot\}$ denotes the expectation operator). Here K_n is the Rician factor [64]. For all simulations in this chapter, we will treat K_n as invariant to n and set $K_n = 27$dB. Eve is assumed to be located anywhere on the ground and can have LoS with the UAV but less likely to have LoS with GS due to terrain and obstacles as illustrated in fig. 3.2. If Eve knows the exact distance between the antenna of UAV and the antenna of GS at all times, she could try to estimate θ_n for all n. But due to limited precision in estimated distance relative to wavelength, it is reasonable to assume that Eve is completely blind to θ_n. Because of multipath fading, Eve is also completely blind to ξ_n (except for its distribution). The estimates of h_n by Alice and Bob may be denoted by $\hat{h}_{n,A}$ and $\hat{h}_{n,B}$ respectively. Then the amount of secrecy available from $\hat{h}_{n,A}$ and $\hat{h}_{n,B}$ is the mutual information between them. For notational convenience and with no serious loss of generality, we will from now on let $\hat{h}_{n,A} = h_n$ and $\hat{h}_{n,B} = h'_n = h_n + w_n$ where $w_n \sim \mathcal{CN}(0, \frac{1}{\mathrm{SNR}_h})$. For a time window of $N/2$ samples, we also let $\mathbf{h} = [h_1, h_2, \cdots, h_{\frac{N}{2}}]^T$ and $\mathbf{h'} = [h'_1, h'_2, \cdots, h'_{\frac{N}{2}}]^T$. (Note that the index n in h_n can be also used to represent the index of any spatial or frequency subchannel between UAV and GS.) Furthermore, we define the ERCVs obtained by Alice and Bob as

$\mathbf{x} = [\mathcal{R}e\{\mathbf{h}\}^T, \mathcal{I}m\{\mathbf{h}^T\}]^T$ and $\mathbf{x}' = [\mathcal{R}e\{\mathbf{h}'\}^T, \mathcal{I}m\{\mathbf{h}'\}^T]^T$. It follows that $\mathbf{x}' = \mathbf{x} + \mathbf{w}_x$ where $\mathbf{w}_x \sim \mathcal{N}(\mathbf{0}, \frac{1}{\text{SNR}_x}\mathbf{I}_N)$ and $\text{SNR}_x = \text{SNR}_h$

3.3 Brief Description of Applied CEF

In this section we provide a brief description of the SVD-based continuous encryption function (SVD-CEF) used for proposed physical layer encryption in this chapter. Let $\mathbf{x}$ be an $N \times 1$ real-valued zero-mean random vector, denoted by $\mathbf{x} \in \mathcal{R}^{N\times 1}$. As discussed in chapter 2, CEF of $\mathbf{x}$ is an easy-to-compute (i.e., with a polynomial complexity in terms of N) map from $\mathbf{x}$ to a sequence of real-valued numbers $y_1, y_2, \cdots$, which is expressed as $y_k = f_k(\mathbf{x}) \in \mathcal{R}$ for all $k \geq 1$. A good CEF as defined in section 2.6 in chapter 2 is such that (1) it is hard (i.e., with an exponential complexity in terms of N) to compute $\mathbf{x}$ from y_k with all $k \geq 1$; (2) there is no such k-invariant function of $\mathbf{x}$, i.e., $g(\mathbf{x})$, that "y_k is an easy-to-compute function of $g(\mathbf{x})$, and $g(\mathbf{x})$ is also easy to compute from y_k with all $k \geq 1$"; (3) the signal-to-noise ratio (SNR) in y_k caused by a noise in $\mathbf{x}$ is not much smaller than the SNR in $\mathbf{x}$; and (4) y_k for all $k \geq 1$ can only have very weak correlations when the entries of a random $\mathbf{x}$ have zero correlations.

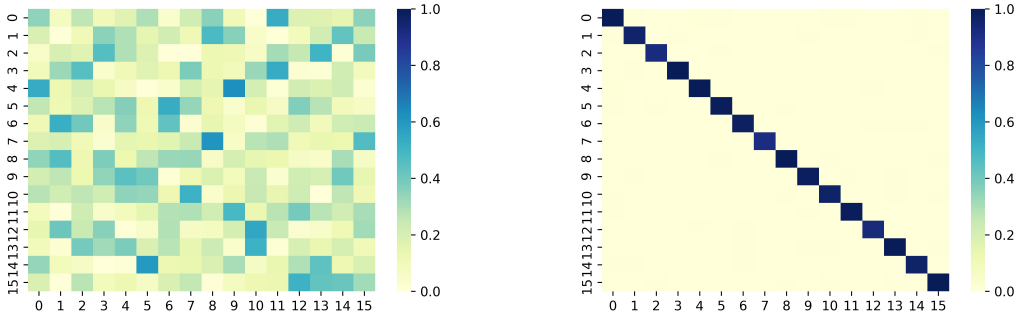
The first two properties of a good CEF have been empirically established in section 2.7 although a formal proof seems hard if not impossible. The third property of a good CEF can be measured by comparison to a unitary random projection (URP), i.e., $\mathbf{g}_k = \mathbf{R}_k\mathbf{x} \in \mathcal{R}^{N\times 1}$ where $\mathbf{R}_k$ for each index k is a pseudo random unitary matrix (governed by a seed). The noise sensitivity of URP is considered to be optimal since the norm of the perturbation vector in $\mathbf{g}_k$ for each k is always the same as the norm of the corresponding perturbation

vector in $\mathbf{x}$. But URP is not hard to invert (if the seed is known or $\mathbf{R}_k$ for any k is known). The fourth property of a good CEF can be verified via simulations. For URP, there is in general a significant correlation between $\mathbf{g}_k$ and $\mathbf{g}_l$ for $k \neq l$ (subject to fixed $\mathbf{R}_k$ and $\mathbf{R}_l$) even if the correlation matrix of $\mathbf{x}$ is the identity matrix $\mathbf{I}_N$.

The SVD-CEF proposed in section 2.6 is based on components of singular value decomposition (SVD) of a pseudo-randomly modulated matrix of $\mathbf{x}$. Specifically, let $\mathbf{Q}_{k,l} \in \mathcal{R}^{N\times N}$ for all pairs of k and l be pseudo-random unitary matrices; $\mathbf{M}_{k,\mathbf{x}} = [\mathbf{Q}_{k,1}\mathbf{x}, \cdots, \mathbf{Q}_{k,N}\mathbf{x}]$ where each column of $\mathbf{M}_{k,\mathbf{x}}$ is a pseudo-random rotation of $\mathbf{x}$; and then we let: $\mathbf{u}_{k,x} = \arg\max_{\mathbf{u},\|\mathbf{u}\|=1} \mathbf{u}^T\mathbf{M}_{k,\mathbf{x}}\mathbf{M}_{k,\mathbf{x}}^T\mathbf{u}$, which is the principal left singular vector of $\mathbf{M}_{k,\mathbf{x}}$. Finally, choose y_k to be a particular (e.g. the first) element of $\mathbf{u}_{k,x}$ for each of $k \geq 1$. We can view this SVD-CEF as a scrambler that turns a finite number of real-valued random numbers in $\mathbf{x}$ into an infinite number of QCPRNs y_k for $k \geq 1$. Unlike any of the conventional PRN generators, here y_k is a *continuous* function of $\mathbf{x}$.

To illustrate the correlations among the output values of URP-CEF and SVD-CEF, we now consider the input vector $\mathbf{x}$ described in section 3.2. The normalized correlation matrix of this $\mathbf{x}$ is $\mathbf{C}_\mathbf{x} = 2\mathcal{E}_\mathbf{x}\{\mathbf{x}\mathbf{x}^T\} = \mathbf{I}_N$. For URP-SVD, we know that $\mathbf{C}_{\mathbf{g}_k} = 2\mathcal{E}_\mathbf{x}\{\mathbf{g}_k\mathbf{g}_k^T\} = \mathbf{I}_N$ for each k. But the corresponding cross-correlation matrix $\mathbf{C}_{\mathbf{g}_k,\mathbf{g}_l} = 2\mathcal{E}_\mathbf{x}\{\mathbf{g}_k\mathbf{g}_l^T\}$ for $k \neq l$ is not small in general. The absolute values of all entries in $\mathbf{C}_{\mathbf{g}_k,\mathbf{g}_l}$ for a random realization of $\mathbf{R}_k$ and $\mathbf{R}_l$ are illustrated in the left of Fig. 3.3 where $N = 16$. For SVD-CEF, we let $\mathbf{y} = [y_1, \cdots, y_N]^T$. Since $\mathbf{u}_{k,x}$ has the norm one, the variance of y_k can be shown to be $\frac{1}{N}$. Then the normalized correlation matrix of $\mathbf{y}$ is $\mathbf{C}_\mathbf{y} = N\mathcal{E}_\mathbf{x}\{\mathbf{y}\mathbf{y}^T\}$. The absolute values of all entries of $\mathbf{C}_\mathbf{y}$ for a random realization of $\{\mathbf{Q}_{k,l}; k = 1, \cdots, N; l = 1, \cdots, N\}$ with $N = 16$

are illustrated in the right of Fig. 3.3. In both cases, the average is done over 10^5 random realizations of $\mathbf{x}$. Here we see near-zero correlations among entries of $\mathbf{y}$. Since $\mathbf{Q}_{k,l}$ are randomly chosen, the observed phenomenon of near-zero correlations holds for all y_k with $k \geq 1$.



URP-CEF SVD-CEF

Figure 3.3: Correlation "heatmaps" of the output of URP-CEF and the output of SVD-CEF. SVD-CEF offers near zero correlations among the output while URP-CEF suffers from significant correlation.

We will next apply SVD-CEF in a physical layer encryption method where the seed used to construct the pseudo-random unitary matrices, $\mathbf{Q}_{k,l}$ for all $k \geq 1$ and $1 \leq l \leq N$, is assumed to be a public information known to Alice, Bob and Eve.

3.4 A Physical Layer Encryption Method

Assume that Alice wants to transmit a sequence of K complex information symbols to Bob. For many commonly used QAM symbol constellations, the real and imaginary parts of the sequence can be each treated as a sequence of M-PAM (real-valued) symbols. Hence we can focus on how to encrypt a sequence of M-PAM symbols, denoted by s_k with

$k = 1, 2, \cdots, K$. Also assume that the constellation of s_k for each k is a discrete set of M points equally spaced within $[-1, 1]$. The spacing between two adjacent points is denoted by $\Delta = \frac{2}{M}$, i.e., $\min_{s_k^{(i)} \neq s_k^{(j)}} |(s_k^{(i)} - s_k^{(j)})_{modulo-[-1,1]}| = \Delta$ where $s_k^{(i)}$ and $s_k^{(j)}$ are two distinct realizations of s_k.

3.4.1 Basic Approach

Let z_k be a function of y_k such that z_k is uniformly distributed over $[-1, 1]$. Then an encrypted symbol for transmission from Alice is defined as $\hat{s}_k = (s_k + z_k)_{modulo-[-1,1]}$. Clearly, given any s_k, $\hat{s}_k$ has the same uniform distribution as z_k. In this case, no method is able to detect s_k based on $\hat{s}_k$ alone. This is because $I(s_k; \hat{s}_k) = h(\hat{s}_k) - h(\hat{s}_k|s_k) = h(\hat{s}_k) - h(\hat{s}_k) = 0$ where $I(\cdot;\cdot)$ denotes mutual information and $h(\cdot)$ denotes the differential entropy.

At Bob, the received symbol corresponding to $\hat{s}_k$ can be written as $\hat{s}'_k = \hat{s}_k + n_k$ where n_k is the (normalized) channel noise with its power inversely proportional to the transmission power from Alice. Since Bob has $\mathbf{x}' = \mathbf{x} + \mathbf{w}_x$, he can compute y'_k from $\mathbf{x}'$ in the same way as Alice computes y_k from $\mathbf{x}$. Furthermore, Bob can compute z'_k from y'_k in the same way as Alice computes z_k from y_k. Let $z'_k = z_k + w_{z_k}$ with w_{z_k} being the encryption noise.

To decrypt $\hat{s}'_k$ (at the physical layer), Bob computes $s'_k \doteq (\hat{s}'_k - z'_k)_{modulo-[-1,1]} = (s_k + n_k - w_{z_k})_{modulo-[-1,1]}$. As long as the channel-and-encryption combined noise $n_k - w_{z_k}$ is small compared to Δ, Bob can detect the information in s'_k with a small error rate.

In theory, z_k for each k can have a continuous uniform distribution over $[-1, 1]$. But in practice, there is a limited numerical resolution and hence z_k for each k should be

discrete over $[-1, 1]$. But the constellation size of z_k must be larger than M in order to hide the constellation size of s_k. If z_k and s_k have the same constellation size M, the symbol s_k is still protected. The power of $\hat{s}_k$ is generally larger than that of s_k. But the difference approaches zero quickly as M increases.

In the next section, we will discuss how to generate the UD-QCPRNs z_k from the output y_k of CEF and discuss the impact of the noise in $\mathbf{x}'$ on the noise in z'_k. In section 3.5, the impact of the combined noise $n_k - w_{z_k}$ on the performance at Bob is investigated via simulation.

3.4.2 Obtaining UD-QCPRNs from SVD-CEF

It is shown in section 2.8 in chapter 2 that if $\mathbf{x}$ consists of i.i.d. Gaussian random variables, then the probability density function (PDF) of each element of the output of SVD-CEF can be approximated by $f_Y(y) = C_N(1-y^2)^{\frac{N-3}{2}}$ where $-1 < y < 1$ and $C_N = \frac{\Gamma(\frac{N}{2})}{\sqrt{\pi}\Gamma(\frac{N-1}{2})}$. We have also found that if $\mathbf{x}$ is the $N \times 1$ random vector constructed as discussed in section 3.2, the output of SVD-CEF can be also approximated by the same PDF. In fact, the PDF of the output of SVD-CEF is rather robust to a range of variations in the statistics of $\mathbf{x}$. This is because of the construction of $\mathbf{M}_{k,\mathbf{x}}$ from $\mathbf{x}$ where each vector $\mathbf{Q}_{k,l}\mathbf{x}$ tends to be Gaussian distributed for a moderate to large N, which follows the well-known large-sample theory of Gaussian random variables.

To obtain z_k with the uniform distribution $\mathcal{U}(-\frac{B}{2}, \frac{B}{2})$, it can be shown that $z_k = T_{SVD}(y_k)$ with

$$T_{SVD}(y) = \int_{-1}^{y} B f_Y(u)du - \frac{B}{2} = BC_N \int_0^{\theta_y} \cos^{N-2}\theta d\theta \tag{3.3}$$

$$\int_0^{\theta_y} \cos^n \theta d\theta = \frac{\cos^{n-1}\theta_y \sin\theta_y}{n} + \frac{n-1}{n}\int_0^{\theta_y} \cos^{n-2}\theta d\theta. \tag{3.4}$$

Where $\theta_y = \sin^{-1} y$. Note that for Alice, we have a process of $\mathbf{x} \to y_k \to z_k$, and for Bob, we have a similar process of $\mathbf{x}' \to y'_k \to z'_k$.

To quantify the relationship between the noise in $\mathbf{x}'$ and the noise in z'_k, we can evaluate $\eta_{x,z} = \eta_{x,y}\eta_{y,z}$ with $\eta_{x,y} = \sqrt{\frac{\text{SNR}_x}{\text{SNR}_y}}$ and $\eta_{y,z} = \sqrt{\frac{\text{SNR}_y}{\text{SNR}_z}}$. Here, SNR_x is the signal to noise ratio in $\mathbf{x}'$, and SNR_y and SNR_z are defined similarly.

Analysis of $\eta_{x,y}$:

Assume $\mathbf{x}' = \mathbf{x} + \mathbf{w}_x$ with $\mathbf{w}_x \sim \mathcal{N}(\mathbf{0}, \sigma_w^2 \mathbf{I}_N)$. Then for a given $\mathbf{x}$ and a given set of $\mathbf{Q}_{k,l}$, we can write

$$\eta_{x,y} = \sqrt{\frac{\text{SNR}_x}{\text{SNR}_y}} = \sqrt{\frac{\left(\frac{\mathcal{E}_{\mathbf{w}_x}\{\|\mathbf{x}\|^2\}}{\mathcal{E}_{\mathbf{w}_x}\{\|\mathbf{w}_x\|^2\}}\right)}{\left(\frac{\mathcal{E}_{\mathbf{w}_x}\{\|\mathbf{y}_k\|^2\}}{\mathcal{E}_{\mathbf{w}_x}\{\|\mathbf{w}_y\|^2\}}\right)}}. \tag{3.5}$$

Since the output of SVD-CEF is invariant to the scaling of $\mathbf{x}$, we can choose $\|\mathbf{x}\|^2 = 1$, $\|\mathbf{x}'\|^2 = 1$, $\|\mathbf{y}_k\|^2 = 1$ and $\|\mathbf{y}'_k\|^2 = 1$. Hence,

$$\eta_{x,y} = \sqrt{\frac{\mathcal{E}_{\mathbf{w}_x}\{\|\mathbf{w}_y\|^2\}}{\mathcal{E}_{\mathbf{w}_x}\{\|\mathbf{w}_x\|^2\}}} \tag{3.6}$$

which is equivalent to $\eta_{k,x}$ in (2.65) in section 2.8.

A closed form of $\eta_{k,x}$ for small σ_w^2 is available in (2.65) and (2.63), which is dependent on $\mathbf{x}$ and $\mathbf{Q}_{k,l}$. For a given $\mathbf{x}$, an upper bound of $\eta_{k,x}$ can be set by pruning $\mathbf{Q}_{k,l}$. In Table 3.1, the percentages of pseudo-randomly generated $\mathbf{Q}_{k,l}$ that satisfy the condition $\eta_{x,y} < \eta_T$ for SVD-CEF are shown. To maintain a high percentage, we will choose $\eta_T = 2.5$ in the remainder of the chapter. Note that the choices of $\mathbf{Q}_{k,l}$ are publicly known and can be locally generated from a common seed.

Table 3.1: Empirically obtained % of $\mathbf{Q}_{k,l}$ that satisfies $\eta_{x,y} < \eta_T$ for different N

$\eta_T \rightarrow$	0.8	1	1.5	2	2.5
$N = 16$	4.62	21.98	63.25	80.78	88.26
$N = 32$	0.29	6.75	48.46	73.10	84.03
$N = 64$	0.007	0.84	31.61	63.32	78.35

Analysis of $\eta_{y,z}$:

Next we evaluate $\eta_{y,z}$ for small σ_w^2. We first recall (ignoring k for convenience) $z = \int_{-1}^{y} Bf_Y(u)du - \frac{B}{2}$ and $z' = \int_{-1}^{y'} Bf_Y(u)du - \frac{B}{2}$ where $y' = y + w_y$ and $z' = z + w_z$. It follows that for small σ_w^2,

$$w_z = z' - z = T_{SVD}(y') - T_{SVD}(y) \approx Bw_y f_Y(y) \tag{3.7}$$

and hence

$$\begin{aligned} \mathcal{E}_Y\{w_z^2\} &= \mathcal{E}_Y\left\{B^2 {w_y}^2 f_Y(y)^2\right\} \\ &= B^2 {w_y}^2 \int_{-1}^{1} f_Y(u)^3 du \\ &= B^2 w_y^2 D_N \end{aligned} \tag{3.8}$$

where $D_N = \frac{\Gamma(\frac{N}{2})^3 \Gamma(\frac{3N-7}{2})}{\pi \Gamma(\frac{N-1}{2})^3 \Gamma(\frac{3N-6}{2})}$. Hence, $\frac{var(w_z)}{var(w_y)} = B^2 D_N$. Furthermore, since $var(y) = \frac{1}{N}$ and $var(z) = \frac{B^2}{12}$, then $\eta_{y,z} = \sqrt{\frac{\text{SNR}_y}{\text{SNR}_z}} = \sqrt{\frac{var(y)}{var(z)} \times \frac{var(w_z)}{var(w_y)}} = \sqrt{\frac{12 D_N}{N}}$ which is invariant to B but dependent on N.

The above theoretical results of $\eta_{y,z}$ are compared with simulation/empirical results (with $B = 2$) in Fig. 3.4. In simulation, we used 10^6 realizations of $\mathbf{y}'_k = \sqrt{1-\alpha}\mathbf{y}_k +$

$\sqrt{\alpha}\mathbf{w}_k$ with random $\mathbf{y}_k$ and $\mathbf{w}_k$ satisfying $\|\mathbf{y}_k\| = 1$, $\|\mathbf{w}_k\| = 1$ and $\mathbf{w}_k^T\mathbf{y}_k = 0$. We see that the two results are reasonably close to each other. We also see that for a large N (such as $N \geq 16$), $\eta_{y,z}$ is close to one and hence $\eta_{x,z}$ is dominated by $\eta_{x,y}$. Which implies that for SVD-CEF, $\eta_{x,z} = \eta_{x,y}\eta_{y,z} \approx \eta_{x,y} \leq \eta_T$

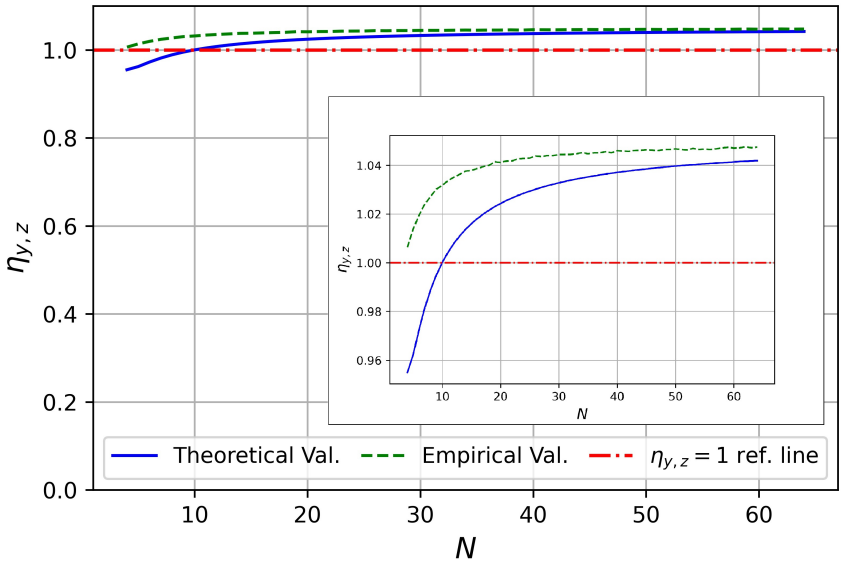
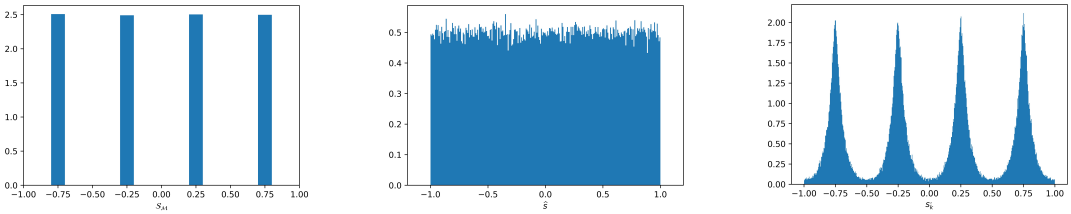


Figure 3.4: The plot of $\eta_{y,z}$ vs N (both theoretical and empirical) for $\alpha = 10^{-5}$.

3.5 Simulation

In this section, we show simulation results of the proposed method. These results illustrate the effects of the channel noise and the encryption noise on the performance at Bob. As explained before, we can focus on M-PAM only.



Dist. of s_k Dist. of $\hat{s}_k$ Dist. of s'_k

Figure 3.5: Distributions of s_k, $\hat{s}_k$ and s'_k where $\text{SNR}_x = 20\text{dB}$ and $1/\sigma_n^2 = 37\text{dB}$

We assume that the channel noise n_k is i.i.d. Gaussian $\mathcal{N}(0, \sigma_n^2)$. The variance of the encryption noise w_{z_k} can be expressed as $\frac{\eta_{x,z}^2}{3\,\text{SNR}_x}$ where $\frac{1}{3}$ is the variance of z_k. In Fig. 3.5, we illustrate the distributions of the ideal 4-PAM symbol s_k (for which the width of each vertical bar is exaggerated for visual purposes), the encrypted symbol $\hat{s}_k$, and the decrypted symbol s'_k.

It is clear that the distribution of $\hat{s}_k$ does not reveal any information about s_k and its constellation.

Once Bob has obtained enough samples of s'_k, he can use any of the existing constellation detection methods [52–55] to detect M (if unknown to Bob) and then detect the secret symbols using a minimum distance method. The simulation results of the symbol error rates (SER) at Bob under different sets of parameters are illustrated in Figs 3.6, 3.7 and 3.8.

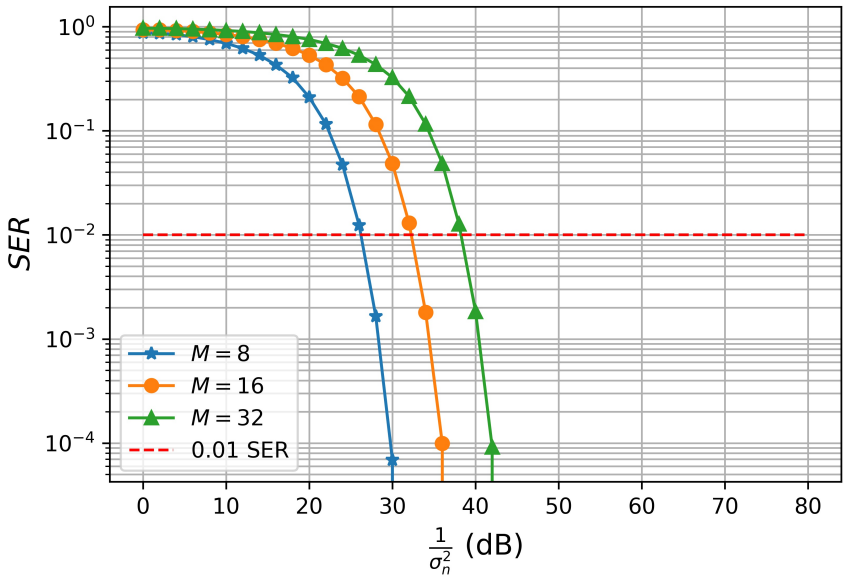


Figure 3.6: Plot of SER vs $\frac{1}{\sigma_n^2}$ with no encryption noise for different M

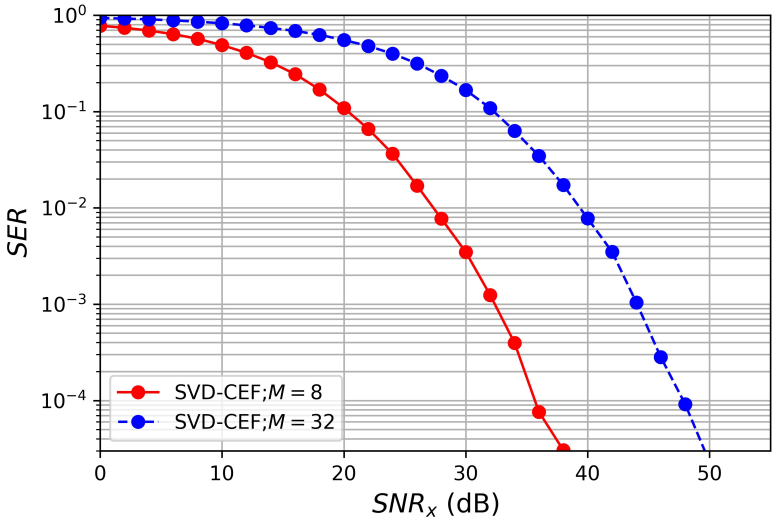


Figure 3.7: Plot of SER vs SNR_x with negligible (i.e., $\sigma_n^2 \approx 0$) channel noise for $N = 16$ and different M

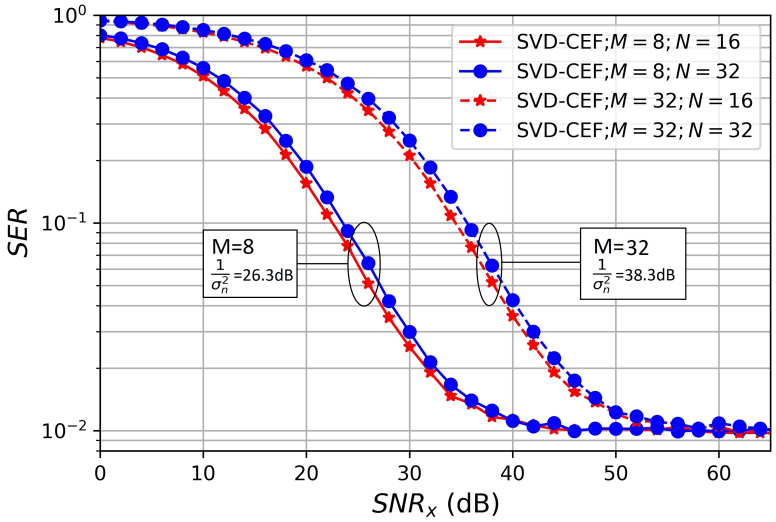


Figure 3.8: Plot of SER vs SNR_x. The value of σ_n^2 for each choice of M is such that SER is 1% in Fig. 3.6

In the simulation it is assumed that Bob has correctly detected the constellation size M from the samples of s'_k. For each chosen set of N, M, σ_n^2 and SNR_x, 10^5 random realizations of z_k, z'_k, s_k, s'_k, $\hat{s}_k$, $\hat{s}'_k$ were generated and the corresponding SER was obtained for each set of parameters. We see that the performance of SVD-CEF degrades slightly as N increases, but empirical study in chapter 2 shows that increasing N exponentially increases hardness to attack by the adversaries. This is a trade-off in choosing CEF between the performance at Bob and the hardness to attack by adversaries. It is important to stress here that even if an adversary with unlimited computing power can attack the computation-based secrecy due to SVD-CEF, there is no way for the adversary to attack the information-theoretic secrecy due to the ERCVs $\mathbf{x}$ and $\mathbf{x}'$ used in the physical layer encryption method.

3.6 Further Discussions

Computing a continuous (subject to machine precision) uniform random variable z_k from the output y_k of CEF may be costly in practice. To reduce the complexity, we can compute a discrete uniform random number $\bar{z}_k$ from y_k, which is equivalent to a uniform quantization of z_k or a non-uniform equiprobable quantization of y_k. The latter is feasible to implement. The procedures of encryption and decryption at Alice and Bob respectively are given below. The effect on Eve is discussed at the end.

Encryption at Alice

- Assume $L = 2^l$, $M' = 2^{m'}$ and $M = 2^m$ where $l > m' > m$ are integers.
- Alice constructs the true-symbol constellation $\mathcal{S}_M = \left\{\pm\frac{1}{M}(2i+1); i = 0, \cdots, \frac{M}{2} - 1\right\}$, and also in a similar way constructs the encrypted-symbol constellation $\mathcal{S}_{M'}$.
- For a known PDF $f_Y(y)$ of y_k, Alice chooses an equiprobable over-quantizer of y_k with the corresponding set of $L+1$ thresholds $\mathcal{T}_L = \{t_i; i = 0, \cdots, L\}$ where $\int_{-1}^{t_i} f_Y(y)dy = \frac{i}{L}$. Note that $\mathcal{T}_L$ also corresponds to a set $\mathcal{I}_L$ of L intervals. Each y_k is quantized into l bits by $\mathcal{T}_L$. The first m' bits of each y_k are used to determine an integer $\bar{z}_k \in \mathcal{S}_{M'}$, and the last $l - m'$ bits of each y_k are transmitted to Bob.
- For each true symbol $s_k \in \mathcal{S}_M$, Alice chooses a $\bar{z}_k \in \mathcal{S}_{M'}$ and transmits the encrypted symbol $\hat{s}_k \doteq (s_k + \bar{z}_k)_{modulo-[-1,1]}$ to Bob.

Decryption at Bob

- Assume that Bob knows l, m' and $\mathcal{T}_L$ as they are in the public domain. For each k, Bob knows $y'_k = y_k + w_{y_k}$ and also $\hat{s}'_k = \hat{s}_k + n_k = s_k + \bar{z}_k + n_k$.

- From the last $l - m'$ bits (received from Alice) of each y_k, Bob determines a corresponding set of $2^{m'}$ intervals $\mathcal{I}'_k \subset \mathcal{I}_L$. Then each y'_k is quantized into an integer $\bar{z}'_k$ of m' bits by $\mathcal{I}'_k$ according to minimum distance.

- The decrypted symbol by Bob is $s'_k \doteq (\hat{s}'_k - \bar{z}'_k)_{modulo-[-1,1]} = (s_k + n_k + \bar{z}_k - \bar{z}'_k)_{modulo-[-1,1]}$. Provided that $n_k + \bar{z}_k - \bar{z}'_k$ is small, Bob is able to detect the constellation of s_k and also the symbol s_k.

- We have observed from simulation that with $l - m' \geq 3$, the quantized scheme shown above has virtually the same performance as the continuous scheme.

Effect on Eve

All transmitted $\hat{s}_k$ from Alice are now assumed to be received by Eve without noise. It can be shown that for $\forall M' = 2iM$ where i is any positive integer, $\hat{s}_k \in \mathcal{S}_{M'}$. Without a good estimate of $\mathbf{x}$, Eve is unable to determine a good estimate of $\bar{z}_k$. In this case, Eve is unable to decrypt her received $\hat{s}_k$. Even if Eve's random guess of s_k for $k = 1, \cdots, L$ with any $L \geq 1$ is correct and hence Eve knows $\bar{z}_k$ for $k = 1, \cdots, L$, there is currently no known method with a polynomial complexity in terms of N that Eve can use to compute $\mathbf{x}$ (chapter 2) and hence Eve may still be unable to compute $\bar{z}_k$ for $k > L$ in order to decrypt $\bar{s}_k$ for $k > L$. For example, let $M' = 8M$ and $L = 8M'$. Then Fig. 3.9 and Fig. 3.10 show a consistency between the quantized scheme and the continuous scheme.

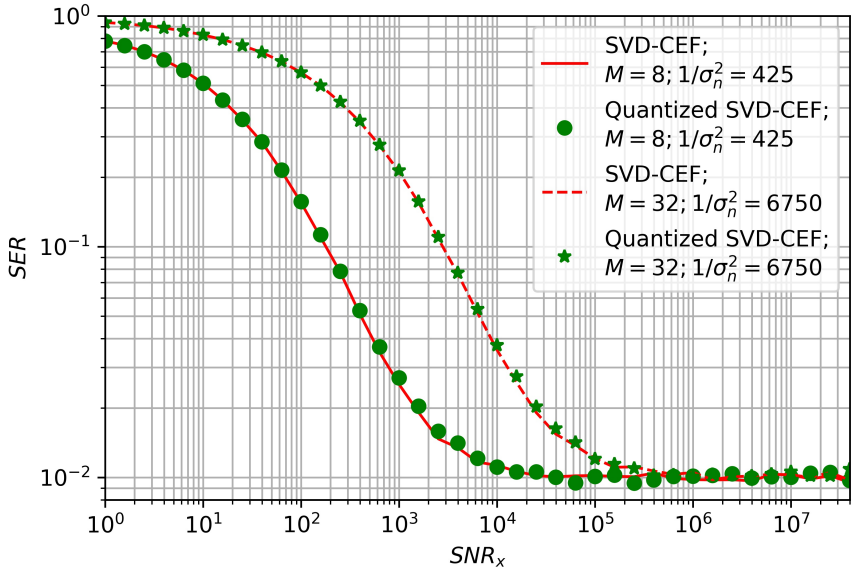


Figure 3.9: Plot of SER vs SNR_x. The value of σ_n^2 for each M was chosen such that SER in Fig. 3.6 is 1%.

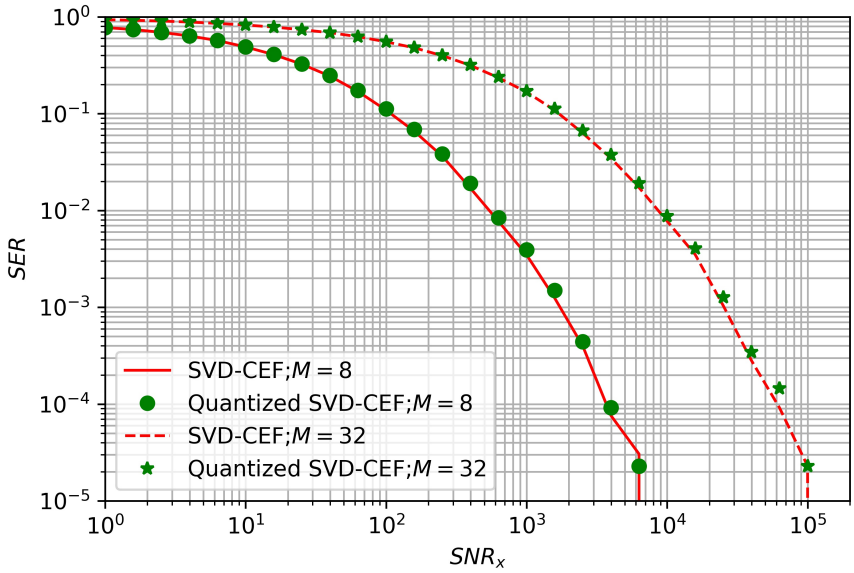


Figure 3.10: Plot of SER vs SNR_x with zero channel noise, i.e., $\sigma_n^2 = 0$

3.7 Conclusion

In this chapter, we have developed a novel physical layer encryption method for symbol and/or constellation hiding against any possible detection methods by adversaries. Our method exploits the information-theoretic secrecy in the reciprocal channel between Alice and Bob and at the same time adds a computation-based secrecy to protect any amount of information against adversaries. Our method uses a singular value decomposition based SVD-CEF that transforms a secret real-valued vector of limited dimension into an unlimited-length sequence of QCPRNs. We have found that the statistics of these QCPRNs is rather robust against a range of variations of the statistics of the ERCVs by Alice and Bob. This is an important advantage for many environments where the true statistics of ERCVs is unknown. The proposed method exploits the stable statistics of these QCPRNs to obtain uniformly distributed UD-QCPRNs, which are then superimposed onto transmitted information symbols for encryption, and/or onto received encrypted-symbols for decryption. The effect of various noises on the performance at Bob has also been investigated.

Chapter 4

Continuous Encryption Before Quantization

4.1 Introduction

Secret key generation (SKG) is a long standing problem for network security applications. For wireless security, a pair of nodes (Alice and Bob) in a wireless network can exploit their reciprocal channel state information to generate a secret key, e.g., see [2, 16, 17, 30, 31, 65–78]. Such a key shared by Alice and Bob can be then used as a symmetric key for information encryption between them over any networks [1, 32]. For biometric security, a biometric feature of a person can be collected to generate a secret key for future authentication of this person over any networks, e.g., see [25–27, 29, 42, 79].

We can view the biometric feature vector collected from a person at one time as the secret vector of "Alice" and a corresponding biometric feature vector collected from the same

person at a future time as the secret vector of "Bob". This connection allows the problem of SKG to transcend both fields of wireless security and biometric security. However, unlike many prior works on information-theoretic capacities of SKG, e.g., see [28, 80, 81], this chapter focuses on practical algorithms for SKG with useful tradeoffs.

A central issue of SKG is how to best transform a pair of highly correlated secret vectors (SVs) at Alice and Bob respectively into a pair of nearly identical sequences of binary bits (i.e., keys). The SVs are in practice quasi-continuous (due to finite precision of real number representation). Since the two SVs collected at Alice and Bob are generally not equal due to noise, the probability of the generated keys being unequal, i.e., key error rate (KER), is generally nonzero. So a central objective of SKG is to minimize KER.

The major steps of SKG for both wireless security and biometric security are: extraction of SVs which should be maximally correlated with each other and contain the minimal amount of non-secret, quantization of SVs with KER as small as possible, and reconciliation and privacy amplification for improved key, e.g., see [66, 67]. In this chapter, we focus on the problem of quantization to turn a pair of SVs into a pair of keys with any length, small KER and sufficient randomness.

To reduce KER caused by quantization of SVs, there are two approaches: guardband quantization (GQ) and adaptive quantization (AQ). The GQ approach was proposed in [30] and further studied in [2, 17, 65, 68–71, 73], where the range of each parameter in a feature vector is partitioned into a number of quantization regions separated by guardbands. When the realization of a parameter falls onto a guardband, that realization is discarded. Using guardbands helps to reduce KER but at a cost of key size. The AQ approach was

proposed in [31] and further studied in [16, 17, 73–77] under such names as coset source coding and over-quantization, where each parameter of a feature vector is assigned with multiple interleaved sub-quantizers. For each realization of a parameter, Alice determines the best sub-quantizer based on her measurement, and Bob is informed of this via public channel. Both Alice and Bob effectively apply the same sub-quantizer to quantize their measurements of the corresponding parameter respectively. Unlike the GQ approach, the AQ approach allows more cooperation between Alice and Bob and is more adaptive to each realization of a random parameter. Prior studies such as [17] suggest that the AQ approach in general outperforms the GQ approach in terms of robustness against the noises in SVs.

The above works on quantization all apply a direct quantization (DQ) on SVs. But many prior works on biometric template security, such as random projection (RP) [25, 29], dynamic random projection (DRP) [26,79] and others [27,42], advocate indirect quantization on SVs, e.g., quantization on the output of a continuous one-way transformation of a secret biometric feature vector to produce cancellable passwords. However, the keys from RP and DRP fail to pass randomness tests.

In this chapter, we present a generalized approach for SKG as shown in Fig. 4.1, also referred to as continuous encryption before quantization (CEbQ). By CEbQ, we first use a continuous encryption function (CEF) to encrypt a pair of highly correlated SVs of limited dimension into a pair of sequences of quasi-continuous pseudorandom numbers (QCPRNs) of any desired length. Unlike conventional PRN-generators, the desired CEF must be continuous function of SV. These QCPRNs are then quantized into keys. Quality of keys will be measured by not only KER but also correlation tests and randomness tests.

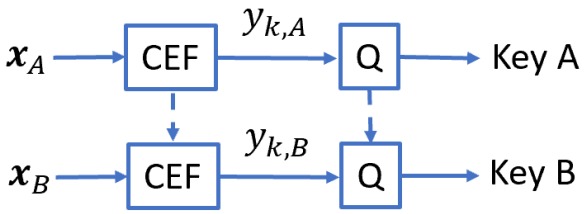


Figure 4.1: Illustration of CEbQ for SKG

Contrast between Direct Quantization (DQ) for Secret Key Generation and Continuous Encryption Before Quantization (CEBQ) is illustrated in fig. 4.2. In DQ approach, the shared Secret Vector (SV) is first quantized and converted into secret bits which then go through reconciliation and privacy amplification process. Then a secret key is obtained which is subsequently used to generate pseudo random bit stream for data encryption. In the CEBQ approach, the SV is first encrypted to generate stream of continuous pseudo random numbers which then passes through a quantizer to generate pseudo random bit stream. Information reconciliation and privacy amplification are crucial for DQ approach as any bit mismatch in the shared keys will result in almost all the bits being different in the later stage. Unlike the DQ approach, the CEBQ approach does not require these steps hence, reducing the latency. Moreover, by expanding the SV in the continuous domain, we can improve the quantization process by extracting fewer bits (as low as 1 or even fractional bits as discussed later in this chapter in section 4.5) per sample. This significantly reduces Bit Error Rate (or Key Error Rate if pseudo key is generated). However, the generated random bits from the CEBQ approach are not error free and should paired with error correction coding for transmission (which is also necessary for DQ approach due to channel noise).

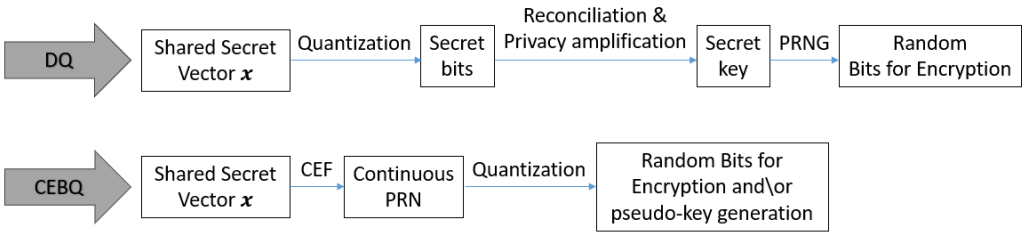


Figure 4.2: Contrast between Direct Quantization (DQ) for Secret Key Generation and use of Continuous Encryption Before Quantization (CEbQ)

In section 4.2, we further discuss the properties of a desired CEF needed in Fig. 4.1 and 4.2, revisit singular-value-decomposition (SVD) based CEF discussed in chapter 2, and explain why the SVD-CEF yields the desired QCPRNs. In section 4.4, we provide simulation results to demonstrate the advantage of CEbQ using SVD-CEF over DQ and two other indirect quantization methods. A fractional quantization method to extract 1 bit from multiple samples is also proposed in section 4.5. We will also highlight the impact of leakage of SVs (due to over-quantization) on KER for several methods.

4.2 Desired Properties of CEF

4.2.1 CEF and QCPRNs

As illustrated in Fig. 4.1, the proposed SKG method requires a CEF to produce QCPRNs from SVs. We will use the expression $y_k = f_k(\mathbf{x})$ to denote a CEF with $\mathbf{x} \in \mathcal{R}^{N\times 1}$ as its $N \times 1$ real-valued input vector and y_k as its kth real-valued output sample with $k \geq 1$. We say that a CEF can produce a sequence of QCPRNs if the following conditions are met:

1) The output of the CEF has a practically indefinite length;

2) The CEF is continuous, i.e., the output of the CEF has a finite sensitivity to a small perturbation on the input of the CEF;

3) The CEF is hard to invert, i.e., given any parts of y_k with $k \geq 1$, there is no known method with a polynomial complexity in terms of N (i.e., N^p for $p < \infty$) to determine an estimate $\hat{\mathbf{x}}$ of the input (or an estimate $\hat{s}(\mathbf{x})$ of a substitute input) such that $y_k = f_k(\mathbf{x}) \approx f_k(\hat{\mathbf{x}})$ (or $y_k = g_k(s(\mathbf{x})) \approx g_k(\hat{s}(\mathbf{x}))$) for all $k \geq 1$;

4) It can be verified empirically that the distribution of y_k is invariant to k when $\mathbf{x}$ consists of N independent and identically distributed (i.i.d.) entries; and

5) It can be verified empirically that y_k with $1 \leq k \leq K$ have near-zero correlations with $K \gg N$ and $\mathbf{x}$ consisting of i.i.d. entries.

The above notions of QCPRNs as the output of a desired CEF are similar to those defined for a good CEF in section 2.5. Such a good CEF is from a family of SVD-CEFs discussed in detail in section 2.6, which is briefly described as follows.

Let $\mathbf{Q}_{k,l} \in \mathcal{R}^{N \times N}$ be a pseudorandom unitary matrix for each pair of k and l where $l = 1, \cdots, N$ and $k \geq 1$. Both the seed and algorithm for generating these pseudorandom matrices are assumed to be in the public domain. For each of $k \geq 1$, define a modulated matrix of $\mathbf{x}$ as

$$\mathbf{M}_{k,x} = [\mathbf{Q}_{k,1}\mathbf{x}, \cdots, \mathbf{Q}_{k,N}\mathbf{x}]. \tag{4.1}$$

An SVD-CEF could define its kth output sample as any component of the SVD of $\mathbf{M}_{k,x}$, which would make $\mathbf{x}$ generally hard to compute from the output samples. But for desired statistical properties, we choose the kth output sample y_k of the SVD-CEF to be an entry (such as the 1st entry) of the left principal singular vector $\mathbf{u}_{k,\mathbf{x},1}$ of $\mathbf{M}_{k,x}$.

Note that $\mathbf{M}_{k,x}\mathbf{M}_{k,x}^T = \sum_{l=1}^{N} \mathbf{Q}_{k,l}\mathbf{x}\mathbf{x}^T\mathbf{Q}_{k,l}^T$. So, $\mathbf{X} \doteq \mathbf{x}\mathbf{x}^T$ is a valid substitute input of the SVD-CEF.

It is obvious from the above description of the SVD-CEF that y_k is a nonlinear function of $\mathbf{x}$, y_k is invariant to the norm $\|\mathbf{x}\|$, y_k is a continuous function of $\mathbf{x}$ for almost all $\mathbf{x}$ subject to a typical (randomly chosen) set $\mathbf{Q}_{k,1:N} \doteq \{\mathbf{Q}_{k,l}; 1 \leq l \leq N\}$, and the sensitivity of y_k to a perturbation on $\mathbf{x}$ depends on the corresponding $\mathbf{Q}_{k,1:N}$. Other properties of the SVD-CEF are discussed below.

4.2.2 Hardness to Invert SVD-CEF

It is shown empirically in section 2.7 in chapter 2 that the SVD-CEF is hard to invert due to the fact that finding the solution of the input $\mathbf{x}$ (up to a scalar and sign) or the substitute input $\mathbf{X}$ from any subset of y_k for $k \geq 1$ amounts to solving a set of 2nd-order multivariate polynomial equations in more than N unknowns.

4.2.3 Noise Sensitivity of SVD-CEF

Without loss of generality, we can write $\mathbf{x}_B = \mathbf{x}_A + \partial\mathbf{x}$ where $\partial\mathbf{x}$ is the difference between the two secret vectors at Alice and Bob. Using the same SVD-CEF, if $y_{k,A}$ is the output at Alice from $\mathbf{x}_A$, then the output at Bob from $\mathbf{x}_B$ can be written as $y_{k,B} = y_{k,A} + \partial y_k$. It is clear that we do not want ∂y_k to be too sensitive to $\partial\mathbf{x}$. Let $\mathtt{SNR}_{\mathtt{in}} = \frac{\|\mathbf{x}_A\|^2}{\mathcal{E}_{\partial\mathbf{x}}\{\|\partial\mathbf{x}\|^2\}}$ be the input signal-to-noise ratio (SNR) at Bob, and $\mathtt{SNR}_{\mathtt{out},k} = \frac{\|y_{k,A}\|^2}{\mathcal{E}_{\partial\mathbf{x}}\{\|\partial y_k\|^2\}}$ be the output SNR at Bob for each k. Here $\mathcal{E}_{\partial\mathbf{x}}\{\cdot\}$ is the expectation taken over $\partial\mathbf{x}$. A figure-of-merit (FoM) of the SVD-CEF for each k can be defined as $\eta_{k,\mathbf{x}_A} \doteq \sqrt{\frac{\mathtt{SNR}_{\mathtt{in}}}{\mathtt{SNR}_{\mathtt{out},k}}}$, which measures how much the input noise for Bob (relative to Alice's input) is amplified at the output for Bob.

Theorem 1 *Assume that $\partial\mathbf{x}$ consists of i.i.d. entries with zero mean and an arbitrarily small variance. Then, $\eta_{k,\mathbf{x}_A} = \sqrt{\frac{1}{N}\sum_{j=1}^{N-1}\sigma_j^2}$ where $\sigma_1 > \cdots > \sigma_N = 0$ are the singular values of:*

$$\mathbf{T} = \left(\sum_{j=2}^{N} \frac{1}{\lambda_1 - \lambda_j}\mathbf{u}_{k,\mathbf{x}_A,j}\mathbf{u}_{k,\mathbf{x}_A,j}^T\right) \cdot \left(\sum_{l=1}^{N} \mathbf{Q}_{k,l}\left[(\mathbf{x}_A^T\mathbf{Q}_{k,l}^T\mathbf{u}_{k,\mathbf{x}_A,1})\mathbf{I}_N + \mathbf{x}_A\mathbf{u}_{k,\mathbf{x}_A,1}^T\mathbf{Q}_{k,l}\right]\right) \tag{4.2}$$

which has the rank $N-1$, and $\mathbf{u}_{k,\mathbf{x}_A,j}$ and λ_j are the jth pair of eigenvector and eigenvalue of $\mathbf{M}_{k,\mathbf{x}_A}\mathbf{M}_{k,\mathbf{x}_A}^T$. Here, λ_j is in the descending order.

For proof, see section 2.8 in chapter 2. It is important to note that for given $\mathbf{x}_A$ and $\mathbf{Q}_{k,1:N}$, $\eta_{k,\mathbf{x}_A}$ can be computed by Alice. For example, if $\eta_{k_0,\mathbf{x}_A}$ is larger than a threshold, Alice can inform Bob (i.e., the left dash arrow line in Fig. 4.1) so that they can both avoid the use of the corresponding $\mathbf{Q}_{k_0,1:N}$. In this way, the noise amplification by the SVD-CEF is under control. In theory, an attacker may gain some information about $\mathbf{x}_A$ from knowing $\eta_{k_0,\mathbf{x}_A}$ exceeding a threshold. But computing $\mathbf{x}_A$ from this knowledge does not seem trivial.

The above theorem also explains why an entry of the principal eigenvector (instead of other eigenvectors) of $\mathbf{M}_{k,\mathbf{x}}\mathbf{M}_{k,\mathbf{x}}^T$ is chosen as the output of the SVD-CEF. See $\frac{1}{\lambda_1-\lambda_j}$ in (4.2).

4.2.4 Statistics of QCPRNs from SVD-CEF

It is shown empirically via simulation in section 2.8 in chapter 2 that if N is moderate or large (such as $N \geq 15$), $\mathbf{Q}_{k,1:N}$ is typical and $\mathbf{x}$ has the Gaussian distribution $\mathcal{N}(0, \sigma_x^2\mathbf{I}_N)$, then the probability density function (PDF) of y_k is approximately given by

$$f_{y_k}(y) = C_N(1-y^2)^{\frac{N-3}{2}} \tag{4.3}$$

with $C_N = \frac{\Gamma(N/2)}{\sqrt{\pi}\Gamma((N-1)/2)}$ and $-1 < y < 1$. This known PDF of y_k is important for optimal quantization on y_k. (In fact, the PDF of y_k is relatively invariant to the PDF of the i.i.d. entries of $\mathbf{x}$ because each entry of $\mathbf{M}_{k,\mathbf{x}}$ is a weighted sum of the entries in $\mathbf{x}$ and hence tends to be Gaussian in general. This gives an additional advantage to CEbQ over direct quantization (DQ) on $\mathbf{x}$. Without a good knowledge of the PDF of $\mathbf{x}$, the performance of a DQ on $\mathbf{x}$ generally suffers.) Furthermore, we have observed via simulation that for a typical set $\mathbf{Q}_{1:K,1:N} \doteq \{\mathbf{Q}_{1,1:N}, \cdots, \mathbf{Q}_{K,1:N}\}$, the output samples of the SVD-CEF, i.e., $y_1, y_2, \cdots, y_K$, have near-zero (normalized) correlations.

The above can be explained by the following analysis. Assume $\mathbf{x} \sim \mathcal{N}(0, \sigma_x^2 \mathbf{I}_N)$. Then $\mathcal{E}\{\mathbf{X}\} = \mathcal{E}\{\mathbf{x}\mathbf{x}^T\} = \sigma_x^2 \mathbf{I}_N$. Let $\mathbf{R}_{k,\mathbf{x}} = \mathbf{M}_{k,\mathbf{x}} \mathbf{M}_{k,\mathbf{x}}^T$ and $\mathbf{W} = \mathbf{X} - \sigma_x^2 \mathbf{I}_N$. It follows that $\mathbf{R}_{k,\mathbf{x}} = \mathbf{R}'_{k,\mathbf{x}} + N\sigma_x^2 \mathbf{I}_N$ with $\mathbf{R}'_{k,\mathbf{x}} = \sum_{l=1}^N \mathbf{Q}_{k,l} \mathbf{W} \mathbf{Q}_{k,l}^T$. Clearly, $\mathbf{R}_{k,\mathbf{x}}$ and $\mathbf{R}'_{k,\mathbf{x}}$ have the same eigenvectors. It also follows that $\mathcal{E}\{\mathbf{W}\} = 0$, $\mathcal{E}\{w_{i,i}^2\} = 2\sigma_x^4$, $\mathcal{E}\{w_{i,j}^2\} = \sigma_x^4$ for $i \neq j$, and $\mathcal{E}\{w_{i,j} w_{l,m}\} = 0$ for $(i,j) \neq (l,m)$. Here $\mathcal{E}\{\cdot\}$ is the expectation operator. Let $\mathbf{q}_{k,l,i}$ be the ith column of $\mathbf{Q}_{k,l}$. Then:

$$\mathbf{R}'_{k,\mathbf{x}} = \sum_{s=1}^N \sum_{v=1}^N \left(\sum_{l=1}^N \mathbf{q}_{k,l,s} \mathbf{q}_{k,l,v}^T \right) w_{s,v} \tag{4.4}$$

Let $\mathbf{G}_{k,s,v} = \sum_{l=1}^N \mathbf{q}_{k,l,s} \mathbf{q}_{k,l,v}^T$. Then

$$\mathbf{R}'_{k,\mathbf{x}} = \sum_{s=1}^N \mathbf{G}_{k,s,s} w_{s,s} + \sum_{N \geq s > v \geq 1} (\mathbf{G}_{k,s,v} + \mathbf{G}_{k,s,v}^T) w_{s,v} \tag{4.5}$$

where we have applied $w_{s,v} = w_{v,s}$. We see that $\mathbf{R}'_{k,\mathbf{x}}$ consists of $N(N+1)/2$ uncorrelated terms corresponding to $w_{s,v}$ for $s \geq v$. Each term typically has the full rank N.

The principal eigenvector of $\mathbf{R}'_{k,\mathbf{x}}$ is therefore highly dependent on $\mathbf{W}$. Based on the variances of $w_{s,v}$, we see that the $N(N+1)/2$ uncorrelated terms in (4.5) have about the same weight on $\mathbf{R}'_{k,\mathbf{x}}$. For this reason, we can conjecture that the principal eigenvector $\mathbf{u}_{k,\mathbf{x}}$

of $\mathbf{R}'_{k,\mathbf{x}}$ tends to appear uniformly on the $N-1$ dimensional sphere of unit radius $S^{N-1}(1)$. Assuming that $\mathbf{u}_{k,\mathbf{x}}$ is uniform on $S^{N-1}(1)$, the PDF of y_k as shown in (4.3) can be proven (see proof of (82) in [46]).

Also, since $\mathbf{R}'_{m,\mathbf{x}}$ depends on a set of basis matrices totally different from those of $\mathbf{R}'_{k,\mathbf{x}}$ for $k \neq m$, the trajectory of $\mathbf{u}_{m,\mathbf{x}}$ is hence uncorrelated with that of $\mathbf{u}_{k,\mathbf{x}}$ as $\mathbf{W}$ or equivalently $\mathbf{x}$ changes.

This explains why the output values of the SVD-CEF, i.e., y_k for $1 \leq k \leq K$ (even with a large $K \gg N$), have near-zero correlations.

4.3 Proposed Adaptive Quantization

We now discuss an adaptive quantizer (AQ) or over-quantization algorithm shown in [17]. This algorithm is summarized below:

For each pair of $y_{k,A}$ and $y_{k,B}$ with $k = 1, 2, \cdots, K$, the number of desired bits per QCPRN sample is set to be m. Alice and Bob share the same L-level equiprobable quantizer $\mathcal{Q}_L$ where $L = 2^{m+l}$ and the boundary values, t_i for $i = 0, \cdots, L$, satisfy $\int_{-1}^{t_i} f_{y_k}(y)dy = \frac{i}{L}$. A sample that falls into $[t_i, t_{i+1})$ will be quantized to the integer i represented by the standard $m + l$ bits.

For each k, Alice uses $\mathcal{Q}_L$ to quantize $y_{k,A}$ into $m + l$ bits. She keeps the m most significant bits (MSBs) as the kth part of her key $\mathcal{K}_A$ of total key length $L_{key} = mK$ and transmits to Bob publicly the l least significant bits (LSBs). The l LSBs do not reveal any information about the m MSBs if $y_{k,A}$ for all k are independent. In simulation, we will choose $l = 0, 1, 3$.

For each k, Bob obtains $\mathcal{C}_{2^m,k}$ consisting of the center points of a subset of 2^m intervals from $\mathcal{Q}_L$, corresponding to the l LSBs received from Alice. Bob then determines $j_k = \arg\min_{c_j \in \mathcal{C}_{2^m,k}} |y_{k,B} - c_j|$. The m-bit representation of j_k are the kth part of his key $\mathcal{K}_B$ of total key length $L_{key} = mK$.

If $\mathcal{K}_A = \mathcal{K}_B$, there is no key error. Otherwise, a key error occurs. The key error rate (KER) will be measured by the percentage of such errors over $R = 10^4$ realizations of $y_{k,A}$ and $y_{k,B}$ with $k = 1, 2, \cdots, K$, which correspond to $R = 10^4$ random realizations of each of $\mathbf{x}_A$ and $\mathbf{w}$ subject to $\mathbf{x}_B = \mathbf{x}_A + \mathbf{w}$ and a fixed $\mathbf{Q}_{1:K,1:N}$. We will choose $\mathbf{x}_A \sim \mathcal{N}(0, \mathbf{I}_N)$ and $\mathbf{w} \sim \mathcal{N}(0, \frac{1}{\mathtt{SNR}_x}\mathbf{I})$ where $\mathtt{SNR}_x$ denotes the SNR in $\mathbf{x}_B$ relative to $\mathbf{x}_A$.

Note that for DQ on $\mathbf{x}_A$ and $\mathbf{x}_B$, $y_{k,A}$ and $y_{k,B}$ for $k = 1, \cdots, K$ in the algorithm will be replaced by the entries of $\mathbf{x}_A$ and $\mathbf{x}_B$ respectively, and hence $K = N$ and $L_{key} = mN$. Also, the PDF $f_{y_k}(y)$ in the algorithm needs to be replaced by the PDF of the entry of $\mathbf{x}$. For most applications, $\mathbf{x}_A$ and $\mathbf{x}_B$ have the same PDF.

4.4 Simulation Results and Comparisons

In this section, we will refer to CEbQ using SVD-CEF simply as SVD-CEF.

4.4.1 Prior Methods for SKG Using Indirect Quantization

In the field of biometric template security, there have been many efforts on using continuous one-way functions to transform a secret vector before quantization to obtain so called cancellable passwords, e.g., see [29] and [79]. Two notable such transformations are random projection (RP) [25] and dynamic random projection (DRP) [26]. But these two

transformations can be both inverted with a polynomial complexity. Moreover, the output samples of RP are highly correlated with each other, which after quantization results in a key with highly correlated bits. So, we will not further consider RP. For DRP, we will consider the "Function II" version in [26], which can be described as follows. An $N \times 1$ secret vector $\mathbf{x}$ is first transformed into a $K \times 1$ vector $\mathbf{v} = \mathbf{R}\mathbf{x}$ with $\mathbf{R}$ being an orthonormal pseudorandom matrix, then the kth entry v_k of $\mathbf{v}$ is quantized (or "indexed") into an integer l_k subject to $1 \leq l_k \leq L$ which is then used to determine one of L pseudorandom Gaussian vectors $\mathbf{a}_{1,k}, \cdots, \mathbf{a}_{L,k}$. The kth output of DRP is $y_k = \mathbf{a}_{l_k,k}^T \mathbf{x}$, which can be then quantized into a key. In our simulation, we will use the AQ for quantizing v_k with $L = N/2$, and also for quantizing y_k for each k into 1 bit, which is an improved version from [26]. The resulting key for each realization of $\mathbf{x}$ has the size $L_{key} = K$.

Another method to turn $\mathbf{x}$ into a key is called index-of-max hashing (IoM-2) [27]. For each of $1 \leq k \leq K$, IoM-2 first generates V pseudorandom permutations of $\mathbf{x}$, then produces a vector $\mathbf{v}_k$ by computing the element-wise products of the V vectors, and finally determines the index of the largest entry in $\mathbf{v}_k$. The resulting key has the size $L_{key} = K \log_2 N$. As shown in chapter 2, IoM-2 can be inverted with a complexity no more than $\mathcal{O}(2^N)$, and its performance in terms of KER is not as good as the SVD-CEF. In this section, we will provide further results on IoM-2 assuming $V = 3$.

4.4.2 Correlation Tests

A basic requirement on a generated key is that the bits in the key should be practically uncorrelated with each other subject to $\mathbf{x}$ consisting of independent entries and all used pseudorandom transformations being fixed. To test the correlation, we map each

Table 4.1: Peak Correlation Values of Bits in Keys

$L_{key} \rightarrow$	32	64	128	256	512
DQ	0.0224	0.0263	0.0246	0.0361	NA
SVD-CEF	0.0231	0.0277	0.0279	0.0306	0.0306
DRP	0.1057	0.1058	0.1201	0.1149	0.1308
IoM	0.1593	0.1700	0.2127	0.2178	0.2543

key of L_{key} bits, generated from $\mathbf{x}$, onto an $L_{key} \times 1$ vector $\mathbf{b}$ consisting of 1's and -1's (corresponding to 1's and 0's). We are interested in the largest off-diagonal element in $\mathbf{C_b} = \mathcal{E}_{\mathbf{x}}\{\mathbf{b}\mathbf{b}^T\}$, i.e., $c_{\texttt{max}} = \max_{i \neq j} |(\mathbf{C_b})_{i,j}|$. Table 4.1 compares the values of $c_{\texttt{max}}$. For $L_{key} = 512$ (and $N = 16$), DQ would need to extract out 32 bits per entry of $\mathbf{x}$ and was not feasible on our computer. For other choices of L_{key}, we see that DQ and SVD-CEF have comparable values of $c_{\texttt{max}}$, which are substantially smaller than those of DRP and IoM. This result is based on $R = 2 \times 10^4$ realizations of $\mathbf{x} \sim \mathcal{N}(0, \mathbf{I}_N)$ with $N = 16$.

4.4.3 Key Error Rate

To compare the KERs, we set $L_{key} = \frac{N}{2} \log_2(1 + \texttt{SNR}_x)$ which is the theoretical limit, i.e., mutual information between $\mathbf{x}_A$ and $\mathbf{x}_B = \mathbf{x}_A + \mathbf{w}$ where $\mathbf{x}_A \sim \mathcal{N}(0, \mathbf{I}_N)$ and $\mathbf{w} \sim \mathcal{N}(0, \frac{1}{\texttt{SNR}_x}\mathbf{I}_N)$. Fig. 4.3 is based on $R = 10^4$ realizations of $\mathbf{x}_A$ and $\mathbf{w}$ with $N = 16$. In Fig. 4.3, m is the number of secret bits per y_k, and l is the number of over-quantized bits. The latter also corresponds to a leakage of $\mathbf{x}_A$ for DQ, a leakage of v_k and y_k for DRP, and a leakage of y_k for SVD-CEF. We see that the DQ fails badly in terms of KER for all $\texttt{SNR}_x$ with or without leakage, and so does DRP without leakage. With some leakage, both DRP

and SVD-CEF can have rather small KERs at a high $\mathtt{SNR}_x$. In principle, the leakage for DQ does not reduce the secrecy of the key assuming statistical independence of the entries in $\mathbf{x}_A$. But the leakage for DRP and SVD-CEF potentially does due to the use of CEF. But unlike DRP, SVD-CEF is hard to invert from y_k, and hence the leakage for SVD-CEF is hard to be exploited by attacker. For pruned SVD-CEF, the realizations of $\mathbf{Q}_{1:K,1:N}$ with $\eta_{k,\mathbf{x}_A} > 2.5$ were dropped. Note that the quality of the keys from DRP in terms of the peak correlation was shown to be bad. It is shown next that DRP also fails on standardized randomness tests.

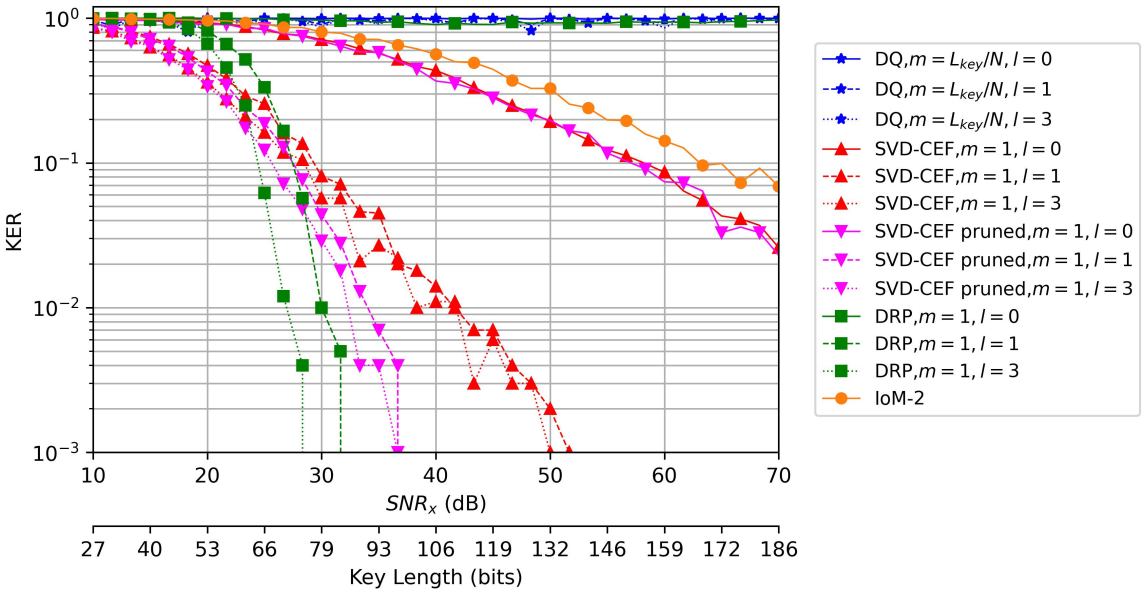


Figure 4.3: KER versus $L_{key} = \frac{N}{2}\log_2(1 + \mathtt{SNR}_x)$.

4.4.4 Randomness Tests

Finally, we consider 15 tests of randomness [82]: T1-Frequency test (monobit); T2-Frequency test within a block; T3-Run Test; T4-Longest Run of ones in a block; T5-Binary matrix rank test; T6-Discrete Fourier transform (spectral) test; T7-Non-overlapping tem-

plate matching test; T8-Overlapping template matching test; T9-Maurer's universal statistical test; T10-Linear complexity test; T11-Serial test A; T12-Serial test B; T13-Approximate entropy test; T14-Cummulative sums (forward) test; T15-Cummulative sums (reverse) test. Each test was done on a binary sequence of RL_{key} bits, consisting of concatenated R keys from R realizations of $\mathbf{x} \sim \mathcal{N}(0, \mathbf{I}_N)$ with $R = 4 \times 10^4$ and $N = 16$ (and all other parameters are fixed). The p-values of these tests are shown in Fig. 4.4. We see that DRP and IoM failed on a number of tests while DQ and SVD-CEF passed all tests with their p-values larger than 0.01. More interestingly, for $L_{key} = 512$, while DQ could not deliver any key, the key from SVD-CEF still passed all randomness tests (including the random excursions test [82] not shown here).

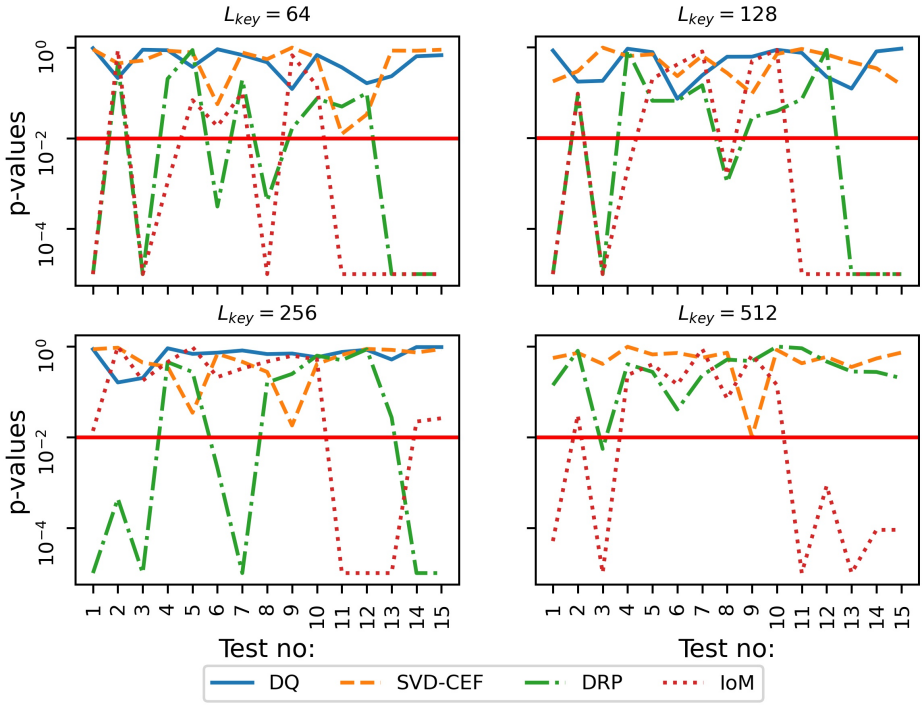


Figure 4.4: The p-values of 15 randomness tests.

4.5 Proposed Fractional Quantization

4.5.1 Methodology

We define a positive integer Q to be the dimension of fractional quantization, which indicates the size of vector $\mathbf{y}_p$ to be used to extract one bit of the secret key, i.e., Q samples of y_k will be used to construct $\mathbf{y}_p \in \mathcal{R}^{Q\times 1}$ to extract one bit. Which implies $\mu = \frac{1}{Q}$. To extract secret key of length $L_{\mathcal{K}}$, we will need $L_{\mathcal{K}}$ such vectors $\mathbf{y}_1, \mathbf{y}_2, \cdots, \mathbf{y}_{L_{\mathcal{K}}}$, thus total $QL_{\mathcal{K}}$ output samples y_k of CEF are needed. We also define another positive integer r to be the quantization bits per dimension which implies that a 2^r level equiprobable quantizer $\mathcal{Q}$ will be used to extract r bits from each sample $y_{q,p}$ of $\mathbf{y}_p$. So Q samples in $\mathbf{y}_p$ are equiprobably quantized to generate total rQ bits. By using the quantizer $\mathcal{Q}$ on $\mathbf{y}_p$, one can obtain an index vector $\mathbf{i}_p = [i_{1,p}, i_{2,p}, \cdots, i_{Q,p}]^T$ where $i_{q,p}$ is the bin index of $\mathcal{Q}$ where $y_{q,p}$ falls and $i_{q,p} \in \{0, 1, \cdots, 2^r - 1\}$. Alice and Bob can both obtain the vectors $\mathbf{y}_p$ and $\mathbf{y}'_p$ for $p = 1, \cdots, L_{\mathcal{K}}$ from their realizations of y_k and y'_k, and Alice can construct index vector $\mathbf{i}_p$

For Alice and Bob to use $\mathbf{y}_p$ and $\mathbf{y}'_p$ to extract pth secret bit, we propose the use of a 'helper tensor' $\mathbb{T}$. $\mathbb{T}$ is a Q dimensional tensor with 2^r elements in each dimension i.e. $\mathbb{T} \in \mathcal{R}^{R_1\times R_2\times,\cdots,\times R_Q}$ where $R_1 = R_2 = \cdots = R_Q = 2^r$. This implies $\mathbb{T}$ consists of 2^{rQ} elements where each element can be denoted as $t_{j_1,j_2,\cdots,j_Q}$ where $j_q \in \{0, 1, \cdots, 2^r - 1\}$. Also the elements of $\mathbb{T}$ consists of all the non-negetive integers $[0, 1, \cdots, 2^{rQ} - 1]$. Now we define an 'MSB pair' of integers $(i, \tilde{i}) \in \{0, \cdots, 2^{rQ} - 1\}$ where their binary representation $\mathbf{b}_i$, $\mathbf{b}_{\tilde{i}}$ differs only by the most significant bit (MSB), i.e. $(i, \tilde{i})$ satisfies following conditions:

$$\tilde{i} = (i + 2^{rQ-1})_{mod-(0,2^{rQ}-1)} \tag{4.6}$$

$$i = (\tilde{i} + 2^{rQ-1})_{mod-(0,2^{rQ}-1)} \tag{4.7}$$

For example $(4, 36), (24, 56)$ are such pairs for $rQ = 6$. The integer elements in $\mathbb{T}$ are arranged in such a way that L_1 distance of the positions of such MSB pairs in $\mathbb{T}$ is constant for all pairs i.e. if $t_{j_1,j_2,\cdots,j_Q} = i$ and $t_{\tilde{j_1},\tilde{j_2},\cdots,\tilde{j_Q}} = \tilde{i}$ then for all MSB pairs $(i, \tilde{i})$:

$$\sum_{q=1}^{Q} ||j_q - \tilde{j_q}||_1 = Q2^{r-1} \tag{4.8}$$

Example of such helper tensor $\mathbb{T}$ for $Q = 3$ and $r = 2$ is given in fig. 4.5 and its construction in described in section 4.5.2. $\mathbb{T}$ can be generated offline given the parameters Q and r. Alice after obtaining the index vector $\mathbf{i}_p$ by quantizing $\mathbf{y}_p$ and Bob after obtaining corresponding $\mathbf{y}'_p$ can generate the pth bit of their secret key $\mathcal{K}$ and $\mathcal{K}'$ as the following:

First, using the index vector $\mathbf{i}_p = [i_{1,p}, i_{2,p}, \cdots, i_{Q,p}]^T$, Alice retrieves the corresponding integer $t_{i_{1,p},\cdots,i_{Q,p}}$ from $\mathbb{T}$ denoted as i^*_p. Alice then keeps the MSB of i^*_p as the pth bit of $\mathcal{K}$ and sends the remaining bits to Bob. On the other hand, Bob upon receiving the bits from Alice, determines the integer pair $(i^*_p, \tilde{i_p}^*)$ where all the bits except the MSB are same as the received bits. Bob then retrieves the index vectors $\mathbf{i}_p$ and $\tilde{\mathbf{i}_p}$ from $\mathbb{T}$ corresponding to the locations of the integer pair $(i^*_p, \tilde{i_p}^*)$. Then Bob constructs $\widehat{\mathbf{y}_p}$ and $\widehat{\tilde{\mathbf{y}}_p}$ using midpoints of the bins in $\mathcal{Q}$ corresponding to the indices $\mathbf{i}_p$ and $\tilde{\mathbf{i}_p}$. Finally, for the pth bit of $\mathcal{K}'$, Bob chooses between the MSB of integers i^*_p and $\tilde{i_p}^*$ depending on which of the two vectors $\widehat{\mathbf{y}_p}$ and $\widehat{\tilde{\mathbf{y}}_p}$ have minimum L_2 distance with $\mathbf{y}'_p$.

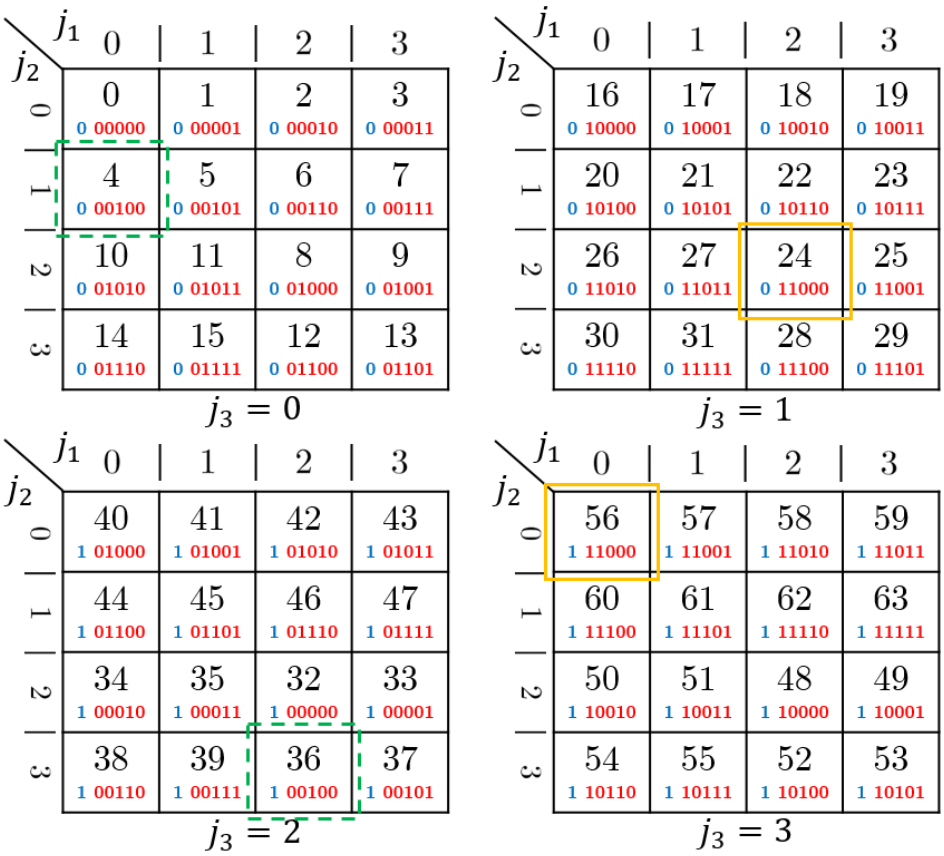


Figure 4.5: Illustration of $\mathbb{T}$ for $Q = 3$ and $r = 2$. The integers from 0 to $2^6 - 1$ are arranged in such a way that all integer pairs whose binary is the same except the MSB have same L_1 distance of 6 according to (4.8). For example, integers of pair $(4, 36)$ lives in $t_{0,1,0}$ and $t_{2,3,2}$ respectively where the L_1 distance is apparently 6. Another such pair is $(24, 56)$

4.5.2 Construction of Tensor $\mathbb{T}$

For given Q and r, the helping tensor $\mathbb{T}$ consists of all the integers $i = 0, 1, \cdots, 2^{rQ} - 1$ and the elements in $\mathbb{T}$ can be denoted as $t_{j_1, j_2, \cdots, j_Q}$ where $j_q \in \{0, 1, \cdots, 2^r - 1\}$. Initially, the elements of $\mathbb{T}$ are filled up with integers from 0 to $2^{rQ} - 1$ sequentially by iterating through the first indices to the last indices i.e. an element in $\mathbb{T}$ is obtained as:

$$t_{j_1, j_2, \cdots, j_Q} = \sum_{q=1}^{Q} j_q 2^{r(q-1)} \tag{4.9}$$

With the initial $\mathbb{T}$ we can recursively swap different blocks within $\mathbb{T}$. Let $\mathbb{T}_{j_1,:,:,\cdots,:}$ be denoted as a $Q-1$ dimensional sub-tensor of Q dimensional tensor $\mathbb{T}$ by keeping its first index constant and iterating through all other indices thoroughly. Also let $\mathbb{T}_{j_{1,1}:j_{1,2},j_{2,1}:j_{2,2},:,:,\cdots,:}$ be denoted as Q dimensional sub-tensor of $\mathbb{T}$ where first two indices are iterated from $j_{1,1}$ to $j_{1,2}$ and from $j_{2,1}$ to $j_{2,2}$ respectively where all other indices are iterated thoroughly. After taking the initially prepared $\mathbb{T}$, we swap the sub-tensors $\mathbb{T}_{2^{r-1}:2^r-1,0:2^{r-1}-1,:,:,\cdots,:}$ and $\mathbb{T}_{2^{r-1}:2^r-1,2^{r-1}:2^r-1,:,:,\cdots,:}$ with each other (graphically illustrated in fig. 4.6). After that we iterate through the first index j_1, extract each sub-tensor $\mathbb{T}_{j_1,:,:,\cdots,:}$ and carry on same swapping operation on them considering them as a different $Q-1$ dimensional tensor until we end up having 2 dimensional matrix in which case we swap $\mathbb{T}_{2^{r-1}:2^r-1,0:2^{r-1}-1}$ and $\mathbb{T}_{2^{r-1}:2^r-1,2^{r-1}:2^r-1}$ with each other. Algorithm 1 illustrates how to construct $\mathbb{T}$:

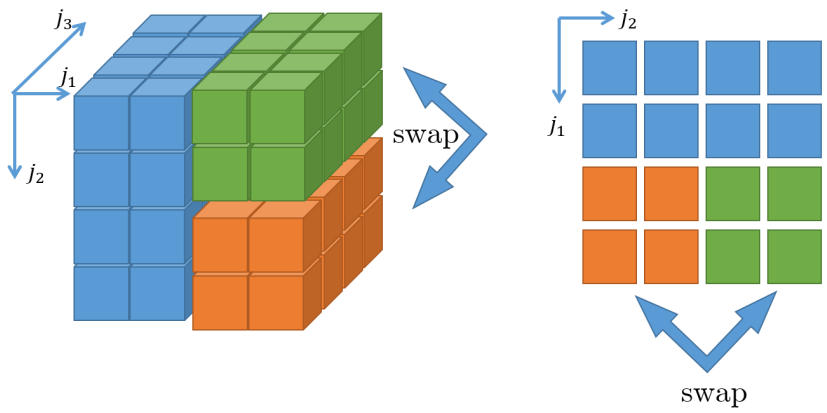


Figure 4.6: Illustration of sub-tensor swap in $\mathbb{T}$ for $Q=3;\ r=2$ (left) and $Q=2;\ r=2$ (right)

Algorithm 1 Algorithm for constructing tensor $\mathbb{T}$

Require: Dimension Q, bits per dimension r

Ensure: Q Dimensional tensor $\mathbb{T}$ having 2^r integer elements in each dimension

Initiate Q dimensional tensor $\mathbb{T}$ with 2^r elements in each dimension where elements can be accessed by $t_{j_1,j_2,\cdots,j_Q}$ where $j_q \in \{0, 1, \cdots, 2^r - 1\}$

Fill $\mathbb{T}$ with integers from 0 to $2^{rQ} - 1$ according to (4.9) by iterating over all the indices

Execute function `BuildMyT`($\mathbb{T}$) using initial $\mathbb{T}$. The function will make changes in $\mathbb{T}$ by moving around the entries. Finally, we will obtain desired helper tensor $\mathbb{T}$

Function `BuildMyT`($\mathbb{T}'$)

    $Q' \leftarrow$ `dimension`($\mathbb{T}'$)

if $Q' \leq 1$ **then**

    return

end if

if $Q' = 2$ **then**

    Swap $\mathbb{T}_{2^{r-1}:2^r-1,0:2^{r-1}-1}$ with $\mathbb{T}_{2^{r-1}:2^r-1,2^{r-1}:2^r-1}$

    return

end if

Swap $\mathbb{T}_{2^{r-1}:2^r-1,0:2^{r-1}-1,:,\cdots,:}$ with $\mathbb{T}_{2^{r-1}:2^r-1,2^{r-1}:2^r-1,:,\cdots,:}$

for all $j_1 = 0 : 2^r - 1$ **do**

    `BuildMyT`($\mathbb{T}_{j_1,:,:,\cdots,:}$)

end for

return

End Function =0

4.6 Key & Bit Error Rate for Different Quantization Scheme

4.6.1 Error Analysis of CEF Output

Let, for a given k the output of CEF at Alice and Bob is y_k and y'_k respectively where we can write $y'_k = y_k + w_{y,k}$. For convenience of subsequent analysis, we drop the index k. The variance of w_y is discussed in section 2.8, but due to nonlinear transformations in the CEF, PDF of w_y is not easily tractable. It is desirable that the variance of w_y does not become much larger compared to the variance of w_x which can be kept under control by pruning out some of the realizations of $\mathbf{Q}_{k,l}$. In this section we use maximum likelihood estimation (MLE) and expectation maximization (EM) method [83] to fit the PDF of w_y from many realizations of w_y for different N and $\texttt{SNR}_x$ so that the estimated PDF can be used in BER/KER analysis. Obviously the variance of w_y depends on $\texttt{SNR}_x$ and so we normalize w_y as $w_y\sqrt{\texttt{SNR}_x}$. In fig. 4.7 we can see the empirical distributions of normalized w_y for various N and $\texttt{SNR}_x$. It is observable that the distribution is practically invariant to $\texttt{SNR}_x$ for $\texttt{SNR}_x \geq 20\texttt{dB}$ i.e. the noise in the output of CEF increases linearly with noise in its input for high SNR. So we take many realizations of normalized w_y for different $\texttt{SNR}_x$ together and fit the PDF of normalized w_y. We can see that the empirical distribution is zero mean and unimodal, so we can model the distribution with zero mean mixed Gaussian of different variances which can be written as:

$$f_{W_y}(w_y;\sigma_1,\cdots,\sigma_K,c_1,\cdots,c_K)=\sum_{k=1}^{K}c_k\phi_k(w_y) \tag{4.10}$$

$$\text{subject to } \sum_{k=1}^{K}c_k=1;\ c_k\geq 0 \tag{4.11}$$

Here, $\phi_k(\cdot)$ is zero mean Gaussian with variance σ_k^2

$$\phi_k(x)=\frac{1}{\sqrt{2\pi\sigma_k^2}}e^{-\frac{x^2}{2\sigma_k^2}} \tag{4.12}$$

We also denote parameter $\Theta=[\sigma_1,\cdots,\sigma_K,c_1,\cdots,c_K]$ for notational convenience. Assuming different i.i.d. realizations of $w_{y,i}$ for $i=1,2,\cdots,n$ we can write the log-likelihood function as:

$$l(\Theta|W_y)=\sum_{i=1}^{n}\log\left[\sum_{k=1}^{K}c_k\phi_k(w_{y,i})\right] \tag{4.13}$$

In (4.10), $c_k\phi_k(w_{y,i})\mathrm{d}w_y$ is the probability of drawing a data point around $w_{y,i}$ from the component $\phi_k(\cdot)$. We can denote the probability of $w_{y,i}$ belonging to the component $\phi_k(\cdot)$ as $\Delta_{i,k}$ and for given Θ and $w_{y,i}$ for $i=1,2,\cdots,n$ we can estimate $\Delta_{i,k}$ as:

$$\Delta_{i,k}=\frac{c_k\phi_k(w_{y,i})}{\sum_{k'=1}^{K}c_{k'}\phi_{k'}(w_{y,i})} \tag{4.14}$$

Clearly, $\sum_{k=1}^{K}\Delta_{i,k}=1$. To maximize the log-likelihood function, from (4.13) we find the derivative of l w.r.t. σ_k and c_k:

$$\frac{\partial l}{\partial\sigma_k} = \sum_{i=1}^{n}\frac{\partial}{\partial\sigma_k}\left[\log\sum_{k'=1}^{K}c_{k'}\phi_{k'}(w_{y,i})\right] \tag{4.15}$$

$$= \sum_{i=1}^{n}\left[\frac{1}{\sum_{k'=1}^{K}c_{k'}\phi_{k'}(w_{y,i})}\left\{\sum_{k''=1}^{K}c_{k''}\frac{\partial}{\partial\sigma_k}\phi_{k''}(w_{y,i})\right\}\right] \tag{4.16}$$

It can be easily shown that:

$$\frac{\partial\phi_j(w_{y,i})}{\partial\sigma_k}=\begin{cases}\left(\frac{w_{y,i}^2}{\sigma_k^3}-\frac{1}{\sigma_k}\right)\phi_k(w_{y,i}) & \text{if } k=j\\ 0 & \text{otherwise}\end{cases} \tag{4.17}$$

So, we can write (4.16) as:

$$\frac{\partial l}{\partial \sigma_k} = \sum_{i=1}^{n}\left[\frac{1}{\sum_{k'=1}^{K} c_{k'}\phi_{k'}(w_{y,i})}\left\{\left(\frac{w_{y,i}^2}{\sigma_k^3}-\frac{1}{\sigma_k}\right)c_k\phi_k(w_{y,i})\right\}\right] \tag{4.18}$$

$$= \frac{c_k\phi_k(w_{y,i})}{\sum_{k'=1}^{K} c_{k'}\phi_{k'}(w_{y,i})}\left(\frac{w_{y,i}^2}{\sigma_k^3}-\frac{1}{\sigma_k}\right) \tag{4.19}$$

$$= \Delta_{i,k}\left(\frac{w_{y,i}^2}{\sigma_k^3}-\frac{1}{\sigma_k}\right) \tag{4.20}$$

Solving (4.20) for 0 we get:

$$\widehat{\sigma_k} = \sqrt{\frac{\sum_{i=1}^{n}\Delta_{i,k}w_{y,i}^2}{\sum_{i=1}^{n}\Delta_{i,k}}} \tag{4.21}$$

As c_k has constraints as in (4.11), this can be handled by writing c_k as a function of unconstrained variable μ_k [84]:

$$c_k = \frac{e^{\gamma_k}}{\sum_{k'=1}^{K} e^{\gamma_k'}}\text{; and it can be shown that:} \tag{4.22}$$

$$\frac{\partial c_k}{\partial \gamma_j} = \begin{cases} c_k - c_k^2 & \text{if } k = j \\ -c_k c_j & \text{otherwise} \end{cases} \tag{4.23}$$

Now, we can write:

$$\frac{\partial l}{\partial \gamma_k} = \sum_{i=1}^{n}\left[\frac{1}{\sum_{k'=1}^{K} c_{k'}\phi_{k'}(w_{y,i})}\left\{\sum_{k''=1}^{K}\phi_{k''}(w_{y,i})\frac{\partial c_{k''}}{\partial \gamma_k}\right\}\right] \tag{4.24}$$

$$= \sum_{i=1}^{n}\left[\frac{1}{\sum_{k'=1}^{K} c_{k'}\phi_{k'}(w_{y,i})}\left\{c_k\phi_k(x_n) - c_k\sum_{k''=1}^{K} c_{k''}\phi_{k''}(x_n)\right\}\right] \tag{4.25}$$

$$= \sum_{i=1}^{n}\left[\Delta_{i,k} - c_k\right] \tag{4.26}$$

Solving (4.26) for 0 we get:

$$\widehat{c_k} = \frac{1}{n}\sum_{i=1}^{n}\Delta_{i,k} \tag{4.27}$$

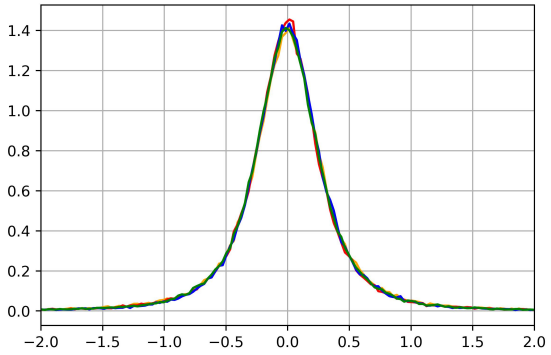
Using (4.14),(4.21) and (4.27) we can estimate the fitting parameters $\sigma_1, \cdots, \sigma_K$ and c_1, $\cdots, c_K$ by iteration until convergence. Having many realizations of $w_{y,i}$ for $i = 1, 2, \cdots, n$ we start of with random values of σ_k and c_k (subject to $\sum_{k=1}^{K} c_k = 1;\ c_k \geq 0$) and then obtain $\Delta_{i,k};\ \forall i, \forall k$ using (4.14). Then we update σ_k and c_k for $\forall k$ using (4.21) and (4.27). We carry on this update process until the values of the parameters converge. Algorithm 2 illustrates how to estimate the parameters:

Algorithm 2 Algorithm for fitting mixed Gaussian over $w_{y,i}$

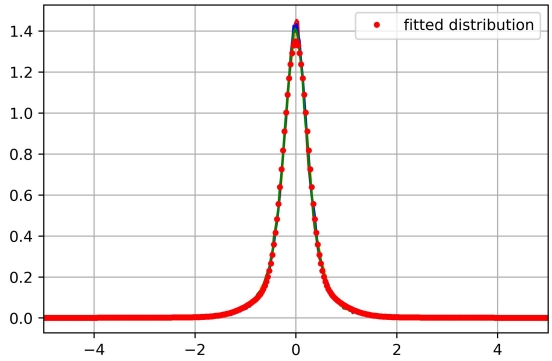
Require: Samples of data $w_{y,i}$ for $i = 1, 2, \cdots, n$, number of parameters K, random initializations of $\sigma_1, \cdots, \sigma_K$ and $c_1, \cdots, c_K$ subject to $\sum_{k=1}^{K} c_k = 1;\ c_k \geq 0$

Ensure: Parameters $\sigma_1, \cdots, \sigma_K$ and $c_1, \cdots, c_K$ after fitting the distribution on data $w_{y,i}$

1: **while** Parameters not converged **do**
2: **for** $i = 1$ to n **do**
3: **for** $k = 1$ to K **do**
4: Update $\Delta_{i,k}$ using (4.14)
5: **end for**
6: **end for**
7: **for** $k = 1$ to K **do**
8: Update $\widehat{\sigma_k}$ using (4.21)
9: Update c_k using (4.27)
10: **end for**
11: Check for convergence by comparing σ_k and c_k with their previous value
12: **end while**=0



Histogram of $w_y\sqrt{\texttt{SNR}_x}$ for $N = 16$ and $\texttt{SNR}_x = 20, 30, 40, 50\texttt{dB}$. It can be seen that histogram of normalized w_y for different $\texttt{SNR}_x$ is practically same.



Fitted distribution over the histogram using the above discussed method for $K = 3$

Figure 4.7: Fitting distribution over histogram data obtained from 10^5 realizations of w_y for different $\texttt{SNR}_x$

After obtaining $\sigma_1, \cdots, \sigma_K$ and $c_1, \cdots, c_K$ we can use the expression in (4.10) as the PDF of w_y for given N. From experiments, for $N = 16$ and $K = 3$ the parameters are obtained to be the following: $\sigma_1 = 0.234$, $\sigma_2 = 0.622$, $\sigma_3 = 2.634$ and $c_1 = 0.679$, $c_2 = 0.298$, $c_3 = 0.023$

4.6.2 KER and BER analysis

We discuss here the bit error rate (BER) and key error rate (KER) of the schemes discussed above using fitted mixed Gaussian distribution in (4.10) and parameters obtained by algorithm 2. We assume the estimation noise in $\mathbf{x}$ is Gaussian with variance $\frac{1}{\texttt{SNR}_x}$ i.e. $\mathbf{x}$ and $\mathbf{x}'$ being $N \times 1$ channel estimation of Alice and Bob respectively, we can write $\mathbf{x}' = \mathbf{x} + \mathbf{w}_x$ where $\mathbf{w} \in \mathcal{R}^{N\times 1}$. Under this circumstance, the PDF and CDF of the output y of the CEF is known to be [46]:

$$f_Y(y) = \frac{\Gamma(\frac{N}{2})}{\sqrt{\pi}\Gamma(\frac{N-1}{2})}(1-y^2)^{\frac{N-3}{2}} \tag{4.28}$$

$$F_Y(y) = \frac{\Gamma(\frac{N}{2})}{\sqrt{\pi}\Gamma(\frac{N-1}{2})}\int \cos^{N-2}\theta d\theta + \frac{1}{2} \tag{4.29}$$

Where $-1 \leq y \leq 1$ and $\theta = \sin^{-1} y$. For a given BER, KER of a key with length $L_{\mathcal{K}}$ can be written as:

$$KER = 1 - (1 - BER)^{L_{\mathcal{K}}} \tag{4.30}$$

$Q = 1;\ \mu = 1$ **case**

Lets assume a case where $Q = 1;\ \mu = 1$ i.e. 1 bit extracted from 1 sample of y. For convenience of analysis, we also assume an arbitrarily high number r of over quantized bits. We are using equiprobable quantizer which means the CDF of y, $0 \leq F_Y(y) \leq 1$ is uniformly divided into 2^{r+1} regions where the lower half $0 \leq F_Y(y) \leq \frac{1}{2}$ $(-1 \leq y \leq 0)$ is divided into 2^r regions and upper half $\frac{1}{2} \leq F_Y(y) \leq 1$ $(0 \leq y \leq 1)$ is also divided into 2^r regions. For sufficiently high r, the regions will be small enough to assume that they represent a particular value of $F_Y(y)$. For a given r over quantized bits, there are 2 realizations of $F_Y(y)$ which are $\frac{1}{2}$ distance apart from each other i.e. if $-1 \leq y_1 \leq 0$ corresponds to given r over quantized bits of lower half of $F_Y(y)$ and $0 \leq y_2 \leq 1$ corresponds to same r over quantized bits of upper half of $F_Y(y)$, we can write $F_Y(y_2) - F_Y(y_1) = \frac{1}{2}$. So Alice sending r over quantized bits to Bob can be imagined as Alice sending the pair (y_1, y_2) where $y_2 = F_Y^{-1}\{F_Y(y_1) + \frac{1}{2}\}$. Bob on the other hand selects between y_1 and y_2 to identify the secret bit, the one that is closer to Bob's observation y'. Given Alice's observation y

and mixed Gaussian noise w_y, the BER is governed by the distance d between y_1 and y_2; $d = y_2 - y_1$.

Now, the distance d is not constant and depends on y_1 (or y_2). For BER analysis, we estimate the mean d. Here the distribution of y given in (4.28), is divided into equal parts for $y_1 \leq 0$ and for $y_2 \geq 0$ which after normalizing can be written as:

$$f_{Y'}(y_{(1,2)}) = 2\frac{\Gamma(\frac{N}{2})}{\sqrt{\pi}\Gamma(\frac{N-1}{2})}(1 - y_{(1,2)}^2)^{\frac{N-3}{2}};\ -1 \leq y_1 \leq 0;\ 0 \leq y_2 \leq 1 \tag{4.31}$$

Now we can estimate the mean d as:

$$\overline{d} = \mathbb{E}_{y_1}\{y_2 - y_1\} = \mathbb{E}_{y_1}\{y_2\} - \mathbb{E}_{y_1}\{y_1\} \tag{4.32}$$

As $f_Y(y)$ in (4.28) is symmetric and unimodal, y_1 and y_2 are one-to-one related which means we can write $\mathbb{E}_{y_1}\{y_2\} = \mathbb{E}_{y_2}\{y_2\}$. Now, from (4.31) we can write:

$$\begin{aligned} \mathbb{E}_{y_2}\{y_2\} &= \int_0^1 2\frac{\Gamma(\frac{N}{2})}{\sqrt{\pi}\Gamma(\frac{N-1}{2})}(1 - y^2)^{\frac{N-3}{2}} y\, dy & (4.33)\\ &= \frac{2\Gamma(\frac{N}{2})}{(N-1)\sqrt{\pi}\Gamma(\frac{N-1}{2})} & (4.34) \end{aligned}$$

Similarly, we can evaluate $\mathbb{E}_{y_1}\{y_1\} = -\frac{2\Gamma(\frac{N}{2})}{(N-1)\sqrt{\pi}\Gamma(\frac{N-1}{2})}$. From (4.32) and normalizing for $\mathtt{SNR}_x$, we ca write:

$$\overline{d}_{N,\mathtt{SNR}_x} = \frac{4\Gamma(\frac{N}{2})}{(N-1)\sqrt{\pi}\Gamma(\frac{N-1}{2})}\sqrt{\mathtt{SNR}_x} \tag{4.35}$$

From the PDF $f_{W_y}(w_y;\Theta)$ of w_y in (4.10) in section 4.6.1 where $\Theta = [\sigma_1,\cdots,\sigma_K, c_1,\cdots,c_K]$, we can the write the BER as:

$$\begin{aligned} \overline{\mathrm{BER}}_{th,N,\mathtt{SNR}_x} &= \sum_{k=1}^{K} c_k \mathbb{Q}\left(\frac{\overline{d}_{N,\mathtt{SNR}_x}}{2\sigma_k}\right) & (4.36)\\ \text{where; } \mathbb{Q}(x) &= \int_x^\infty \frac{1}{\sqrt{2\pi}} e^{-\frac{u^2}{2}}\, du & (4.37) \end{aligned}$$

We can also estimate the upper bound of BER by evaluating the minimum possible distance between y_1 and y_2. As $f_Y(y)$ is a symmetric unimodal function,the minimum distance between y_1 and y_2 occurs when $y_1 = -y_2$ which implies $F_Y(y_1) = \frac{1}{4};\ F_Y(y_2) = \frac{3}{4}$. So, the minimum possible distance $\widetilde{d}$ can be obtained as $\widetilde{d} = 2F_Y^{-1}\left(\frac{3}{4}\right)$. (For $N = 16;\ \widetilde{d} = 0.3514$). Using similar argument for (4.36), we can write the upper bound of BER as:

$$\widetilde{\mathrm{BER}}_{th,N,\mathtt{SNR}_x} = \sum_{k=1}^{K} c_k \mathbb{Q}\left(\frac{\widetilde{d}_{N,\mathtt{SNR}_x}}{2\sigma_k}\right) \tag{4.38}$$

$Q > 1;\ \mu < 1$ **case**

In case of vector quantization, Q samples of y generate 1 bit of secret key. The distance between two realizations $\mathbf{y}_1$ and $\mathbf{y}_2$ like discussed in section 4.6.2 can be written as $\mathbb{E}_{\mathbf{y_1}}\{||\mathbf{y}_2 - \mathbf{y}_1||_2\}$. As the samples of y are uncorrelated, the expectation along each dimension is the same and equal to (4.32). So, considering the effect of Q, (4.36) and (4.38) can be written as:

$$\overline{\mathrm{BER}}_{th,N,\mathtt{SNR}_x} = \sum_{k=1}^{K} c_k \mathbb{Q}\left(\frac{\overline{d}_{N,\mathtt{SNR}_x}\sqrt{Q}}{2\sigma_k}\right) \tag{4.39}$$

$$\widetilde{\mathrm{BER}}_{th,N,\mathtt{SNR}_x} = \sum_{k=1}^{K} c_k \mathbb{Q}\left(\frac{\widetilde{d}_{N,\mathtt{SNR}_x}\sqrt{Q}}{2\sigma_k}\right) \tag{4.40}$$

Corresponding KER can be found using (4.30). In the following plot, we can see the theoretical and experimental results.

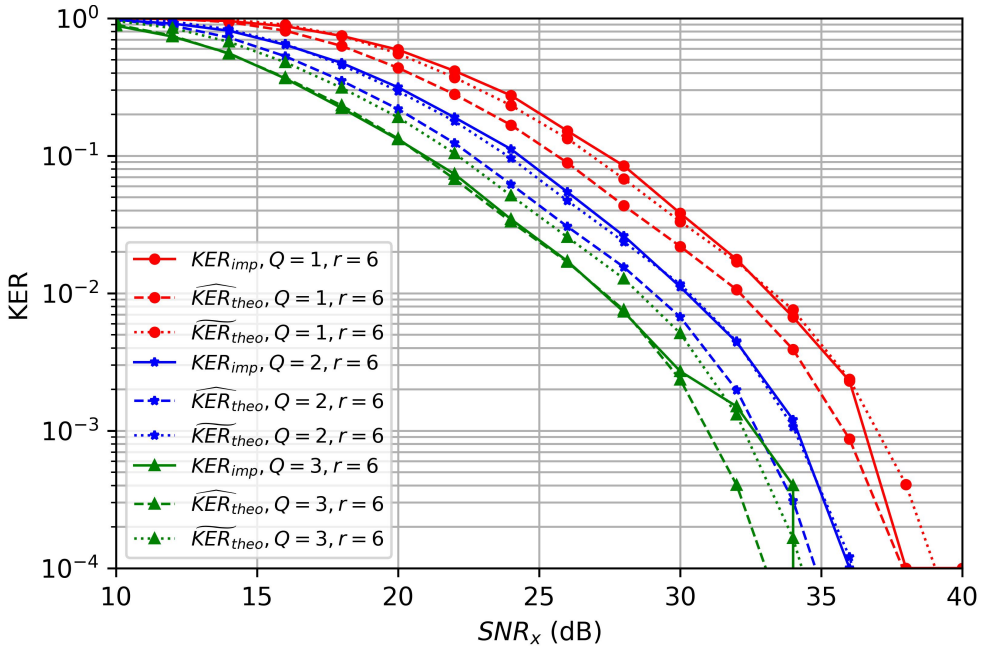


Figure 4.8: Empirical and theoretical KER VS $\mathtt{SNR}_x$ for $N = 16$ and different parameters. For given $\mathtt{SNR}_x$, 10^4 realizations of $\mathbf{x}$, $\mathbf{x}'$ and corresponding keys were generated to estimate empirical KER. To approximate the theoretical KER, $K = 25$ Gaussian components were used.

4.7 Conclusion

We have presented a novel method for SKG. Simulation results show that at a moderate or high SNR, subject to a required quality of key randomness, the proposed method called CEbQ has the best reliability in terms of KER compared to other methods based on DQ, DRP and IoM. The main reason for this improved reliability is that after continuous encryption, a lower rate quantizer per encrypted sample can be applied without reduction of key size. Furthermore, the SVD-CEF used with CEbQ is a good QCPRN-generator, which ensures sufficient randomness of a long key generated from a secret vector of a limited dimension. A controlled leakage for both DRP and SVD-CEF due to over-

quantization yields a major improvement or reduction of KER. Future research on the security impact of such leakage is needed.

Chapter 5

Secret-Key Capacity From MIMO Channel Probing

5.1 Introduction

Secret keys are essential for confidentiality, integrity and authenticity in both military and civilian networks. Secret key generation (SKG) between two wireless mobile nodes in dynamic channel environments is useful for situations where a secret key between the nodes needs to be established or enhanced on the fly.

Prior research on SKG has spanned more than three decades [85]. When nodes A and B need to establish a secret key based on their respective observations $\mathcal{X}$ and $\mathcal{Y}$ in the presence of an eavesdropper with the observation $\mathcal{Z}$, the secret-key capacity (SKC) C_S in bits per independent realization of $\{\mathcal{X}, \mathcal{Y}, \mathcal{Z}\}$ is known [14] to be bounded as follows $C_L \leq C_S \leq C_U$ with $C_L = I(\mathcal{X};\mathcal{Y}) - \min(I(\mathcal{X};\mathcal{Z}), I(\mathcal{Y};\mathcal{Z}))$ and $C_U = \min(I(\mathcal{X};\mathcal{Y}), I(\mathcal{X};\mathcal{Y}|\mathcal{Z}))$. Here

$I(a;b|c)$ denotes the mutual information between a and b given c. Although formal proofs of these bounds are all based on discrete $\{\mathcal{X}, \mathcal{Y}, \mathcal{Z}\}$, by using the argument of generalized mutual information [86], the above bounds also hold for continuous $\{\mathcal{X}, \mathcal{Y}, \mathcal{Z}\}$.

Despite the utility of the above bounds which were pioneered by Maurer, Ahlswede and Csiszar (MAC), few of the prior works on secret-key capacity make use of the MAC bounds to gather deeper insights into various possible channel probing schemes for SKG. Recently in [33], the MAC bounds are used to reveal the degree of freedom (DoF) of SKC based on MIMO channel probing schemes. By utilizing the MAC bounds, one does not need to repeat the information-theoretic formal description and/or derivation of the same or slightly different bounds for secret-key capacity but rather can focus on new discoveries. It should be noted that the MAC bounds hold for both weak and strong secret-key capacity [85].

For continuous observations $\{\mathcal{X}, \mathcal{Y}, \mathcal{Z}\}$, there is in general a nominal signal-to-noise ratio ($\mathtt{SNR}$), and for $\mathtt{SNR} \gg 1$, $C_S \approx d \log \mathtt{SNR} + c$. More precisely, $d = \lim_{\mathtt{SNR}\to\infty} \frac{C_S}{\log \mathtt{SNR}}$ and $c = \lim_{\mathtt{SNR}\to\infty}(C_S - d \log \mathtt{SNR})$. The value of d is called the DoF of C_S. In this chapter, we will refer to $d \log \mathtt{SNR}$ and c respectively as the first-order term ($\mathtt{FoT}$) and second-order term ($\mathtt{SoT}$) of C_S. The corresponding analyses will be called first- and second-order analyses although the latter is in general intertwined with the former.

In [33], a first-order analysis of C_S was done for a few MIMO based schemes for SKG. In this chapter, we make a contribution beyond [33]. Specifically, we will derive both the $\mathtt{FoT}$ and $\mathtt{SoT}$ of C_L and C_U (lower and upper bound on C_S) assuming a MIMO channel probing scheme (or called MIMO-Hybrid-Probing) where both public pilots and random

symbols are used. While the FoT of the bounds are the same and coincides with that in [33] (as expected), the SoTs of the bounds reveals novel insights. In particular, our result on SoTs shows that the SKC in bits per channel coherence period based on MIMO-Hybrid-Probing increases linearly with the number of random transmissions from one node to another in each coherence period regardless of the number of antennas on Eve. This result goes beyond those shown in [33, 71, 87] for example.

In Section 5.2, we describe the system model, i.e., the MIMO-Hybrid-Probing scheme from which $\{\mathcal{X}, \mathcal{Y}, \mathcal{Z}\}$ are generated. In Section 5.3, we discuss the main results in this chapter. Section 5.5 provides the details of the analysis behind the main results, for which random-matrix theory is also applied. Section 5.6 shows simulation results to validate our analysis. Section 5.7 concludes the chapter.

Notations: The column-wise stacks of a matrix $\mathbf{A}$ and its transpose $\mathbf{A}^T$ are respectively denoted by $\mathbf{a} = vec(\mathbf{A})$ and $\mathbf{a}^t = vec(\mathbf{A}^T)$. $\mathbb{E}_x\{\cdot\}$ denotes the expectation over x. The logarithm log is of the base 2. The circular complex Gaussian distribution with mean $\mathbf{m}$ and covariance matrix $\mathbf{C}$ is denoted by $\mathcal{CN}(\mathbf{m}, \mathbf{C})$. The differential entropy of $\mathbf{x}$ given $\mathbf{y}$ is denoted by $h(\mathbf{x}|\mathbf{y})$. We also use $(x)^+ = \max(0, x)$ and $\log^\dagger x = \log x$ for $x > 0$ and $\log^\dagger x = 0$ for $x = 0$. Also, $\delta_{|\rho|=1} = 1$ if $|\rho| = 1$, and $\delta_{|\rho|=1} = 0$ if $|\rho| < 1$.

5.2 System Model

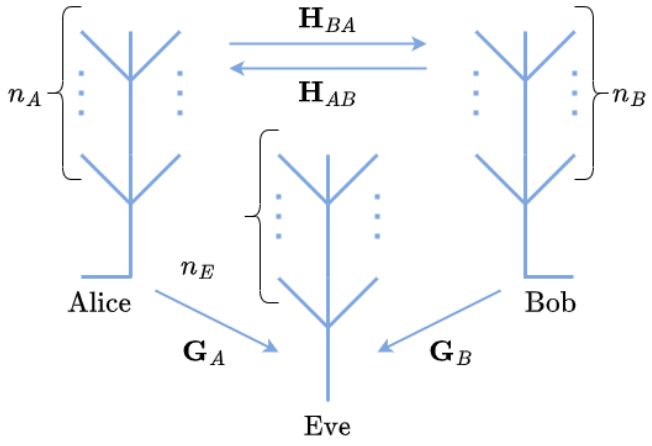


Figure 5.1: Illustration of channel model

We consider the MIMO channel (Fig. 5.1) between legitimate nodes A and B (Alice and Bob) in the presence of an Eavesdropper (Eve). The numbers of antennas on these nodes are respectively n_A, n_B, and n_E. The channel response matrices from Alice to Bob and Bob to Alice are denoted by $\mathbf{H}_{BA}$ and $\mathbf{H}_{AB}$ respectively, and channel response matrices from Alice to Eve and Bob to Eve are denoted by $\mathbf{G}_A$ and $\mathbf{G}_B$ respectively. Each of the channel coherence periods is divided into four regions. In region 1, Alice transmits an orthogonal public pilot matrix $\mathbf{\Pi}_A \in \mathcal{C}^{n_A \times n_A}$ of power ψ_A over n_A antennas and n_A time slots. Here $\mathbf{\Pi}_A^H \mathbf{\Pi}_A = \psi_A \mathbf{I}_{n_A}$. In region 2, Alice transmits a random matrix $\mathbf{X}_A \in \mathcal{C}^{n_A \times v_A}$ of unit power over n_A antennas and v_A time slots. Similarly, Bob transmits an orthogonal public pilot matrix $\mathbf{\Pi}_B \in \mathcal{C}^{n_B \times n_B}$ of power ψ_B in region 3, and a random matrix $\mathbf{X}_B \in \mathcal{C}^{n_B \times v_B}$ of unit power in region 4. Here $\mathbf{\Pi}_B^H \mathbf{\Pi}_B = \psi_B \mathbf{I}_{n_B}$. We assume $\psi_A, \psi_B \gg 1$. Then, the signals received by Alice, Bob, and Eve can be expressed as follows:

$$\left[\mathbf{Y}_A^{(1)}, \mathbf{Y}_A^{(2)}\right] = \sqrt{\gamma_{BA}}\left[\mathbf{H}_{AB}\mathbf{\Pi}_B, \mathbf{H}_{AB}\mathbf{X}_B\right] + \left[\mathbf{W}_A^{(1)}, \mathbf{W}_A^{(2)}\right] \tag{5.1a}$$

$$\left[\mathbf{Y}_B^{(1)}, \mathbf{Y}_B^{(2)}\right] = \sqrt{\gamma_{AB}}\left[\mathbf{H}_{BA}\mathbf{\Pi}_A, \mathbf{H}_{BA}\mathbf{X}_A\right] + \left[\mathbf{W}_B^{(1)}, \mathbf{W}_B^{(2)}\right] \tag{5.1b}$$

$$\left[\mathbf{Y}_{EA}^{(1)}, \mathbf{Y}_{EA}^{(2)}\right] = \sqrt{\gamma_{AE}}\left[\mathbf{G}_A\mathbf{\Pi}_A, \mathbf{G}_A\mathbf{X}_A\right] + \left[\mathbf{W}_{EA}^{(1)}, \mathbf{W}_{EA}^{(2)}\right] \tag{5.1c}$$

$$\left[\mathbf{Y}_{EB}^{(1)}, \mathbf{Y}_{EB}^{(2)}\right] = \sqrt{\gamma_{BE}}\left[\mathbf{G}_B\mathbf{\Pi}_B, \mathbf{G}_B\mathbf{X}_B\right] + \left[\mathbf{W}_{EB}^{(1)}, \mathbf{W}_{EB}^{(2)}\right] \tag{5.1d}$$

Here the entries of $\mathbf{X}_A$, $\mathbf{X}_B$ and $\mathbf{H}_{BA}$ are i.i.d. $\mathcal{CN}(0,1)$, i.e., $\mathbb{E}_{\mathbf{x}_A}[\mathbf{x}_A\mathbf{x}_A^H] = \mathbf{I}_{n_A v_A}$ and $\mathbb{E}_{\mathbf{x}_B}[\mathbf{x}_B\mathbf{x}_B^H] = \mathbf{I}_{n_B v_B}$. The relationship between $\mathbf{H}_{BA}$ and $\mathbf{H}_{AB}$ is modelled as jointly Gaussian with zero mean and $\mathbb{E}\{\mathbf{h}_{AB}^t\mathbf{h}_{BA}^H\} = \rho\mathbf{I}_{n_A n_B}$. Let $\mathbf{C}_{\mathbf{x}|\mathbf{y}}$ denote the conditional covariance matrix of $\mathbf{x}$ given $\mathbf{y}$. It follows that $\mathbf{C}_{\mathbf{h}_{AB}|\mathbf{h}_{BA}^t} = \mathbf{C}_{\mathbf{h}_{BA}^t|\mathbf{h}_{AB}} = (1-|\rho|^2)\mathbf{I}_{n_A n_B}$. Here, $|\rho| = 1$ if the channel is perfectly reciprocal, and $|\rho| = 0$ if the channel is completely non-reciprocal. We also assume that $\mathbf{G}_A \in \mathcal{C}^{n_E \times n_A}$ and $\mathbf{G}_B \in \mathcal{C}^{n_E \times n_B}$ are independent of $\mathbf{X}_A$, $\mathbf{X}_B$, $\mathbf{H}_{BA}$ and $\mathbf{H}_{AB}$, and have i.i.d. $\mathcal{CN}(0,1)$ entries. The entries of the noise matrices (i.e., the $\mathbf{W}$ matrices) are also assumed (due to a normalization) to be i.i.d. $\mathcal{CN}(0,1)$. Hence, we can write $\gamma_{AB} = \frac{\alpha_A P}{\lambda_B}$, $\gamma_{BA} = \frac{\alpha_B P}{\lambda_A}$, $\gamma_{AE} = \frac{\alpha_A P}{\lambda_{EA}}$ and $\gamma_{BE} = \frac{\alpha_B P}{\lambda_{EB}}$ where P is a nominal signal power or a nominal `SNR`. Also, for example, α_A indicates a relative power gain from Alice to Bob, and λ_B indicates a relative noise variance at Bob. We will assume that all the values of α and λ are deterministic and constant of P.

Therefore, the overall data sets available to Alice, Bob, and Eve in each coherence period are as follows:

$$\text{Alice: } \mathcal{X} = \left\{ \mathbf{X}_A, \mathbf{Y}_A^{(1)}, \mathbf{Y}_A^{(2)} \right\}, \tag{5.2a}$$

$$\text{Bob: } \mathcal{Y} = \left\{ \mathbf{X}_B, \mathbf{Y}_B^{(1)}, \mathbf{Y}_B^{(2)} \right\}, \tag{5.2b}$$

$$\text{Eve: } \mathcal{Z} = \left\{ \mathbf{Y}_{EA}^{(1)}, \mathbf{Y}_{EA}^{(2)}, \mathbf{Y}_{EB}^{(1)}, \mathbf{Y}_{EB}^{(2)} \right\}. \tag{5.2c}$$

Without loss of generality, we assume $n_A \geq n_B$. Let $C_A = I(\mathcal{X};\mathcal{Y}) - I(\mathcal{X};\mathcal{Z}) = h(\mathcal{X}|\mathcal{Z}) - h(\mathcal{X}|\mathcal{Y})$ and $C_B = I(\mathcal{X};\mathcal{Y}) - I(\mathcal{Y};\mathcal{Z}) = h(\mathcal{Y}|\mathcal{Z}) - h(\mathcal{Y}|\mathcal{X})$. Then, we have $\max(C_A, C_B) = C_L \leq C_S \leq C_U$. It is shown in [33] that for the data sets defined in (5.2a)-(5.2c), the degrees of freedom (DoF) of C_L, C_S and C_U relative to $\log P$ are equal, i.e., $\texttt{DoF}(C_L) = \texttt{DoF}(C_S) = \texttt{DoF}(C_U)$, and if $n_A \geq n_B$, $\texttt{DoF}(C_L) = \texttt{DoF}(C_B) \geq \texttt{DoF}(C_A)$. Therefore, to analyze C_L subject to $n_A \geq n_B$, we will focus on C_B. We also know that $C_U \leq C_Z = I(\mathcal{X};\mathcal{Y}|\mathcal{Z}) = h(\mathcal{X}|\mathcal{Z}) - h(\mathcal{X}|\mathcal{Y},\mathcal{Z})$. Thus for the analysis of C_U, we will focus on C_Z.

5.3 Main Results

The upper and lower bounds can be expressed in terms of FoT and SoT. Assuming $P, \psi_A, \psi_B \gg 1$, $n_A \geq n_B \gg 1$ and $n_E \gg 1$, we will show that

$$C_B \approx \texttt{FoT} + \texttt{SoT}^{(1)} + \texttt{SoT}_{C_B}^{(2)} \tag{5.3a}$$

$$C_Z \approx \texttt{FoT} + \texttt{SoT}^{(1)} + \texttt{SoT}_{C_Z}^{(2)} \tag{5.3b}$$

where $\texttt{FoT} = d \log P$ and SoTs are invariant to P, i.e.,

$$\texttt{FoT} = v_A \min[n_B, (n_A - n_E)^+] \log P + v_B \left(n_B - n_E\right)^+ \log P + \delta_{|\rho|=1} n_A n_B \log P \tag{5.4}$$

and $\mathtt{SoT}^{(1)}$ is given in (5.5). Also $\mathtt{SoT}^{(2)}_{C_B} = v_B\theta(n_A, n_B)$ and $\mathtt{SoT}^{(2)}_{C_Z} = v_B\mu(n_A, n_E, n_B, \lambda_A/\lambda_{EB})$. Here $\theta(N, K)$, $\mu(N_1, N_2, K, \xi)$ and κ are given by (5.16), (5.19) and (5.21) respectively.

$$\begin{aligned}\mathtt{SoT}^{(1)} =& v_A[\min\{n_B, (n_A - n_E)^+\}\log\alpha_A - \{\min(n_B + n_E, n_A)\log\lambda_B - \min(n_E, n_A)\log\lambda_{EA}\}] \\ &+ v_B[(n_B - n_E)^+\log\alpha_B - \{n_B\log\lambda_A - \min(n_E, n_B)\log\lambda_{EB}\}] \\ &+ \kappa - (1-\delta_{|\rho|=1})\log(1-|\rho|^2) + v_A[\mu(n_B, n_E, n_A, \lambda_B/\lambda_{EA}) - \theta(n_E, n_A)] - v_B\theta(n_E, n_B)\end{aligned} \tag{5.5}$$

As expected, $\frac{\mathtt{FoT}}{\log P}$ shown above equals the DoF of C_S shown in [33] subject to $n_A \geq n_B$. It is clear that if $n_E \geq n_B$, the $\mathtt{FoT}$ is not affected by choosing $v_B = 0$. For more discussions of $\mathtt{FoT}$ and its comparison with prior works, see [33]. We also see that the only different term between C_B and C_Z is $\mathtt{SoT}^{(2)}_{C_B}$ and $\mathtt{SoT}^{(2)}_{C_Z}$ which vanishes if we choose $v_B = 0$.

In the case of $n_E \geq n_A \geq n_B$, (5.4) reduces to $\mathtt{FoT} = \delta_{|\rho|=1} n_A n_B \log P$ which is invariant to both v_A and v_B, but (5.5) reduces to resulting in:

$$\mathtt{SoT}^{(1)} + \mathtt{SoT}^{(2)}_{C_B} = \omega_A v_A + \omega_B v_B + \kappa - (1 - \delta_{|\rho|=1}) n_A n_B \log(1 - |\rho|^2)$$

where

$$\begin{aligned}\omega_A =& n_A[\log\lambda_{EA} - \log\lambda_B] \\ &+ n_B\log(1 + n_A\tau') + n_E\log(1 + \xi' n_A\tau') \\ &+ (n_E - n_A)\log(n_E - n_A) - n_E\log n_E - n_A\log\tau', \end{aligned} \tag{5.6a}$$

$$\begin{aligned}\omega_B =& n_B[\log\lambda_{EB} - \log\lambda_A] \\ &+ \{(n_E - n_B)\log(n_E - n_B) - (n_A - n_B)\log(n_A - n_B)\} \\ &+ (n_A\log n_A - n_E\log n_E), \end{aligned} \tag{5.6b}$$

Here $\xi' = \frac{\lambda_B}{\lambda_{EA}}$ and $\tau' > 0$ is the solution to:

$$\tau'^2 + \tau' \frac{\xi'(n_A - n_E) + (n_A - n_B)}{(n_A - n_B - n_E)\xi' n_A} + \frac{1}{(n_A - n_B - n_E)\xi' n_A} \tag{5.7}$$

For $n_E \to \infty$, we have $\tau' \to \frac{1}{\xi' n_E}$ (see the discussion at the end of section 5.8.3 of this chapter) and hence (easy to verify) $\omega_A \to n_E \log(1 + \frac{n_A}{n_E}) + n_E \log(1 - \frac{n_A}{n_E}) - n_A \log(1 - \frac{n_A}{n_E}) \to 0$. Note that $\lim_{n\to\infty}(1 + 1/n)^n = e$ and $\lim_{n\to\infty}(1 - 1/n)^n = 1/e$. Also it is shown in Appendix 5.8.5 that $\frac{\partial \omega_A}{\partial n_E} < 0$ for $n_E \geq n_A \geq n_B$. Hence ω_A must be positive for all finite n_E.

On the other hand, for $n_E \to \infty$, we have $\omega_B \to n_B \log \frac{\lambda_{EB}}{\lambda_A} - n_B \log e - n_B \log(n_E - n_B) - (n_A - n_B)\log(n_A - n_B) + n_A \log n_A \approx -n_B \log n_E < 0$. And if $n_E = n_A$, we have $\omega_B = n_B \log \frac{\lambda_{EB}}{\lambda_A}$, which would be negative if the channel from Bob to Eve is less noisy than the channel from Bob to Alice. Therefore, we should generally treat ω_B as negative and hence the best choice of v_B is $v_B = 0$. This is consistent for both `FoT` and `SoT`.

Illustrated in Fig. 5.2 and Fig. 5.3 are ω_A and $-\omega_B$ as functions of n_E. We see that ω_A is positive and decreases as n_E increases but ω_B stays negative for these particular sets of realizations and its magnitude increases as n_E increases.

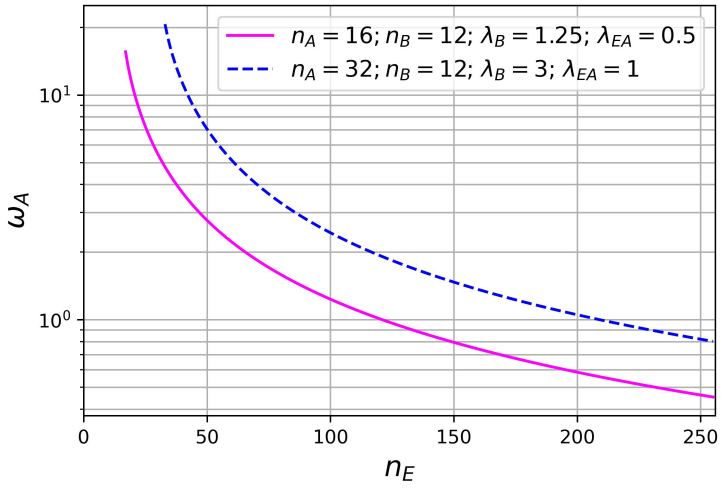


Figure 5.2: The coefficient ω_A of v_A in SoT versus $n_E \geq n_A$.

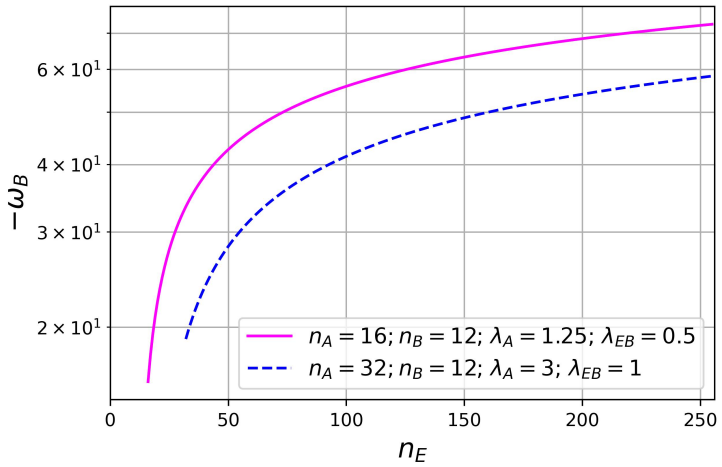


Figure 5.3: The coefficient $-\omega_B$ of $-v_B$ in SoT versus $n_E \geq n_A$.

Proposition 1 *If $n_A \geq n_B$ and $\lambda_{EB} \leq \lambda_A$, the* `FoT` *and* `SoT` *of C_B are maximized by $v_B = 0$, the* `FoT` *is a positive increasing function of v_A subject to $n_A > n_E$, and the* `SoT` *is a positive increasing function of v_A for any $n_E \geq n_A \geq n_B \gg 1$ subject to $v_B = 0$.*

Also, according to this proposition, for $v_B = 0$, the gap between upper and lower bounds which is $v_B[\mu(n_A, n_E, n_B, \lambda_A/\lambda_{EB}) - \theta(n_A, n_B)]$ vanishes. Similar phenomenon is also observed for the SISO case in [34].

5.3.1 More Analysis on the Bounds

From expressions obtained from (5.57) and (5.58), we have the following theorem:

Theorem 2 *Assume large ψ_A and ψ_B, and any $n_A \geq 1$, $n_B \geq 1$ and $n_E \geq 1$. The gap between C_Z and C_B is*

$$C_Z - C_B = v_B \mathbb{E}\{\log|\mathbf{I}_{n_B} + \gamma_{AB}\tilde{\mathbf{H}}_{AB}^H \tilde{\mathbf{H}}_{AB}|\} - v_B \mathbb{E}\{\log|\mathbf{I}_{n_B} + \gamma_{AB}\mathbf{H}_{AB}^H \mathbf{H}_{AB}|\} \tag{5.8}$$

where $\tilde{\mathbf{H}}_{AB}^H \tilde{\mathbf{H}}_{AB} = \mathbf{H}_{AB}^H \mathbf{H}_{AB} + \frac{\lambda_A}{\lambda_{EB}} \mathbf{G}_B^H \mathbf{G}_B$. Equivalently,

$$C_Z - C_B = v_B \mathbb{E}\left\{\log\left|\mathbf{I}_{n_B} + \gamma_{AB}\frac{\lambda_A}{\lambda_{EB}}\mathbf{G}_B^H\mathbf{G}_B \cdot \left(\mathbf{I}_{n_B} + \gamma_{AB}\mathbf{H}_{AB}^H\mathbf{H}_{AB}\right)^{-1}\right|\right\} \geq 0 \tag{5.9}$$

with equality if and only if $v_B = 0$ (provided $\gamma_{AB} > 0$ and $\frac{\lambda_A}{\lambda_{EB}} > 0$). Furthermore,

$$C_B = C_S^{(1)} + v_A \xi_B - v_B \mathbb{E}\{\log|\gamma_{EB}\mathbf{G}_B^H\mathbf{G}_B + \mathbf{I}_{n_B}|\} + v_B \mathbb{E}\{\log|\gamma_{AB}\mathbf{H}_{AB}^H\mathbf{H}_{AB} + \mathbf{I}_{n_B}|\} \tag{5.10}$$

with

$$\xi_B = \mathbb{E}\{\log|\gamma_{EA}\tilde{\mathbf{G}}_A^H\tilde{\mathbf{G}}_A + \mathbf{I}_{n_A}|\} - \mathbb{E}\{\log|\gamma_{EA}\mathbf{G}_A^H\mathbf{G}_A + \mathbf{I}_{n_A}|\} \tag{5.11}$$

and $\tilde{\mathbf{G}}_A^H\tilde{\mathbf{G}}_A = \mathbf{G}_A^H\mathbf{G}_A + \frac{\lambda_{EA}}{\lambda_B}\mathbf{H}_{BA}^H\mathbf{H}_{BA}$. Equivalently,

$$\xi_B = \mathbb{E}\left\{\log\left|\mathbf{I}_{n_A} + \gamma_{BA}\mathbf{H}_{BA}^H\mathbf{H}_{BA} \cdot \left(\gamma_{BA}(\lambda_B/\lambda_{EA})\mathbf{G}_A^H\mathbf{G}_A + \mathbf{I}_{n_A}\right)^{-1}\right|\right\} \geq 0 \tag{5.12}$$

with equality only if $\frac{\lambda_B}{\lambda_{EA}} = \infty$ (provided $\gamma_{BA} > 0$).

5.3.2 Discussion of Theorem 2

Theorem 2 does not require $n_A \geq n_B$. But if $n_A \geq n_B$, we see that both $\mathbf{H}_{AB}$ and $\tilde{\mathbf{H}}_{AB}$ have the full column rank n_B for all $n_E \geq 1$ and hence (one can verify) $\texttt{DoF}(C_Z - C_B) =$

0 for all $v_A \geq 1$, $v_B \geq 1$ and $n_E \geq 1$. This is consistent with a previous result shown in [33]. If $v_A \geq 1$ and $v_B = 0$ (i.e., one-way channel probing from Alice to Bob), then $C_B = C_Z$ and hence

$$\frac{1}{v_A} C_S = \frac{1}{v_A} C_B = \frac{1}{v_A} C_Z = \frac{1}{v_A} C_S^{(1)} + \xi_B \geq \xi_B \tag{5.13}$$

with equality if $\rho = 0$ or $v_A \to \infty$. Since Theorem 2 does not require $n_A \geq n_B$, it also follows that if $v_A = 0$ and $v_B \geq 1$ then $C_S = C_A = C_Z$ (by symmetry between C_A and C_B). In other words, if the channel probing is done only in one direction, the secret-key capacity C_S based on the corresponding data sets always coincides with the corresponding Maurer's lower and upper bounds.

But the channel probing from a node with more antennas to another node with fewer antennas should generally result in a larger C_S in the regime of high power. This is because for $n_A \geq n_B$, $\mathtt{DoF}(C_S) = v_A \min[n_B, (n_A - n_E)^+] + v_B (n_B - n_E)^+ + \delta_\rho n_A n_B$ [33] where $\delta_\rho = 1$ if $|\rho| = 1$, and $\delta_\rho = 0$ if $|\rho| < 1$. Then subject to $v_A + v_B \leq v^*$, $\mathtt{DoF}(C_S)$ is maximized by $v_A = v^*$ and $v_B = 0$.

Theorem 2 also implies that for one-way channel probing from Alice to Bob, the resulting secret-key capacity $\frac{C_S}{v_A}$ in bits per probing instant is always lower bounded by ξ_B which is positive as long as $\lambda_{EA} > 0$ (i.e., the signals received by Eve from Alice are not noiseless).

Numerical illustrations of ξ_B are shown in Figs 5.4 and 5.5. Fig. 5.4 illustrates $\xi_B > 0$ in all cases under $\lambda_B/\lambda_{EA} < \infty$. Fig. 5.5 confirms the theory $\mathtt{DoF}(\xi_B) = \min[n_B, (n_A - n_E)^+]$; i.e., $\mathtt{DoF}(\xi_B) \doteq \lim_{P\to\infty} \frac{\xi_B}{\log P} = 2$ for $n_A = 8$, $n_B = 4$ and $n_E = 10$, and $\mathtt{DoF}(\xi_B) = 0$ for $n_A = 8$, $n_B = 4$ and $n_E = 6$.

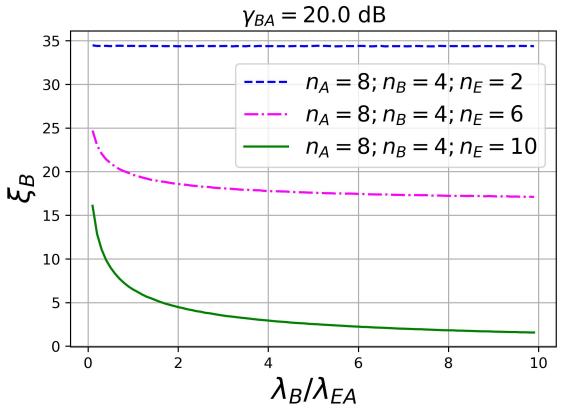


Figure 5.4: ξ_B vs λ_B/λ_{EA}.

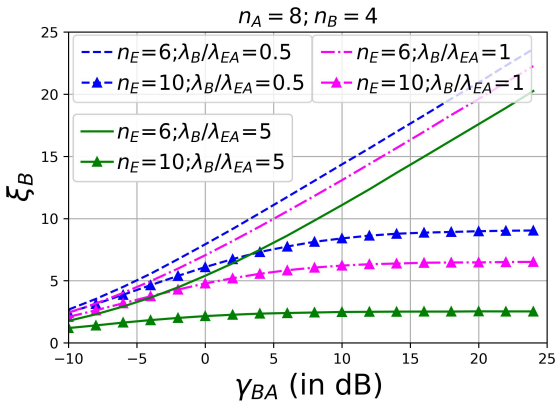


Figure 5.5: ξ_B vs $\gamma_{BA} = \alpha_A P/\lambda_B$.

The contribution of $v_B > 0$ to C_B is either positive or negative, depending on whether or not $|\gamma_{AB}\mathbf{H}_{AB}^H\mathbf{H}_{AB} + \mathbf{I}_{n_B}| > |\gamma_{EB}\mathbf{G}_B^H\mathbf{G}_B + \mathbf{I}_{n_B}|$, i.e., whether or not the MIMO capacity from Bob to Alice is larger than that from Bob to Eve (subject to uniform power scheduling).

5.4 Preliminaries

For the detailed analysis, the following lemmas will be needed.

Lemma 1 *If* $\mathbf{Y} = \sqrt{\gamma}\mathbf{H}\mathbf{\Pi} + \mathbf{W}$ *where* $\mathbf{H} \in \mathcal{C}^{N\times K}$, $\mathbf{\Pi} \in \mathcal{C}^{K\times K}$, $\mathbf{\Pi}^H\mathbf{\Pi} = \psi\mathbf{I}_K$, $\mathbf{Y}, \mathbf{W} \in \mathcal{C}^{N\times K}$, *and all entries in* $\mathbf{H}, \mathbf{W}$ *are i.i.d.* $\mathcal{CN}(0,1)$. *Then for large* γ *and large* ψ, $\mathbf{H}$ *is accurately given if* $\mathbf{Y}$ *and* $\mathbf{\Pi}$ *are given.*

For proof see Appendix 5.8.1.

Lemma 2 *If* $\mathbf{Y} = \sqrt{\gamma}\mathbf{H}\mathbf{X} + \mathbf{W}$ *where* $\mathbf{H} \in \mathcal{C}^{N\times K}$, $\mathbf{X} \in \mathcal{C}^{K\times M}$, $\mathbf{Y}, \mathbf{W} \in \mathcal{C}^{N\times M}$, *all entries in* $\mathbf{X}, \mathbf{W}$ *are i.i.d.* $\mathcal{CN}(0,1)$, *and all entries in* $\mathbf{H}$ *are i.i.d. with zero mean and unit variance. Then for large* N, K *and* γ*:*

$$h(\mathbf{Y}|\mathbf{H}) = NM\log(\pi e) + M\mathbb{E}\{\log|\gamma\mathbf{H}\mathbf{H}^H + \mathbf{I}_N|\} \tag{5.14}$$

Here;

$$\mathbb{E}\{\log|\gamma\mathbf{H}\mathbf{H}^H + \mathbf{I}_N|\} = \min(N,K)\log\gamma + \theta(N,K), \tag{5.15}$$

$$\theta(N,K) = \max(N,K)\log\max(N,K) - \min(N,K)\log e - |N-K|\log^{\dagger}|N-K| \tag{5.16}$$

For proof see Appendix 5.8.2.

Lemma 3 *If* $\mathbf{Y} = \sqrt{\gamma}\tilde{\mathbf{H}}\mathbf{X} + \mathbf{W}$ *where* $\tilde{\mathbf{H}} = [\mathbf{H}_1^T, \sqrt{\xi}\mathbf{H}_2^T]^T$. *Here,* $\mathbf{H}_1 \in \mathcal{C}^{N_1\times K}$, $\mathbf{H}_2 \in \mathcal{C}^{N_2\times K}$, $\mathbf{X} \in \mathcal{C}^{K\times M}$, $\mathbf{Y}, \mathbf{W} \in \mathcal{C}^{(N_1+N_2)\times M}$, *all entries in* $\mathbf{X}, \mathbf{W}$ *are i.i.d.* $\mathcal{CN}(0,1)$, *and all entries in* $\mathbf{H}_1, \mathbf{H}_2$ *are i.i.d. with zero mean and unit variance. Then for large* N_1, N_2, K, *and* γ*:*

$$h(\mathbf{Y}|\tilde{\mathbf{H}}) = (N_1+N_2)M\log(\pi e) + M\mathbb{E}\{\log|\gamma\tilde{\mathbf{H}}\tilde{\mathbf{H}}^H + \mathbf{I}_{N_1+N_2}|\} \tag{5.17}$$

Here;

$$\mathbb{E}\{\log|\gamma\tilde{\mathbf{H}}\tilde{\mathbf{H}}^H + \mathbf{I}_{N_1+N_2}|\} = \min(N_1+N_2,K)\log\gamma + \mu(N_1,N_2,K,\xi) \tag{5.18}$$

$$\mu(N_1,N_2,K,\xi) = \begin{cases} K\log K + N_2\log\xi \\ -(K-N_1-N_2)\log^{\dagger}(K-N_1-N_2) \\ -(N_1+N_2)\log e; \quad \textit{if } K-N_1-N_2 \geq 0, \\ \\ N_1\log(1+K\tau) + N_2\log(1+\xi K\tau) \\ -K\log(e\tau); \quad \textit{if } K-N_1-N_2 < 0. \end{cases} \tag{5.19}$$

Where τ *is the positive solution to the quadratic equation* $\tau^2 + b\tau + c = 0$ *with* $b = \frac{\xi(K-N_2)+(K-N_1)}{(K-N_1-N_2)\xi K}$ *and* $c = \frac{1}{(K-N_1-N_2)\xi K}$

For proof see Appendix 5.8.3.

Lemma 4 *For* $\mathbf{Y}_A^{(1)}$ *and* $\mathbf{Y}_B^{(1)}$ *defined in section 5.2,*

$$
\begin{aligned}
& I(\mathbf{Y}_A^{(1)}; \mathbf{Y}_B^{(1)}) \\
&= h(\mathbf{Y}_A^{(1)}) - h(\mathbf{Y}_A^{(1)}|\mathbf{Y}_B^{(1)}) = h(\mathbf{Y}_B^{(1)}) - h(\mathbf{Y}_B^{(1)}|\mathbf{Y}_A^{(1)}) \\
&= n_A n_B \log(\frac{(\gamma_{AB}\psi_B + 1)(\gamma_{BA}\psi_A + 1)}{(1-|\rho|^2)\gamma_{AB}\psi_B\gamma_{BA}\psi_A + \gamma_{AB}\psi_B + \gamma_{BA}\psi_A + 1} \\
&= \delta_{|\rho|=1} n_A n_B \log \frac{\psi_A\psi_B\alpha_A\alpha_B}{\alpha_A\lambda_A\psi_B + \alpha_B\lambda_B\psi_A} + \delta_{|\rho|=1} n_A n_B \log P - (1-\delta_{|\rho|=1}) n_A n_B \log(1-|\rho|^2)
\end{aligned}
\tag{5.20}
$$

For proof see Appendix 5.8.4. We will denote the first term in (5.20) as κ, i.e.,

$$
\kappa = \delta_{|\rho|=1} n_A n_B \log \frac{\psi_A\psi_B\alpha_A\alpha_B}{\psi_A\alpha_A\lambda_A + \psi_B\alpha_B\lambda_B} \tag{5.21}
$$

5.5 Analysis

The final expressions of the bounds given in section 5.3 can be found by obtaining the differential entropies $h(\mathcal{Y}|\mathcal{X})$, $h(\mathcal{Y}|\mathcal{Z})$, $h(\mathcal{X}|\mathcal{Z})$ and $h(\mathcal{X}|\mathcal{Y},\mathcal{Z})$. The following discussion provides a detailed analysis

5.5.1 Analysis of $h(\mathcal{Y}|\mathcal{X})$

We can write by applying chain rule:

$$\begin{aligned} h(\mathcal{Y}|\mathcal{X}) &= h(\mathbf{X}_B, \mathbf{Y}_B^{(1)}, \mathbf{Y}_B^{(2)} | \mathbf{X}_A, \mathbf{Y}_A^{(1)}, \mathbf{Y}_A^{(2)}) \\ &= h(\mathbf{X}_B | \mathbf{X}_A, \mathbf{Y}_A^{(1)}, \mathbf{Y}_A^{(2)}) + h(\mathbf{Y}_B^{(1)}, \mathbf{Y}_B^{(2)} | \mathbf{X}_B, \mathbf{X}_A, \mathbf{Y}_A^{(1)}, \mathbf{Y}_A^{(2)}) \end{aligned} \tag{5.22}$$

Analysis of 1st term in (5.22)

Here $\mathbf{X}_A$ is independent of $\{\mathbf{X}_B, \mathbf{Y}_A^{(1)}, \mathbf{Y}_A^{(2)}\}$. For large P and ψ_B, we can replace $\mathbf{Y}_A^{(1)}$ by $\mathbf{H}_{AB}$ using Lemma 1 and write,

$$\begin{aligned} h(\mathbf{X}_B | \mathbf{X}_A, \mathbf{Y}_A^{(1)}, \mathbf{Y}_A^{(2)}) &\approx h(\mathbf{X}_B | \mathbf{H}_{AB}, \mathbf{Y}_A^{(2)}) \\ &= h(\mathbf{X}_B) + h(\mathbf{Y}_A^{(2)} | \mathbf{H}_{AB}, \mathbf{X}_B) - h(\mathbf{Y}_A^{(2)} | \mathbf{H}_{AB}) \end{aligned} \tag{5.23}$$

Analysis of 2nd term in (5.22)

We can write the 2nd term in (5.22) as:

$$\begin{aligned} h(\mathbf{Y}_B^{(1)}, \mathbf{Y}_B^{(2)} | \mathbf{X}_B, \mathbf{X}_A, \mathbf{Y}_A^{(1)}, \mathbf{Y}_A^{(2)}) &= h(\mathbf{Y}_B^{(1)} | \mathbf{X}_B, \mathbf{X}_A, \mathbf{Y}_A^{(1)}, \mathbf{Y}_A^{(2)}) \\ &+ h(\mathbf{Y}_B^{(2)} | \mathbf{Y}_B^{(1)}, \mathbf{X}_B, \mathbf{X}_A, \mathbf{Y}_A^{(1)}, \mathbf{Y}_A^{(2)}) \end{aligned} \tag{5.24}$$

For the second term in (5.24), we can replace $\mathbf{Y}_B^{(1)}$ by $\mathbf{H}_{BA}$ for large P and ψ_A using Lemma 1. Also, given $\{\mathbf{H}_{BA}, \mathbf{X}_A\}$, $\mathbf{Y}_B^{(2)}$ is independent of $\{\mathbf{X}_B, \mathbf{Y}_A^{(1)}, \mathbf{Y}_A^{(2)}\}$. So we can write,

$$h(\mathbf{Y}_B^{(2)} | \mathbf{Y}_B^{(1)}, \mathbf{X}_B, \mathbf{X}_A, \mathbf{Y}_A^{(1)}, \mathbf{Y}_A^{(2)}) \approx h(\mathbf{Y}_B^{(2)} | \mathbf{H}_{BA}, \mathbf{X}_A) \tag{5.25}$$

For the first term in (5.24), $\mathbf{X}_A$ is independent of $\{\mathbf{Y}_B^{(1)}, \mathbf{X}_B, \mathbf{Y}_A^{(1)}, \mathbf{Y}_A^{(2)}\}$. So we can write,

$$h(\mathbf{Y}_B^{(1)} | \mathbf{X}_B, \mathbf{X}_A, \mathbf{Y}_A^{(1)}, \mathbf{Y}_A^{(2)}) = h(\mathbf{Y}_B^{(1)} | \mathbf{X}_B, \mathbf{Y}_A^{(1)}, \mathbf{Y}_A^{(2)}) \tag{5.26}$$

Using chain rule, we can further write:

$$h(\mathbf{Y}_B^{(1)}|\mathbf{X}_B, \mathbf{Y}_A^{(1)}, \mathbf{Y}_A^{(2)}) = h(\mathbf{Y}_A^{(1)}) + h(\mathbf{Y}_B^{(1)}|\mathbf{Y}_A^{(1)}) + h(\mathbf{X}_B, \mathbf{Y}_A^{(2)}|\mathbf{Y}_A^{(1)}, \mathbf{Y}_B^{(1)}) - h(\mathbf{X}_B, \mathbf{Y}_A^{(1)}, \mathbf{Y}_A^{(2)}) \quad (5.27)$$

Here $\mathbf{X}_B$, $\mathbf{Y}_A^{(1)}$ and $\mathbf{Y}_B^{(1)}$ are independent of each other, and we can replace $\mathbf{Y}_A^{(1)}$ by $\mathbf{H}_{AB}$ for large P and ψ_B using Lemma 1. So, we obtain, $h(\mathbf{X}_B, \mathbf{Y}_A^{(1)}, \mathbf{Y}_A^{(2)}) \approx h(\mathbf{X}_B) + h(\mathbf{Y}_A^{(1)}) + h(\mathbf{Y}_A^{(2)}|\mathbf{H}_{AB}, \mathbf{X}_B)$ and $h(\mathbf{X}_B, \mathbf{Y}_A^{(2)}|\mathbf{Y}_A^{(1)}, \mathbf{Y}_B^{(1)}) \approx h(\mathbf{X}_B) + h(\mathbf{Y}_A^{(2)}|\mathbf{H}_{AB}, \mathbf{X}_B)$. Using the above discussion on (5.27) we obtain:

$$h(\mathbf{Y}_B^{(1)}|\mathbf{X}_B, \mathbf{Y}_A^{(1)}, \mathbf{Y}_A^{(2)}) \approx h(\mathbf{Y}_B^{(1)}|\mathbf{Y}_A^{(1)}) \quad (5.28)$$

Summary

Using (5.28) and (5.25) in (5.24), and then using (5.23) and (5.24) in (5.22) we obtain:

$$h(\mathcal{Y}|\mathcal{X}) \approx h(\mathbf{X}_B) + h(\mathbf{Y}_A^{(2)}|\mathbf{H}_{AB}, \mathbf{X}_B) - h(\mathbf{Y}_A^{(2)}|\mathbf{H}_{AB}) + h(\mathbf{Y}_B^{(1)}|\mathbf{Y}_A^{(1)}) + h(\mathbf{Y}_B^{(2)}|\mathbf{H}_{BA}, \mathbf{X}_A) \quad (5.29)$$

5.5.2 Analysis of $h(\mathcal{Y}|\mathcal{Z})$

We can write

$$\begin{aligned} h(\mathcal{Y}|\mathcal{Z}) &= h(\mathbf{X}_B, \mathbf{Y}_B^{(1)}, \mathbf{Y}_B^{(2)}|\mathbf{Y}_{EA}^{(1)}, \mathbf{Y}_{EA}^{(2)}, \mathbf{Y}_{EB}^{(1)}, \mathbf{Y}_{EB}^{(2)}) \\ &= h(\mathbf{X}_B|\mathbf{Y}_{EA}^{(1)}, \mathbf{Y}_{EA}^{(2)}, \mathbf{Y}_{EB}^{(1)}, \mathbf{Y}_{EB}^{(2)}) \\ &\quad + h(\mathbf{Y}_B^{(1)}, \mathbf{Y}_B^{(2)}|\mathbf{X}_B, \mathbf{Y}_{EA}^{(1)}, \mathbf{Y}_{EA}^{(2)}, \mathbf{Y}_{EB}^{(1)}, \mathbf{Y}_{EB}^{(2)}) \end{aligned} \quad (5.30)$$

Analysis of 1st term in (5.30)

Here $\{\mathbf{X}_B, \mathbf{Y}_{EB}^{(2)}\}$ is independent of $\{\mathbf{Y}_{EA}^{(1)}, \mathbf{Y}_{EA}^{(2)}\}$. For a large P and ψ_B, we can use Lemma 1 to replace $\mathbf{Y}_{EB}^{(1)}$ by $\mathbf{G}_B$:

$$h(\mathbf{X}_B|\mathbf{Y}_{EA}^{(1)},\mathbf{Y}_{EA}^{(2)},\mathbf{Y}_{EB}^{(1)},\mathbf{Y}_{EB}^{(2)}) \approx h(\mathbf{X}_B|\mathbf{G}_B,\mathbf{Y}_{EB}^{(2)}) \tag{5.31}$$

Similar to $h(\mathbf{X}_B|\mathbf{H}_{AB}, \mathbf{Y}_A^{(2)})$ in (5.23), we can write:

$$h(\mathbf{X}_B|\mathbf{G}_B,\mathbf{Y}_{EB}^{(2)}) = h(\mathbf{X}_B) + h(\mathbf{Y}_{EB}^{(2)}|\mathbf{G}_B,\mathbf{X}_B) - h(\mathbf{Y}_{EB}^{(2)}|\mathbf{G}_B) \tag{5.32}$$

Analysis of 2nd term in (5.30)

Here, $\{\mathbf{X}_B,\mathbf{Y}_{EB}^{(1)},\mathbf{Y}_{EB}^{(2)}\}$ is independent of $\{\mathbf{Y}_B^{(1)}, \mathbf{Y}_B^{(2)}, \mathbf{Y}_{EA}^{(1)}, \mathbf{Y}_{EA}^{(2)}\}$. We can write

$$\begin{aligned}
&h(\mathbf{Y}_B^{(1)}, \mathbf{Y}_B^{(2)}|\mathbf{X}_B, \mathbf{Y}_{EA}^{(1)}, \mathbf{Y}_{EA}^{(2)}, \mathbf{Y}_{EB}^{(1)}, \mathbf{Y}_{EB}^{(2)}) \\
&= h(\mathbf{Y}_B^{(1)}, \mathbf{Y}_B^{(2)}|\mathbf{Y}_{EA}^{(1)}, \mathbf{Y}_{EA}^{(2)}) = h(\mathbf{Y}_B^{(1)}|\mathbf{Y}_{EA}^{(1)}, \mathbf{Y}_{EA}^{(2)}) + h(\mathbf{Y}_B^{(2)}|\mathbf{Y}_B^{(1)}, \mathbf{Y}_{EA}^{(1)}, \mathbf{Y}_{EA}^{(2)})
\end{aligned} \tag{5.33}$$

For the first term in (5.33), $\mathbf{Y}_B^{(1)}$ is independent of $\{\mathbf{Y}_{EA}^{(1)}, \mathbf{Y}_{EA}^{(2)}\}$ conditioned on the public $\mathbf{\Pi}_A$. So, we can write:

$$h(\mathbf{Y}_B^{(1)}|\mathbf{Y}_{EA}^{(1)}, \mathbf{Y}_{EA}^{(2)}) = h(\mathbf{Y}_B^{(1)}) \tag{5.34}$$

Now for the second term in (5.33), we can replace $\mathbf{Y}_B^{(1)}$ and $\mathbf{Y}_{EA}^{(1)}$ by $\mathbf{H}_{BA}$ and $\mathbf{G}_A$ using Lemma 1 for large P and ψ_A. Then using chain rule, we have

$$\begin{aligned}
&h(\mathbf{Y}_B^{(2)}|\mathbf{Y}_B^{(1)}, \mathbf{Y}_{EA}^{(1)}, \mathbf{Y}_{EA}^{(2)}) \\
&\approx h(\mathbf{Y}_{EA}^{(2)}, \mathbf{Y}_B^{(2)}|\mathbf{G}_A, \mathbf{H}_{BA}) - h(\mathbf{Y}_{EA}^{(2)}|\mathbf{G}_A, \mathbf{H}_{BA}) \\
&= h(\mathbf{Y}_{EA}^{(2)}, \mathbf{Y}_B^{(2)}|\mathbf{G}_A, \mathbf{H}_{BA}) - h(\mathbf{Y}_{EA}^{(2)}|\mathbf{G}_A)
\end{aligned} \tag{5.35}$$

Here we have used the fact that $\mathbf{H}_{BA}$ is independent of $\{\mathbf{Y}_{EA}^{(2)}, \mathbf{G}_A\}$.

Summary

Using (5.34) and (5.35) in (5.33) and then using (5.31) and (5.33) in (5.30) we can write:

$$h(\mathcal{Y}|\mathcal{Z}) \approx h(\mathbf{X}_B) + h(\mathbf{Y}_{EB}^{(2)}|\mathbf{G}_B, \mathbf{X}_B) - h(\mathbf{Y}_{EB}^{(2)}|\mathbf{G}_B) + h(\mathbf{Y}_B^{(1)}) + h(\mathbf{Y}_{EA}^{(2)}, \mathbf{Y}_B^{(2)}|\mathbf{G}_A, \mathbf{H}_{BA}) - h(\mathbf{Y}_{EA}^{(2)}|\mathbf{G}_A) \tag{5.36}$$

5.5.3 Analysis of $h(\mathcal{X}|\mathcal{Z})$

We can write using chain rule

$$\begin{aligned} h(\mathcal{X}|\mathcal{Z}) &= h(\mathbf{X}_A, \mathbf{Y}_A^{(1)}, \mathbf{Y}_A^{(2)}|\mathbf{Y}_{EA}^{(1)}, \mathbf{Y}_{EA}^{(2)}, \mathbf{Y}_{EB}^{(1)}, \mathbf{Y}_{EB}^{(2)}) \\ &= h(\mathbf{X}_A|\mathbf{Y}_{EA}^{(1)}, \mathbf{Y}_{EA}^{(2)}, \mathbf{Y}_{EB}^{(1)}, \mathbf{Y}_{EB}^{(2)}) \\ &\quad + h(\mathbf{Y}_A^{(1)}, \mathbf{Y}_A^{(2)}|\mathbf{X}_A, \mathbf{Y}_{EA}^{(1)}, \mathbf{Y}_{EA}^{(2)}, \mathbf{Y}_{EB}^{(1)}, \mathbf{Y}_{EB}^{(2)}) \end{aligned} \tag{5.37}$$

Analysis of 1st term in (5.37)

Here $\{\mathbf{Y}_{EB}^{(1)}, \mathbf{Y}_{EB}^{(2)}\}$ is independent of $\{\mathbf{X}_A, \mathbf{Y}_{EA}^{(1)}, \mathbf{Y}_{EA}^{(2)}\}$. Also, for a large P and ψ_A, we can use Lemma 1 to replace $\mathbf{Y}_{EA}^{(1)}$ by $\mathbf{G}_A$:

$$h(\mathbf{X}_A|\mathbf{Y}_{EA}^{(1)}, \mathbf{Y}_{EA}^{(2)}, \mathbf{Y}_{EB}^{(1)}, \mathbf{Y}_{EB}^{(2)}) \approx h(\mathbf{X}_A|\mathbf{G}_A, \mathbf{Y}_{EA}^{(2)}) \tag{5.38}$$

Similar to $h(\mathbf{X}_B|\mathbf{H}_{AB}, \mathbf{Y}_A^{(2)})$ in (5.23), we can write:

$$h(\mathbf{X}_A|\mathbf{G}_A, \mathbf{Y}_{EA}^{(2)}) = h(\mathbf{X}_A) + h(\mathbf{Y}_{EA}^{(2)}|\mathbf{G}_A, \mathbf{X}_A) - h(\mathbf{Y}_{EA}^{(2)}|\mathbf{G}_A) \tag{5.39}$$

Analysis of 2nd term in (5.37)

Here, $\{\mathbf{X}_A, \mathbf{Y}_{EA}^{(1)}, \mathbf{Y}_{EA}^{(2)}\}$ is independent of $\{\mathbf{Y}_A^{(1)}, \mathbf{Y}_A^{(2)}, \mathbf{Y}_{EB}^{(1)}, \mathbf{Y}_{EB}^{(2)}\}$. We can write

$$\begin{aligned} & h(\mathbf{Y}_A^{(1)}, \mathbf{Y}_A^{(2)} | \mathbf{X}_A, \mathbf{Y}_{EA}^{(1)}, \mathbf{Y}_{EA}^{(2)}, \mathbf{Y}_{EB}^{(1)}, \mathbf{Y}_{EB}^{(2)}) \\ &= h(\mathbf{Y}_A^{(1)}, \mathbf{Y}_A^{(2)} | \mathbf{Y}_{EB}^{(1)}, \mathbf{Y}_{EB}^{(2)}) \\ &= h(\mathbf{Y}_A^{(1)} | \mathbf{Y}_{EB}^{(1)}, \mathbf{Y}_{EB}^{(2)}) + h(\mathbf{Y}_A^{(2)} | \mathbf{Y}_A^{(1)}, \mathbf{Y}_{EB}^{(1)}, \mathbf{Y}_{EB}^{(2)}) \end{aligned} \tag{5.40}$$

For the first term in (5.40), $\mathbf{Y}_A^{(1)}$ is independent of $\{\mathbf{Y}_{EB}^{(1)}, \mathbf{Y}_{EB}^{(2)}\}$ conditioned on the public $\mathbf{\Pi}_B$. So, we can write:

$$h(\mathbf{Y}_A^{(1)} | \mathbf{Y}_{EB}^{(1)}, \mathbf{Y}_{EB}^{(2)}) = h(\mathbf{Y}_A^{(1)}) \tag{5.41}$$

Now for the 2nd term in (5.40), we can replace $\mathbf{Y}_A^{(1)}$ and $\mathbf{Y}_{EB}^{(1)}$ by $\mathbf{H}_{AB}$ and $\mathbf{G}_B$ for large P and ψ_B using Lemma 1. Then using the chain rule, we have

$$\begin{aligned} & h(\mathbf{Y}_A^{(2)} | \mathbf{Y}_A^{(1)}, \mathbf{Y}_{EB}^{(1)}, \mathbf{Y}_{EB}^{(2)}) \\ &\approx h(\mathbf{Y}_{EB}^{(2)}, \mathbf{Y}_A^{(2)} | \mathbf{G}_B, \mathbf{H}_{AB}) - h(\mathbf{Y}_{EB}^{(2)} | \mathbf{G}_B, \mathbf{H}_{AB}) \\ &= h(\mathbf{Y}_{EB}^{(2)}, \mathbf{Y}_A^{(2)} | \mathbf{G}_B, \mathbf{H}_{AB}) - h(\mathbf{Y}_{EB}^{(2)} | \mathbf{G}_B) \end{aligned} \tag{5.42}$$

Here we have used the fact that $\mathbf{H}_{AB}$ is independent of $\{\mathbf{Y}_{EB}^{(2)}, \mathbf{G}_B\}$.

Summary

Using (5.41) and (5.42) in (5.40) and then using (5.38) and (5.40) in (5.37) we can write:

$$\begin{aligned} h(\mathcal{Y}|\mathcal{Z}) \approx & h(\mathbf{X}_A) + h(\mathbf{Y}_{EA}^{(2)} | \mathbf{G}_A, \mathbf{X}_A) - h(\mathbf{Y}_{EA}^{(2)} | \mathbf{G}_A) + h(\mathbf{Y}_A^{(1)}) \\ & + h(\mathbf{Y}_{EB}^{(2)}, \mathbf{Y}_A^{(2)} | \mathbf{G}_B, \mathbf{H}_{AB}) - h(\mathbf{Y}_{EB}^{(2)} | \mathbf{G}_B) \end{aligned} \tag{5.43}$$

5.5.4 Analysis of $h(\mathcal{X}|\mathcal{Y},\mathcal{Z})$

We can write:

$$\begin{aligned}
&h(\mathcal{X}|\mathcal{Y},\mathcal{Z}) \\
&= h(\mathbf{X}_A,\mathbf{Y}_A^{(1)},\mathbf{Y}_A^{(2)}|\mathbf{X}_B,\mathbf{Y}_B^{(1)},\mathbf{Y}_B^{(2)},\mathbf{Y}_{EA}^{(1)},\mathbf{Y}_{EA}^{(2)},\mathbf{Y}_{EB}^{(1)},\mathbf{Y}_{EB}^{(2)}) \\
&= h(\mathbf{X}_A|\mathbf{X}_B,\mathbf{Y}_B^{(1)},\mathbf{Y}_B^{(2)},\mathbf{Y}_{EA}^{(1)},\mathbf{Y}_{EA}^{(2)},\mathbf{Y}_{EB}^{(1)},\mathbf{Y}_{EB}^{(2)}) \\
&+ h(\mathbf{Y}_A^{(1)},\mathbf{Y}_A^{(2)}|\mathbf{X}_A,\mathbf{X}_B,\mathbf{Y}_B^{(1)},\mathbf{Y}_B^{(2)},\mathbf{Y}_{EA}^{(1)},\mathbf{Y}_{EA}^{(2)},\mathbf{Y}_{EB}^{(1)},\mathbf{Y}_{EB}^{(2)}) \\
&= h(\mathbf{X}_A|\mathbf{Y}_B^{(1)},\mathbf{Y}_B^{(2)},\mathbf{Y}_{EA}^{(1)},\mathbf{Y}_{EA}^{(2)}) \\
&+ h(\mathbf{Y}_A^{(1)},\mathbf{Y}_A^{(2)}|\mathbf{X}_A,\mathbf{X}_B,\mathbf{Y}_B^{(1)},\mathbf{Y}_B^{(2)}) \qquad (5.44)
\end{aligned}$$

Where we have used the fact that $\{\mathbf{X}_B,\mathbf{Y}_{EB}^{(1)},\mathbf{Y}_{EB}^{(2)}\}$ is independent of $\{\mathbf{X}_A,\mathbf{Y}_B^{(1)},\mathbf{Y}_B^{(2)},\mathbf{Y}_{EA}^{(1)},\mathbf{Y}_{EA}^{(2)}\}$ and given $\{\mathbf{X}_A,\mathbf{X}_B\}$, $\{\mathbf{Y}_{EA}^{(1)},\mathbf{Y}_{EA}^{(2)},\mathbf{Y}_{EB}^{(1)},\mathbf{Y}_{EB}^{(2)}\}$ is independent of $\{\mathbf{Y}_A^{(1)},\mathbf{Y}_A^{(2)},\mathbf{Y}_B^{(1)},\mathbf{Y}_B^{(2)}\}$.

Analysis of the first term in (5.44)

We can replace $\mathbf{Y}_B^{(1)}$ by $\mathbf{H}_{BA}$ and $\mathbf{Y}_{EA}^{(1)}$ by $\mathbf{G}_A$ for large P and ψ_A using Lemma 1 and write:

$$\begin{aligned}
h(\mathbf{X}_A|\mathbf{Y}_B^{(1)},\mathbf{Y}_B^{(2)},\mathbf{Y}_{EA}^{(1)},\mathbf{Y}_{EA}^{(2)}) &\approx h(\mathbf{X}_A|\mathbf{H}_{BA},\mathbf{G}_A,\mathbf{Y}_B^{(2)},\mathbf{Y}_{EA}^{(2)}) \\
&= h(\mathbf{X}_A) + h(\mathbf{Y}_{EA}^{(2)},\mathbf{Y}_B^{(2)}|\mathbf{G}_A,\mathbf{H}_{BA},\mathbf{X}_A) - h(\mathbf{Y}_{EA}^{(2)},\mathbf{Y}_B^{(2)}|\mathbf{G}_A,\mathbf{H}_{BA}) \qquad (5.45)
\end{aligned}$$

Analysis of the second term in (5.44)

Using similar analysis for (5.24) in 5.5.1, we can write:

$$\begin{aligned} & h(\mathbf{Y}_A^{(1)}, \mathbf{Y}_A^{(2)} | \mathbf{X}_A, \mathbf{X}_B, \mathbf{Y}_B^{(1)}, \mathbf{Y}_B^{(2)}) \\ & \approx h(\mathbf{Y}_A^{(1)} | \mathbf{Y}_B^{(1)}) + h(\mathbf{Y}_A^{(2)} | \mathbf{H}_{AB}, \mathbf{X}_B) \end{aligned} \tag{5.46}$$

Summary

Using (5.45) and (5.46) in (5.44), we can finally write:

$$\begin{aligned} h(\mathcal{X}|\mathcal{Y}, \mathcal{Z}) \approx & \, h(\mathbf{Y}_{EA}^{(2)}, \mathbf{Y}_B^{(2)} | \mathbf{G}_A, \mathbf{H}_{BA}, \mathbf{X}_A) + h(\mathbf{Y}_A^{(1)} | \mathbf{H}_{BA}) \\ & + h(\mathbf{X}_A) - h(\mathbf{Y}_{EA}^{(2)}, \mathbf{Y}_B^{(2)} | \mathbf{G}_A, \mathbf{H}_{BA}) + h(\mathbf{Y}_A^{(2)} | \mathbf{H}_{AB}, \mathbf{X}_B) \end{aligned} \tag{5.47}$$

5.5.5 Analysis of C_B and C_Z

From the quantities obtained from sections 5.5.1 to 5.5.4, we arrive at the following expressions for C_B and C_Z:

$$\begin{aligned} C_B &= h(\mathcal{Y}|\mathcal{Z}) - h(\mathcal{Y}|\mathcal{X}) \\ &\approx h(\mathbf{Y}_B^{(1)}) - h(\mathbf{Y}_B^{(1)} | \mathbf{Y}_A^{(1)}) \\ &+ h(\mathbf{Y}_{EB}^{(2)} | \mathbf{G}_B, \mathbf{X}_B) - h(\mathbf{Y}_{EB}^{(2)} | \mathbf{G}_B) \\ &+ h(\mathbf{Y}_A^{(2)} | \mathbf{H}_{AB}) - h(\mathbf{Y}_A^{(2)} | \mathbf{H}_{AB}, \mathbf{X}_B) \\ &+ h(\mathbf{Y}_{EA}^{(2)}, \mathbf{Y}_B^{(2)} | \mathbf{G}_A, \mathbf{H}_{BA}) - h(\mathbf{Y}_{EA}^{(2)} | \mathbf{G}_A) \\ &- h(\mathbf{Y}_B^{(2)} | \mathbf{H}_{BA}, \mathbf{X}_A), \end{aligned} \tag{5.48}$$

$$
\begin{aligned}
C_Z &= h(\mathcal{X}|\mathcal{Z}) - h(\mathcal{X}|\mathcal{Y},\mathcal{Z}) \\
&\approx h(\mathbf{Y}_A^{(1)}) - h(\mathbf{Y}_A^{(1)}|\mathbf{Y}_B^{(1)}) \\
&+ h(\mathbf{Y}_{EA}^{(2)}|\mathbf{G}_A,\mathbf{X}_A) - h(\mathbf{Y}_{EA}^{(2)}|\mathbf{G}_A) \\
&+ h(\mathbf{Y}_{EA}^{(2)},\mathbf{Y}_B^{(2)}|\mathbf{G}_A,\mathbf{H}_{BA}) - h(\mathbf{Y}_{EA}^{(2)},\mathbf{Y}_B^{(2)}|\mathbf{G}_A,\mathbf{H}_{BA},\mathbf{X}_A) \\
&+ h(\mathbf{Y}_{EB}^{(2)},\mathbf{Y}_A^{(2)}|\mathbf{G}_B,\mathbf{H}_{AB}) - h(\mathbf{Y}_{EB}^{(2)}|\mathbf{G}_B) \\
&- h(\mathbf{Y}_A^{(2)}|\mathbf{H}_{AB},\mathbf{X}_B) \qquad (5.49)
\end{aligned}
$$

Now, for the first two terms in (5.48) and (5.49), we can refer to (5.20), i.e., they are the mutual information between $\mathbf{Y}_A^{(1)}$ and $\mathbf{Y}_B^{(1)}$. According to (5.1d), $vec(\mathbf{Y}_{EB}^{(2)}|\mathbf{G}_B,\mathbf{X}_B) \sim \mathcal{CN}(*,\mathbf{I}_{n_E v_B})$, where $*$ denotes a quantity of no importance. So, $h(\mathbf{Y}_{EB}^{(2)}|\mathbf{G}_B,\mathbf{X}_B) = n_E v_B \log(\pi e)$. Similar argument can be applied to obtain the following,

$$h(\mathbf{Y}_A^{(2)}|\mathbf{H}_{AB},\mathbf{X}_B) = n_A v_B \log(\pi e) \qquad (5.50a)$$

$$h(\mathbf{Y}_B^{(2)}|\mathbf{H}_{BA},\mathbf{X}_A) = n_B v_A \log(\pi e) \qquad (5.50b)$$

$$h(\mathbf{Y}_{EA}^{(2)}|\mathbf{G}_A,\mathbf{X}_A) = n_E v_A \log(\pi e) \qquad (5.50c)$$

$$h(\mathbf{Y}_{EA}^{(2)},\mathbf{Y}_B^{(2)}|\mathbf{G}_A,\mathbf{H}_{BA},\mathbf{X}_A) = (n_B+n_E) v_A \log(\pi e) \qquad (5.50d)$$

Also, according to (5.1d), we can apply (5.14) from Lemma 2 and obtain $h(\mathbf{Y}_{EB}^{(2)}|\mathbf{G}_B) = n_E v_B \log(\pi e) + v_B \mathbb{E}\{\log|\gamma_{BE}\mathbf{G}_B\mathbf{G}_B^H + \mathbf{I}_{n_E}|\}$. Similar argument can be applied to obtain:

$$
\begin{aligned}
h(\mathbf{Y}_A^{(2)}|\mathbf{H}_{AB}) &= n_A v_B \log(\pi e) \\
&\quad + v_B \mathbb{E}\{\log|\gamma_{BA}\mathbf{H}_{AB}\mathbf{H}_{AB}^H + \mathbf{I}_{n_A}|\} \qquad (5.51a) \\
h(\mathbf{Y}_{EA}^{(2)}|\mathbf{G}_A) &= n_E v_A \log(\pi e) \\
&\quad + v_A \mathbb{E}\{\log|\gamma_{AE}\mathbf{G}_A\mathbf{G}_A^H + \mathbf{I}_{n_E}|\} \qquad (5.51b)
\end{aligned}
$$

For $h(\mathbf{Y}_{EA}^{(2)}, \mathbf{Y}_{B}^{(2)}|\mathbf{G}_A, \mathbf{H}_{BA})$, we can rewrite (5.1b) and (5.1c) as follows,

$$\begin{bmatrix} \mathbf{Y}_{B}^{(2)} \\ \mathbf{Y}_{EA}^{(2)} \end{bmatrix} = \begin{bmatrix} \sqrt{\gamma_{AB}}\mathbf{H}_{BA} \\ \sqrt{\gamma_{AE}}\mathbf{G}_A \end{bmatrix} \mathbf{X}_A + \begin{bmatrix} \mathbf{W}_{B}^{(2)} \\ \mathbf{W}_{EA}^{(2)} \end{bmatrix} \tag{5.52}$$

$$\mathbf{Y}_1 = \sqrt{\gamma_{AB}}\tilde{\mathbf{H}}_1\mathbf{X}_A + \mathbf{W}_1 \tag{5.53}$$

Where $\tilde{\mathbf{H}}_1 = [\mathbf{H}_{BA}^T, \sqrt{\lambda_B/\lambda_{EA}}\mathbf{G}_A^T]^T$. Then we can apply (5.17) from Lemma 3 on (5.53) and obtain:

$$\begin{aligned} h(\mathbf{Y}_{EA}^{(2)}, \mathbf{Y}_{B}^{(2)}|\mathbf{G}_A, \mathbf{H}_{BA}) &= (n_B + n_E)v_A \log(\pi e) \\ &\quad + v_A\mathbb{E}\{\log|\gamma_{AB}\tilde{\mathbf{H}}_1\tilde{\mathbf{H}}_1^H + \mathbf{I}_{n_B+n_E}|\} \end{aligned} \tag{5.54}$$

Similar argument can be applied for $h(\mathbf{Y}_{EB}^{(2)}, \mathbf{Y}_{A}^{(2)}|\mathbf{G}_B, \mathbf{H}_{AB})$ with:

$$\tilde{\mathbf{H}}_2 = [\mathbf{H}_{AB}^T, \sqrt{\lambda_A/\lambda_{EB}}, \mathbf{G}_B^T]^T \tag{5.55}$$

and obtain the following,

$$\begin{aligned} h(\mathbf{Y}_{EB}^{(2)}, \mathbf{Y}_{A}^{(2)}|\mathbf{G}_B, \mathbf{H}_{AB}) &= (n_A + n_E)v_B \log(\pi e) \\ &\quad + v_B\mathbb{E}\{\log|\gamma_{BA}\tilde{\mathbf{H}}_2\tilde{\mathbf{H}}_2^H + \mathbf{I}_{n_A+n_E}|\} \end{aligned} \tag{5.56}$$

Using the above discussion on (5.48) and (5.49), we obtain the following:

$$\begin{aligned} C_B \approx \kappa &+ \delta_{|\rho|=1} n_A n_B \log P - (\delta_{1-|\rho|=1}) n_A n_B \log(1-|\rho|^2) \\ &- v_B\mathbb{E}\{\log|\gamma_{BE}\mathbf{G}_B\mathbf{G}_B^H + \mathbf{I}_{n_E}|\} \\ &+ v_B\mathbb{E}\{\log|\gamma_{BA}\mathbf{H}_{AB}\mathbf{H}_{AB}^H + \mathbf{I}_{n_A}|\} \\ &+ v_A\mathbb{E}\{\log|\gamma_{AB}\tilde{\mathbf{H}}_1\tilde{\mathbf{H}}_1^H + \mathbf{I}_{n_B+n_E}|\} \\ &- v_A\mathbb{E}\{\log|\gamma_{AE}\mathbf{G}_A\mathbf{G}_A^H + \mathbf{I}_{n_E}|\}, \end{aligned} \tag{5.57}$$

$$
\begin{aligned}
C_Z \approx \kappa &+ \delta_{|\rho|=1} n_A n_B \log P - (1-\delta_{|\rho|=1}) n_A n_B \log(1-|\rho|^2) \\
&- v_A \mathbb{E}\{\log|\gamma_{AE}\mathbf{G}_A\mathbf{G}_A^H + \mathbf{I}_{n_E}|\} \\
&+ v_A \mathbb{E}\{\log|\gamma_{AB}\tilde{\mathbf{H}}_1\tilde{\mathbf{H}}_1^H + \mathbf{I}_{n_B+n_E}|\} \\
&+ v_B \mathbb{E}\{\log|\gamma_{BA}\tilde{\mathbf{H}}_2\tilde{\mathbf{H}}_2^H + \mathbf{I}_{n_A+n_E}|\} \\
&- v_B \mathbb{E}\{\log|\gamma_{BE}\mathbf{G}_B\mathbf{G}_B^H + \mathbf{I}_{n_E}|\}. \qquad (5.58)
\end{aligned}
$$

Now, we can again apply Lemma 2 and Lemma 3 on the terms involving expectations ($\mathbb{E}\{\cdot\}$) in (5.57) and (5.58). For terms involving the matrices $\tilde{\mathbf{H}}_1\tilde{\mathbf{H}}_1^H$ and $\tilde{\mathbf{H}}_2\tilde{\mathbf{H}}_2^H$, we apply (5.17) from Lemma 3, for all other matrices we apply (5.15) from Lemma 2. Finally, we obtain the expressions given in (5.4) and (5.5).

5.6 Simulation Results

In this section, we provide some empirical values of the bounds and compare them with closed form expressions where random matrix theory and large power approximation were used. The empirical values of the bounds can be obtained using (5.57) and (5.58) where we can apply Monte-Carlo simulation to obtain the empirical values of the expectation terms. The closed form expressions are given by (5.3), (5.4) and (5.5).

For each given set of the parameters $\{n_A, n_B, n_E, v_A, v_B, \alpha_A, \alpha_B, \lambda_A, \lambda_B, \lambda_{EA}, \lambda_{EB}, \rho\}$, 2000 Gaussian realizations of the matrices $\mathbf{H}_{AB}, \mathbf{H}_{BA}, \mathbf{G}_A, \mathbf{G}_B$ were generated and then empirical C_B and C_Z for different power P was obtained from (5.57) and (5.58). For same set, closed form for C_B and C_Z were obtained using (5.3), (5.4) and (5.5). In Fig. 5.6, the simulation results of C_B are compared with the closed-form results of C_B shown in

(5.3), (5.4) and (5.5). We see a very good agreement between the simulation results and the analytical results at high power even for moderate numbers of antennas.

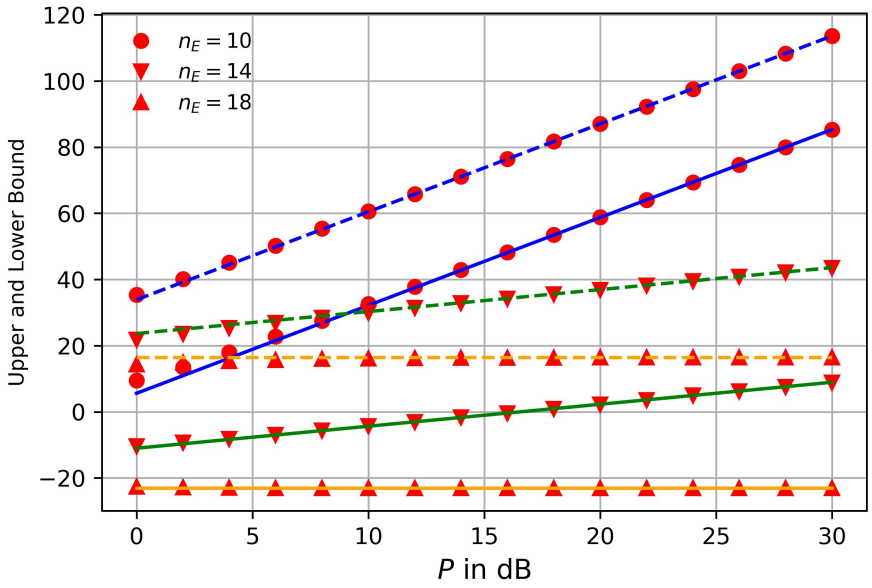


Figure 5.6: Simulation results of (5.57) and (5.58) (markers) versus the closed-form results in (5.3), (5.4) and (5.5) (solid lines for lower-bound and dashed lines for upper-bounds). Parameters: $n_A = 16, n_B = 12, v_A = 1, v_B = 1, \alpha_A = 1.25, \alpha_B = 1.75, \alpha_{EA} = 0.5, \alpha_{EB} = 0.25, \rho = 0.5$

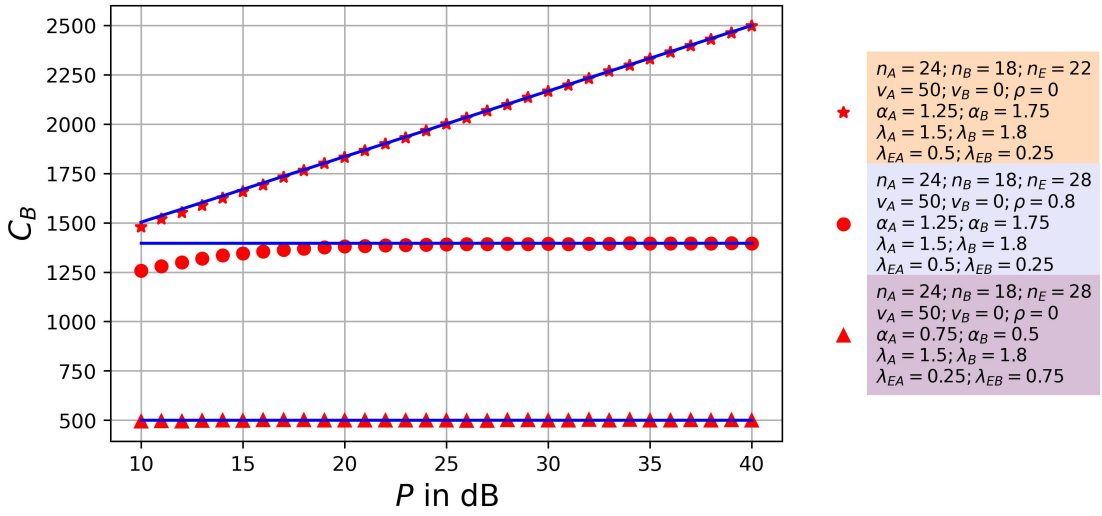


Figure 5.7: Simulation results of (5.57) (markers) versus the closed-form results in (5.3), (5.4) and (5.5) (solid lines) for different parameters

5.7 Conclusion

We have analyzed a MAC upper bound C_Z and lower bound C_B on SKC from MIMO-Hybrid-Probing. Assuming large signal powers from Alice and Bob, and large numbers of antennas on Alice, Bob and Eve, we derived both the `FoT` and `SoT` of C_B and C_Z. The analytical results are also validated by simulations. The `FoT` is the same as in the prior work [33], which becomes zero if neither Alice nor Bob has more antennas than Eve and $|\rho| < 1$. But the `SoT` is entirely novel, which shows in particular that even if Eve has many more antennas than Alice and Bob, the SKC in bits per channel coherence period increases linearly with the number of random transmissions from one node to another in each coherent period. See Proposition 1.

5.8 Appendix

5.8.1 Proof of Lemma 1

We have $\mathbf{Y} = \sqrt{\gamma}\mathbf{H}\mathbf{\Pi} + \mathbf{W}$ or $\mathbf{y} = \sqrt{\gamma}(\mathbf{\Pi}^T \otimes \mathbf{I}_N)\mathbf{h} + \mathbf{w}$. The MSE of MMSE estimation $\widehat{\mathbf{h}}$ of $\mathbf{h}$ can be written as:

$$\begin{aligned}\mathbf{C}_{\mathbf{h}|\mathbf{y},\mathbf{\Pi}} &= \mathbf{I}_{NK} - \gamma(\mathbf{\Pi}^T \otimes \mathbf{I}_N)^H[\gamma(\mathbf{\Pi}^T \otimes \mathbf{I}_N)(\mathbf{\Pi}^T \otimes \mathbf{I}_N)^H + \mathbf{I}_{NK}]^{-1}(\mathbf{\Pi}^T \otimes \mathbf{I}_N) \\ &= \mathbf{I}_{NK} - \gamma(\mathbf{\Pi}^* \otimes \mathbf{I}_N)[(\psi\gamma + 1)\mathbf{I}_{NK}]^{-1}(\mathbf{\Pi}^T \otimes \mathbf{I}_N) \\ &= \frac{1}{\psi\gamma+1}\mathbf{I}_{NK} \end{aligned} \tag{5.59}$$

So we can write $\mathbf{H} = \widehat{\mathbf{H}} + \Delta\mathbf{H}$ where $vec(\Delta\mathbf{H}) \sim \mathcal{CN}(0, \frac{1}{1+\psi\gamma}\mathbf{I}_{NK})$. For very large ψ, γ, $\Delta\mathbf{H}$ is negligible and we can make the following approximation:

$$\begin{aligned}\mathbf{Y}' &= \mathbf{H}\mathbf{X} + \mathbf{W}'; \text{ entries of } \mathbf{W}' \text{ are i.i.d. } \mathcal{CN}(0,1) \\ &= \widehat{\mathbf{H}}\mathbf{X} + \Delta\mathbf{H}\mathbf{X} + \mathbf{W}' \approx \widehat{\mathbf{H}}\mathbf{X} + \mathbf{W}' \end{aligned} \tag{5.60}$$

Instead of transmitting pilots with much higher power than the random symbols, the same pilots can be repeated multiple times and the same result can be achieved. For that, we now assume $\mathbf{\Pi}^H\mathbf{\Pi} = \mathbf{I}_K$ and the pilot $\mathbf{\Pi}$ is transmitted ψ times (ψ being an integer), i.e., let $\mathbf{P} = [\mathbf{\Pi}, \cdots, \mathbf{\Pi}] = \mathbf{e}_\psi^T \otimes \mathbf{\Pi}$ and $\tilde{\mathbf{Y}} = \sqrt{\gamma}\mathbf{H}\mathbf{P} + \tilde{\mathbf{W}}$ where $\mathbf{e}_\psi$ is a vector containing all 1. We can write $\tilde{\mathbf{y}} = \sqrt{\gamma}(\mathbf{P}^T \otimes \mathbf{I}_N)\mathbf{h} + \tilde{\mathbf{w}}$ and for MMSE estimation, we can write,

$$\mathbf{C}_{\mathbf{h}|\tilde{\mathbf{y}},\mathbf{P}} = \mathbf{I}_{NK} - \gamma(\mathbf{P}^T \otimes \mathbf{I}_N)^H[\gamma(\mathbf{P}^T \otimes \mathbf{I}_N)(\mathbf{P}^T \otimes \mathbf{I}_N)^H + \mathbf{I}_{\psi NK}]^{-1}(\mathbf{P}^T \otimes \mathbf{I}_N) \tag{5.61}$$

The matrix inverse part can be written as

$$[\gamma(\mathbf{P}^T \otimes \mathbf{I}_N)(\mathbf{P}^T \otimes \mathbf{I}_N)^H + \mathbf{I}_{\psi NK}]^{-1} = (\mathbf{I}_\psi - \frac{\gamma}{1+\psi\gamma}\mathbf{e}_\psi\mathbf{e}_\psi^T) \otimes \mathbf{I}_{NK} \tag{5.62}$$

Here we have used the fact that $\mathbf{P}^H\mathbf{P} = \mathbf{e}_\psi\mathbf{e}_\psi^T \otimes \mathbf{I}_K$ and $\mathbf{P}\mathbf{P}^H = \psi\mathbf{I}_K$. Then we can write:

$$\begin{aligned}\mathbf{C}_{\mathbf{h}|\tilde{\mathbf{y}},\mathbf{P}} &= \mathbf{I}_{NK} - \gamma(\mathbf{P}^T \otimes \mathbf{I}_N)^H[\{(\mathbf{I}_\psi - \frac{\gamma}{1+\psi\gamma}\mathbf{e}_\psi\mathbf{e}_\psi^T) \otimes \mathbf{I}_K\}\{\mathbf{e}_\psi \otimes \mathbf{\Pi}^T\}] \otimes \mathbf{I}_N \\ &= \mathbf{I}_{NK} - \gamma(\mathbf{P}^T \otimes \mathbf{I}_N)^H[(1 - \frac{\psi\gamma}{1+\psi\gamma})\mathbf{e}_\psi \otimes \mathbf{\Pi}^T] \otimes \mathbf{I}_N \\ &= \mathbf{I}_{NK} - \frac{\gamma\mathbf{e}_\psi^T\mathbf{e}_\psi}{1+\psi\gamma} \otimes \mathbf{\Pi}^*\mathbf{\Pi}^T \otimes \mathbf{I}_N = \frac{1}{1+\psi\gamma}\mathbf{I}_{NK} \end{aligned} \tag{5.63}$$

Which is the same as in (5.59)

5.8.2 Proof of Lemma 2

We can write

$$\mathbf{y} = vec(\mathbf{Y}) = \sqrt{\gamma}(\mathbf{I}_M \otimes \mathbf{H})\mathbf{x} + \mathbf{w} \tag{5.64}$$

which, given $\mathbf{H}$, is Gaussian with the conditional covariance matrix $\mathbf{C}_{\mathbf{y}|\mathbf{H}} = \mathbb{E}_{\mathbf{x}}[\mathbf{y}\mathbf{y}^H] = \mathbf{I}_M \otimes (\gamma\mathbf{H}\mathbf{H}^H + \mathbf{I}_N)$. Then,

$$h(\mathbf{y}|\mathbf{H}) = NM\log(\pi e) + M\mathbb{E}_{\mathbf{H}}[\log|\gamma\mathbf{H}\mathbf{H}^H + \mathbf{I}_N|] \tag{5.65}$$

According to (1.14) and (1.17) in [88], for large N and K, we can write

$$\begin{aligned} V \doteq \mathbb{E}_{\mathbf{H}}[\log \mid \gamma\mathbf{H}\mathbf{H}^H + \mathbf{I}_N \mid] &\approx K\log[1 + N\gamma - A(\gamma,\beta)] \\ &+ N\log[1 + K\gamma - A(\gamma,\beta)] - \frac{1}{\gamma}A(\gamma,\beta)\log e \end{aligned} \tag{5.66}$$

with $\beta = \frac{K}{N}$ and

$$A(\gamma,\beta) = \frac{1}{4}\left[\sqrt{\gamma N\left(1+\sqrt{\beta}\right)^2 + 1} - \sqrt{\gamma N\left(1-\sqrt{\beta}\right)^2 + 1}\right]^2. \tag{5.67}$$

Carrying out the square operations in (5.67), we have

$$\begin{aligned} A(\gamma,\beta) &= \frac{1}{2}\gamma N(1+\beta) + \frac{1}{2} \\ &\quad - \frac{1}{2}\sqrt{\gamma^2N^2(1-\beta)^2 + 2\gamma N + 2\gamma N\beta + 1}. \end{aligned} \tag{5.68}$$

For large γ and $\beta = 1$, we can write

$$A(\gamma, \beta) \approx \gamma N - \sqrt{\gamma N}. \tag{5.69}$$

For large γ and $\beta \neq 1$, we use the first-order approximation (like $\sqrt{\gamma^2 + \gamma + 1} \approx \gamma(1 + \frac{1}{2}\frac{\gamma+1}{\gamma^2}) \approx \gamma + \frac{1}{2}$) to the square-root in (5.68) to obtain

$$A(\gamma, \beta) \approx \frac{1}{2}\left(\gamma N(1+\beta) - \gamma N|1-\beta| + 1 - \frac{1+\beta}{|1-\beta|}\right). \tag{5.70}$$

It follows that for $\beta = 1$, $V \approx N\log(\gamma N) - N\log e$ where we have used such approximation $1 + \frac{1}{\sqrt{\gamma}} \approx 1$. Similarly, for $\beta < 1$, $V \approx K\log[\gamma(N-K)] + N\log(\frac{N}{N-K}) - K\log e$, and for $\beta > 1$, $V \approx K\log(\frac{K}{K-N}) + N\log[\gamma(K-N)] - N\log e$. Using the above results in (5.66) and then in (5.65) results in (5.14), (5.15) and (5.16).

5.8.3 Proof of Lemma 3

This lemma is a special case of Theorem 2.39 in [88] and is also similar to Lemma 1 in [6]. By applying (5.65) on $\mathbf{Y} = \sqrt{\gamma}\tilde{\mathbf{H}}\mathbf{X} + \mathbf{W}$ we obtain (5.17). Here, $\tilde{\mathbf{H}} = [\mathbf{H}_1^T, \sqrt{\xi}\mathbf{H}_2^T]^T$ where all the entries in $\mathbf{H}_1$ and $\mathbf{H}_2$ are i.i.d. zero mean and unit variance. We can rewrite $\tilde{\mathbf{H}} \doteq \sqrt{K}\overline{\mathbf{H}} = \sqrt{K}[\sqrt{\frac{1}{K}}\mathbf{H}_1^T, \sqrt{\frac{\xi}{K}}\mathbf{H}_2^T]^T$ and let $\overline{\mathbf{H}}_1 = \sqrt{\frac{1}{K}}\mathbf{H}_1$ and $\overline{\mathbf{H}}_2 = \sqrt{\frac{1}{K}}\mathbf{H}_2$. Now all the entries in $\overline{\mathbf{H}}_1$ and $\overline{\mathbf{H}}_2$ are i.i.d. zero mean and variance equal to $\frac{1}{K}$. Now we can write the following

$$|\gamma\tilde{\mathbf{H}}\tilde{\mathbf{H}}^H + \mathbf{I}_{N_1+N_2}| = |\gamma K\overline{\mathbf{H}}\overline{\mathbf{H}}^H + \mathbf{I}_{N_1+N_2}| = |\gamma K\overline{\mathbf{H}}^H\overline{\mathbf{H}} + \mathbf{I}_K| = |\gamma K\overline{\mathbf{H}}'\mathbf{\Theta}\overline{\mathbf{H}}'^H + \mathbf{I}_K| \tag{5.71}$$

Here $\overline{\overline{\mathbf{H}}}'$ is a $K \times (N_1 + N_2)$ matrix of i.i.d. entries having zero mean and variance equal to $\frac{1}{K}$, and $\mathbf{\Theta} = diag\,[\mathbf{I}_{N_1}, \xi \mathbf{I}_{N_2}]$. So we can rewrite (5.17) as

$$h(\mathbf{Y}|\tilde{\mathbf{H}}) = (N_1 + N_2)M \log(\pi e) + M\mathbb{E}\{\log|\gamma K \overline{\overline{\mathbf{H}}}' \mathbf{\Theta} \overline{\overline{\mathbf{H}}}'^H + \mathbf{I}_K|\} \tag{5.72}$$

As $K, N_1 + N_2 \to \infty$ with $\frac{N_1+N_2}{K} \to \beta$, we can rewrite (17) in [6] for $\gamma > 0$ as:

$$\frac{1}{K}\mathbb{E}_{\overline{\overline{\mathbf{H}}}'}\{\log|\gamma K \overline{\overline{\mathbf{H}}}' \mathbf{\Theta} \overline{\overline{\mathbf{H}}}'^H + \mathbf{I}_K|\} \xrightarrow{a.s.} \beta \mathcal{V}_{\mathbf{\Theta}}(\gamma K \eta) - \log \eta + (\eta - 1)\log e \tag{5.73}$$

with

$$\begin{aligned} \mathcal{V}_{\mathbf{\Theta}}(\gamma K \eta) &= \frac{1}{N_1 + N_2} \sum_{j=1}^{N_1+N_2} \log(1 + \gamma K \eta \mathbf{\Theta}_{j,j}) \\ &= \frac{N_1}{N_1+N_2}\log(1+\gamma K \eta) + \frac{N_2}{N_1+N_2}\log(1+\xi\gamma K\eta) \end{aligned} \tag{5.74}$$

and

$$\begin{aligned} \eta_{\mathbf{\Theta}}(\gamma K \eta) &= \frac{1}{N_1 + N_2} \sum_{j=1}^{N_1+N_2} \frac{1}{1 + \gamma K \eta \mathbf{\Theta}_{j,j}} \\ &= \frac{N_1}{N_1+N_2}\frac{1}{1+\gamma K\eta} + \frac{N_2}{N_1+N_2}\frac{1}{1+\xi\gamma K\eta}, \end{aligned} \tag{5.75}$$

and $0 < \eta \leq 1$ satisfying

$$\beta = \frac{1 - \eta}{1 - \eta_{\mathbf{\Theta}}(\gamma K \eta)}. \tag{5.76}$$

Using (5.74) on (5.73) with $\beta = \frac{N_1+N_2}{K}$, we obtain,

$$\begin{aligned} \mathbb{E}_{\overline{\overline{\mathbf{H}}}'}\{\log|\gamma K \overline{\overline{\mathbf{H}}}' \mathbf{\Theta} \overline{\overline{\mathbf{H}}}'^H + \mathbf{I}_K|\} &= N_1 \log(1 + K\gamma\eta) \\ &+ N_2 \log(1 + K\xi\gamma\eta) + \log\eta + (\eta - 1)\log e \end{aligned} \tag{5.77}$$

Also, using (5.75) and (5.76) with $\beta = \frac{N_1+N_2}{K}$ we obtain the following equation to solve for η,

$$\frac{N_1/K}{1 + K\gamma\eta} + \frac{N_2/K}{1 + K\xi\gamma\eta} + 1 - \eta - \frac{N_1 + N_2}{K} = 0 \tag{5.78}$$

Analysis of η under large γ

If $\gamma \to \infty$ implies $\gamma\eta \to \infty$, then for large γ we can use the approximation $1+\gamma\eta \approx \gamma\eta$ and (5.78) becomes,

$$\eta^2 + b'\eta - c' \approx 0 \tag{5.79}$$

with $b' = \frac{N_1+N_2}{K} - 1$ and $c' = \frac{\xi N_1+N_2}{K^2\gamma\xi}$. Note that $c' \to 0$ as $\gamma \to \infty$. If $K > N_1 + N_2$, then $b' < 0$ and hence $\eta \approx -b' = 1 - \frac{N_1+N_2}{K}$, which satisfies $\gamma\eta \to \infty$ as $\gamma \to \infty$. If $K = N_1 + N_2$, then $b' = 0$ and hence $\eta \approx \sqrt{c'}$, which satisfies $\gamma\eta = \mathcal{O}(\sqrt{\gamma}) \to \infty$ as $\gamma \to \infty$. But if $K < N_1 + N_2$, then $b' > 0$ and (5.79) would imply $\eta \approx \frac{1}{2}(-b' + \sqrt{b'^2 + 4c'}) \approx \frac{c'}{b'}$, which does not satisfy $\gamma\eta \to \infty$ as $\gamma \to \infty$. So, for $K < N_1 + N_2$, we now consider $\tau \doteq \gamma\eta$ in (5.79) as $\gamma \to \infty$. It can be shown that τ increases with γ, and as $\gamma \to \infty$, τ is upper bounded. This implies $\eta \to 0$ as $\gamma \to \infty$. And in this case, (5.79) becomes

$$\tau^2 + b\tau + c \approx 0 \tag{5.80}$$

with $b = \frac{\xi(K-N_2)+(K-N_1)}{(K-N_1-N_2)K\xi}$, $c = \frac{1}{(K-N_1-N_2)K\xi}$. Note that τ is the positive solution to (5.80), which is finite and does not go to zero as $\gamma \to \infty$.

Using the values of η for different cases on (5.72), we obtain (5.18) and (5.19) for large γ.

Note that if $N_2 \to \infty$, we have $b \to \frac{1}{K}$ and $c \to \frac{-1}{\xi N_2 K}$ and thus from (5.80), $\tau \to \frac{1}{\xi N_2}$.

5.8.4 Proof of Lemma 4

$$I(\mathbf{Y}_A^{(1)};\mathbf{Y}_B^{(1)}) = h(\mathbf{Y}_A^{(1)}) + h(\mathbf{Y}_B^{(1)}) - h(\mathbf{Y}_A^{(1)},\mathbf{Y}_B^{(1)}). \tag{5.81}$$

It follows from (5.1a) and (5.1b) that

$$\mathbf{y}_A^{(1)} \doteq vec(\mathbf{Y}_A^{(1)}) = \sqrt{\gamma_{AB}}(\mathbf{\Pi}_B^T \otimes \mathbf{I}_{n_A})\mathbf{h}_{AB} + \mathbf{w}_A^{(1)},$$

$$\mathbf{y}_B^{(1)t} \doteq vec(\mathbf{Y}_B^{(1)T}) = \sqrt{\gamma_{BA}}(\mathbf{I}_{n_B} \otimes \mathbf{\Pi}_A^T)\mathbf{h}_{BA}^t + \mathbf{w}_B^{(1)t},$$

$$\mathbf{A} \doteq \mathbb{E}\{\mathbf{y}_A^{(1)}\mathbf{y}_A^{(1)H}\} = \gamma_{AB}(\mathbf{\Pi}_B^T\mathbf{\Pi}_B^* \otimes \mathbf{I}_{n_A}) + \mathbf{I}_{n_A\phi_B}, \tag{5.83}$$

$$\mathbf{B} \doteq \mathbb{E}\{\mathbf{y}_B^{(1)}\mathbf{y}_B^{(1)tH}\} = \gamma_{BA}(\mathbf{I}_{n_B} \otimes \mathbf{\Pi}_A^T\mathbf{\Pi}_A^*) + \mathbf{I}_{n_B\phi_A}, \tag{5.84}$$

$$\mathbf{C} \doteq \mathbb{E}\{\mathbf{y}_A^{(1)}\mathbf{y}_B^{(1)tH}\} = \rho\sqrt{\gamma_{BA}\gamma_{AB}}(\mathbf{\Pi}_B^T \otimes \mathbf{\Pi}_A^*). \tag{5.85}$$

Then it follows from (5.81) that

$$I(\mathbf{Y}_A^{(1)};\mathbf{Y}_B^{(1)}) = \log|\mathbf{A}| + \log|\mathbf{B}| - \log\begin{vmatrix} \mathbf{A} & \mathbf{C} \\ \mathbf{C}^H & \mathbf{B} \end{vmatrix}. \tag{5.86}$$

We will use $\begin{vmatrix} \mathbf{A} & \mathbf{C} \\ \mathbf{C}^H & \mathbf{B} \end{vmatrix} = |\mathbf{A}| \cdot |\mathbf{B} - \mathbf{C}^H\mathbf{A}^{-1}\mathbf{C}|$. Also recall the facts $|\mathbf{I} + \mathbf{M}_1\mathbf{M}_2| = |\mathbf{I} + \mathbf{M}_2\mathbf{M}_1|$ and $\mathbf{M}_1(\mathbf{M}_2\mathbf{M}_1 + \mathbf{I})^{-1}\mathbf{M}_3 = (\mathbf{M}_1\mathbf{M}_2 + \mathbf{I})^{-1}\mathbf{M}_1\mathbf{M}_3$ for compatible matrices. Then

$$I(\mathbf{Y}_A^{(1)};\mathbf{Y}_B^{(1)}) = -\log|\mathbf{I}_{n_A\phi_A} - \mathbf{B}^{-1}\mathbf{C}^H\mathbf{A}^{-1}\mathbf{C}|. \tag{5.87}$$

Here

$$\mathbf{C}^H\mathbf{A}^{-1}\mathbf{C} = |\rho|^2\gamma_{BA}\gamma_{AB}(\mathbf{\Pi}_B^* \otimes \mathbf{\Pi}_A^T)\cdot(\gamma_{AB}(\mathbf{\Pi}_B^T\mathbf{\Pi}_B^* \otimes \mathbf{I}_{n_A}) + \mathbf{I}_{n_A\phi_B})^{-1}(\mathbf{\Pi}_B^T \otimes \mathbf{\Pi}_A^*)$$

$$= |\rho|^2\gamma_{BA}\gamma_{AB}(\mathbf{I}_{n_B} \otimes \mathbf{\Pi}_A^T)\cdot(\gamma_{AB}(\mathbf{\Pi}_B^*\mathbf{\Pi}_B^T \otimes \mathbf{I}_{n_A}) + \mathbf{I}_{n_An_B})^{-1}(\mathbf{\Pi}_B^*\mathbf{\Pi}_B^T \otimes \mathbf{\Pi}_A^*)$$

$$= |\rho|^2\frac{\gamma_{BA}\gamma_{AB}\psi_B}{\gamma_{AB}\psi_B + 1}(\mathbf{I}_{n_B} \otimes \mathbf{\Pi}_A^T\mathbf{\Pi}_A^*). \tag{5.88}$$

Then, from (5.87),

$$I(\mathbf{Y}_A^{(1)};\mathbf{Y}_B^{(1)})$$

$$= -\log\left|\mathbf{I}_{n_A\phi_B} - |\rho|^2(\gamma_{BA}(\mathbf{I}_{n_B} \otimes \mathbf{\Pi}_A^T\mathbf{\Pi}_A^*) + \mathbf{I}_{n_B\phi_A})^{-1}\cdot\frac{\gamma_{BA}\gamma_{AB}\psi_B}{\gamma_{AB}\psi_B + 1}(\mathbf{I}_{n_B} \otimes \mathbf{\Pi}_A^T\mathbf{\Pi}_A^*)\right|$$

$$= -\log\left|\mathbf{I}_{n_An_B} - |\rho|^2(\gamma_{BA}(\mathbf{I}_{n_B} \otimes \mathbf{\Pi}_A^*\mathbf{\Pi}_A^T) + \mathbf{I}_{n_Bn_A})^{-1}\cdot\frac{\gamma_{BA}\gamma_{AB}\psi_B}{\gamma_{AB}\psi_B + 1}(\mathbf{I}_{n_B} \otimes \mathbf{\Pi}_A^*\mathbf{\Pi}_A^T)\right|$$

$$= -\log\left|\mathbf{I}_{n_An_B} - |\rho|^2\frac{\gamma_{BA}\gamma_{AB}\psi_B\psi_A}{(\gamma_{AB}\psi_B + 1)(\gamma_{BA}\psi_A + 1)}\mathbf{I}_{n_An_B}\right| \tag{5.89}$$

5.8.5 Proof of $\frac{\partial\omega_A}{\partial n_E} < 0$ for $n_E \geq n_A \geq n_B$

Let, $\Delta_\mu = n_A - n_B - n_E < 0$ and τ' is the solution of the following quadratic equation;

$$\tau'^2 + \tau'\frac{\xi'(n_A - n_E) + (n_A - n_B)}{\Delta_\mu\xi' n_A} + \frac{1}{\Delta_\mu\xi' n_A} \tag{5.90}$$

$$\text{or, } \frac{1}{1 + n_A\tau'}\frac{n_B}{n_A} + \frac{1}{1 + \xi' n_A\tau'}\frac{n_E}{n_A} + \frac{\Delta_\mu}{n_A} = 0 \tag{5.91}$$

Where, $\xi' = \frac{\lambda_B}{\lambda_{EA}}$. Now,

$$\begin{aligned}\omega_A &= n_A[\log\lambda_{EA} - \log\lambda_B] + n_B\log(1+n_A\tau') \\ &+ n_E\log(1+\xi' n_A\tau') + (n_E - n_A)\log(n_E - n_A) \\ &- n_E\log n_E - n_A\log\tau' \end{aligned} \tag{5.92}$$

$$\begin{aligned}\frac{\partial\omega_A}{\partial n_E} &= \left[\frac{n_A n_B}{1+n_A\tau'} + \frac{\xi' n_A n_E}{1+\xi' n_A\tau'} - \frac{n_A}{\tau'}\right]\frac{\partial\tau'}{\partial n_E} \\ &+ \log(n_E - n_A) - \log n_E + \log(1+\xi' n_A\tau') \end{aligned} \tag{5.93}$$

It will suffice to show $\frac{n_A n_B}{1+n_A\tau'} + \frac{\xi' n_A n_E}{1+\xi' n_A\tau'} - \frac{n_A}{\tau'} = 0$ and $\frac{(1+\xi' n_A\tau')(n_E - n_A)}{n_E} < 1$ for $\forall n_E \geq n_A \geq n_B$. Now we can write,

$$\frac{n_A n_B}{1+n_A\tau'} + \frac{\xi' n_A n_E}{1+\xi' n_A\tau'} - \frac{n_A}{\tau'} = \frac{n_A\mathcal{A}}{\tau'(1+n_A\tau')(1+\xi' n_A\tau')} \tag{5.94}$$

where,

$$\begin{aligned}\mathcal{A} &= n_B\tau'(1+\xi' n_A\tau') + \xi' n_E\tau'(1+n_A\tau') \\ &- (1+n_A\tau')(1+\xi' n_A\tau') \end{aligned} \tag{5.95}$$

It can be shown that $\frac{\mathcal{A}}{-\Delta_\mu\xi' n_A} = \tau'^2 + \tau'\frac{\xi'(n_A - n_E) + (n_A - n_B)}{\Delta_\mu\xi' n_A} + \frac{1}{\Delta_\mu\xi' n_A} = 0$ and hence $\frac{n_A n_B}{1+n_A\tau'} + \frac{\xi' n_A n_E}{1+\xi' n_A\tau'} - \frac{n_A}{\tau'} = 0$. Now, from (5.91) we can write:

$$\begin{aligned}\frac{n_B}{1+n_A\tau'} + \frac{n_E}{1+\xi' n_A\tau'} + \Delta_\mu &= 0 \\ \rightarrow (1+\xi' n_A\tau') &= \frac{n_E(1+n_A\tau')}{-n_B - \Delta_\mu(1+n_A\tau')} \\ &= \frac{n_E(1+n_A\tau')}{(n_E - n_A)(1+n_A\tau') + n_B n_A\tau'} < \frac{n_E}{n_E - n_A} \end{aligned} \tag{5.96}$$

So, $\frac{(1+\xi' n_A\tau')(n_E - n_A)}{n_E} < 1$ and thus we can finally conclude, $\frac{\partial\omega_A}{\partial n_E} < 0$ for $\forall n_E \geq n_A \geq n_B$.

Chapter 6

Secure Multi-Carrier Communication using STEEP

6.1 Introduction

Secret message transmission between two nodes in the presence of an eavesdropper is a long-standing problem in wireless network security. The secret key generation (SKG) model [14] and wiretap channel (WTC) model [18] are the two main pillars of wireless physical layer secrecy. However, possibility of achieving positive secrecy rate in static environments against strong eavesdropper (with more antennas and better SNR), with or without channel reciprocity was not revealed until [33, 89, 90]. The newly proposed scheme called STEEP [91] asymptotically achieves secrecy rate shown in [34]. In [92], STEEP is shown as a method for low latency secure multiple access. STEEP is a two-way round-trip communication scheme that allows positive achievable secrecy rates against a powerful

eavesdropper for static non-reciprocal channels. [93] presents further insights into STEEP for MISO setups.

Multi-carrier communication, namely OFDM, is a widely used protocol for modern wireless systems, especially in urban and high-scattering regions. In this chapter, we focus on applying STEEP for a multi-carrier link between two single-antenna nodes, Alice and Bob. STEEP, being a two-phase round-trip scheme, can benefit from the pairing of different carriers for phase-1 and phase-2. We specifically focus on different pairing policies and power scheduling across carriers in both phases. We evaluate the performance of different policies through simulations based on many random realizations of user and eavesdropper channel state information (CSI). We also compare the results with the classical wiretap channel, and observe that STEEP equipped with good policies offers significantly higher secrecy rates against a strong eavesdropper. Through simulation, we also compared the achievable secrecy rates of STEEP with secret key capacity and observed that achievable secrecy rate approaches Secret key capacity with high echoing power.

6.2 System Model

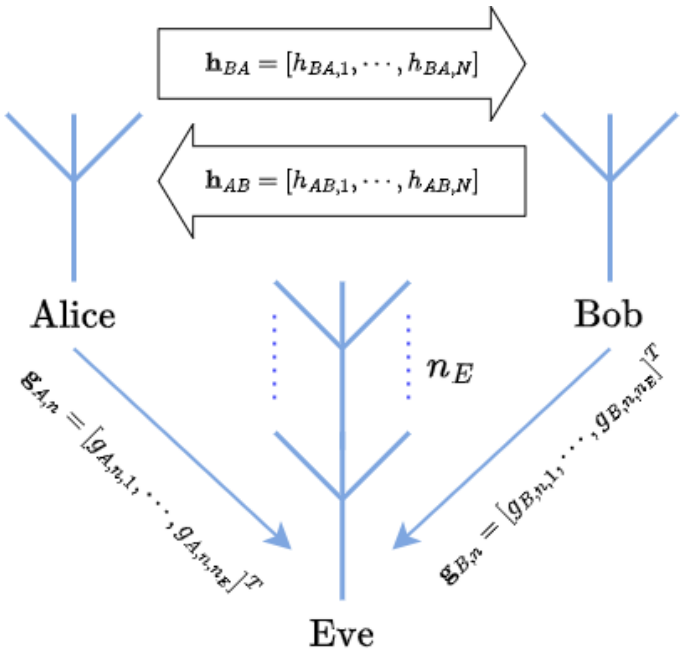


Figure 6.1: Illustration of system model for Secure MC-STEEP.

We consider two single antenna nodes, Alice and Bob operating in multi-carrier mode (namely OFDM) under flat fading in the presence of an eavesdropper (Eve) equipped with $n_E \geq 1$ antennas. The nodes are operating in Half-Duplex mode where they transmit in orthogonal time slots (in same or different coherence blocks). We assume the channel is divided into N_c sub-carriers where the small-scale fading from Alice to Bob and Bob to Alice is denoted as $\mathbf{h}_{BA} = [h_{BA,1}, \cdots, h_{BA,N_c}]$ and $\mathbf{h}_{AB} = [h_{AB,1}, \cdots, h_{AB,N_c}]$. The small-scale fading from Alice to Eve and Bob to Eve for n^{th} carrier is denoted as $\mathbf{g}_{A,n} = [g_{A,n,1}, \cdots, g_{A,n,n_E}]^T$ and $\mathbf{g}_{B,n} = [g_{B,n,1}, \cdots, g_{B,n,n_E}]^T$. These small-scale fading parameters, i.e., the Channel State Information (CSI) are assumed to be i.i.d. standard complex Gaussian, i.e., $\mathcal{CN}(0,1)$. This corresponds to non-reciprocal channel between Al-

ice and Bob. For reciprocal channel, the channel reciprocity can be modeled accordingly (see section 5.2 in chapter 5). We assume all the receive channels are known to the receiver through proper channel sounding. The large-scale fading at Eve from Alice and Bob is denoted as α and β respectively where the large-scale fading between Alice and Bob is normalized to unity. The noise power at all nodes and sub-carriers are normalized to unity and corresponding transmit power at Alice and Bob in n^{th} carrier is denoted as $p_{A,n}$ and $p_{B,n}$. We can immediately write the following for carrier-n:

$$\texttt{SNR} \text{ at Alice: } b_n = p_{B,n}|h_{AB,n}|^2 \tag{6.1a}$$

$$\texttt{SNR} \text{ at Bob: } a_n = p_{A,n}|h_{BA,n}|^2 \tag{6.1b}$$

$$\texttt{SNR} \text{ at Eve from Alice: } c_n = \alpha p_{A,n}||\mathbf{g}_{A,n}||^2 \tag{6.1c}$$

$$\texttt{SNR} \text{ at Eve from Bob: } d_n = \beta p_{B,n}||\mathbf{g}_{B,n}||^2 \tag{6.1d}$$

The quantities defined above are raw SNR at the nodes given the CSI at a given coherence block.

6.3 Brief Description of STEEP

STEEP is a two-phase, round-trip scheme to transmit secret message from one node to another (one-way). Without loss of generality, here we assume the secret message being transmitted from Bob to Alice. In phase 1 (also called the probing phase), Alice sends random symbols i.e., probe $x_{A,n}(k)$ to Bob. Bob obtains the estimate $\widehat{x}_{A,n}(k)$ from the received signal and then constructs an echo signal $\widehat{x}_{A,n}(k) + s_n(k)$ using the *secret message* $s_n(k)$. In phase 2, (also called the echoing phase) Bob sends the echo signal to

Alice. The received signals at each node in both phases can be written as follows (dropping time index (k) for convenient notation):

phase-1:

$$\text{Bob receives: } y_{B,n} = h_{BA,n}x_{A,n} + w_{B,n} \tag{6.2a}$$

$$\text{Eve receives: } \mathbf{y}_{EA,n} = \mathbf{g}_{A,n}x_{A,n} + \mathbf{w}_{EA,n} \tag{6.2b}$$

phase-2:

$$\text{Alice receives: } y_{A,n} = h_{AB,n}(\widehat{x}_{A,n} + s_n) + w_{A,n} \tag{6.2c}$$

$$\text{Eve receives: } \mathbf{y}_{EB,n} = \mathbf{g}_{B,n}(\widehat{x}_{A,n} + s_n) + \mathbf{w}_{EB,n} \tag{6.2d}$$

After phase-2, Alice obtains the estimate $\widehat{s}_{A,n}$ from the received signal and available clean $x_{A,n}$ and Eve also obtains the estimate $\widehat{s}_{E,n}$ from corresponding received signals. We can say that a virtual wiretap channel forms from Bob to Alice with respect to the secret message s_n. The MSE of MMSE estimate (which is optimal for standard Gaussian x and s) of $\widehat{s}_{A,n}$ and $\widehat{s}_{E,n}$ can be written in terms of the SNRs in (6.1) [34]:

$$\sigma^2_{\Delta\widehat{s}_{A,n}} = \frac{\frac{(a_n+1)^2}{a_n b_n} + 2}{b_n} \tag{6.3a}$$

$$\sigma^2_{\Delta\widehat{s}_{E,n}} = \frac{\frac{d_n a_n (a_n+c_n+1)}{(a_n+1)^2(c_n+1)} + 2}{d_n} \tag{6.3b}$$

We can subsequently obtain the achievable average secrecy rate (AASR) in bits per carrier per round-trip symbol interval which can be defined as:

$$C_{s,steep} \doteq \frac{1}{N_c}\sum_{n=1}^{N_c}\left[\log\left(1 + \frac{b_n}{\frac{a_n b_n}{(a_n+1)^2} + 2}\right) - \log\left(\frac{d_n}{\frac{d_n a_n (a_n+c_n+1)}{(a_n+1)^2(c_n+1)} + 2}\right)\right]^+ \tag{6.4}$$

For contrast, we also note the average secrecy rate for the classic wiretap channel where Alice and Bob both transmit (in different time slots) to each other:

$$C_{s,classic} = \frac{1}{N_c}\sum_{n=1}^{N_c}\{[\log(1+a_n) - \log(1+c_n)]^+ + [\log(1+b_n) - \log(1+d_n)]^+\} \qquad (6.5)$$

We also note some important characteristics of $C_{s,steep}$ in (6.4) from [34]:

- $C_{s,steep}$ is increasing function of b_n
- $C_{s,steep}$ increases and then decreases with a_n
- For given a_n and b_n, $C_{s,steep}$ decreases as α and β increases

6.4 Improving Secrecy Rate for MC-STEEP

In this section, we propose some policies by which we can achieve improved AASR for STEEP operating in multi-carrier mode. STEEP being a round-trip scheme where probing and echoing are performed in phase-1 and phase-2, it is not necessary to perform both probing and echoing in the same carrier. The round-trip communication can consist of probing in one carrier and echoing in another carrier. This provides the opportunity of pairing policies for probing and echoing carriers to potentially increase the AASR. However, the classical wiretap channel cannot benefit from this pairing as the secrecy rates from Alice to Bob and from Bob to Alice are independent of each other. Due to the complex expression in (6.4), power allocation algorithm (water filing) for OFDM systems will not provide the best results for STEEP. So, we also propose power allocation policies to further improve AASR.

Without the knowledge of Eavesdropper CSI, closed form expression of (6.4) is not available to guarantee optimality of the policies. However, the proposed policies are based on heuristics and their performances are evaluated and compared through simulation.

6.4.1 Paring of Probe and Echo carriers

In a given coherent block, the small scale fading (CSI) for N_c carriers from Alice to Bob and Bob to Alice is denoted as:

$\mathbf{h}_{BA} = [h_{BA,1}, \cdots, h_{BA,N_c}]$ and $\mathbf{h}_{AB} = [h_{AB,1}, \cdots, h_{AB,N_c}]$. The proposed pairing policies are as follows:

Policy-1

We also refer to this as the baseline policy. We do not perform any additional pairing, i.e., we pair the carriers according to their original sequence.

So the pairing will be: $\{h_{BA,1}, h_{AB,1}\}, \{h_{BA,2}, h_{AB,2}\}, \cdots, \{h_{BA,N_c}, h_{AB,N_c}\}$.

Policy-2

We pair the strongest probing carrier with the strongest echoing carrier and so on. We first sort both the probing and echo sub-carriers in descending order of their gain to obtain: $\tilde{\mathbf{h}}_{BA} = [\tilde{h}_{BA,1}, \tilde{h}_{BA,2}, \cdots, \tilde{h}_{BA,N_c}]$, $\tilde{\mathbf{h}}_{AB} = [\tilde{h}_{AB,1}, \tilde{h}_{AB,2}, \cdots, \tilde{h}_{AB,N_c}]$ where $|\tilde{h}_{BA,1}| \geq |\tilde{h}_{BA,2}| \geq \cdots \geq |\tilde{h}_{BA,N_c}|$ and $|\tilde{h}_{AB,1}| \geq |\tilde{h}_{AB,2}| \geq \cdots \geq |\tilde{h}_{AB,N_c}|$.

Then the pairing will be: $\{\tilde{h}_{BA,1}, \tilde{h}_{AB,1}\}, \{\tilde{h}_{BA,2}, \tilde{h}_{AB,2}\}, \cdots, \{\tilde{h}_{BA,N_c}, \tilde{h}_{AB,N_c}\}$

Policy-3

Here, we pair the strongest probing carrier with the weakest echoing carrier and so on. We first sort the probing sub-carriers in descending order of their gain to obtain: $\tilde{\mathbf{h}}_{BA} = [\tilde{h}_{BA,1}, \tilde{h}_{BA,2}, \cdots, \tilde{h}_{BA,N_c}]$ where $|\tilde{h}_{BA,1}| \geq |\tilde{h}_{BA,2}| \geq \cdots \geq |\tilde{h}_{BA,N_c}|$ and then we sort the echoing carrier in ascending order of their gain to obtain $\widehat{\mathbf{h}}_{AB} = [\tilde{h}_{AB,1}, \tilde{h}_{AB,2}, \cdots, \tilde{h}_{AB,N_c}]$ where $|\tilde{h}_{AB,1}| \leq |\tilde{h}_{AB,2}| \leq \cdots \leq |\tilde{h}_{AB,N_c}|$.

Then pair the carriers as: $\{\tilde{h}_{BA,1}, \tilde{h}_{AB,1}\}, \{\tilde{h}_{BA,2}, \tilde{h}_{AB,2}\}, \cdots, \{\tilde{h}_{BA,N_c}, \tilde{h}_{AB,N_c}\}$

Complexity of Different Policies

Policy-1 does not introduce any additional complexity. Policy-2 and policy-3 require sorting of two sets of CSI having N_c elements each which introduces additional $\mathcal{O}(2N_c \log N_c)$ complexity. To execute policy-2,3 Alice and Bob only need to make publicly available the permutation indices of the sorted carriers to achieve the pairing which is also responsible for some overhead.

6.4.2 Simulation Results of Policies-1,2,3

To illustrate the performance of policies discussed above, we obtained mean AASR as: $\bar{C}_{s,steep} \doteq \mathbb{E}_{h_{BA,n}, h_{AB,n}, \mathbf{g}_{A,n}, \mathbf{g}_{B,n}; \forall n}\{C_{s,steep}\}$. For given $N_c, \alpha, \beta, n_E, p_A, p_B$ we generated 2000 random realizations of user and eavesdropper CSI to obtain $\bar{C}_{s,steep}$.

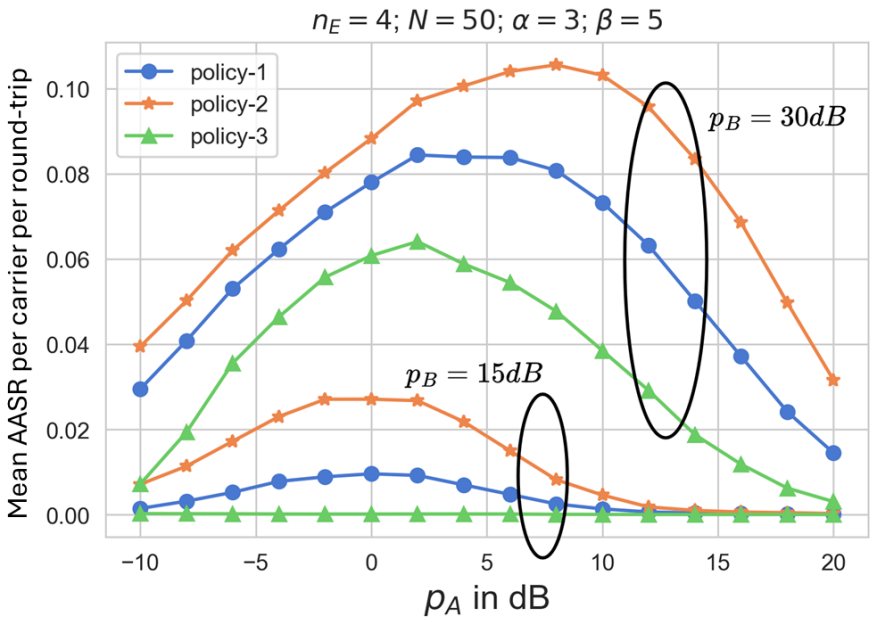


Figure 6.2: Achievable secrecy rate for policy-1,2,3 for different probing power p_A and echoing power p_B.

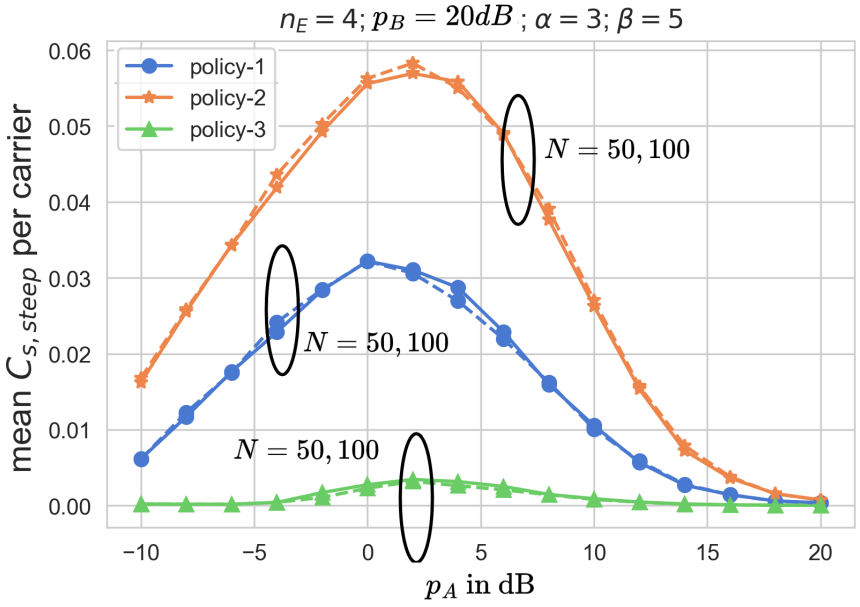


Figure 6.3: Achievable secrecy rate for policy-1,2,3 for different probing power p_A and number of carriers N_c.

We can see from the above illustrations that policy-2 performs considerably better than the other two. Although above figures only illustrate the mean secrecy rate throughout the channel realizations, simulation results show policy-2 also performs better in each individual realizations. It is also observed in fig. 6.3 that for the large N_c, the number of carrier has negligible effect on AASR. As policy-2 performs best among the above mentioned policies, it will be used as the default pairing policy for subsequent discussion.

6.4.3 Power Allocation over Sub-Carriers

Now we discuss proposed power allocation policies at Alice (probing phase) and Bob (echoing phase) to further improve the AASR. Let, Alice and Bob have available power budget of $N_c p_A$ and $N_c p_B$ for total N_c carriers, and allocated power for n^{th} carrier is denoted as $p_{A,n}$ and $p_{B,n}$. Please note that although we do not assume that the users have the knowledge of eavesdropper CSI, we assume that the users know the statistics of eavesdropper CSI which is used in the power allocation policies. First we discuss echoing power allocation:

Power allocation at Bob (echo phase); policy-4

We can rewrite (6.4) in terms of $p_{B,n}$ and other terms:

$$C_{s,steep} \doteq \frac{1}{N_c} \sum_{n=1}^{N_c} \left[\log \left(1 + \frac{\tilde{q}_n p_{B,n}}{\tilde{r}_n p_{B,n} + 2} \right) - l_n \right]^+ \tag{6.6}$$

Where $\tilde{q}_n = |\tilde{h}_{AB,n}|^2$ and $\tilde{r}_n = \frac{\tilde{a}_n \tilde{q}_n}{(\tilde{a}_n+1)^2}$ and $\tilde{a}_n = p_{A,n}|\tilde{h}_{BA,n}|^2$. Without knowing Eve's CSI, the exact l_n is not available. However, using the knowledge of the statistics of Eve's CSI, Bob can statistically obtain l_n using the following:

$$l_n = \mathbb{E}_{\mathbf{g}_A,\mathbf{g}_B} \log\left(1 + \frac{d_n}{\frac{d_n\tilde{a}_n(\tilde{a}_n+c_n+1)}{(\tilde{a}_n+1)^2(c_n+1)}+2}\right) \tag{6.7}$$

where $c_n = \alpha p_A ||\mathbf{g}_A||^2$ and $d_n = \beta p_B ||\mathbf{g}_B||^2$. We also define the following:

$$v_n(p_{B,n}) \doteq \log(1 + \frac{\tilde{q}_n p_{B,n}}{\tilde{r}_n p_{B,n}+2}) - l_n \tag{6.8a}$$

$$g_n(p_{B,n}) \doteq \frac{\partial v_n(p_{B,n})}{\partial p_{B,n}} = \frac{\log e}{1+\frac{\tilde{q}_n p_{B,n}}{\tilde{r}_n p_{B,n}+2}} \frac{\tilde{q}_n}{(\tilde{r}_n p_{B,n}+2)^2} \tag{6.8b}$$

Here $v_n(p_{B,n})$ is monotonically increasing and $g_n(p_{B,n})$ is monotonically decreasing function of $p_{B,n}$. by considering $\tilde{q}_n$, $\tilde{r}_n$, l_n as constants, we can formulate the following the convex optimization problem which should have a unique solution from its KKT conditions:

$$\mathbb{P}1: \max_{p_{B,n},\forall n} C_{s,steep}; \qquad s.t.: \sum_{n=1}^{N_c} p_{B,n} \leq N_c p_B \tag{6.9}$$

We let $J = -C_{s,steep} + \mu(\sum_{n=1}^{N_c} p_{B,n} - N_c p_B)$. The KKT conditions are:

$$g_n^{'}(p_{B,n}) = \mu;\ \forall n \ \texttt{where}\ g_n^{'}(p_{B,n}) > 0 \tag{6.10a}$$

$$\sum_{n=1}^{N_c} p_{B,n} = N_c p_B \tag{6.10b}$$

Here $g_n^{'}(p_{B,n}) = g_n(p_{B,n})$ if $v_n(p_{B,n}) > 0$ or 0 otherwise. We can use bisection method to solve the above problem. As $g_n(p_{B,n})$ is monotonically decreasing, we can also use bisection to obtain $p_{B,n}$ for $g_n^{'}(p_{B,n}) = \mu$. Algorithm 3 shows the steps to solve the echoing power allocation (at Bob). Intuitively, Bob removes power from carriers with smaller channel gain that result in zero achievable secrecy rate. The total power is then allocated to the remaining carriers that have positive secrecy rate according to their gradients ($g_n(p_{B,n})$).

This power allocation introduces additional complexity of $\mathcal{O}\left\{N_c \log\left(\frac{N_c p_B}{\texttt{tol}}\right)\right\} \times \mathcal{O}\left\{\log\left(\frac{\mu_{\max}}{\texttt{tol}}\right)\right\}$. The first term is due to the bisection search for $p_{B,i};\ \forall i \in I$ ranging from 0

Algorithm 3 Bisection Algorithm to find $p_{B,n}$; $\forall n$

Input: $p_T = N_c p_B$ and function $v_n(\cdot), g_n^{'}(\cdot)$; $\forall n$. **Output:** $p_{B,n}$; $\forall n$

Initiate: $\mu_1 = 0, \mu_2 = \texttt{Null}, \mu = 10^{-6}, I = \{1, \cdots, N_c\}$

$\texttt{thres} = 0, P_{th} = 10^{-1}, \texttt{iter}_{\texttt{max}} = 500, \texttt{iter} = 1$

for $\texttt{iter} \leq \texttt{iter}_{\texttt{max}}$ **do**

for each $i \in I$ **do**

Compute $p_{B,i}$ for $g_i^{'}(p_{B,i}) = \mu$ using bisection, then Compute $v_i(p_{B,i})$

if $v_i(p_{B,i}) \leq \texttt{thres}$ **then**

Set $p_{B,i} = 0$, $\mu_1 = 0$, $\mu_2 = \texttt{Null}$, then Remove i from I

end if

end for

Compute $P_G = \sum_{j=1}^{N_c} p_{B,j} - P_T$

If $|P_G| < P_{\texttt{th}}$: **return** $p_{B,n}$; $\forall n$

elif $P_G > 0$: Set $\mu_1 = \mu$, $\mu = \frac{1}{2}(\mu_1 + \mu_2)$; or $\mu = 2\mu$ if μ_2 is $\texttt{Null}$

elif $P_G < 0$:Set $\mu_2 = \mu$ and $\mu = \frac{1}{2}(\mu_1 + \mu_2)$

Check if $|P_G|$ is same for 5 consecutive iterations (algorithm is stuck for set I)

if $|P_G|$ is same for 5 consecutive iterations **then**

Increase $\texttt{thres}$ by 10^{-2} (this will remove element from I that is causing problem)

end if

Increase $\texttt{iter}$ by 1

end for

return $\texttt{Null}$ =0

to $N_c p_B$ and the second term is due to bisection search for μ. Here, $\texttt{tol}$ is the tolerance of the bisection searches and $\mu_{\texttt{max}}$ is the maximum range of μ.

Power allocation at Alice (probing phase); policy-5

Subsequent to echoing power allocation at Bob, probing power allocation at Alice can be done. Unlike the echoing power, the AASR given by (6.4) is not monotonically increasing with probing power. From fig. 6.2, it is apparent that AASR increases then decreases with $p_{A,n}$ and without the knowledge of Eve's channel, the closed form for (6.4) is not feasible. First, we propose a simple method for probing power allocation which is as follows: After Bob allocates power to $N_c' \leq N_c$ echoing carriers, Alice equally distributes total power $N_c p_A$ only to the corresponding N_c' probing carriers, i.e., Alice also removes power from corresponding carriers where Bob removed power from and distribute total power equally to remaining carriers. We denote the policy as 'policy-5-uniform'. We also propose another probing power allocation policy denoted as 'policy-5-waterfil' where Alice distributes total power $N_c p_A$ only to the corresponding N_c' probing carriers using water-filing algorithm according to the CSI (maximize $\sum_{n\in\mathcal{N}_c'} \log(1+p_{A,n}|\tilde{h}_{BA,n}|^2)$ s.t. $\sum_{n\in\mathcal{N}_c'} p_{A,n} \leq N_c p_A$, here $\mathcal{N}_c'$ is set of carriers with nonzero echo power). Policy-5-uniform introduces additional $\mathcal{O}\{N_c\}$ complexity and policy-5-waterfil introduces additional $\mathcal{O}\{N_c \log(\mu_{\texttt{max}}/\texttt{tol})\}$ complexity.

Simulation Results of Policies-4,5

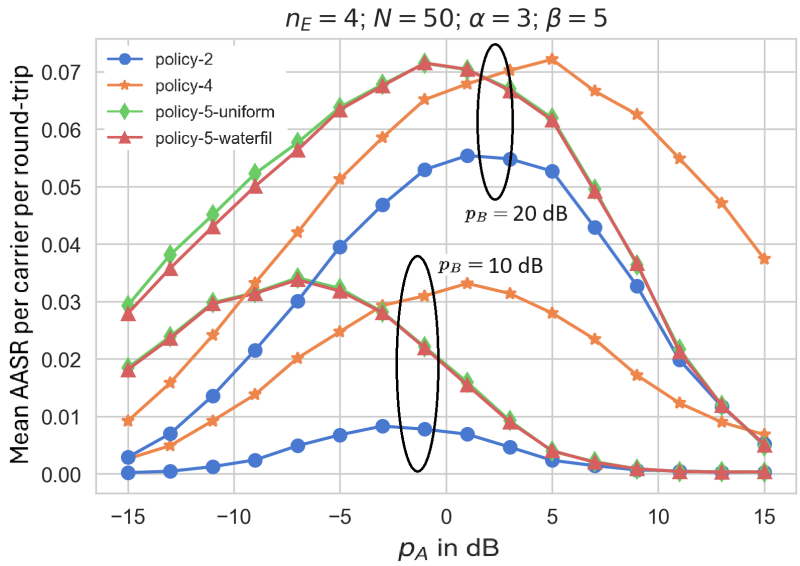


Figure 6.4: Achievable secrecy rate for policies-2,4,5 for different probing power p_A and echoing power p_B.

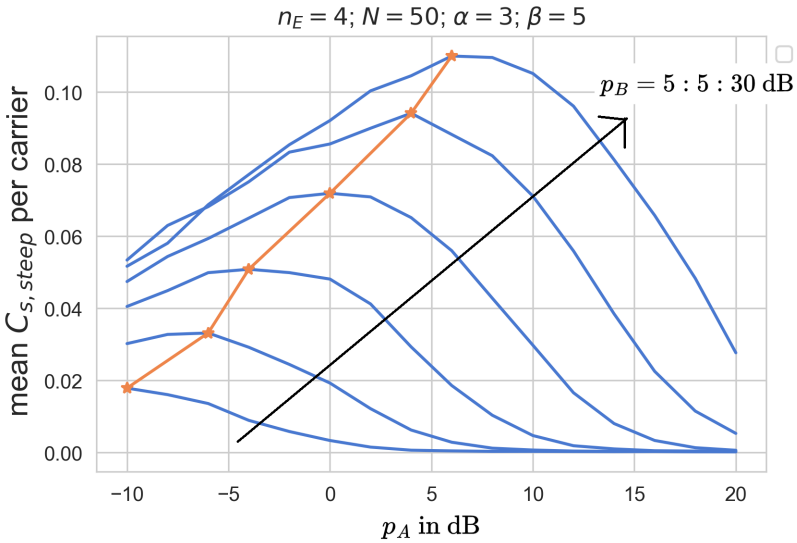


Figure 6.5: Achievable secrecy rate for policy-5 for different probing power p_A and echoing power p_B. For a given echo power, probing power should not exceed a threshold (orange line).

Fig. 6.4 shows the mean AASR for policy-4 and -5 (policy-2 which is the best among 1,2,3, is also included for comparison). Like fig. 6.2, mean AASR was obtained from 2000 random realizations of user and eavesdropper CSI. From the illustrations, it is observed that policy-4 significantly improves AASR compared to policy-2. Although policy-5 does not improve the AASR, it allows achieving the same AASR for much lower p_A. We also observe that the two variants of policy-5 provide almost identical results. As the water filling algorithm only optimizes for the channel capacity, it does not have much effect on the achievable secrecy rate which also involves eavesdropper CSI and a different expression than the channel capacity. As both variants of policy-5 produce nearly identical performance and policy-5-uniform requires much lower complexity than its variant, we will use policy-5-uniform for subsequent analysis and refer to it as 'policy-5'.

Also it is evident from the plots that AASR does not monotonically increase with probing power p_A, which implies that there is an optimal probing power beyond which the AASR would decrease. The optimal probing power may not be the maximum available power and probing should not be done at larger power than a threshold (shown in fig. 6.5).

6.5 Comparison with SKC and Classic WTC

In this section, we present some simulation results to compare STEEP policies with the classic WTC channel and secret key capacity.

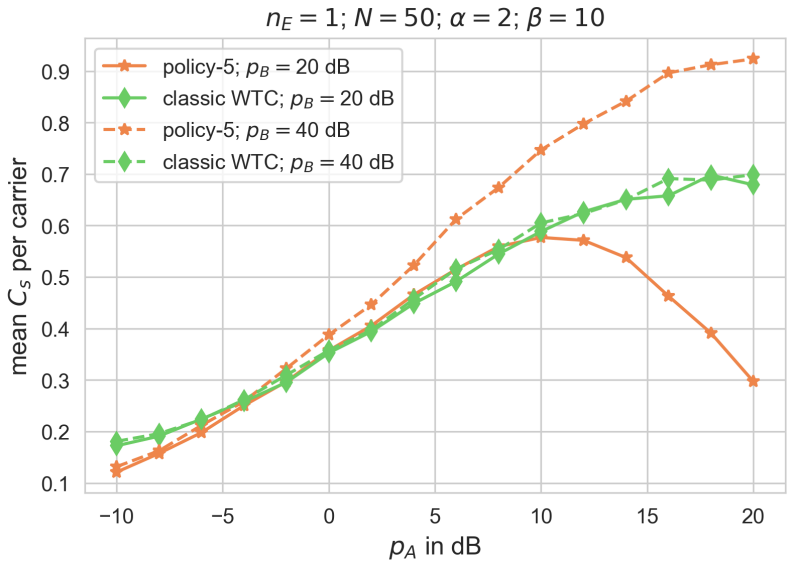


Figure 6.6: Achievable secrecy rate of STEEP policy-5 VS classic water-filing for $n_E = 1$, $N_c = 50$, $\alpha = 2$, $\beta = 10$ and different p_A and p_B.

6.5.1 Classic WTC

In the classic WTC scheme, for given N_c, α, β, n_E, we have used 2000 channel realizations to obtain AASR using (6.5) which captures both way transmissions from Alice and Bob. We have also applied water-filling algorithm for power allocation at Alice and Bob which is optimal for this scheme. From the plots in fig. 6.6, 6.7 and 6.8, we can see that the STEEP with proper pairing and power scheduling (policy-5), performs significantly better than classic WTC in terms of average achievable secrecy rate. Especially against Eavesdropper with more antennas, AASR for classical WTC vanishes but MC-STEEP provides higher rate. Also, STEEP benefits from increasing the echoing power (p_B) whereas the classical WTC is indifferent to different echoing power.

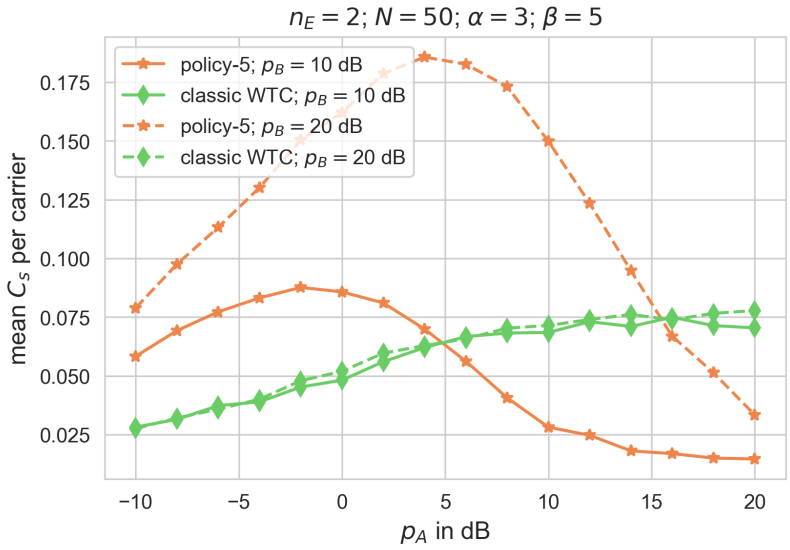


Figure 6.7: Achievable secrecy rate of STEEP policy-5 VS classic water-filing for $n_E = 2$, $N_c = 50$, $\alpha = 3$, $\beta = 5$ and different p_A and p_B.

6.5.2 Secret Key Capacity

For a given realization of $h_{BA,n}$ and $\mathbf{g}_{A,n}$, $C_{key,n}$ in bits per probing interval for a given carrier n is given by [34]:

$$C_{key,n} = \log\left(1 + \frac{p_{A,n}|h_{BA,n}|^2}{1 + \alpha p_{A,n}||\mathbf{g}_{A,n}||^2}\right) = \log\left(1 + \frac{a_n}{1 + c_n}\right) \tag{6.11}$$

Here, $p_{A,n}, \forall n$ can be obtained by uniformly distributing total available probing power to all carriers (similar to policy-5-uniform) as well as distributing total available power according to water filing based on $h_{BA,n}$ (similar to policy-5-waterfil).

Mean AASR of policy-5 (both variants) and C_{key} for both uniform and waterfil power allocation are plotted in fig. 6.9. For given set of parameters, we used $R = 2000$ realizations of user and eavesdropper CSI to obtain the plots. We can see that for increasing echo power p_B, policy-5 seems to approach C_{key} and C_{key} saturates with increasing probing power. Here it is observable that using water-filing for probing power to maximize channel

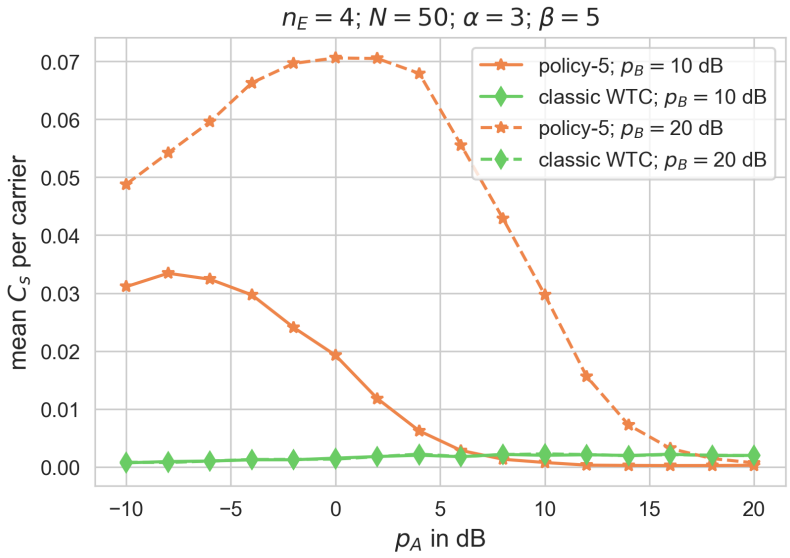


Figure 6.8: Achievable secrecy rate of STEEP policy-5 VS classic water-filing for $n_E = 4$, $N_c = 50$, $\alpha = 3$, $\beta = 5$ and different p_A and p_B.

capacity results in somewhat reduced C_{key}. As C_{key} in (6.11) involves both user and eavesdropper CSI, optimizing for channel capacity using water filing method does not necessarily result in higher key capacity. Distributions of AASR for policy-5 (both variants) and C_{key} (for both uniform and waterfil power allocation) for different probing and echoing power illustrated in fig. 6.10. For a given echoing power, the probing power is chosen where mean AASR is maximized. It is observed that the distributions with uniform power allocation and optimized power allocation with water filing are almost identical. Power allocation for maximizing channel capacity does not affect achievable average secrecy rate in STEEP and secrecy capacity.

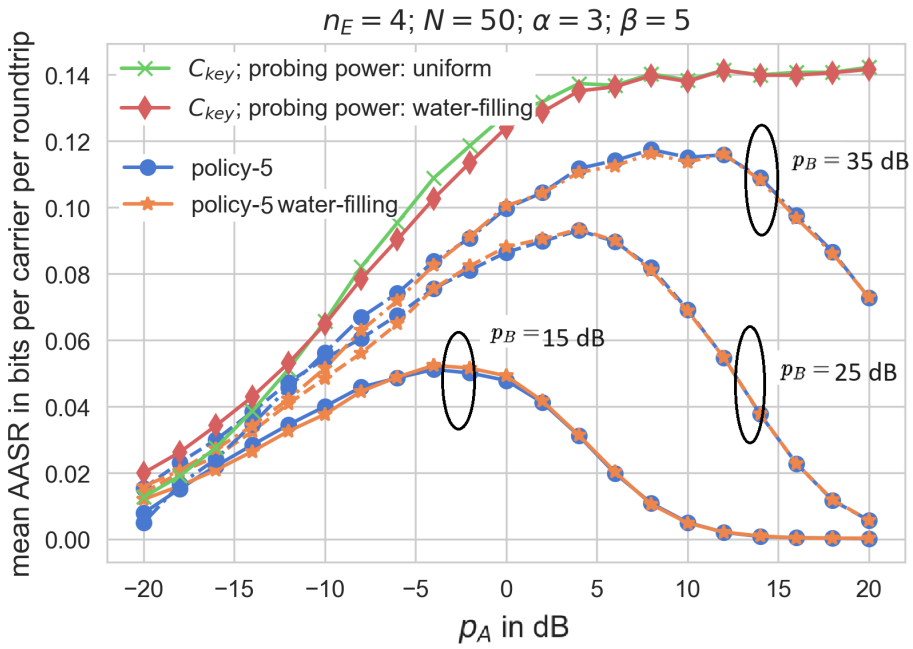


Figure 6.9: Secrecy Capacity and Achievable secrecy rate for policies-5 for different p_A and p_B.

6.6 Conclusion

In this chapter, we have presented different policies for STEEP in multi-carrier setup. In terms of Average Achievable Secrecy Rate (AASR), STEEP can benefit from accordingly pairing the probing and echoing carriers whereas, the classical WTC cannot. As the closed form for the AASR is hard to obtain without knowing the eavesdropper channel, we proposed some pairing policies based on heuristics. From simulations it is evident that pairing the strongest probing carrier with the strongest echoing carrier and so on (policy-2), improves AASR significantly. We have proposed echoing power scheduling algorithm where power is cut off from the weak carriers that do not contribute to AASR and redistributed to remaining carriers using KKT conditions. For probing power allocation, we have seen that distributing total power to active carriers uniformly and by using water filing

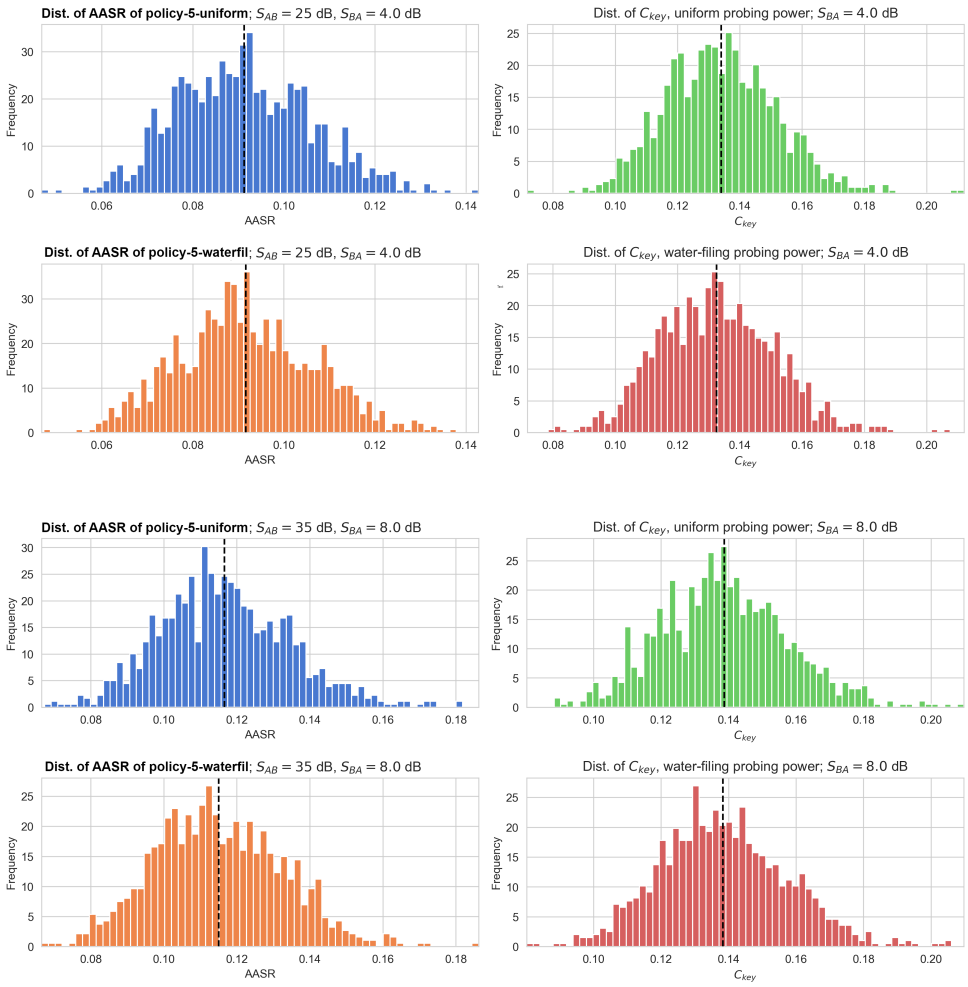


Figure 6.10: Distribution of Achievable secrecy rate for policy-5 and C_{key} for different probing and echoing power.

algorithm to maximize channel capacity gives almost identical results which is getting same AASR at lower probing power budget. It is also observed that for a given echoing power, probing power should not be increased arbitrarily. Finally, we compared the best policy (policy-5) with classical wire-tap channel and secret key capacity. Simulation results show STEEP policy-5 performing significantly better than WTC channel model especially against eavesdropper with large number of antennas. Also, it is observed that AASR of STEEP policy-5 approaches secret key capacity with large echoing power. Further investigation

can be carried out to explore better pairing and power scheduling policies with optimality guarantees and potential extension of MIMO links and short-block-length regime.

Chapter 7

Conclusions

In this thesis, we have explored various facets of physical layer security (PLS) to address the increasing concerns regarding data transmission and storage in wireless networks. The research highlighted the vulnerabilities and challenges posed by the broadcast nature of wireless communication and proposed advanced methods to ensure robust security against strong eavesdroppers.

In chapter 2 we began by introducing the concept of Unconditional Secrecy (UNS) and exploring the promising method RRCM for secure transmission. We explored the concept of continuous encryption which provides light-weight low-latency solution to the encryption-decryption problem without establishing secret key. We also propose SVD-CEF which encrypts shared secret vector (SV) into a long sequence of continuous random outputs. SVD-CEF is shown to be robust against Newton's search algorithm and exhaustive search and possesses desirable statistical properties.

In chapter 3 we proposed PLE techniques for secure UAV to Ground communication. Our method converts the shared secret vector into uniform RVs using SVD-CEF which is then used to encrypt symbols and hide the constellation. We derived the noise propagation of different transformations and provided simulation results that show the performance in terms of symbol error rate.

In chapter 4 we proposed a generalized secret key generation technique with the help of SVD-CEF. Unlike direct quantization method (DQ), our proposed method allows one or even fractional bits to be extracted from a sample, which significantly reduces Key Error Rate. We also tested the randomness using NIST randomness test suite and the generated keys showed sufficient randomness.

In chapter 5 we derived Secret Key Capacity (SKC) bounds for MIMO channel with Gaussian probing. While the First-order-Terms (FoTs) are the same as previously reported, the Second-order-Terms (SoTs) reveal useful insights. It is shown that SKC is positive if the channel between users is non-reciprocal and Eve has more antenna and better SNR.

In chapter 6 we discussed different policies to apply STEEP, a novel secret message transmission scheme for SISO multi-carrier setup. STEEP being a round-trip scheme can benefit from pairing of probing and echoing carriers. We also proposed power scheduling techniques to further improve the performance.

While the methods presented in this thesis provide valuable insights into methods for network security, they represent only a fraction of the potential problems in this field. It is essential to acknowledge that there is still much to explore and discover. In conclusion,

we briefly outline several logical extensions of this work that can pave the way for future research. These extensions hold promise for further advancements in the field and offer potential directions for future investigations.

- Further test the robustness of CEF against potential attack schemes
- Extensive comparison between traditional key generation based encryption and continuous encryption.
- Exploration of SKC for relay channel
- Exploration of STEEP for multi-carrier setup for MIMO channels
- Exploration of Achievable Secrecy Rate for finite block length regime

In conclusion, this thesis has significantly advanced the understanding and implementation of physical layer security in wireless networks. The proposed methods and theoretical insights offer practical solutions for enhancing the security of modern communication systems. Future research could further refine these techniques and explore their applicability in emerging technologies and more complex network scenarios.

Bibliography

[1] Jonathan Katz and Yehuda Lindell. *Introduction to modern cryptography: principles and protocols*. Chapman and Hall/CRC, 2007.

[2] Zijie Ji, Yan Zhang, Zunwen He, Phee Lep Yeoh, Bin Li, Hao Yin, Yonghui Li, and Branka Vucetic. Wireless secret key generation for distributed antenna systems: A joint space-time-frequency perspective. *IEEE Internet of Things Journal*, 9(1):633–647, 2021.

[3] Yuexing Peng, Peng Wang, Wei Xiang, and Yonghui Li. Secret key generation based on estimated channel state information for TDD-OFDM systems over fading channels. *IEEE Transactions on Wireless Communications*, 16(8):5176–5186, 2017.

[4] Claude E Shannon. Communication theory of secrecy systems. *The Bell System Technical Journal*, 28(4):656–715, 1949.

[5] Mahdi Shakiba-Herfeh, Arsenia Chorti, and H Vincent Poor. Physical layer security: Authentication, integrity, and confidentiality. *Physical Layer Security*, pages 129–150, 2021.

[6] Reza Sohrabi, Qiping Zhu, and Yingbo Hua. Secrecy analyses of a full-duplex MIMOME network. *IEEE Transactions on Signal Processing*, 67(23):5968–5982, 2019.

[7] Yingbo Hua. Advanced properties of full-duplex radio for securing wireless network. *IEEE Transactions on Signal Processing*, 67(1):120–135, 2018.

[8] Changick Song. Leakage rate analysis for artificial noise assisted massive MIMO with non-coherent passive eavesdropper in block-fading. *IEEE Transactions on Wireless Communications*, 18(4):2111–2124, 2019.

[9] Ashish Khisti, Gregory Wornell, Ami Wiesel, and Yonina Eldar. On the gaussian MIMO wiretap channel. In *2007 IEEE International Symposium on Information Theory (ISIT)*, pages 2471–2475. IEEE, 2007.

[10] H Vincent Poor and Rafael F Schaefer. Wireless physical layer security. *Proceedings of the National Academy of Sciences*, 114(1):19–26, 2017.

[11] Matthieu Bloch and Joao Barros. *Physical-layer security: from information theory to security engineering*. Cambridge University Press, 2011.

[12] Yongpeng Wu, Ashish Khisti, Chengshan Xiao, Giuseppe Caire, Kai-Kit Wong, and Xiqi Gao. A survey of physical layer security techniques for 5G wireless networks and challenges ahead. *IEEE Journal on Selected Areas in Communications*, 36(4):679–695, 2018.

[13] Henri Hentila, Visa Koivunen, H Vincent Poor, and Rick S Blum. Secure key generation for distributed inference in IoT invited presentation. In *2019 Annual Conference on Information Sciences and Systems (CISS)*, pages 1–6. IEEE, 2019.

[14] Ueli M Maurer. Secret key agreement by public discussion from common information. *IEEE Transactions on Information Theory*, 39(3):733–742, 1993.

[15] Chunxuan Ye, Alex Reznik, and Yogendra Shah. Extracting secrecy from jointly gaussian random variables. In *2006 IEEE International Symposium on Information Theory (ISIT)*, pages 2593–2597. IEEE, 2006.

[16] Chan Chen and Michael A Jensen. Secret key establishment using temporally and spatially correlated wireless channel coefficients. *IEEE Transactions on Mobile Computing*, 10(2):205–215, 2010.

[17] Jon W Wallace and Rajesh K Sharma. Automatic secret keys from reciprocal MIMO wireless channels: Measurement and analysis. *IEEE Transactions on Information Forensics and Security*, 5(3):381–392, 2010.

[18] Aaron D Wyner. The wire-tap channel. *The Bell System Technical Journal*, 54(8):1355–1387, 1975.

[19] Satashu Goel and Rohit Negi. Guaranteeing secrecy using artificial noise. *IEEE Transactions on Wireless Communications*, 7(6):2180–2189, 2008.

[20] Xiaohua Li, Juite Hwu, E Paul Ratazzi, et al. Using antenna array redundancy and channel diversity for secure wireless transmissions. *J. Commun.*, 2(3):24–32, 2007.

[21] Chunxuan Ye, Suhas Mathur, Alex Reznik, Yogendra Shah, Wade Trappe, and Narayan B Mandayam. Information-theoretically secret key generation for fading wireless channels. *IEEE Transactions on Information Forensics and Security*, 5(2):240–254, 2010.

[22] Kai Zeng, Daniel Wu, An Chan, and Prasant Mohapatra. Exploiting multiple-antenna diversity for shared secret key generation in wireless networks. In *2010 IEEE Conference on Computer Communications (INFOCOM)*, pages 1–9. IEEE, 2010.

[23] Y-W Peter Hong, Lin-Ming Huang, and Hou-Tung Li. Vector quantization and clustered key mapping for channel-based secret key generation. *IEEE Transactions on Information Forensics and Security*, 12(5):1170–1181, 2017.

[24] Yingbo Hua. Reliable and secure transmission for future networks. In *2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5260–5264. IEEE, 2020.

[25] Andrew Beng Jin Teoh and Chong Tze Yuang. Cancelable biometrics realization with multispace random projections. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 37(5):1096–1106, 2007.

[26] Bian Yang, Daniel Hartung, Koen Simoens, and Christoph Busch. Dynamic random projection for biometric template protection. In *2010 IEEE International Conference on Biometrics: Theory, Applications and Systems (BTAS)*, pages 1–7. IEEE, 2010.

[27] Zhe Jin, Jung Yeon Hwang, Yen-Lung Lai, Soohyung Kim, and Andrew Beng Jin Teoh. Ranking-based locality sensitive hashing-enabled cancelable biometrics: Index-of-max hashing. *IEEE Transactions on Information Forensics and Security*, 13(2):393–407, 2017.

[28] Lifeng Lai, Siu-Wai Ho, and H Vincent Poor. Privacy–security trade-offs in biometric security systems—part I: Single use case. *IEEE Transactions on Information Forensics and Security*, 6(1):122–139, 2010.

[29] AK Jain, K Nandakumar, and A Nagar. Biometric template security. *EURASIP Journal on Advances in Signal Processing*, 579416:17, 2008.

[30] Suhas Mathur, Wade Trappe, Narayan Mandayam, Chunxuan Ye, and Alex Reznik. Radio-telepathy: Extracting a secret key from an unauthenticated wireless channel. In *2008 ACM International Conference on Mobile Computing and Networking*, pages 128–139, 2008.

[31] S Sandeep Pradhan and Kannan Ramchandran. Distributed source coding using syndromes (DISCUS): Design and construction. *IEEE Transactions on Information Theory*, 49(3):626–643, 2003.

[32] Simon Heron. Advanced encryption standard (AES). *Network Security*, 2009(12):8–12, 2009.

[33] Yingbo Hua. Generalized channel probing and generalized pre-processing for secret key generation. *IEEE Transactions on Signal Processing*, 71:1067–1082, 2023.

[34] Yingbo Hua. On secret-message transmission by echoing encrypted probes. *IEEE Transactions on Communication*, 2024.

[35] Yeow-Khiang Chia and Abbas El Gamal. Wiretap channel with causal state information. *IEEE Transactions on Information Theory*, 58(5):2838–2849, 2012.

[36] Anil K Jain, Arun Ross, and Umut Uludag. Biometric template security: Challenges and solutions. In *2005 European Signal Processing Conference*, pages 1–4. IEEE, 2005.

[37] Vishal M Patel, Nalini K Ratha, and Rama Chellappa. Cancelable biometrics: A review. *IEEE Signal Processing Magazine*, 32(5):54–65, 2015.

[38] Dima Grigoriev and Sergey Nikolenko. Continuous hard-to-invert functions and biometric authentication. *Groups Complexity Cryptology*, 2012.

[39] Yingbo Hua and Ahmed Maksud. Unconditional secrecy and computational complexity against wireless eavesdropping. In *2020 IEEE International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*, pages 1–5. IEEE, 2020.

[40] Jaishanker K. Pillai, Vishal M. Patel, Rama Chellappa, and Nalini K. Ratha. Secure and robust iris recognition using random projections and sparse representations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(9):1877–1893, 2011.

[41] Gene H. Golub and Charles F. Van Loan. *Matrix computations*. Johns Hopkins University Press, 2013.

[42] Simon Kirchgasser, Christof Kauba, Yen-Lung Lai, Jin Zhe, and Andreas Uhl. Finger vein template protection based on alignment-robust feature description and index-of-maximum hashing. *IEEE Transactions on Biometrics, Behavior, and Identity Science*, 2(4):337–349, 2020.

[43] Leonid A Levin. The tale of one-way functions. *Problems of Information Transmission*, 39(1):92–103, 2003.

[44] M Bridson, T Gowers, J Barrow-Green, and I Leader. Geometric and combinatorial group theory. *The Princeton Companion to Mathematics*, page 10, 2008.

[45] Anne Greenbaum, Ren-cang Li, and Michael L Overton. First-order perturbation theory for eigenvalues and eigenvectors. *SIAM review*, 62(2):463–482, 2020.

[46] Yingbo Hua and Ahmed Maksud. Continuous encryption functions for security over networks. *Signal Processing*, 203:108807, 2023.

[47] Jan R. Magnus and Heinz Neudecker. *Matrix differential calculus with applications in statistics and econometrics*. John Wiley & Sons, 2019.

[48] Yong Zeng, Rui Zhang, and Teng Joon Lim. Wireless communications with unmanned aerial vehicles: Opportunities and challenges. *IEEE Communications Magazine*, 54(5):36–42, 2016.

[49] Ning Xie, ZhuoYuan Li, Jie Tan, and Alex X Liu. Detection of information hiding at physical layer in wireless communications. *IEEE Transactions on Dependable and Secure Computing*, 19(2):1104–1117, 2020.

[50] Guangchi Zhang, Qingqing Wu, Miao Cui, and Rui Zhang. Securing UAV communications via joint trajectory and power control. *IEEE Transactions on Wireless Communications*, 18(2):1376–1389, 2019.

[51] Xiaofang Sun, Derrick Wing Kwan Ng, Zhiguo Ding, Yanqing Xu, and Zhangdui Zhong. Physical layer security in UAV systems: Challenges and opportunities. *IEEE Wireless Communications*, 26(5):40–47, 2019.

[52] Ananthram Swami and Brian M Sadler. Hierarchical digital modulation classification using cumulants. *IEEE Transactions on Communications*, 48(3):416–429, 2000.

[53] Fahed Hameed, Octavia A Dobre, and Dimitrie C Popescu. On the likelihood-based approach to modulation classification. *IEEE Transactions on Wireless Communications*, 8(12):5884–5892, 2009.

[54] Van-Sang Doan, Thien Huynh-The, Cam-Hao Hua, Quoc-Viet Pham, and Dong-Seong Kim. Learning constellation map with deep CNN for accurate modulation recognition. In *2020 IEEE Global Communications Conference (GLOBECOM)*, pages 1–6. IEEE, 2020.

[55] Shengliang Peng, Hanyu Jiang, Huaxia Wang, Hathal Alwageed, Yu Zhou, Marjan Mazrouei Sebdani, and Yu-Dong Yao. Modulation classification based on signal constellation diagrams and deep learning. *IEEE Transactions on Neural Networks and Learning Systems*, 30(3):718–727, 2018.

[56] Muhammad Zaid Hameed, András György, and Deniz Gündüz. The best defense is a good offense: Adversarial attacks to avoid modulation detection. *IEEE Transactions on Information Forensics and Security*, 16:1074–1087, 2020.

[57] Meysam Sadeghi and Erik G Larsson. Adversarial attacks on deep-learning based radio signal classification. *IEEE Wireless Communications Letters*, 8(1):213–216, 2018.

[58] Haojun Zhao, Yun Lin, Song Gao, and Shui Yu. Evaluating and improving adversarial attacks on DNN-based modulation recognition. In *2020 IEEE Global Communications Conference (GLOBECOM)*, pages 1–5. IEEE, 2020.

[59] Bryse Flowers, R Michael Buehrer, and William C Headley. Communications aware adversarial residual networks for over the air evasion attacks. In *2019 IEEE Military Communications Conference (MILCOM)*, pages 133–140. IEEE, 2019.

[60] Alex Berian, Kory Staab, Gregory Ditzler, Tamal Bose, and Ravi Tandon. Adversarial filters for secure modulation classification. In *2021 Asilomar Conference on Signals, Systems, and Computers*, pages 361–367. IEEE, 2021.

[61] Krystian Grzesiak, Zbigniew Piotrowski, and Jan M Kelner. A wireless covert channel based on dirty constellation with phase drift. *Electronics*, 10(6):647, 2021.

[62] Changsheng You and Rui Zhang. 3D trajectory optimization in rician fading for UAV-enabled data harvesting. *IEEE Transactions on Wireless Communications*, 18(6):3192–3207, 2019.

[63] JM Meredith. Study on enhanced LTE support for aerial vehicles. *3GPP, Sophia Antipolis, France, Rep. TR*, 36, 2017.

[64] David W Matolak and Ruoyu Sun. Air-ground channel characterization for unmanned aircraft systems: The near-urban environment. In *2015 IEEE Military Communications Conference (MILCOM)*, pages 1656–1660. IEEE, 2015.

[65] Sriram Nandha Premnath, Suman Jana, Jessica Croft, Prarthana Lakshmane Gowda, Mike Clark, Sneha Kumar Kasera, Neal Patwari, and Srikanth V Krishnamurthy. Secret key extraction from wireless signal strength in real environments. *IEEE Transactions on Mobile Computing*, 12(5):917–930, 2012.

[66] Christopher Huth, René Guillaume, Thomas Strohm, Paul Duplys, Irin Ann Samuel, and Tim Güneysu. Information reconciliation schemes in physical-layer security: A survey. *Computer Networks*, 109:84–104, 2016.

[67] Junqing Zhang, Trung Q Duong, Alan Marshall, and Roger Woods. Key generation from wireless channels: A review. *IEEE Access*, 4:614–626, 2016.

[68] Guyue Li, Aiqun Hu, Junqing Zhang, Linning Peng, Chen Sun, and Daming Cao. High-agreement uncorrelated secret key generation based on principal component analysis preprocessing. *IEEE Transactions on Communications*, 66(7):3022–3034, 2018.

[69] Chan Dai Truyen Thai, Jemin Lee, Jay Prakash, and Tony QS Quek. Secret group-key generation at physical layer for multi-antenna mesh topology. *IEEE Transactions on Information Forensics and Security*, 14(1):18–33, 2018.

[70] Weitao Xu, Sanjay Jha, and Wen Hu. LoRa-key: Secure key generation system for LoRa-based network. *IEEE Internet of Things Journal*, 6(4):6404–6416, 2018.

[71] Nasser Aldaghri and Hessam Mahdavifar. Physical layer secret key generation in static environments. *IEEE Transactions on Information Forensics and Security*, 15:2692–2705, 2020.

[72] Dengke Guo, Kuo Cao, Jun Xiong, Dongtang Ma, and Haitao Zhao. A lightweight key generation scheme for the internet of things. *IEEE Internet of Things Journal*, 8(15):12137–12149, 2021.

[73] Guyue Li, Yinghao Xu, Wei Xu, Eduard Jorswieck, and Aiqun Hu. Robust key generation with hardware mismatch for secure MIMO communications. *IEEE Transactions on Information Forensics and Security*, 16:5264–5278, 2021.

[74] Robert Wilson, David Tse, and Robert A Scholtz. Channel identification: Secret sharing using reciprocity in ultrawideband channels. *IEEE Transactions on Information Forensics and Security*, 2(3):364–375, 2007.

[75] Akbar Sayeed and Adrian Perrig. Secure wireless communications: Secret keys through multipath. In *2008 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3013–3016. IEEE, 2008.

[76] Neal Patwari, Jessica Croft, Suman Jana, and Sneha K Kasera. High-rate uncorrelated bit extraction for shared secret key generation from channel measurements. *IEEE Transactions on Mobile Computing*, 9(1):17–30, 2009.

[77] Lei Huang, Dengke Guo, Jun Xiong, and Dongtang Ma. An improved CQA quantization algorithm for physical layer secret key extraction. In *2020 International Conference on Wireless Communications and Signal Processing (WCSP)*, pages 829–834. IEEE, 2020.

[78] Guyue Li, Chen Sun, Eduard A Jorswieck, Junqing Zhang, Aiqun Hu, and You Chen. Sum secret key rate maximization for TDD multi-user massive MIMO wireless networks. *IEEE Transactions on Information Forensics and Security*, 16:968–982, 2020.

[79] Vishal M. Patel, Nalini K. Ratha, and Rama Chellappa. Cancelable biometrics: A review. *IEEE Signal Processing Magazine*, 32(5):54–65, 2015.

[80] Holger Boche, Rafael F Schaefer, Sebastian Baur, and H Vincent Poor. On the algorithmic computability of the secret key and authentication capacity under channel, storage, and privacy leakage constraints. *IEEE Transactions on Signal Processing*, 67(17):4636–4648, 2019.

[81] Shuo Wu and Yingbo Hua. Total secrecy from anti-eavesdropping channel estimation. *IEEE Transactions on Signal Processing*, 70:1088–1103, 2022.

[82] Andrew Rukhin, Juan Soto, James Nechvatal, Miles Smid, Elaine Barker, Stefan Leigh, Mark Levenson, Mark Vangel, David Banks, Alan Heckert, et al. *A statistical test suite for random and pseudorandom number generators for cryptographic applications*, volume 22. US Department of Commerce, Technology Administration, National Institute of Standards and Technology, 2001.

[83] Carlo Tomasi. Estimating gaussian mixture densities with EM: A tutorial. *Duke University*, pages 1–8, 2004.

[84] Christopher M Bishop. *Neural networks for pattern recognition*. Oxford University Press, 1995.

[85] Matthieu Bloch. *Physical-layer security*. Georgia Institute of Technology, 2008.

[86] Thomas M Cover. *Elements of information theory*. John Wiley & Sons, 1999.

[87] Guyue Li, Haiyu Yang, Junqing Zhang, Hongbo Liu, and Aiqun Hu. Fast and secure key generation with channel obfuscation in slowly varying environments. In *2022 IEEE Conference on Computer Communications (INFOCOM)*, pages 1–10. IEEE, 2022.

[88] Antonia M Tulino, Sergio Verdú, et al. Random matrix theory and wireless communications. *Foundations and Trends® in Communications and Information Theory*, 1(1):1–182, 2004.

[89] Ahmed Maksud and Yingbo Hua. Second-order analysis of secret-key capacity from a MIMO channel. In *2023 IEEE Military Communications Conference (MILCOM)*, pages 833–838. IEEE, 2023.

[90] Masahito Hayashi and Ángeles Vázquez-Castro. Two-way physical layer security protocol for gaussian channels. *IEEE Transactions on Communications*, 68(5):3068–3078, 2020.

[91] Yingbo Hua and Ahmed Maksud. Secret-key capacity from MIMO channel probing. *IEEE Wireless Communications Letters*, 2024.

[92] Yingbo Hua, Md Saydur Rahman, and Ananthram Swami. A method for low-latency secure multiple access. In *2024 IEEE International Symposium on Local and Metropolitan Area Networks (LANMAN)*, 2024.

[93] Yingbo Hua and Md Saydur Rahman. Unification of secret key generation and wiretap channel transmission. *2022 IEEE International Conference on Communications (ICC)*, 2024.

[94] Haji M Furqan, Jehad M Hamamreh, and Huseyin Arslan. Secret key generation using channel quantization with SVD for reciprocal MIMO channels. In *2016 International Symposium on Wireless Communication Systems (ISWCS)*, pages 597–602. IEEE, 2016.

[95] Kaidi Xu, Ming-Min Zhao, Yunlong Cai, and Lajos Hanzo. Low-complexity joint power allocation and trajectory design for UAV-enabled secure communications with power splitting. *IEEE Transactions on Communications*, 69(3):1896–1911, 2020.

[96] Edward Gerjuoy. Shor's factoring algorithm and modern cryptography. an illustration of the capabilities inherent in quantum computers. *American Journal of Physics*, 73(6):521–540, 2005.

[97] Lei Chen, Qiping Zhu, Weixiao Meng, and Yingbo Hua. Fast power allocation for secure communication with full-duplex radio. *IEEE Transactions on Signal Processing*, 65(14):3846–3861, 2017.

[98] Ashish Khisti and Gregory W Wornell. Secure transmission with multiple antennas—part II: The MIMOME wiretap channel. *IEEE Transactions on Information Theory*, 56(11):5515–5532, 2010.

[99] Hui-Ming Wang, Tongxing Zheng, and Xiang-Gen Xia. Secure MISO wiretap channels with multiantenna passive eavesdropper: Artificial noise vs. artificial fast fading. *IEEE Transactions on Wireless Communications*, 14(1):94–106, 2014.

[100] Tsung-Hui Chang, Wei-Cheng Chiang, Y-W Peter Hong, and Chong-Yung Chi. Training sequence design for discriminatory channel estimation in wireless MIMO systems. *IEEE Transactions on Signal Processing*, 58(12):6223–6237, 2010.

[101] Rohit Negi and Satashu Goel. Secret communication using artificial noise. In *2005 IEEE Vehicular Technology Conference*, volume 62, page 1906. Citeseer, 2005.

[102] Amitav Mukherjee, S Ali A Fakoorian, Jing Huang, and A Lee Swindlehurst. Principles of physical layer security in multiuser wireless networks: A survey. *IEEE Communications Surveys & Tutorials*, 16(3):1550–1573, 2014.

[103] Nan Yang, Lifeng Wang, Giovanni Geraci, Maged Elkashlan, Jinhong Yuan, and Marco Di Renzo. Safeguarding 5G wireless communication networks using physical layer security. *IEEE Communications Magazine*, 53(4):20–27, 2015.

[104] Yulong Zou, Jia Zhu, Xianbin Wang, and Lajos Hanzo. A survey on wireless security: Technical challenges, recent advances, and future trends. *Proceedings of the IEEE*, 104(9):1727–1765, 2016.

[105] Haiyang Zhang and Lingjie Duan. Beyond secrecy rate in miso wiretap channels: An information jamming approach. *IEEE Transactions on Communications*, 68(5):3057–3067, 2020.

[106] Thomas R Dean and Andrea J Goldsmith. Physical-layer cryptography through massive MIMO. *IEEE Transactions on Information Theory*, 63(8):5419–5436, 2017.

[107] Robb J Muirhead. *Aspects of multivariate statistical theory*. John Wiley & Sons, 2009.

[108] Hans Delfs, Helmut Knebl, and Helmut Knebl. *Introduction to cryptography*, volume 2. Springer, 2002.