REPORT DOCUMENTATION PAGE

1

Form Approved OMB NO. 0704-0188

1. REPORT DATE (DD-MM-YYYY)	2. REPORT TYPE	3. DATES COVERED (From - To)
12-09-2014	Ph.D. Dissertation	-

4. TITLE AND SUBTITLE
Adversaries in Networks

5a. CONTRACT NUMBER
W911NF-10-1-0419

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER
611102

6. AUTHORS
Oliver Kosut

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAMES AND ADDRESSES
Cornell University
373 Pine Tree Road

Ithaca, NY 14850 -2820

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS (ES)
U.S. Army Research Office
P.O. Box 12211
Research Triangle Park, NC 27709-2211

10. SPONSOR/MONITOR'S ACRONYM(S)
ARO

11. SPONSOR/MONITOR'S REPORT NUMBER(S)
58094-NS.16

12. DISTRIBUTION AVAILIBILITY STATEMENT
Approved for public release; distribution is unlimited.

13. SUPPLEMENTARY NOTES
The views, opinions and/or findings contained in this report are those of the author(s) and should not contrued as an official Department of the Army position, policy or decision, unless so designated by other documentation.

14. ABSTRACT
As systems become more distributed, they are vulnerable to new forms of attack. An adversary could seize control of several nodes in a network and reprogram them, unbeknownst to the rest of the network. Strategies are needed that can ensure robust performance in the presence of these sorts of attacks. This thesis studies the adversarial problem in three scenarios.

15. SUBJECT TERMS
network security, network coding, information theory

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	15. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			Lang Tong
UU	UU	UU	UU		19b. TELEPHONE NUMBER 607-255-3900

Report Title

Adversaries in Networks

ABSTRACT

As systems become more distributed, they are vulnerable to new forms of attack. An adversary could seize control of several nodes in a network and reprogram them, unbeknownst to the rest of the network. Strategies are needed that can ensure robust performance in the presence of these sorts of attacks. This thesis studies the adversarial problem in three scenarios.

First is the problem of network coding, in which a source seeks to send data to a destination through a network of intermediate nodes that may perform arbitrarily complicated coding functions. When an adversary controls nodes in the network, achievable rates and upper bounds on capacity are found, and Polytope Codes are introduced, which are a nonlinear class of codes specially designed to handle adversaries in a network coding framework.

Second, multiterminal source coding is studied, in which several nodes make correlated measurements, independently encode them, and transmit their encodings to a common decoder, which attempts to recover some information. Two special cases of this problem are studied when several of the nodes may be controlled by an adversary: the problem of Slepian and Wolf, in which the decoder attempts to perfectly decode all measurements, and the CEO Problem, in which the decoder attempts to estimate a source correlated with the measurements.

Finally, adversarial attacks are studied against power system sensing and estimation. In this problem, a control center receives various measurements from meters in a power grid, and attempts to recover information about the state of the system. Attacks of various degrees of severity are studied, as well as countermeasures that the control center may employ to prevent these attacks.

ADVERSARIES IN NETWORKS

A Dissertation

Presented to the Faculty of the Graduate School

of Cornell University

in Partial Fulfillment of the Requirements for the Degree of

Doctor of Philosophy

by

Oliver Eli Kosut

August 2010

ADVERSARIES IN NETWORKS

Oliver Eli Kosut, Ph.D.

Cornell University 2010

As systems become more distributed, they are vulnerable to new forms of attack. An adversary could seize control of several nodes in a network and reprogram them, unbeknownst to the rest of the network. Strategies are needed that can ensure robust performance in the presence of these sorts of attacks. This thesis studies the adversarial problem in three scenarios.

First is the problem of network coding, in which a source seeks to send data to a destination through a network of intermediate nodes that may perform arbitrarily complicated coding functions. When an adversary controls nodes in the network, achievable rates and upper bounds on capacity are found, and Polytope Codes are introduced, which are a nonlinear class of codes specially designed to handle adversaries in a network coding framework.

Second, multiterminal source coding is studied, in which several nodes make correlated measurements, independently encode them, and transmit their encodings to a common decoder, which attempts to recover some information. Two special cases of this problem are studied when several of the nodes may be controlled by an adversary: the problem of Slepian and Wolf, in which the decoder attempts to perfectly decode all measurements, and the CEO Problem, in which the decoder attempts to estimate a source correlated with the measurements.

Finally, adversarial attacks are studied against power system sensing and estimation. In this problem, a control center receives various measurements from meters in a power grid, and attempts to recover information about the state of the

system. Attacks of various degrees of severity are studied, as well as countermeasures that the control center may employ to prevent these attacks.

BIOGRAPHICAL SKETCH

Oliver Kosut was born in California in 1982. He grew up near San Francisco, then did his undergraduate work at MIT. He graduated in 2004 with bachelor's degrees in Electrical Engineering and Mathematics. He came to Cornell as a PhD student in August 2005, and joined the Adaptive Communications and Signal Processing (ACSP) group under Prof. Lang Tong in April 2006. He spent the 2008–2009 academic year visiting UC Berkeley when Prof. Tong took his sabbatical there, where he had the privilege of working with Prof. David Tse. Oliver enjoys juggling, puzzles, sailing, change-ringing, procrastinating, and long walks on the beach.

For the best parents in the world, who happen to be mine

and for Sergio

ACKNOWLEDGEMENTS

Above all, I would like to thank my advisor, Lang Tong, without whom this thesis would consist of exactly zero words. He was never unwilling to discuss a problem, always encouraging and helpful in suggesting new directions, and constantly happy to indulge me as I plunged into the tall grass of information theory and random walks in five dimensions. He was also the one who, at one of our first meetings in May 2006, first spoke the magic word "Byzantine," and, when I asked him what exactly he meant by that, the even more magic words "I don't know."

I also want to thank the other members of my thesis committee. Rick Johnson, for being a constant guide on the adventure that is grad school and academia. Aaron Wagner, for knowing everything there is to know about information theory. And Robert Kleinberg, for being interested and encouraging about my research every time I spoke to him.

I thank David Tse, without whom Chapter 2 would definitely not exist. His perspectives and suggestions were always spot-on and enlightening, and he greatly helped me in being clear and logical in my thinking, writing, and speaking. He was without doubt worth traveling 3000 miles to work with.

I thank Robert Thomas, for not laughing at us when we thought we could say something about power grids. Chapter 5 would not be the same without his constantly helpful input. Liyan Jia also contributed significantly to the work in Chapter 5.

I thank other Cornell professors who taught me in classes: Toby Berger, Steve Strogatz, Sergio Servetto, Salman Avestimehr, Mark Psiaki, Mark Campbell, Eugene Dynkin, and David Hammer. I love to learn, and they taught well. Also Scott Coldren, without whom the department would consist mostly of students and professors running madly around the engineering quad, each yelling at the top

of their lungs about something technical, hoping that someone might crash into them and hear what they have to say.

I thank teachers I have had in the past, who first showed me that I might like some of this stuff, especially Bruce Cohen, Mary Laycock, Lori Lambertson, Manny Dominguez, Tom Murray, Steve Leeb, and George Verghese.

I thank other members and former members of my group, for discussions, help with problems, and for arguing with me at group meetings: Parv Venkitasubramaniam, Ting He, Anima Anandkumar, Saswat Misra, Stefan Geirhofer, Abhishek Nadamani, Matt Ezovski, John Sun, Andrey Turovsky, Jinsub Kim, Brandon Jones, Gael Hatchue, Ameya Agaskar, Aaron Lei, Amine Laourine, Guilherme Pinto, Shiyao Chen, Meng Wang, Liyan Jia, and Pouya Tehrani.

I thank other friends of mine in Ithaca, for allowing me to retain some modicum of a life, and thereby keeping me from going insane: Ben Kelly, Mikhail Lisovich, Igors Gorbovickis, Ania Kacewicz, Nithin Michael, Frank Ciaramello, Caroline Andrews, Laura Fegeley, Amir Sadovnik, Katherine Lai, Natalia Buitrago, Alisa Blinova, George Khachatryan, Anna Tarakanova, Ilan Shomorony, Alireza Vahid, Ebad Ahmed, Saifur Rahman, and Yucel Altug.

I thank friends of mine who live outside of Ithaca, for reminding me that the world extends beyond its borders, and that I might one day escape from them: Mira and James Whiting, Andrew Thomas, Kate Baker, Joel Corbo, Allen Rabinovich, Jennifer Tu, Katherine Reid, Laura Boylan, Lanya da Silva, Nick Martin, Jen Selby, Bobak Nazer, Krish Eswaran, Cat Miller, John Danaher, Jim Wnorowski, Stephanie Fried, Dianne Cermak, and many many others. If I didn't list you, it's not because I don't love you; the orchestra is playing me off.

I thank Molly Peeples, for accompanying me on a bit of this life thing.

I thank my brother Alexei, for telling me that there was a monster in our closet, and then showing me that there wasn't. Also my extended family, for being there for me: of you there are too many to list.

Finally, I would like to thank my parents Robert and Cynthia. My father for buying me toys, and teaching me everything I know. My mother for always believing in me, and teaching me everything my father didn't.

This work is supported in part by the National Science Foundation under Award CCF-0635070, the U. S. Army Research Laboratory under the Collaborative Technology Alliance Program DAAD19-01-2-0011, and TRUST (The Team for Research in Ubiquitous Secure Technology) sponsored by the National Science Foundation under award CCF-0424422.

This work is supported in part by the National Science Foundation under Award CCF-0635070 and the Army Research Office under Grant AROW911NF- 06-1-0346.

This work is supported in part by the National Science Foundation under Award CCF-0728872.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 Motivation and Overview

Increasingly, we are surrounded by distributed systems comprised of many nodes interacting with one another. From the internet to cell phones to sensor networks, everything is made up of many small pieces. This trend creates new security problems, which require new methods to build systems that are robust against various forms of attack. In this thesis, we consider one potential form of an attack against a network, that of a malicious adversary entering the network, seizing and controlling a group of nodes, unbeknownst to the rest of the network. We study this scenario in several contexts, analyzing the impact of the adversary, and designing strategies to counteract its presence. In particular, we focus on two problems from information theory: multiterminal source coding and network coding, in addition to a problem in power system sensing and estimation.

There are several applications in communication networks in which one user may wish to relay data through other nodes toward a second user, when those relay nodes may not be reliable or trustworthy. Consider, for example, a wireless ad hoc network. In such a network, nodes may enter and exit the network often, and messages need to be transmitted through. Nodes need to learn about each other, establish communication paths, and update them as the network changes. It is easy to imagine that a node could enter the network without any intention of following the agreed-upon protocol. It could at first appear to act honestly, so as to establish itself as a relay point in the network, but then it could forward messages incorrectly, jam its neighbor's signals, or eavesdrop on others' communication.

Even in the wired setting, nodes may be vulnerable to malicious reprogramming. Internet routers can be hacked into and compromised, or simply fail and transmit unreliable information. These concerns motivate our study of network coding in the presence of adversarial nodes.

Network coding is a concept in network information theory that allows nodes in a network to perform potentially elaborate operations to transmit data through a network. In Chapter 2, we study this problem with adversarial nodes. We give upper bounds on communication rates, and present a class of nonlinear codes called Polytope Codes, which is the first class of codes capable of achieving capacity for a general class of networks with adversarial nodes. In particular, we show that these codes achieve capacity for a certain class of planar networks.

Now consider a sensor network. This could involve a large number of cheap nodes gathering data to be collected by a central receiver that acts as a fusion center, organizing and analyzing the aggregate information. Should some of the sensors be seized by an adversary, the fusion center should use strategies to make its decisions robust against these attacks. The topology of a sensor network made up of many nodes communicating directly to a single fusion center is exactly that of multiterminal source coding, which is our second major area of study. We are mostly interested in the tradeoff between the adversary's impact on the quality of the information collected at the fusion center (referred to as the decoder in the sequel), and the amount of data is transmitted from the sensors to the fusion center. We study two main subcases of the multiterminal source coding problem with adversarial nodes: in Chapter 3, the problem of Slepian and Wolf, in which the decoder attempts to recover all data that was available at the sensors; and in Chapter 4, the CEO Problem, in which the decoder estimates a quantity observed

by each sensor through a noisy channel. For both these problems, we give achievable schemes and outer bounds on the sets of achievable communication rates. In some cases, these bounds match.

Finally, we consider the power system. The power grid in this country—and most others—serves to deliver reliable electricity to millions of homes and offices, and its continued operation is vital part of the infrastructure of our society. Therefore, any potential vulnerabilities are a serious concern. The system itself is a vast network of generators, transmission lines, transformers, and switches. It is important for the continuing operation of the grid that operators in control centers have reliable up-do-date information about the current state of the system. To this end, numerous meters are deployed throughout the grid, measuring voltage and/or power flow. These meters report their findings back to control centers, who use the gathered data to make decisions. If an adversary were able to manipulate the meter readings sent to the control center, then it could potentially influence the trajectory of the power state, and even cause blackouts. In Chapter 5, we present some results that allow us to identify vulnerable parts of the power system to these attacks, and detection strategies to find them if they occur.

1.2 Byzantine Attack

The notion of an adversary controlling a subset of nodes in a network, unbeknownst to the other nodes, is sometimes known as *Byzantine attack*. The term *Byzantine* is conspicuous, and deserves a moment's explanation. According to Greek legend, in the 7th century BC lived King Byzas, who in 667 BC founded the city of Byzantium on the shores of the Bosphorus Strait connecting the Mediterranean

sea to the Black sea, the location of present day Istanbul. Byzantium kept its name and became a chief city of the Roman Empire, which by the 4th century AD had become so large and difficult to govern that it began to fracture between east and west. In 330 AD, the emperor Constantine I moved the capital of the eastern part to Byzantium, and renamed the city Constantinople. In its day, it was usually called the Eastern Roman Empire, or simply the Roman Empire, since it survived for almost a millennium longer than the western half. However, partly out of confusion, and partly out of a desire to differentiate it from the earlier and unified Roman Empire, by the nineteenth century the eastern empire came to be known by historians as the Byzantine Empire, even though the empire came into being at the very moment that Byzantium was renamed.

Long after the empire collapsed after Constantinople fell to the Ottomans in 1453, the Byzantine Empire became known for being excessively beaurocratic and decadent. Hence the word "Byzantine" came to mean overly complicated, hard to understand, or unnecessarily obtuse. In its entry on the word "byzantine", the Oxford English Dictionary sites the 1937 book *Spanish Testament* by Arthur Koestler as an early written example of this use of the word. He wrote "In the old days people often smiled at the Byzantine structure of the Spanish army" [1]. By the latter half of the 20th century, this meaning of the word was common.

In 1980, Marshall Pease, Leslie Lamport, and Robert Shostak wrote [2], titled "Reaching Agreement in the Presence of Faults," which was based partially on earlier work by Lamport and others [3, 4]. Two years later, the same authors wrote [5], in which they renamed the same problem "the Byzantine Generals' Problem." This version of the problem is described follows. A number of generals of the Byzantine army are separately encamped outside an enemy city. They must

come to an agreement about whether to attack the city. They do this by sending messengers from one to another, indicating each general's opinion or preference for their course of action. This is complicated by the fact that some of the generals are *traitors*; that is, they may send inconsistent or meaningless messages, and therefore make it more difficult for the honest generals to reach agreement. The result of [2] and [5] is that consensus among the honest generals can be reached as long as fewer than one third of the generals are traitors.

On his website, Lamport describes the process leading to the more colorful naming of the problem:

> I have long felt that, because it was posed as a cute problem about philosophers seated around a table, Dijkstra's dining philosopher's problem received much more attention than it deserves. (For example, it has probably received more attention in the theory community than the readers/writers problem, which illustrates the same principles and has much more practical importance.) I believed that the problem introduced in [2] was very important and deserved the attention of computer scientists. The popularity of the dining philosophers problem taught me that the best way to attract attention to a problem is to present it in terms of a story.
>
> There is a problem in distributed computing that is sometimes called the Chinese Generals Problem, in which two generals have to come to a common agreement on whether to attack or retreat, but can communicate only by sending messengers who might never arrive. I stole the idea of the generals and posed the problem in terms of a group of generals, some of whom may be traitors, who have to reach

> a common decision. I wanted to assign the generals a nationality that would not offend any readers. At the time, Albania was a completely closed society, and I felt it unlikely that there would be any Albanians around to object, so the original title of this paper was The Albanian Generals Problem. Jack Goldberg was smart enough to realize that there were Albanians in the world outside Albania, and Albania might not always be a black hole, so he suggested that I find another name. The obviously more appropriate Byzantine generals then occurred to me.

When he says "obviously more appropriate," he is evidently referring to the fact that "Byzantine" can describe the generals in two ways: first, it is their nationality; second, some of their actions are undoubtedly byzantine.

A critical component of the problem description in the original Byzantine Generals' Problem is that the traitors may send arbitrary messages to other generals, and the honest generals must reach agreement no matter what the traitors do. This notion of robust performance in the face of arbitrary behavior is at the heart of Byzantine attack, and at the heart of the adversary model for the work in this thesis.

An important distinction should be made between two possible interpretations of this sort of model. The interpretation originally intended by [2, 5] is that of errors; that is, the generals represent identical units which should in principle produce the same, unless one suffers from a random fault. If a system is designed to be robust against Byzantine failures, then it will always come to the correct decision even if the faulty unit behaves in an arbitrary manner. The second interpretation, and the one we mostly use in this thesis, is that of a true adversary: an intelligent

entity motivated to defeat the aims of the network if it can, one that will study the network operation and search for a vulnerability. These two interpretations are usually mathematically equivalent, but our choice of the second one does motivate some choices we make in our modeling of the problem. For example, in our work on network coding, discussed in Chapter 2, we adopt a model in which the adversary controls nodes in the network. As we will discuss our network coding literature review in Section 1.3.1, this differs from some earlier work on adversarial attacks in network coding. In particular, [6, 7] studied the problem of an adversary controlling links in the network, as opposed to nodes. They seem to be using the first interpretation of Byzantine attack, such that adversarial actions represent errors on communication channels between nodes, and as long as the number of these errors is small, no matter what each error is, they can guarantee performance. Our view, instead, is that the attacks represent an adversary taking control of nodes in a network, and therefore able to alter any transmission made by those nodes. This leads to a mathematically different problem, and, it turns out, a harder one.

Another important element in studying Byzantine and adversarial attacks has to do with placing a limit the adversary's power. The problem should be designed so that successful strategies are robust against attacks of a certain size. Obviously, if the adversary controls the entire network, then no strategy could ever defeat it. Therefore, we allow the adversary to perform arbitrary actions, but subject to being able to control only a certain number of nodes in the network. The honest users of network cannot know for certain that the number of nodes to come under the adversary's control will not exceed the threshold, but it they can at least make a performance guarantee if it does not exceed the threshold. Should the adversary size exceed the threshold, performance could degrade. We can therefore think of the limit on adversary size not as a priori knowledge of the power of the adversary,

but rather as a parameter with which we can trade off robustness to attacks with performance. As we will see, handling more adversaries requires more redundancy in the system, which means performance decreases.

1.3 Network Coding

1.3.1 Related Work

A classical problem in graph theory is the maximal flow problem. That is, given a directed graph composed of nodes and capacity-limited edges, we wish to find the flow of maximum size from a source to a sink. A flow is given by a quantity associated with each edge, representing the amount of some commodity flowing through that edge, and upper bounded by the edge capacity. Flow must be conserved at each node, except for the source, which produces the commodity, and the sink, which consumes it. In 1956, Ford and Fulkerson [8] showed that the flow maximizing the amount of the commodity that travels from the source to the destination is given by the minimum cut of the graph. This is known as the max-flow min-cut theorem. By a *cut*, we mean a way to split the network into two parts, such that the source is in one part and the sink in the other. The value of a cut is given by the total capacity of all edges from the part with the source to the part with the sink. The *min-cut* is the minimum cut value over all cuts separating the source from the sink.

Even though the problem studied by Ford and Fulkerson was purely mathematical in nature—the commodity is an abstract notion, and is often imagined to be, for example, water flowing through pipes—the result can immediately be applied

to communicating in a network. Nodes in the graph represent machines able to receive and transmit messages along communication links, which are represented by edges. The edge capacities represent communication limits of the communication links. A flow through the graph can be converted into a routing strategy, whereby the numbers of data packets received and transmitted by intermediate nodes is given by the flow. In this setup, nodes in the network do nothing except copy received information to their outgoing communication links.

The classical max-flow min-cut result cannot be applied to the problem of multicast: the case that a single source wishes to transmit the same message to more than one destination. Here the "water as information" metaphor breaks down, because data packets, unlike water, can be duplicated, so a node with an incoming bit stream can reproduce it one several outgoing links. More significantly, data can be combined in nontrivial ways. In particular, more intelligent intermediate nodes can do *coding*: in principle, a node's output can be an arbitrary function of its input. In the landmark paper [9], it was found that if this so-called *network coding* is allowed, then for multicast, the min-cut can be achieved to each destination simultaneously.

In the last decade, network coding has become one of the pillars of network information theory. While the achievability proof used in [9] relied on a random coding argument over arbitrary coding functions, it was shown in [10] that for multicast it is sufficient to use only linear codes: that is, the values transmitted on each link are elements taken from a finite field, and each node takes linear combinations over that field of its input to produce its output. In [11], an algebraic framework for network coding was presented, which led to a necessary and sufficient conditions for the success of a linear code in a general setting, as well as

polynomial time encoding and decoding. The idea of *random linear network coding* was first suggested in [12] and elaborated in [13]. In this approach, linear coding is performed with the coefficients chosen randomly; with high probability, the result is a good network code that can achieve the network capacity for multicast. Random linear coding does not require an outside authority which knows the complete network topology in order to design good codes; instead, nodes may work in a more distributed manner without losing any communication rate. A polynomial time algorithm for finding good linear network codes was given in [14]. Network coding for practical use has been studied and/or demonstrated in [15, 16, 17, 18].

While linear coding is sufficient to achieve capacity for multicast, for some problems with multiple sources, linear codes are insufficient. It was shown in [19] that standard network coding problems fall into three categories: (1) coding is unnecessary, and routing is enough to achieve capacity; (2) linear coding is sufficient, and optimal linear codes can be found in polynomial time, and (3) determining whether a linear code can achieve a given communication rate is NP-hard. They also gave an example of a network in the third category for which a nonlinear code can outperform any linear code. It was pointed out in [20] that even this code is not far from linear, and [20] introduced the class of *vector linear* codes, whereby several elements from a finite field can be transmitted on each link. However, [21] provided an example for which even these codes are insufficient for general multi-source multi-destination problems. The work in this thesis on network coding with adversaries shows that even for the single-source single-destination problem, nonlinear network coding is required to achieve capacity. This indicates that the general adversary problem may differ substantially from the standard network coding problem.

Another branch of study on network coding involves the so-called entropic region. For n correlated random variables, one may calculate the joint entropy using Shannon's entropy measure for any subset of the variables. There are 2^{n-1} non-trivial subsets, so any set of variables can be associated with a 2^{n-1} dimensional vector. Any vector for which there exists such a set of random variables is called entropic. The closure of the set of all entropic vectors is often written $\bar{\Gamma}^*$. Any linear bound on $\bar{\Gamma}^*$ is known as an information inequality. The framework of the entropic region and information inequalities was introduced in [22]. The positivity of conditional entropy and conditional mutual information compose a set of information inequalities known as the Shannon type inequalities. It was first shown in [23] that there exist non-Shannon type inequalities: that is, $\bar{\Gamma}^*$ is strictly smaller than the set of vectors satisfying the Shannon type inequalities. It can be shown that any network coding problem can be expressed in terms of $\bar{\Gamma}^*$; if $\bar{\Gamma}^*$ were completely known, then all network coding problems would be immediately solved. Moreover, it was shown in [24] that non-Shannon type inequalities can be relevant in network coding problems. This indicates that the general network coding problem is identical to that of characterizing $\bar{\Gamma}^*$. In [25], it was shown that $\bar{\Gamma}^*$ is identical to the set of group characterizable vectors derived from subgroups of a finite group. Therefore, so-called coset codes based on finite groups can in principle solve any network coding problem. Linear codes are special cases of these codes. An interesting property of our Polytope Codes used to defeat adversaries, discussed in Chapter 2, is that they do not appear to be special cases of coset codes.

The first consideration of network coding with security concerns was [26], which considered the problem of an eavesdropper able to overhear the messages sent on a fixed number of communication links in a network. This was based partially on

the foundational work on information-theoretic security by Shannon [27] as well as Wyner's wiretap channel [28]. In [26], it is shown that when the eavesdropper's capabilities are always identical, linear codes are sufficient to achieve the highest possible communication rate without allowing the eavesdropping to learn anything about the message. The same problem but with communication links of differing capacity was studied in [29]. In this setup, the eavesdropper has varying power depending on which links it is able to overhear, and [29] finds that many standard linear coding techniques fail, and one must be more careful in designing the code so as to maximize secure communication rate. This is a different sort of adversary to the ones we consider, but it is a similar finding, in that when the adversary has different levels of power depending on where it is in the network, the problem becomes harder.

Adversarial attacks on network coding were first considered in [30], which looked at detecting adversaries in a random linear coding environment. The first major work on correcting adversaries in network coding was [6, 7]. This two-part paper looked at the multicast network coding problem in which the adversary controls exactly z unit-capacity links in the network. This was introduced as "network error correction", and, as mentioned above, considered the errors as channel failures rather than adversarial actions. In [31], the same problem is studied, providing distributed and low complexity coding algorithms to achieve the same asymptotically optimal rates. In addition, [31] looks at two adversary models slightly different from the omniscient one considered in [6, 7] and in this thesis. They show that higher rates can be achieved under these alternate models. In our study of multiterminal source coding, we explore similar ways of slightly reducing the power of the adversary, but for the rest of this thesis, we always assume the worst case adversary that is completely omniscient. In [32], a more general view

of the adversary problem is given, whereby the network itself is abstracted into an arbitrary linear transformation.

These works seek to correct for the adversarial errors at the destination. An alternative strategy known as the watchdog, studied for wireless network coding in [33], is for nodes to police downstream nodes by overhearing their messages to detect modifications. In [34], a similar approach is taken, and they found that nonlinear operations similar to ours can be helpful, just as we do.

The work presented in Chapter 2 on an adversary able to control a fixed number of nodes in a network rather than a fixed number of edges has previously appeared in [35, 36]. Simultaneously with this work, a slightly different adversarial network coding problem was considered in [37, 38]. In these papers, the adversary controls a fixed number of edges, as in [6, 7], but the edges may unequal capacity. They find that this problem also requires nonlinear coding to achieve capacity. It seems that linear coding is sufficient when the adversary has uniform power, no matter where it is—as in the unit-capacity edge problem—but when its power can vary, such as the node problem or the unequal-edge problem, nonlinear coding may be required.

1.3.2 Contributions

Our primary contribution is a class of network codes to defeat adversaries called Polytope Codes. These were originally introduced in [35] under the less descriptive term "bounded-linear codes". Polytope Codes are nonlinear codes, and they improve over linear codes by allowing error detection inside the network. This allows adversaries to be more easily identified, whereby the messages they send can be

ignored. We also prove a cut-set upper bound on achievable rates in networks with node-based adversaries. This cut-set bound is a form of the Singleton bound [39], originally proved for classical error-correcting codes. We show that for a class of planar networks, Polytope Codes can achieve the rate given by this cut-set bound, which means that they achieve the capacity for these networks. We also show that the cut-set bound is not always achievable, by giving an example network with a strictly smaller capacity.

We briefly describe the high-level idea behind Polytope Codes, because the same idea is at the heart of our achievable results for multiterminal source coding. It is easy to grasp and it comprises the majority of this thesis, so we momentarily dwell on it. Consider three nodes in a network, which we name Xander, Yvaine, and Zoe for convenience. Let X and Y be two correlated random variables with joint distribution $p(x, y)$. Suppose Xander and Yvaine observe X and Y respectively, and both independently report their observation to Zoe. One or both of them may be a traitor; i.e. taking instructions from an adversary, so their transmissions to Zoe could be incorrect. From her received information, Zoe can estimate the empirical joint distribution of X and Y, which we denote $q(x, y)$. Since one of Xander and Yvaine may not be trustworthy, $q(x, y)$ could differ from the true empirical distribution. However, if both Xander and Yvaine were honest, then Zoe can expect $q(x, y)$ to be close—or exactly equal to—$p(x, y)$. Therefore, if q is not close to p, then Zoe can conclude that one of her friends must be lying. Note that Zoe may not be able to tell which person has done so, but now both Xander and Yvaine are suspect, which means that if Zoe can gather information from other nodes, those nodes might be more reliable, assuming the adversary has influence over a limited number of nodes. Consider the situation also from the adversary's perspective. If Xander is a traitor, he has two choices in what he tells

Zoe. He could report a value for X that will cause q to be close to p, or not. If the former, then he is constrained in his choice for what he tells Zoe, which means he has reduced ability to cause damage. If the latter, he partially gives away his position. The key in designing strategies to defeat adversaries is to allow checks to be made, like the one Zoe made by comparing q to p. The more checks, the more rock-or-hard-place decisions the adversary must make, thereby diminishing its influence.

The main building block of the Polytope Code is special probability distributions over polytopes in real vector fields. These distribution produce random variables like X and Y that are sent through the network. Their empirical distributions are compared at internal nodes in the network, just as Zoe does. This allows for error detection inside the network. The special polytope structure over the real vector field allows for the internal comparisons to be particular effective, in a way that would not occur with probability distributions over a finite field.

1.4 Multiterminal Source Coding

1.4.1 Related Work

Multiterminal source coding was introduced by Slepian and Wolf in [40]. They considered the situation that two separate encoders observe correlated random variables, and each independently transmit encoded versions of their observations to a common decoder, which attempts to recover the sources exactly, with small probability of error. They found the remarkable result that the sum-rate—the total communication rate from both encoders to the decoder—can be made as small as

if a single encoder could observe both sources simultaneously and compress them jointly. A proof of the same result of the same result was given in [41]. This paper used the technique of *random binning*, whereby a random ensemble of codes is created by placing each possible observed source sequence into bins uniformly at random. Given a particular binning, the encoding process consists simply of transmitting to the encoder the index of the bin containing the observed source sequence. With high probability, the codes created by this process are good.

The notion of source coding with side information was studied in [42, 43]. These papers considered the same setup as the Slepian-Wolf problem, except now the decoder is interested only in recovering one of the two sources. The other source and the associated encoder provides only so-called side information, since it is used only to help recover the target source. The description of the achievable rate region for this problem required the use of an auxiliary random variable, which represents a quantized or degraded version of the side information.

A similar problem in the rate-distortion framework was studied in [44]. Here, the decoder has complete side information (i.e. uncoded), and wishes to recover a target source, but it may accept some degradation in its source estimate, as long as the estimate satisfies a distortion constraint. The solution of this problem gives the trade-off between communication rate from the encoder, and distortion of the source estimate produced at the decoder.

All the above problems involved at most two sources, but achievable strategies used to solve them naturally generalize to many sources, many encoders, and many distortion constraints. This general achievable scheme is sometimes known as the Berger-Tung achievable scheme [45, 46]. Another common term for it—and perhaps more descriptive—as quantize-and-bin. The idea is that each encoder

quantizes its measured source, in a manner prescribed by an auxiliary random variable along the lines of [42, 43]; then, the encoders use random binning, exactly following the proof of the Slepian-Wolf result in [41]. The achievable rate-distortion region given by this strategy has a very intuitive form, but unfortunately it is not always optimal for multiterminal source coding problems. In [47], Korner and Marton provide a surprisingly simple example for which an achievable strategy strictly better than Berger-Tung exists. Despite considerable effort, the most general form of the problem remains unsolved, even for two sources. Still, a steadily growing number of special cases have been solved.

One such special case of the multiterminal source coding problem which additional structure is known as the CEO Problem. It was introduced for discrete memoryless sources in [48]. In the CEO Problem (so-named because the decoder represents a company's CEO that has supposedly dispatched his or her employees as encoders to gather data and report back), the decoder is interested in recovering a single source with some distortion, but this source is not directly observed by any encoder. Instead, the encoders observe noisy versions of the source, such that the noise for each encoder is conditionally independent given the source. This conditional independence structure of the sources comprises a clean structure that appears to make the problem more tractable. In [48], it was found that with a large number of encoders each observing the source through the same noisy channel, the distortion of the estimate found at the decoder falls exponentially fast with the sum-rate from all the sources. Moreover, they exactly characterize the optimal error exponent. Again, the achievable strategy used is Berger-Tung.

A significant sub-class of multiterminal source coding is the quadratic Gaussian setup. Here, sources are Gaussian and distortion constraints are quadratic. These

assumptions tend to make problems more tractable and allow the use of powerful tools, such as the entropy power inequality, which was originally stated by Shannon [49] and proved in [50, 51]. For example, the complete rate-distortion region for the two-terminal source coding problem in the quadratic Gaussian setup was found in [52].

The quadratic Gaussian CEO Problem was introduced in [53]. In a result along the lines of that of [48] it is shown that with many encoders measuring a noisy version of the source with identical noise variance, the achievable distortion falls asymptotically with the sum-rate like K/R, where R is the sum rate and K is a constant depending only on the source characteristics. Moreover, they exactly characterize K. The exact rate-distortion function for finite sum-rate was found in [54]. The rate-distortion region for a finite number of sources and nonidentical encoder measurements was discovered simultaneously in [55] and [56]. All these results again use only the Berger-Tung strategy to prove achievability. The converse arguments make heavy use of the entropy power inequality, and follow the essential argument first proposed in [54], which is also based partially on [57].

There is a modest amount of work in the literature on source coding under adversarial attack. Perhaps the closest commonly-studied relative is the multiple descriptions problem, introduced with early work in [58, 59, 60]. The problem here is that two encoders observe a single source. They must each independently transmit encoded versions to a common decoder. However, the transmissions may fail to arrive, so they should be designed so that each one leads to a quality estimate, but if both arrive, an even better estimate can be produced. This problem has elements of the idea of an attack on source coding: each encoder need to be designed for the possibility that other encoders may fail. A significant general

achievability result was given in [61]. This strategy has been shown to be optimal in the case that there is no excess rate [62] and the quadratic Gaussian case [57]. More recently, [63] studied so-called robust distributed source coding. The problem there was somewhat closer to ours: it is in effect a combination of the multiple descriptions problem and the CEO problem. Nodes observe noisy versions of the source, and must encode these sources in such a way that the more arrive, the better the decoder's estimate.

Prior versions of the work presented in Chapter 3 and 4 of this thesis has appeared in [64, 65, 66, 67, 68].

1.4.2 Contributions

In Chapter 3 we consider the Slepian-Wolf problem, and in Chapter 4 the CEO problem, both under adversarial attack. For the Slepian-Wolf problem—wherein the decoder seeks to exactly recover all sources with small probability of error—we exactly characterize the achievable rate regions for three setups:

1. A variable-rate model, in which the decoder can in real-time allocate transmission rate to the encoders. Here, we place a guarantee on the sum-rate that will be achieved, but cannot promise exactly how this rate is allocated, because it depends on the actions of the adversary.
2. A randomized fixed-rate model, in which the rate for each encoder is fixed beforehand, but the encoders have private randomness that is hidden from the adversary.
3. A deterministic fixed-rate model, in which the encoders do not have private randomness. This is the most pessimistic model, but therefore the most

robust against powerful adversaries. Moreover, this model most closely corresponds to the model used in other chapters of this thesis.

For all these models, we allow a very general model of the information known to the adversary. In particular, we assume the adversary has access to the output of an arbitrary noisy channel, which takes as input the sources observed by the encoders. This model allows for an adversary that knows nothing, an adversary that knows everything, or any in between. We also allow for a very general view of what the decoder knows about which nodes the adversary may control as well as what information the adversary has access to.

Our achievable strategies for the Slepian-Wolf problem are generalizations of the random binning approach of [41]. The variable-rate achievable strategy for the first setup is the most substantially different, in that it involves numerous small messages being sent between encoders and the decoder. After each message the decoder chooses which encoder to hear from next, thereby allocating rate in real time.

One peculiarity about the Slepian-Wolf problem in the presence of an adversary is that it is not reasonable to expect the decoder to recover all the sources exactly, as we can without an adversary. This is because an adversarial node may simply choose not to transmit any useful information about its associated source. Moreover, it may not be possible for the decoder to learn exactly which nodes are the traitors. We therefore require only that the estimates produced by the decoder are accurate only for honest nodes, even if it does not know which ones those are. This allows the decoder to place a guarantee on the number of correct estimates that it produces, but it means that the estimates are arguably not useful without post-processing.

This inherent difficulty with the Slepian-Wolf problem motivates our study of the CEO problem in chapter 4. The advantage of this problem is that no single node has a monopoly on any information about the target source, so we can guarantee quality of the source estimate at the decoder in all cases. We study this problem in both the discrete memoryless case, for which we generalize the results of [48], as well the quadratic Gaussian case, for which we generalize the results of [55, 56]. For the discrete memoryless problem, we present upper and lower bounds on the sum-rate error exponent for many encoders with statistically identical observations. For the quadratic Gaussian problem, we present inner and outer bounds on the rate-distortion region for a finite number of encoders with nonuniform measurements.

For the CEO problem, we focus only on the most pessimistic model, corresponding to the deterministic fixed-rate model discussed above for the Slepian-Wolf problem, and assuming the adversary is omniscient. Our achievable results are derived from a unified achievable scheme for both the discrete memoryless and quadratic Gaussian problems. Our achievable scheme for the adversarial problem is a generalization of the non-adversarial Berger-Tung strategy, and can be applied to a similarly general form of the problem. Our outer bounds are based on a specific type of attack by the adversary, which can be viewed as a form of the Singleton bound [39].

1.5 Power System Sensing and Estimation

1.5.1 Related Work

Power system state estimation was introduced by Schweppe, Wildes, and Rom in [69]. State estimation took as input measurements of power flows taken in the power system and produced an estimate of the voltages and phases on all busses in the system. Ever since this first introduction of state estimation, it has been necessary to deal with bad data. Traditionally, bad data were assumed to be caused by random errors resulting from a fault in a meter and/or its attendant communication system. These errors are modeled by a change of variance in Gaussian noise, which leads to an energy (l_2) detector (see [70, 71, 72, 73, 74]). Another classical detector as the so-called largest normalized residue (LNR) detector [69, 70], which has the form of a test on the l_∞ norm of the normalized measurement residual.

Observability is an important consideration when measuring the system state. A system is observable only if there are enough meters so that there is no bus whose voltage could change without having an effect on some meter. The problem to determine whether the system is observable has been studied in [75, 76]. In [77], a purely topological condition for observability was given.

Recently, Liu, Ning, and Reiter studied the problem that several meters are seized by an adversary that is able to corrupt the measurements from those meters as received by the control center [78]. This differs from previous investigations of the problem in that the false data at various meters can be simultaneously crafted by the adversary to defeat the state estimator, as opposed to independent errors caused by random faults. It is observed in [78] that there exist cooperative and

malicious attacks on meters that all known bad data techniques will fail to detect. The authors of [78] gave a method to adjust measurements at just a few meters in the grid in such a way that bad data detector will fail to perceive the corruption of the data.

Another recent work that is similar to ours is by Gorinevsky, Boyd, and Poll [79]. They study attempt to find a small number of faults in a power system by formulating a convex problem that is likely to lead to a sparse solution. Their work is partially inspired by the recent development of compressed sensing and l_1 minimization techniques [80]. In their problem, the desired sparsity has to do with the small number of faults they expect in the problem. In our work on adversarial attacks, we expect a small number of adversaries in the network; therefore, a similar approach is applicable.

Prior versions of our work on power system sensing in the presence of adversaries have appeared in [81, 82, 83].

1.5.2 Contributions

In Chapter 5, we present several results extending the work of [78]. We note that the observation made therein can be made even stronger: if an adversary has the ability to adjust the measurements from enough meters, then no algorithm at the control center will ever be able to detect that an adjustment has been made. This can be viewed as a fundamental limit on the ability of the classical formulation of state estimation to handle cooperative attacks. We also show that there is a close relationship between the attacks described in [78] and system observability. For this reason, we refer to the attacks of [78] as *unobservable* attacks. This

relationship allows us to extend the topological results of [77] to give an efficient algorithm to calculate attacks of this nature require a small number of adversarial meters. Our algorithm is based on the special structure of the power system, and makes use of techniques to efficiently minimize submodular functions [84, 85, 86]. Our algorithms allows an operator of a power system to find the places in which it is most vulnerable to these attacks.

Unobservable attacks may be executed by the adversary only if it controls enough meters. We also study the problem in the regime that it is not able to perform this attack. Here, we develop a heuristic that allows us to find attacks that minimize the energy of the measurement residual, and therefore are likely to cause the most damage. We also present a decision theoretic formulation in which the control center attempts detect malicious data injections by an adversary. The adversary has the freedom to choose which meters it takes control of, and what sort of attack it performs; therefore, this detection problem cannot be formulated as a simple hypothesis test, and the uniformly most powerful test may not exist. We study the generalized likelihood ratio test (GLRT) for this problem. The GLRT is not optimal in general, but it is known to perform well in practice and its performance has shown to be close to optimal when the detector has access to a large number of data samples [87, 88, 89]. We also find that when there is only a single meter controlled by the adversary, the GLRT is identical to the LNR detector [70], which provides some theoretical underpinning to this already-in-use test.

For large systems and possible many adversaries, it is not feasibly to implement the exact GLRT. Instead, we study a convex relaxation based on the l_1 norm, which is likely to produce sparse solutions. We perform numerical simulations that

demonstrate that the GLRT and its convex relaxation both outperform traditional detectors.

1.6 Organization

Chapter 2: We introduce node-based adversarial attacks on network coding. We give our upper bound on achievable rates. We proceed to introduce Polytope Codes through several examples, culminating in the general theory and the fundamental properties. Then we prove that Polytope Codes achieve the capacity for a class of planar networks. Finally, we provide an example with capacity strictly less than the cut-set bound.

Chapter 3: We study the Slepian-Wolf problem with adversarial nodes. We present our model, then give a simple example illustrating it and our basic technique. We go on to find the exact achievable rate region for the three cases described above: variable rate, randomized fixed-rate, and deterministic fixed-rate.

Chapter 4: We investigate the CEO problem under adversarial attack, for both the discrete memoryless case and the quadratic Gaussian case. We present our unified achievable Berger-Tung-like achievable scheme. We apply it to calculate bounds on the achievable error exponent for the discrete memoryless case and the rate region for the quadratic Gaussian case. Then we find outer bounds for both cases.

Chapter 5: We present our work on power system sensing and estimation, in the presence of malicious attacks on meters. We describe unobservable attacks, and prove the relationship between them and system observability. We go on to

use this to find an efficient algorithm to find these attacks, and show that it is able to find optimal attacks. We present a Bayesian formulation of the problem, which we argue has some advantages as compared with the traditional model. We give a decision theoretic framework for the problem, and find the find the generalized likelihood ratio test that results from it. We perform some numerical simulations on various detectors for these problem, including the GLRT and its convex relaxation.

Chapter 6: We offer some concluding remarks and thoughts on future directions.

CHAPTER 2

NODE-BASED ATTACKS ON NETWORK CODING AND POLYTOPE CODES

2.1 Introduction

This chapter studies network coding in a network with one source and one destination when any s nodes may be controlled by an adversary. These node-based attacks differ from the edge-based attacks first considered in [6, 7]. There, the adversary can control any z unit-capacity links. In [6, 7], it is shown that the capacity with z adversarial links is exactly $2z$ less than the min-cut of the network, which is the capacity with no adversary present. The precise result is quoted in Sec. 2.4.

Defeating node-based attacks is fundamentally different from defeating edge-based attacks. First, the edge problem does not immediately solve the node problem. Consider, for example, the Cockroach network, shown in Fig. 2.1. Suppose we wish to handle any single adversarial node in the network. One simple approach would be to apply to edge result from [6, 7]: no node controls more than two unit-capacity edges, so we can defeat the node-based attack by using a code that can handle an attack on any two edges. However, note that the achievable rate for this network without an adversary is 4, so subtracting twice the number of bad edges leaves us with an achievable rate of 0. As we will show, the actual capacity of the Cockroach network with one traitor node is 2. Relaxing the node attack problem to the edge attack problem is too pessimistic, and we can do better if we treat the node problem differently.

Node-based attacks and edge-based attacks differ in an even more significant way. When the adversary can control any set of z unit-capacity edges, it is clear that it should always take over the edges on the minimum cut of the network. However, if the adversary can control any set of s nodes, it is not so obvious: one node may have many more output edges than another, so depending on which nodes the adversary takes over, it may control various numbers of edges. It may face a choice between a node directly on the min-cut, but with few output edges, and a node away from the min-cut, but with many output edges. For example, in the Cockroach network, node 4 has only one output edge, but it is on the min-cut (which is between nodes $S, 1, 2, 3, 4, 5$ and D); node 1 has two output edges, so it is apparently more powerful, but it is also one step removed from the min-cut, and therefore its ability to influence the destination may be limited. This uncertainty about where a network is most vulnerable seems to make the problem hard. Indeed, we find that linear network coding techniques fail to achieve capacity, so we resort to nonlinear codes, and in particular Polytope Codes, to be described. We further discuss the relationship between the edge problem and the node problem in Sec. 2.3, in which we show that the edge problem is subsumed by the node problem.

Many achievability results in network coding have been proved using linear codes over a finite field. In this chapter we demonstrate that linear codes are insufficient for this problem. Moreover, we develop a class of codes called Polytope Codes, originally introduced in [35] under the less descriptive term "bounded-linear codes". Polytope codes are used to prove that a cut-set bound, stated and proved in Sec. 2.4, is tight for a certain class of networks. Polytope Codes differ from linear codes in three ways:

1. *Comparisons:* A significant tool we use to defeat the adversary is for internal

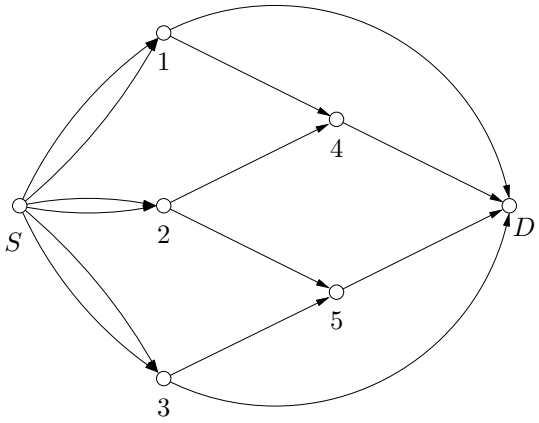


Figure 2.1: The Cockroach Network. All edges have capacity 1. With a single traitor node, the capacity is 2, but no linear code can achieve a rate higher than 4/3. A proof of the linear capacity is given in Sec. 2.13. A capacity-achieving linear code supplemented by nonlinear comparisons is given in Sec. 2.6, and a capacity-achieving Polytope Code is given in Sec. 2.8.

nodes in the network to perform comparisons: they check whether their received data could have occurred if all nodes had been honest. If not, then a traitor must have altered one of the received values, in which case it can be localized. The result of the comparison, a bit representing whether or not it succeeded, can be transmitted downstream through the network. The destination receives these comparison bits and uses them to determine who may be the traitors, and how to decode. These comparison operations are nonlinear, and, as we will demonstrate in Sec. 2.6, incorporating them into a standard finite-field linear code can increase achieved rate. However, even a code composed of a linear code supplemented by these nonlinear comparison operations is insufficient to achieve capacity for some networks; Polytope Codes also incorporate comparisons, but of a more sophisticated variety.

2. *Joint Type Codebooks via Probability Distributions:* Unlike usual linear network codes, Polytope Codes make use of probability distributions. In many ways they are more like random codes, such as those used in the standard

proof of Shannon's channel coding theorem, but they differ from these as well. Each Polytope Code is governed by a joint probability distribution on a set of random variables, one for each edge in the network. Given the distribution, codewords are selected to be sequences with joint type exactly equal to the distribution. Contrast this with randomly generated codewords, which would, with high probability, have joint type close to the base distribution. Here we use an entirely deterministic process to generate the codebook: we simply list all sequences with type equal to the given distribution, and associate each one with a message. The advantage of this method of code construction is that an internal node will know exactly what joint type to expect of its received sequences, because it knows the original distribution. The comparisons discussed above consist of checking whether the observed joint type matches the expected distribution. If it does not, then the adversary must have influenced one of the received sequences, so it can be localized.

3. *Distributions over Polytopes:* The final difference between classical error control codes and Polytope Codes—and the one for which the latter are named—comes from the nature of the probability distributions discussed above. These distributions are uniform over the set of integer lattice points on polytopes in real vector fields. This choice for distribution provides two useful properties. First, the entropy vector for these distributions can be easily calculated merely from properties of the linear space in which the polytope sits. In this sense, they share characteristics with finite-field linear codes. In fact, a Polytope Code can almost always be used in place of a linear code. The second useful property has to do with how the comparisons inside the network are used. The distributions over polytopes are such that if enough comparisons succeed, the adversary is forced to act as an honest node and transmit cor-

rect information. This property will be elaborated in examples in Sec. 2.7 and Sec. 2.8, as well as stated in its most general form in Sec. 2.9.

Our main result, that the cut-set bound can be achieved using Polytope Codes for a class of planar networks, is stated in Sec. 2.5. Planarity requires that the graph can be embedded in a plane such that intersections between edges occur only at nodes. This ensures that enough opportunities for comparisons are available, allowing the code to more well defeat adversarialy attacks. Before proving the result in Sec. 2.10, we develop the theory of Polytope Codes through several examples in Sec. 2.6, 2.7, 2.8; we also discuss some general properties of Polytope Codes in Sec. 2.9.

In Sec. 2.11–2.13, we provide some additional comments on this problem. Sec. 2.11 shows that the cut-set bound is not always tight, by giving an example with a tighter bound. Sec. 2.12 includes a tighter version of the cut-set bound than that stated in Sec. 2.4, along with an illustrating example of the need for a more general bound. Sec. 2.13 provides a proof that linear codes are insufficient for the Cockroach network.

2.2 Problem Formulation

Let (V, E) be an directed acyclic graph. We assume all edges are unit-capacity, and there may be more than one edge connected the same pair of nodes. One node in V is denoted S, the source, and one is denoted D, the destination. We wish to determine the maximum achievable throughput from S to D when any set of s nodes in $V \setminus \{S, D\}$ are *traitors*; i.e. they are controlled by the adversary. Given a rate R and a block-length n, the message W is chosen at random from the set

$\{1, \ldots, 2^{nR}\}$. Because each edge is unit capacity, it holds a value $X_e \in \{1, \ldots, 2^n\}$.

A code is be made up of three components:

1. an encoding function at the source, which produces values to place on all the output edges given the message,
2. a coding function at each internal node $i \in V \setminus \{S, D\}$, which produces values to place on all output edges from i given the values on all input edges to i,
3. and a decoding function at the destination, which produces an estimate $\hat{W}$ of the message given the values on all input edges.

Suppose $T \subseteq V \setminus \{S, D\}$ with $|T| = s$ is the set of traitors. They may subvert the coding functions at nodes $i \in T$ by placing arbitrary values on all the output edges from these nodes. Let Z_T be the set of values on these edges. For a particular code, specifying the message W as well as Z_T determines exactly the values on all edges in the network, in addition to the destination's estimate $\hat{W}$. We say that a rate R is *achievable* if there exists a code operating at that rate with some block-length n such that for all messages, all sets of traitors T, and all values of Z_T, $W = \hat{W}$. That is, the destination always decodes correctly no matter what the adversary does. Let the *capacity* C be the supremum over all achievable rates.

2.3 Node Problem vs. Edge Problem

The first major work on network coding in the presence of adversaries, [6, 7], studied the problem in which a fixed number of unit-capacity edges are controlled by the adversary. A more general form of the problem, in which the adversary

controls a fixed number of edges of possibly differing capacities, was studied in [37, 38]. We argue in this section that even the latter problem is subsumed by the node problem studied in this chapter. In fact, we prove a somewhat stronger fact, that the node problem is equivalent to what we call the *limited-node* problem.

The limited-node problem is a generalization of the node problem, in which a special subset of nodes are designated as potential traitors, and the code must only guard against adversarial control of any s of those nodes. Certainly the limited-node problem subsumes the all-node problem, since we may simply take the set of potential traitors to be all nodes. Furthermore, it subsumes the unequal-edge problem studied in [37, 38], because given an instance of the unequal-edge problem, an equivalent all-node problem can be constructed as follows: create a new network with every edge replaced by a pair of edges of equal capacity with a node between them. Then limit the traitors to be only these interior nodes.

We now show that the all-node problem actually subsumes the limited-node problem, and therefore also the unequal-edge problem. In Sec. 2.11, we construct an instance of the limited-node problem for which the cut-set bound is not tight. Because of the equivalence of these two problem shown in this section, this indicates that for even the all-node problem, the cut-set bound is not tight in general.

Let (V, E) be a network under a limited-node adversarial attack, where there may be at most s traitors constrained to be in $U \subseteq V$, and let C be its capacity. We construct a sequence of all-node problems, such that finding the capacity of these problems is enough to find that of the original limited-node problem. Let $(V^{(M)}, E^{(M)})$ be a network as follows. First make M copies of (V, E). That is, for each $i \in V$, put $i^{(1)}, \ldots, i^{(M)}$ into $V^{(M)}$, and for each edge in E, create M copies of it connected the equivalent nodes, each with the same capacity. Then,

for each $i \in U$, merge $i^{(1)}, \ldots, i^{(M)}$ into a single node i^*, transferring all edges that were previously connected to any of $i^{(1)}, \ldots, i^{(M)}$ to i^*. Let $C^{(M)}$ be the all-node capacity of $(V^{(M)}, E^{(M)})$ with s traitors. For large M, this network will be such that for any $i \notin U$, a traitor taking over one of the respective nodes is almost useless because it commands such a small fraction of the information flow through the network. That is, we may almost assume that the traitors will only ever be nodes in U. This is stated explicitly in the following theorem.

Theorem 1 *For any M, $C^{(M)}$ is related to C by*

$$\frac{1}{M}C^{(M)} \leq C \leq \frac{1}{M-2s}C^{(M)}. \tag{2.1}$$

Moreover,

$$C = \lim_{M\to\infty} \frac{1}{M}C^{(M)} \tag{2.2}$$

and if $C^{(M)}$ can be computed to arbitrary precision for any M in finite time, then so can C.

Proof: We first show that $\frac{1}{M}C^{(M)} \leq C$. Take any code on $(V^{(M)}, E^{(M)})$ achieving rate R when any s nodes may be traitors. We use this to construct a code on (V, E), achieving rate R/M when any s nodes in U may be traitors. We do this by first increasing the block-length by a factor of M, but maintaining the same number of messages, thereby reducing the achieved rate by a factor of M. Now, since each edge in (V, E) corresponds to M edges in $(V^{(M)}, E^{(M)})$, we may place every value transmitted on an edge in the $(V^{(M)}, E^{(M)})$ code to be transmitted on the equivalent edge in the (V, E) code. That is, all functions executed by $i^{(1)}, \ldots, i^{(M)}$ are now executed by i. The original code could certainly handle any s traitor nodes in U. Hence the new code can handle any s nodes in U, since the actions performed

by these nodes have not changed from $(V^{(M)}, E^{(M)})$ to (V, E). Therefore, the new code on (V, E) achieving rate R/M for the limited-node problem.

Now we show that $C \leq \frac{1}{M-2s} C^{(M)}$. Take any code on (V, E) achieving rate R. We will construct a code on $(V^{(M)}, E^{(M)})$ achieving rate $(M - 2s)R$. This direction is slightly more difficult because the new code needs to handle a greater variety of traitors. The code on $(V^{(M)}, E^{(M)})$ is composed of an outer code and M copies of the (V, E) code running in parallel. The outer code is a $(M, M - 2s)$ MDS code with coded output values $w_1, \ldots, w_M$. These values form the messages for the inner codes. Since we use an MDS code, if $w_1, \ldots, w_M$ are reconstructed at the destination such that no more than s are corrupted, the errors can be entirely corrected. The jth copy of the (V, E) code is performed by i^* for $i \in U$, and by $i^{(j)}$ for $i \notin U$. That is, nodes in U are each involved in all M copies of the code, while nodes not in U are involved in only one. Because the (V, E) code is assumed to defeat any attack on only nodes in U, if for some j, no nodes $i^{(j)}$ for $i \notin U$ are traitors, then the message w_j will be recovered correctly at the destination. Therefore, one of the w_j could be corrupted only if $i^{(j)}$ is a traitor for some $i \notin U$. Since there are at most s traitors, at most of the $w_1, \ldots, w_M$ will be corrupted, so the outer code corrects the errors.

From (2.1), (2.2) is immediate. We can easily identify M large enough to compute C to any desired precision.

□

2.4 Cut-Set Upper Bound

It is shown in [6, 7] that, if an adversary controls z unit-capacity edges, the network coding capacity reduces by $2z$. This is a special case of a more general principle: an adversary-controlled part of the network does twice as much damage in rate as it would if that part of the network were merely removed. In particular, the following theorem, proved in [6, 7], gives the capacity for multicast and an adversary controlling z unit-capacity edges:

Theorem 2 (Theorem 4 in [6] and Theorem 4 in [7]) *In a multicast problem with source S and destinations $D_1, \ldots, D_K$, the network coding capacity with an adversary capable of controlling any z unit-capacity edges is*

$$C = \min_k \operatorname{mincut}(S; D_k) - 2z. \tag{2.3}$$

Moreover, the capacity can be achieved using linear codes.

The doubling effect seen in (2.3) is for the same reason that, in a classical error correction code, the Hamming distance between codewords must be at least twice the number of errors that can be corrected; this is the Singleton bound [39]. We now give a cut-set upper bound for node-based adversaries in network coding that makes this explicit.

A *cut* in a network is a subset of nodes $A \subseteq V$ containing the source but not the destination. The cut-set upper bound on network coding without adversaries is the sum of the capacities of all forward-facing edges [9]; that is, edges (i, j) with $i \in A$ and $j \notin A$. All backward edges are ignored.

In the adversarial problem, backward edges are more of a concern. This is because the argument relies on values on certain edges crossing the cut being un-

affected by changes in the values on other edges crossing the cut. This is not guaranteed in the presence of a backwards edge. We give an example of the complication in Sec. 2.12. To avoid the issue, we state here Theorem 3, a simplified cut-set bound that applies only to cuts without backward edges. This bound will be enough to prove our main result, stated in Sec. 2.5, giving the capacity of a certain class of networks, but for the general problem Theorem 3 can be tightened. We expand on the issue of backwards edges, and state a tighter version of the cut-set bound in Sec. 2.12. Unlike the problem without adversaries, we see that there is not necessarily a single cut-set bound. Some more elaborate cut-set bounds are found in [37, 38]. This paper studies the unequal-edge problem, but the bounds can be readily applied to the node problem. It was originally conjectured in [37] that even the best cut-set bound is not tight in general. In Sec. 2.11, we demonstrate that there can be an active upper bound fundamentally unlike a cut-set bound. The example used to demonstrate this, though it is a node adversary problem, can be easily modified to confirm the conjecture stated in [37].

Theorem 3 *Consider a cut $A \subseteq V$ with the source S in A, the destination D not in A, and with no backward edges; that is, there is no edge $(i, j) \in E$ with $i \notin A$ and $j \in A$. If there are s traitor nodes, then for any set $T \subset V$ with $|T| = 2s$, the following upper bound holds on the capacity of the network:*

$$C \leq |\{(i, j) \in E : i \in A \setminus T,\ j \notin A\}|. \tag{2.4}$$

Proof: Divide T into two disjoint sets T_1 and T_2 with $|T_1| = |T_2| = s$. Let E_1 and E_2 be the sets of edges out of nodes in T_1 and T_2 respectively that cross the cut; that is, edges (i, j) with $i \in A \cap T_1$ or $i \in A \cap T_2$, and $j \notin A$. Let $\bar{E}$ be the set of all edges crossing the cut not out of nodes in T_1 or T_2. Observe that the

upper bound in (2.4) is precisely the total number of edges in $\bar{E}$. Since there are no backwards edges for the cut A, the values on edges in $\bar{E}$ are not functions of the values on edges in E_1 or E_2. In particular, if the adversary alters a value on an edge in E_1 or E_2, it cannot change the values in $\bar{E}$.

Suppose (2.4) does not hold. If so, there would exist a code with block-length n achieving a rate R higher than the right hand side of (2.4). For any set of edges $F \subseteq E$, for this code, we can define a function

$$X_F : 2^{nR} \to \prod_{e \in F} 2^n \tag{2.5}$$

such that for a message w, assuming all nodes act honestly, the values on edges in F is given by $X_F(w)$. Since R is greater than $|\bar{E}|$, there exist two messages w_1 and w_2 such that $X_{\bar{E}}(w_1) = X_{\bar{E}}(w_2)$.

We demonstrate that it is possible for the adversary to confuse the message w_1 with w_2. Suppose w_1 were the true message, and the traitors are T_1. The traitors replace the messages going along edges in E_1 with $X_{E_1}(w_2)$. If there are edges out of nodes in T_1 that are not in E_1—i.e. they do not cross the cut—the traitors do not alter the messages on these edges from what would be sent if they were honest. Thus, the values sent along edges in $\bar{E}$ is given by $X_{\bar{E}}(w_1)$. Now suppose w_2 were the true message, and the traitors are T_2. They now replace the messages going along edges in E_2 with $X_{E_2}(w_1)$, again leaving all other edges alone, meaning that the values on $\bar{E}$ are $X_{\bar{E}}(w_2) = X_{\bar{E}}(w_1)$. Note that in both these cases, the values on E_1 are $X_{E_1}(w_2)$, the values on E_2 are $X_{E_2}(w_1)$, and the values on $\bar{E}$ are $X_{\bar{E}}(w_1)$. This comprises all edges crossing the cut, so the destination receives the same values under each case; therefore it cannot differentiate w_1 from w_2. □

We illustrate the use of Theorem 3 on the Cockroach network, reproduced in Fig. 2.2, with a single adversary node. To apply the bound, we choose a cut A

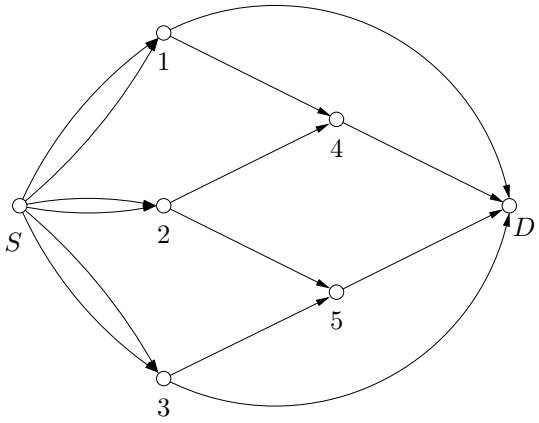


Figure 2.2: The Cockroach Network. All edges have capacity 1. With one traitor, the cut-set bound of Theorem 3 gives an upper bound on capacity of 2 by setting $A = \{S, 1, 2, 3\}$ and $T = \{1, 2\}$.

and a set T with $|T| = 2s = 2$, since we consider a single traitor node. Take $A = \{S, 1, 2, 3, 4, 5\}$, and $T = \{1, 4\}$. Four edges cross the cut, but the only ones not out of nodes T are $(3, D)$ and $(5, D)$, so we may apply Theorem 3 to give an upper bound on capacity of 2. Alternatively, we could take $A = \{S, 1, 2, 3\}$ and $T = \{1, 2\}$, to give again an upper bound of 2. Note that there are 6 edges crossing this second cut, even though the cut-set bound is the same. It is not hard to see that 2 is the smallest upper bound given by Theorem 3 for the capacity of the Cockroach network. In fact, rate 2 is achievable, as will be shown in Sec. 2.6 using a linear code supplemented by comparison operations, and again in Sec. 2.8 using a Polytope Code.

2.5 Capacity of A Class of Planar Networks

Theorem 4 *Let (V, E) be a network with the following properties:*

1. *It is planar.*

2. No node other than the source has more than two unit-capacity output edges.

3. No node other than the source has more output edges than input edges.

If $s = 1$, the cut-set bound given by Theorem 3 is tight for this network.

Polytope Codes are used to prove achievability for this theorem. The complete proof is given in Sec. 2.10, but first we develop the theory of Polytope Codes by means of several examples in Sec. 2.6–2.8 and general properties in Sec. 2.9.

Perhaps the most interesting condition in the statement of Theorem 4 is the planarity condition. Recall that a graph is said to be *embedded* in a surface (usually a two-dimensional manifold) when it is drawn in this surface so that no two edges intersect. A graph is *planar* if it can be embedded in the plane [90].

2.6 A Linear Code with Comparisons for the Cockroach Network

The Cockroach network satisfies the conditions of Theorem 4. Fig 2.1 shows a plane embedding with both S and D on the exterior, and the second condition is easily seen to be satisfied. Therefore, since the smallest cut-set bound given by Theorem 3 for a single traitor node is 2, as we have discussed, Theorem 4 claims that the capacity of the Cockroach network is 2. In this section, we present a capacity-achieving code for the Cockroach network that is a linear code over a finite-field supplemented by nonlinear comparisons. This illustrates the usefulness of comparisons in defeating adversaries against network coding. Before doing so, we

provide an intuitive argument that linear codes are insufficient. A more technical proof that the linear capacity is in fact 4/3 is given in Sec. 2.13.

Is it possible to construct a linear code achieving rate 2 for the Cockroach network? We know from the Singleton bound-type argument—the argument at the heart of the proof of Theorem 3—that, in order to defeat a single traitor node, if we take out everything controlled by two nodes, the destination must be able to decode from whatever remains. Suppose we take out nodes 2 and 3. These nodes certainly control the values on $(5, D)$ and $(3, D)$, so if we hope to achieve rate 2, the values on $(1, D)$ and $(4, D)$ must be uncorruptable by nodes 2 and 3. Edge $(1, D)$ is not a problem, but consider $(4, D)$. With a linear code, the value on this edge is a linear combination of the values on $(1, 4)$ and $(2, 4)$. In order to keep the value on $(4, D)$ uncorruptable by node 2, the coefficient used to construct the value on $(4, D)$ from $(2, 4)$ must be zero. In other words, the value on $(1, 4)$ should be merely forwarded to $(4, D)$. By a symmetric argument removing nodes 1 and 2, the value on $(3, 5)$ should be forwarded to $(5, D)$. But now we can remove nodes 1 and 3, and control everything received by the destination. Therefore no linear code can successfully achieve rate 2.

This argument does not rigorously show that the linear capacity is less than 2, because it shows only that a linear code cannot achieve exactly rate 2, but it does not bound the achievable rate with a linear code away from 2. However, it is meant to be an intuitive explanation for the limitations of linear codes for this problem, as compared with the successful nonlinear codes that we will subsequently present. The complete proof that the linear capacity is 4/3 is given in Sec. 2.13.

We now introduce a nonlinear code to achieve the capacity of 2. We work in the finite field of p elements. Let the message w be a $2k$-length vector split into

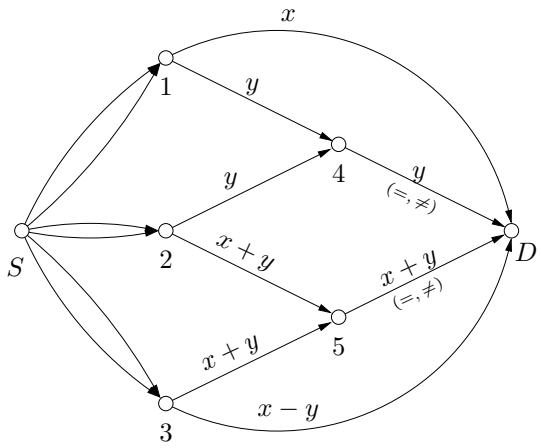


Figure 2.3: A nonlinear code for the Cockroach Network achieving the capacity of 2.

two k-length vectors x and y. We will use a block length large enough to place one of $2p^k$ values on each link. In particular, enough to place on a link some linear combination of x and y plus one additional bit. For large enough k, this extra bit becomes insignificant, so we still achieve a rate of 2.

The scheme is shown in Figure 2.3. Node 4 receives the vector y from both 1 and 2. It forwards one of these copies to D (it does not matter which). In addition, it performs a nonlinear comparison between the two received copies of y, resulting an one additional bit comprised of one of the special symbols $=$ or $\neq$. If the two received copies of y agree, it forwards $=$, otherwise it sends $\neq$. The link $(4, D)$ can accommodate this, since it may have up to $2p^k$ messages placed on it. Node 5 does the same with its two copies of the vector $x + y$.

The destination's decoding strategy depends on which of the two comparison bits sent from nodes 4 and 5 are $=$ or $\neq$, as follows:

- If the bit from node 4 is $\neq$ but the bit from 5 is $=$, then the traitor must be either node 1, 2, or 4. In any case, the vector $x - y$ received from node 3 is certainly trustworthy. However, $x + y$ is trustworthy as well, because even if

node 2 is the traitor, its transmission must have matched whatever was sent by node 3, because if not node 5 would have transmitted $\neq$. Since it did not, the destination can trust both $x+y$ and $x-y$, from which it can decode the message $w = (x, y)$.

- If the message from 5 is $\neq$ but the message from 4 is $=$, then we are in the symmetric situation and can reliably decode w from x and y.
- If both the messages from 4 and 5 are $\neq$, then the traitor must be node 2, in which case x and $x - y$ are trustworthy, so the destination can decode w.
- If both messages are $=$, then the destination cannot eliminate any node as a possible traitor. However, at most one of $x, y, x + y, x - y$ can have been corrupted by the traitor, because no node controls more than one of the vectors received at the destination. For instance, if node 1 is the traitor, it may choose whatever it wants for x, and the destination would never know. However, node 1 cannot impact the value of y without inducing a $\neq$, because its transmission to node 4 is verified against that from node 2. Similarly, node 3 controls $x - y$ but not $x + y$. Nodes 4 and 5 control only y and $x + y$ respectively. Node 2 controls nothing, because both y and $x + y$ are checked against other transmissions. Therefore, if the destination can find three of $x, y, x + y, x - y$ that all agree on the message w, then this message must be the truth because only one of them could be corrupted, and w can be decoded from the other two. Conversely, there must be a group of three of $x, y, x+y, x+2y$ that agree, because at most one has been corrupted. Hence, the destination can always decode w.

Even though our general proof of Theorem 4 uses a Polytope Code, which differs significantly from this one, the manner in which the comparisons comes into play is

essentially the same. The key insight is to consider the code from the perspective of the traitor. Suppose it is node 1, and consider the choice of what value for y to send along edge $(1, 4)$. If it sends a false value for y, then the comparison at node 4 will fail, which will lead the destination to consider the upper part of the network suspect, and thereby ignore all values influenced by node 1. The only other choice for node 1 is to cause the comparison at node 4 to succeed; but this requires sending the true value of y, which means it has no hope to corrupt the decoding process. This is the general principle that makes our codes work: force the traitor to make a choice between acting like an honest node, or acting otherwise and thereby giving away its position.

We make one further note on this code, having to do with why the specific approach used here for the Cockroach network fails on the more general problem. Observe that in order to make an effective comparison, the values sent along edges $(1, 4)$ and $(2, 4)$ needed to be exactly the same. If they had been independent vectors, no comparison could be useful. This highly constrains the construction of the code, and even though it succeeds for this network, it fails for others, such as the Caterpillar network, to be introduced in the next section. The advantage of the Polytope Code is that it deconstrains the types of values that must be available in order to form a useful comparison; in fact, it becomes possible to have useful comparisons between nearly independent variables, which is not possible with a code built on a finite-field.

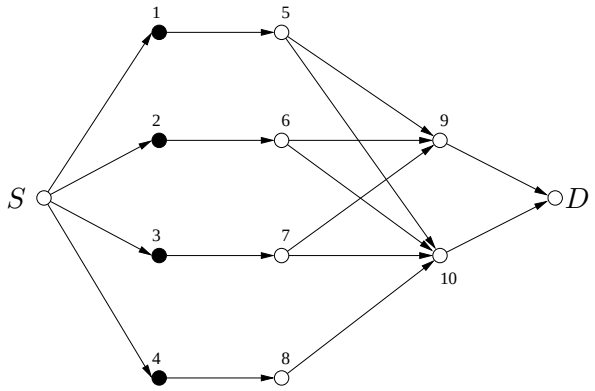


Figure 2.4: The Caterpillar Network. One node may be a traitor, but only one of the black nodes: nodes 1–4.

2.7 An Example Polytope Code: The Caterpillar Network

The Caterpillar Network is shown in Figure 2.4. We consider a slightly different version of the node-based Byzantine attack on this network: at most one node may be a traitor, but only nodes 1–4. This network is not in the class defined in the statement of Theorem 4, but we introduce it in order to motivate the Polytope Code.

Even though this problem differs from the one defined earlier in that not every node in the network may be a traitor, it is easy to see that we may still apply the cut-set bound of Theorem 3 as long as we take the set T to be a subset of the allowable traitors. If we apply Theorem 3 with $A = \{S, 1, 2, 3, 4\}$ and $T = \{1, 2\}$, we find that the capacity of this network is no more than 2. As we will show, the capacity is 2.

Before we demonstrate how rate 2 is achieved, consider what is required to do so for this network. Of the four values on the edges $(1, 5), (2, 6), (3, 7), (4, 8)$, one may be corrupted by the adversary. This means that these four values must form a $(4, 2)$ MDS code. That is, given any uncorrupted pair of these four values, it must

be possible to decode the message exactly. Since each edge has capacity 1, in order to achieve rate 2, the values on each pair of edges must be independent, or nearly independent. For example, we could take the message to be composed of two elements x, y from a finite field, and transmit on these four edges $x, y, x+y, x-y$. However, as we will argue, this choice does not succeed.

Now consider the two edges $(9, D)$ and $(10, D)$. As these are the only edges incident to the destination, to achieve rate 2, both must hold values guaranteed to be uncorrupted by the traitor. We may assume that nodes 5–8 forward whatever they receive on their incoming edges to all their outgoing edges, so node 10 receives all four values sent from nodes 1–4. From these, it can decode the entire message, so it is not a problem to construct a trustworthy value to send along $(10, D)$. However, node 9 has access to only three of the four values sent from nodes 1–4, from which it is not obvious how to construct a trustworthy value. The key problem in designing a successful code is to design the values placed on edges $(1, 5), (2, 6), (3, 7)$ to be pairwise independent, but such that if one value is corrupted, it is always possible to construct a trustworthy value to transmit on $(9, D)$. This is impossible to do using a finite field code. For example, suppose if node 9 receives values for $x, y, x + y$, one of which may be corrupted by the traitor. If the linear constraint among these three values does not hold—that is, if the received value for $x + y$ does not match the sum of the value for x and the value for y—then any of the three values may be the incorrect one. Therefore, from node 9's perspective, any of nodes 1, 2, or 3 could be the traitor. In order to produce a trustworthy symbol, it must rule out at least one node as a possible traitor. If, for example, it could determine that the traitor was either node 1 or 2 but not 3, then the value sent along $(3, 7)$ could be forwarded to $(9, D)$ with a guarantee of correctness. Sending $x, y, x + y$ along the edges $(1, 5), (2, 6), (3, 7)$ does not allow this. In fact, sending

any three elements of a finite field, subject by a single linear constraint, cannot work, but a Polytope Code can.

2.7.1 Coding Strategy

We now begin to describe a capacity-achieving Polytope Code for the Caterpillar network. We do so first by describing how the code is built out of a probability distribution, and the properties we might like this probability distribution to have. Subsequently, we give an explicit construction for a probability distribution derived from a polytope over a real vector field, and show that it has the desired properties.

Let X, Y, Z, W be jointly distributed random variables on the same finite alphabet $\mathcal{X}$. Assume all probabilities on these random variables are rational. For a block length n that is a multiple of the lowest common denominator of the joint distribution of X, Y, Z, W, we may consider the set of all joint sequences $(x^n y^n z^n w^n)$ with joint type exactly equal to this joint distribution. Denote this set $T_p^n(XYZW)$. We know from the theory of types that

$$|T_p^n(XYZW)| \geq \frac{1}{(n+1)^{|\mathcal{X}|^4}} 2^{nH(XYZW)}. \tag{2.6}$$

Our coding strategy will be to associate each element of $T_p^n(XYZW)$ with a distinct message. Given the message, we find the associated four sequences x^n, y^n, z^n, w^n, and transmit them on the four edges out of nodes 1,2,3,4 respectively. Doing this requires placing a sequence in $\mathcal{X}^n$ on each edge. Therefore the rate of this code is

$$\frac{\log |T_p^n(XYZW)|}{n \log |\mathcal{X}|} \geq \frac{H(XYZW)}{\log |\mathcal{X}|} - \frac{|\mathcal{X}|^4 \log(n+1)}{n \log |\mathcal{X}|}. \tag{2.7}$$

Note that for sufficiently large n, we may operate at a rate arbitrarily close to $\frac{H(XYZW)}{\log |\mathcal{X}|}$.

Because of the adversary, the actual sequences sent out of nodes 1–4 may differ from what is sent out of the source. Let $\tilde{x}^n, \tilde{y}^n, \tilde{z}^n, \tilde{w}^n$ be the four sequences as they actually appear on the four edges; at most one of these may differ from x^n, y^n, z^n, w^n. We may now define random variables $\tilde{X}, \tilde{Y}, \tilde{Z}, \tilde{W}$ to have joint distribution equal to the joint type of $(\tilde{x}^n, \tilde{y}^n, \tilde{z}^n, \tilde{w}^n)$. This is a formal definition; these variables do not actually exist, but nodes that have access to these sequences can construct the related random variables. For example, node 9 observes $\tilde{x}^n, \tilde{y}^n, \tilde{z}^n$, so it knows exactly the joint distribution of $\tilde{X}, \tilde{Y}, \tilde{Z}$. The advantage of this coding strategy is that node 9 can now check whether the distribution of these random variables matches that of X, Y, Z. If the distributions differ, a traitor must be present.

The sequences placed on the edges out of nodes 1–4 must be such that nodes 9 and 10 can successfully find trustworthy values to place on edges $(9, D)$ and $(10, D)$. In order for this to be possible, any two of X, Y, Z, W must determine the others. Moreover, as we have discussed, the significant difficulty is allowing node 9 to narrow down the list of possible traitors to just two out of nodes 1–3. The following property on the variables allows this.

Property 1 *The distribution of* (X, Y, Z) *is such that for any three random variables* $(\tilde{X}, \tilde{Y}, \tilde{Z})$ *satisfying*

$$(\tilde{X}, \tilde{Y}) \sim (X, Y) \tag{2.8}$$

$$(\tilde{X}, \tilde{Z}) \sim (X, Z) \tag{2.9}$$

$$(\tilde{Y}, \tilde{Z}) \sim (Y, Z) \tag{2.10}$$

the following holds:

$$(\tilde{X}, \tilde{Y}, \tilde{Z}) \sim (X, Y, Z). \tag{2.11}$$

Suppose we have random variables X, Y, Z, W such that (X, Y, Z) satisfy Property 1. We will show in Sec. 2.7.2 that such a set of random variables exists. The process at node 9 to transmit a message to the destination is as follows. Node 9 observes $\tilde{X}, \tilde{Y}, \tilde{Z}$. If the joint distribution of these three variables matches that of (X, Y, Z), then all three sequences $\tilde{x}^n, \tilde{y}^n, \tilde{z}^n$ are trustworthy, because if a traitor is among nodes 1–3, it must have transmitted the true value of its output sequence, or else the empirical type would not match, due to the fact that any two of the four variables determine the other two. Therefore, node 9 forwards $\tilde{x}^n$ to the destination, confident that it is correct. Meanwhile, node 10 can also observe $\tilde{X}, \tilde{Y}, \tilde{Z}$, and so it forwards $\tilde{y}^n$ to the destination. If the two distributions are different, then by Property 1, one of (2.8), (2.9), or (2.10) must not hold. Suppose, for example, that $(\tilde{X}, \tilde{Y}) \not\sim (X, Y)$. If both node 1 and 2 were honest, then by our code construction, (2.8) would hold. Since it did not, one of nodes 1 or 2 must be the traitor. We have thereby succeeded in reducing the number of nodes that may be the traitor to two, so node 9 may forward $\tilde{z}^n$ to the destination with confidence. Similarly, whichever pairwise distribution does not match, node 9 can always forward the sequence not involved in the mismatch. Meanwhile, node 10 may forward $\tilde{w}^n$ to the destination, since in any case the traitor has been localized to nodes 1–3. The destination always receives two of the four sequences, both guaranteed correct; therefore it may decode.

2.7.2 The Polytope Distribution

All that remains to prove that rate 2 can be achieved for the Caterpillar network is to show that there exists variables X, Y, Z, W such that any two variables determine the other two, satisfying Property 1, and such that $\frac{H(XYZW)}{\log|\mathcal{X}|} = 2$. In fact, this

Table 2.1: A simple distribution satisfying Property 1.

x	y	z	$\Pr(X = x, Y = y, Z = z)$
0	0	0	0
0	0	1	1/3
0	1	0	1/3
0	1	1	0
1	0	0	1/3
1	0	1	0
1	1	0	0
1	1	1	0

is not quite possible. If the entropy requirement holds exactly, then X, Y, Z, W must be pairwise independent, and if so Property 1 cannot hold, because we can take $\tilde{X}, \tilde{Y}, \tilde{Z}$ to be jointly independent with $\tilde{X} \sim X$, $\tilde{Y} \sim Y$, and $\tilde{Z} \sim Z$. This satisfies (2.8)–(2.10) but not (2.11). In fact, we need only show that a suitable set of variables exists such that $\frac{H(XYZW)}{\log |\mathcal{X}|} \geq 2 - \epsilon$ for arbitrarily $\epsilon > 0$. This is possible, and indicates that the set of distributions satisfying Property 1 is not a topologically closed set.

The most unusual aspect of the Polytope Code is Property 1 and its generalization, to be stated as Theorem 5 in Sec. 2.9. Therefore, before constructing a distribution used to achieve rate 2 for the Caterpillar network, we illustrate in Table 2.1 a very simple distribution on three binary variables variables satisfying Property 1. This distribution is only on X, Y, Z; to simplify we momentarily leave out W, because it is not involved in Property 1. We encourage the reader to manually verify Property 1 for this distribution. Observe that X, Y, Z given in Table 2.1 may be alternatively expressed as being uniformly distributed on the set of $x, y, z \in \{0, 1\}$ satisfying $x + y + z = 1$. This set is a polytope, which motivates the more general construction of the distribution to follow.

We now construct the distribution used to achieve rate 2 for the Caterpillar

network. For any positive integer k, consider the set of $x, y, z, w \in \{-k, \ldots, k\}$ satisfying

$$x + y + z = 0 \tag{2.12}$$

$$3x - y + 2w = 0. \tag{2.13}$$

This is the set of integer lattice points in a polytope. Let X, Y, Z, W be uniform over these points. Observe first that this distribution satisfies the requirement that any two variables determine the others. The region of (x, y) pairs with positive probability is shown in Figure 2.5. Note that even though the subspace defined by (2.12)–(2.13) projected onto the (x, y) plane is two-dimensional, X and Y are not statistically independent, because the boundedness of Z and W requires that X and Y satisfy certain linear inequalities. Nevertheless, the area of the polygon shown in Figure 2.5 grows as $\mathcal{O}(k^2)$. Hence the rate of the code resulting from this distribution is

$$\frac{\log H(XYZW)}{\log |\mathcal{X}|} = \frac{\log \mathcal{O}(k^2)}{\log(2k+1)}. \tag{2.14}$$

For large k, this can be made arbitrarily close to 2. When k is large, any pair of the four variables are nearly statistically independent, in that their joint entropy is close to the sum of their individual entropies. We have therefore constructed something like a $(4, 2)$ MDS code. In fact, if we reinterpret (2.12)–(2.13) as constraints on elements x, y, z, w of a finite field, the resulting finite subspace would be exactly a $(4, 2)$ MDS code. This illustrates a general principle of Polytope Codes: any code construction on a finite field can be immediately used to construct a Polytope Code, and many of the properties of the original code will hold over. The resulting code will be substantially harder to implement, in that it involves much longer block-lengths, and more complicated coding functions, but it allows properties like Property 1 to hold.

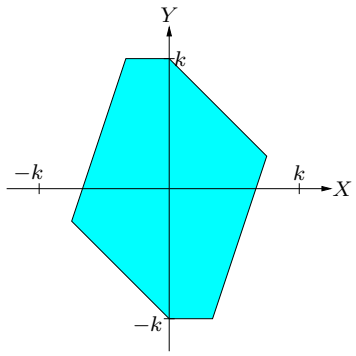


Figure 2.5: An example polytope projected into the (x, y) plane.

All that remains is to verify Property 1 on the polytope distribution. Assuming $\tilde{X}, \tilde{Y}, \tilde{Z}$ satisfy (2.8)–(2.10), we may write

$$\mathbb{E}\big[(\tilde{X}+\tilde{Y}+\tilde{Z})^2\big] = \mathbb{E}\big[\tilde{X}^2+\tilde{Y}^2+\tilde{Z}^2+2\tilde{X}\tilde{Y}+2\tilde{X}\tilde{Z}+2\tilde{Y}\tilde{Z}\big] \tag{2.15}$$

$$= \mathbb{E}\big[X^2+Y^2+Z^2+2XY+2XZ+2YZ\big] \tag{2.16}$$

$$= \mathbb{E}\big[(X+Y+Z)^2\big] \tag{2.17}$$

$$= 0 \tag{2.18}$$

where (2.16) holds from (2.8)–(2.10), and because each term in the some involves at most two of the three variables; and (2.18) holds because $X + Y + Z = 0$ by construction. Now we may write

$$(\tilde{X}, \tilde{Y}, \tilde{Z}) = (\tilde{X}, \tilde{Y}, -\tilde{X}-\tilde{Y}) \tag{2.19}$$

$$\sim (X, Y, -X-Y) \tag{2.20}$$

$$= (X, Y, Z) \tag{2.21}$$

where (2.20) holds by (2.8). This concludes the proof of Property 1.

Observe that the linear constraint $X + Y + Z = 0$ was in no way special; the proof could work just as well under any linear constraint with nonzero coefficients for all three variables. This completes the proof of correctness for the Polytope Code for the Caterpillar network.

2.8 A Polytope Code for the Cockroach Network

We return now to the Cockroach network, and demonstrate a capacity-achieving Polytope Code for it. We do this not to find the capacity for the network, because we have already done so with the simpler code in Sec. 2.6, but to illustrate a Polytope Code on a network satisfying the conditions of Theorem 4, which are somewhat different from the Caterpillar network.

In Sec. 2.6, we illustrated how performing comparisons and transmitting comparison bits through the network can help defeat traitors. In Sec. 2.7, we illustrated how a code can be built out a distribution on a polytope, and how a special property of that distribution comes into play in the operation of the code. To build a Polytope Code for the Cockroach network, we combine these two ideas: the primary data sent through the network comes from the distribution on a polytope, but then comparisons are performed in the network in order to localize the traitor.

The first step in constructing a Polytope Code is to describe a distribution over a polytope. That is, we define a linear subspace in a real vector field, and take a uniform distribution over the polytope defined by the set of vectors with entries in $\{-k, \ldots, k\}$ for some integer k. The nature of this distribution depends on the characteristics of the linear subspace. For our code for the Cockroach network, we need one that is the equivalent of a $(6, 2)$ MDS code. That is, the linear subspace sits in $\mathbb{R}^6$, has dimension 2, and is defined by four constraints such that any two variables determine the others. One choice for the subspace, for example, would

be the set of (a, b, c, d, e, f) satisfying

$$a + b + c = 0 \tag{2.22}$$

$$a - b + d = 0 \tag{2.23}$$

$$a + 2b + e = 0 \tag{2.24}$$

$$2a + b + f = 0. \tag{2.25}$$

Let the random variables A, B, C, D, E, F have joint distribution uniformly distributed over the polytope defined by (2.22)–(2.25) and $a, b, c, d, e, f \in \{-k, \ldots, k\}$. By a similar argument to that in Sec. 2.7, for large k,

$$\frac{H(ABCDEF)}{\log(2k+1)} \approx 2. \tag{2.26}$$

We choose a block length n and associate each message with a joint sequence $(a^n b^n c^n d^n e^n f^n)$ with joint type exactly equal to the distribution of the six variables. For large n and k, we may place one sequence a^n–f^n on each unit capacity edge in the network and operate at rate 2. These six sequences are generated at the source and then routed through the network as shown in Fig. 2.6. For convenience, the figure refers to the variables as scalars instead of vectors, but we always mean them to be sequences.

As in Sec. 2.7, we define $\tilde{A}, \tilde{B}, \tilde{C}, \tilde{D}, \tilde{E}, \tilde{F}$ to have joint distribution equal to the type of the six sequences an they actually appear in the network, which may differ from the sequences sent by the source because of the adversary. In addition to forwarding one sequence as shown in Fig. 2.6, nodes 4 and 5 perform more elaborate operations. In particular, they compare the types of their received sequences with the original distribution. For example, node 4 receives the two sequences b^n and c^n, from which it can construct $\tilde{B}$ and $\tilde{C}$. It checks whether the joint distribution of $(\tilde{B}, \tilde{C})$ matches that of (B, C), and forwards a single bit relaying whether they

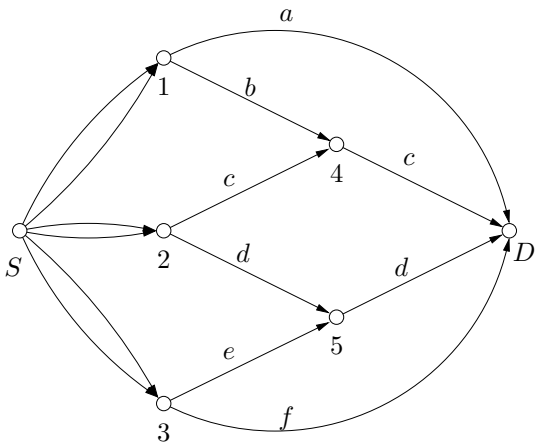


Figure 2.6: A capacity-achieving Polytope Code for the Cockroach Network.

agree along the edge $(4, D)$ in addition to the sequence c^n. This single bit costs asymptotically negligible rate, so it has no effect on the achieved rate of the code for large n and k. Node 5 performs a similar action, comparing the distribution of $(\tilde{D}, \tilde{E})$ with that of (D, E), and transmitting a comparison bit to the destination.

We now describe the decoding operation at the destination. The first step is to compile a list of possible traitors. We denote this list $\mathcal{L} \subseteq \{1, \ldots, 5\}$. The destination does this in the following way. Since the code is entirely known, with no randomness, it can determine whether all its received data could be induced if each node were the traitor. That is, it considers each possible message, each possible traitor, and each possible set of values on the output edges of that traitor. Any given combination of these three things gives a deterministic set of values received at the destination, which may be compared to the set of values that the destination has in fact received. If a node i is such that it could have been the traitor and induced the set of values received at the destination, for any message and any action by node i, then i is put onto $\mathcal{L}$. This process ensures that the true traitor, even though it may not be known by the destination, is surely in $\mathcal{L}$. Note that this procedure could in principle be done for any code, not necessarily

a Polytope Code.

The next step in the decoding process is to use $\mathcal{L}$ to decide from which of the four symbols available at the destination to decode. Since any pair of the six original symbols contain all the information in the message, if at least two of the four symbols a, c, d, f can be determined to be trustworthy by the destination, then it can decode. The destination discards any symbol that was touched by all nodes in $\mathcal{L}$, and decodes from the rest. For example, if $\mathcal{L} = \{2\}$, then the destination discards c, d and decodes from a, f. If $\mathcal{L} = \{2, 4\}$, the destination discards just c—because it is the only symbol touched by both nodes 2 and 4—and decodes from a, d, f. If $\mathcal{L} = \{1, \ldots, 5\}$, then it discards no symbols and decodes from all four.

The prove the correctness of this code, we must show that the destination never decodes from a symbol that was altered by the traitor. This is easy to see if $|\mathcal{L}| = 1$, because in this case the destination knows exactly which node is the traitor, and it simply discards all symbols that may have been influenced by this node. Since no node touches more than two of the symbols available at the destination, there are always at least two remaining from which to decode.

More complicated is when $|\mathcal{L}| \geq 2$. In this case, the decoding process, as described above, sometimes requires the destination to decode from symbols touched by the traitor. For example, suppose node 2 were the traitor, and $\mathcal{L} = \{2, 4\}$. The destination discards c, since it is touched by both nodes 2 and 4, but it decodes from the other available symbols: a, d, f. In particular, the destination uses d to decode, even though it is touched by node 2. Therefore, to prove correctness we must show that it was impossible for node 2 to have transmitted to node 5 anything but the true value of d. What we use to prove this is the fact that $\mathcal{L}$ contains

node 4. That means that node 2 must have acted in a way such that it appears to the destination that node 4 could be the traitor. This induces constraints on the behavior of node 2. For instance, the comparison that occurs at node 5 between d and e must succeed. If it did not, then the destination would receive a bit indicating a failed comparison from node 5. This precludes node 4 being the traitor, because if it were, it could not have induced this failed comparison bit. Therefore the distribution of $(\tilde{D}, \tilde{E})$ must be identical to that of (D, E). This constitutes a constraint on node 2 in its transmission of d. Moreover, $(\tilde{D}, \tilde{F}) \sim (D, F)$, because the destination may observe d and f, so it could detect a difference between these two distributions if it existed. Since both are untouched by node 4, if the distributions did not match then node 4 would not be placed on $\mathcal{L}$. Finally, we have that $(\tilde{E}, \tilde{F} \sim (E, F)$. This holds simply because neither e nor f are touched by the traitor node 2. Summarizing, we have

$$(\tilde{D}, \tilde{E}) \sim (D, E), \tag{2.27}$$

$$(\tilde{D}, \tilde{F}) \sim (D, F), \tag{2.28}$$

$$(\tilde{E}, \tilde{F}) \sim (E, F). \tag{2.29}$$

Given these three conditions, we apply Property 1 to conclude that $(\tilde{D}, \tilde{E}, \tilde{F}) \sim (D, E, F)$. We may do this because, as we argued in Sec. 2.7, Property 1 holds for for any three variables in a polytope subject to a single linear constraint with nonzero coefficients one each one. Since we have constructed the 6 variables to be a $(6, 2)$ MDS code, this is true here (e.g. in the space defined by (2.22)–(2.25), the three variables D, E, F are subject to $D + E - F = 0$). Since e and f together specify the entire message, in order for this three-way distribution to match, the only choice for d is the true value of d. This concludes the proof for this case, because we have shown that in order for node 2 to act in a way so as to cause $\mathcal{L} = \{2, 4\}$, it cannot have altered the value of d at all. Therefore the destination

is justified it using it to decode the message.

The above analysis holds for any $\mathcal{L}$ containing $\{2, 4\}$. That is, if node 2 is the traitor, and $4 \in \mathcal{L}$, then node 2 cannot corrupt d. It is enough to prove correctness of the code to prove a similar fact for every pair of nodes. In particular, we wish to show that if node i is the traitor, and node $j \in \mathcal{L}$, then node i can only corrupt values also touched by node j. This implies that if node i is the traitor, it cannot corrupt any symbol not touched by any node in $\mathcal{L}$. Therefore the destination is justified in only discarding symbols touched by every node in $\mathcal{L}$.

Moreover, it is enough to consider each unordered pair only once. For example, as we have already proven this fact for $i = 2$ and $j = 4$, we do not need to perform a complete proof for $i = 4$ and $j = 2$. This is justified as follows. Suppose node 4 is the traitor and $2 \in \mathcal{L}$. We know from the above argument that when node 2 is the traitor and $4 \in \mathcal{L}$, d is uncorrupted, meaning $(\tilde{A}, \tilde{D}, \tilde{F}) \sim (A, D, F)$. This means that if $(\tilde{A}, \tilde{D}, \tilde{F}) \not\sim (A, D, F)$ and $4 \in \mathcal{L}$, then $2 \notin \mathcal{L}$. Hence, if $2, 4 \in \mathcal{L}$, then $(\tilde{A}, \tilde{D}, \tilde{F}) \sim (A, D, F)$. Since when node 4 is the traitor, a and f are uncorrupted, this implies that the only choice for d transmitted by is the true value of d.

We now complete the proof of correctness of the proposed Polytope Code for the Cockroach network by considering all unordered pairs of potential traitors in the network:

$(1, 2)$ Suppose node 2 is the traitor and $1 \in \mathcal{L}$. Since both these nodes share no symbols, we must show that neither c nor d can be corrupted by node 2. We

have

$$(\tilde{A}, \tilde{B}, \tilde{E}, \tilde{F}) \sim (A, B, E, F), \tag{2.30}$$

$$(\tilde{D}, \tilde{E}) \sim (D, E), \tag{2.31}$$

$$(\tilde{C}, \tilde{D}, \tilde{F}) \sim (C, D, F), \tag{2.32}$$

where (2.30) follows because these symbols are not touched by node 2, (2.31) follows because the comparison at node 5 must succeed, and (2.32) follows because node 1 would be discarded as a possible traitor if $(\tilde{C}, \tilde{D}, \tilde{F})$ did not match at the destination. We may apply Property 1 on D, E, F to conclude that $(\tilde{D}, \tilde{E}, \tilde{F}) \sim (D, E, F)$, therefore d cannot be corrupted. That c cannot be corrupted follows from (2.32).

$(1, 3)$ Suppose node 1 is the traitor and $3 \in \mathcal{L}$. We must show that node 1 cannot corrupt a. We have that $(\tilde{A}, \tilde{C}, \tilde{D}) \sim (A, C, D)$, because these three symbols are not touched by node 3, and are available at the destination. Since c and d determine the message, this single constraint is enough to conclude that node 1 cannot corrupt a. This illustrates a more general principle: when considering the pair of nodes (i, j), if the number of symbols available at the destination untouched by both i or j is at least as large as the rate of the code, we may trivially conclude that no symbols can be corrupted. In fact, this principle works even for finite-field linear codes.

$(1, 4)$**:** Follows exactly as $(1, 3)$.

$(1, 5)$**:** Follows exactly as $(1, 3)$.

$(2, 3)$**:** Follows exactly as $(1, 2)$.

$(2, 4)$**:** Proof above.

$(2, 5)$**:** Follows exactly as $(2, 5)$.

$(3,4)$**:** Follows exactly as $(1,3)$.

$(3,5)$**:** Follows exactly as $(1,3)$.

$(4,5)$**:** Follows exactly as $(1,3)$.

2.9 The Polytope Code

We now describe the general structure of Polytope Codes and state their important properties. Given a matrix $F \in \mathbb{Z}^{u\times m}$, consider the polytope

$$\mathcal{P}_k = \big\{\mathbf{x} \in \mathbb{Z}^m : F\mathbf{x} = 0, |x_i| \leq k \text{ for } i = 1,\ldots,m\big\}. \tag{2.33}$$

We may also describe this polytope in terms of a matrix K whose columns form a basis for the null-space of F. Let $\mathbf{X}$ be an m-dimensional random vector uniformly distributed over $\mathcal{P}_k$. Take n to be a multiple of the least common denominator of the distribution of $\mathbf{X}$ and let $T_p^n(\mathbf{X})$ be the set of sequences $\mathbf{x}^n$ with joint type exactly equal to this distribution. In a Polytope Code, each message is associated with an element of $T_p^n(\mathbf{X})$. By the theory of types, the number of elements in this set is at least $2^{n(H(\mathbf{X})-\epsilon)}$ for any $\epsilon > 0$ and sufficiently large n. Given a message and the corresponding sequence $\mathbf{x}^n$, each edge in the network holds a sequence x_i^n for some $i = 1,\ldots,m$. As we have seen in the example Polytope Codes in Sec. 2.7 and 2.8, the joint entropies of p for large k can be calculated just from the properties of the linear subspace defined by F. The following lemma states this property in general.

Lemma 1 *For any* $S \subseteq \{1,\ldots,m\}$

$$\lim_{k\to\infty} \frac{H(X_S)}{\log k} = \operatorname{rank}(K_S) \tag{2.34}$$

where K_S is the matrix made up of the rows of K corresponding to the elements of S.

Proof: For any $S \subset \{1, \ldots, m\}$, let $\mathcal{P}_k(X_S)$ be the projection of $\mathcal{P}_k$ onto the subspace made up of dimensions S. The number of elements in $\mathcal{P}_k$ is $\Theta(k^{\text{rank}(K_S)})$. That is, there exist constants c_1 and c_2 such that for sufficiently large k

$$c_1 k^{\text{rank}(K_S)} \leq |\mathcal{P}_k(X_S)| \leq c_2 k^{\text{rank}(K_S)}. \tag{2.35}$$

For $S = \{1, \ldots, m\}$, because $\mathbf{X}$ is defined to be uniform on $\mathcal{P}_k$, (2.35) gives

$$\lim_{k\to\infty} \frac{H(\mathbf{X})}{\log k} = \lim_{k\to\infty} \frac{\log |\mathcal{P}_k|}{\log k} = \text{rank}(K). \tag{2.36}$$

Moreover, by the uniform bound

$$\lim_{k\to\infty} \frac{H(X_S)}{\log k} \leq \text{rank}(K_S). \tag{2.37}$$

For any $S \subset \{1, \ldots, m\}$, let $T \subset \{1, \ldots, m\}$ be a minimal set of elements such that $\text{rank}(K_{S,T}) = \text{rank}(K)$; i.e. such that $X_{S,T}$ completely specify X under the constraint $FX = 0$. Note that $\text{rank}(K_T) = \text{rank}(K) - \text{rank}(K_S)$. Hence

$$\lim_{k\to\infty} \frac{H(X_S)}{\log k} = \lim_{k\to\infty} \frac{H(X_{S,T})}{\log k} - \frac{H(X_T|X_S)}{\log k} \tag{2.38}$$

$$\geq \lim_{k\to\infty} \frac{H(X)}{\log k} - \frac{H(X_T)}{\log k} \tag{2.39}$$

$$\geq \text{rank}(K) - \text{rank}(T) \tag{2.40}$$

$$= \text{rank}(K_S). \tag{2.41}$$

Combining (2.37) with (2.41) completes the proof □

Recall that in a linear code operating over the finite field $\mathbb{F}$, we may express the elements on the edges in a network $\mathbf{x} \in \mathbb{F}^m$ as a linear combination of the message $\mathbf{x} = K\mathbf{w}$, where K is a linear transformation over the finite field, and $\mathbf{w}$

is the message vector. Taking a uniform distribution on $\mathbf{w}$ imposes a distribution on $\mathbf{X}$ satisfying

$$H(X_S) = \text{rank}(K_S) \log |\mathbb{F}|. \tag{2.42}$$

This differs from (2.34) only by a constant factor, and also that (2.34) holds only in the limit of large k. Hence, Polytope Codes achieve a similar set of entropy profiles as standard linear codes. They may not be identical, because interpreting a matrix K_S as having integer values as opposed to values from a finite field may cause its rank to change. However, the rank when interpreted as having integer values can never be less than when interpreted as having finite field values, because any linear equality on the integers will hold on a finite field, but not vice versa. The matrix K_S could represent, for example, the source-to-destination linear transformation in a code, so its rank is exactly the achieved rate. Therefore, in fact, the Polytope Code always achieves at least as high a rate as the identical linear code. Often, when designing linear codes, the field size must be made sufficiently large before the code works; here, sending k to infinity serves much the same purpose, albiet in an asymptotic way.

In Sec. 2.7 and 2.8, we saw that Property 1 played an important role in the functionality of the Polytope Codes. The following theorem states the more general version of this property. It compromises the major property that Polytope Codes possess and linear codes do not.

Theorem 5 (Fundamental Property of Polytope Codes) *Let $\mathbf{X} \in \mathbb{R}^m$ be a random vector satisfying $F\mathbf{X} = 0$. Suppose a second random vector $\tilde{\mathbf{X}} \in \mathbb{R}^m$ satisfies the following L constraints:*

$$A_l \tilde{\mathbf{X}} \sim A_l \mathbf{X} \text{ for } l = 1, \ldots, L \tag{2.43}$$

where $A_l \in \mathbb{R}^{u_l \times m}$. The two vectors are equal in distribution if the following properties on F and the A_l hold:

1. *There exists a positive definite matrix C such that*

$$F^T C F = \sum_{l=1}^{L} A_l^T \Sigma_l A_l \tag{2.44}$$

   for some $\Sigma_l \in \mathbb{R}^{u_l \times u_l}$.

2. *There exists an l^* and a matrix G^* such that $F\tilde{\mathbf{X}} = 0$ is equivalent to $\tilde{\mathbf{X}} = G^* A_{l^*} \tilde{\mathbf{X}}$ for any random vector $\tilde{\mathbf{X}}$. This is equivalent to $\begin{bmatrix} A_{l^*} \\ F \end{bmatrix}$ having full column rank.*

Proof: The following proof follows almost exactly the same argument as the proof of Property 1 in Sec. 2.7. We may write

$$\mathbb{E}\big[(F\tilde{\mathbf{X}})^T C (F\tilde{\mathbf{X}})\big] = \sum_{l=1}^{m} \mathbb{E}\big[(A_l\tilde{\mathbf{X}})^T \Sigma_l (A_l\tilde{\mathbf{X}})\big] \tag{2.45}$$

$$= \sum_{l=1}^{m} \mathbb{E}\big[(A_l\mathbf{X})^T \Sigma_l (A_l\mathbf{X})\big] \tag{2.46}$$

$$= \mathbb{E}\big[(F\mathbf{X})^T C (F\mathbf{X})\big] \tag{2.47}$$

$$= 0 \tag{2.48}$$

where (2.45) and (2.47) follow from (2.44); (2.46) follows from (2.43), and because each term in the sum involves $A_l\mathbf{X}$ for some l; and (2.48) follows because $F\mathbf{X} = 0$. Because C is positive definite, we have that $F\tilde{\mathbf{X}} = 0$. Therefore, by the second property in the statement of the theorem, $\tilde{\mathbf{X}} = G^* A_{l^*} \tilde{\mathbf{X}}$. Hence

$$\tilde{\mathbf{X}} = G^* A_{l^*} \tilde{\mathbf{X}} \tag{2.49}$$

$$\sim G^* A_{l^*} \mathbf{X} \tag{2.50}$$

$$= \mathbf{X}. \tag{2.51}$$

This completes the proof. □

As an example of an application of Theorem 5, we use it to prove again Property 1 in Sec. 2.7. Recall that variables $X, Y, Z \in \{-k, \ldots, k\}$ satisfying $X + Y + Z = 0$, and the three pairwise distributions of $\tilde{X}, \tilde{Y}, \tilde{Z}$ match as stated in (2.8)–(2.10). In terms of the notation of Theorem 5, we have $m = 3$, $L = 3$, and

$$F = \begin{bmatrix} 1 & 1 & 1 \end{bmatrix}, \tag{2.52}$$

$$A_1 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}, \tag{2.53}$$

$$A_2 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \tag{2.54}$$

$$A_3 = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{2.55}$$

To satisfy the second condition of Theorem 5, we may set $l^* = 1$, since the single linear constraint $X + Y + Z = 0$ implies that

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \underbrace{\begin{bmatrix} 1 & 0 \\ 0 & 1 \\ -1 & -1 \end{bmatrix}}_{G^*} \begin{bmatrix} X \\ Y \end{bmatrix}. \tag{2.56}$$

In fact, we could just as well have set l^* to 2 or 3. To verify the first condition, we need to check that there exist Σ_l for $l = 1, 2, 3$ and a positive definite C (in this case, a positive scalar, because F has only one row, so $C \in \mathbb{R}^{1\times 1}$) satisfying (2.44). If we let

$$\Sigma_l = \begin{bmatrix} \sigma_{l,11} & \sigma_{l,12} \\ \sigma_{l,21} & \sigma_{l,22} \end{bmatrix} \tag{2.57}$$

then, for instance,

$$A_1^T \Sigma_1 A_1 = \begin{bmatrix} \sigma_{1,11} & \sigma_{1,12} & 0 \\ \sigma_{1,21} & \sigma_{1,22} & 0 \\ 0 & 0 & 0 \end{bmatrix}. \tag{2.58}$$

The right hand side of (2.44) expands to

$$\sum_{l=1}^{3} A_l^T \Sigma_l A_l = \begin{bmatrix} \sigma_{1,11} + \sigma_{2,11} & \sigma_{1,12} & \sigma_{2,12} \\ \sigma_{1,21} & \sigma_{1,22} + \sigma_{3,11} & \sigma_{3,12} \\ \sigma_{2,21} & \sigma_{3,21} & \sigma_{2,22} + \sigma_{3,22} \end{bmatrix}. \tag{2.59}$$

Therefore, for suitable choices of $\{\Sigma_l\}_{l=1}^3$, we can produce any matrix for the right hand side of (2.44). We may simply set $C = 1$ and calculate the resulting matrix for the left hand side, then set $\{\Sigma_l\}_{l=1}^3$ appropriately. This allows us to apply Theorem 5 to conclude that $(\tilde{X}, \tilde{Y}, \tilde{Z}) \sim (X, Y, Z)$.

In our proof of Theorem 4, we will not use Theorem 5 in its most general form. Instead, we state three corollaries that will be more convenient. The first is a generalization of the above argument for more than three variables.

Corollary 1 *Let* $\mathbf{X}$ *satisfy* $F\mathbf{X} = 0$ *for some* $F \in \mathbb{Z}^{1\times m}$ *with all nonzero values. If* $\tilde{\mathbf{X}}$ *satisfies*

$$(\tilde{X}_i, \tilde{X}_j) \sim (X_i, X_j) \text{ for all } i, j = 1, \ldots, m \tag{2.60}$$

$$(\tilde{X_2}, \cdots, \tilde{X_m}) \sim (X_2, \cdots, X_m) \tag{2.61}$$

then $\tilde{\mathbf{X}} \sim \mathbf{X}$.

Proof: We omit the explicit construction of the A_l matrices corresponding to the conditions (2.60), (2.61). The second condition for Theorem 5 is satisfied by (2.61), since the linear constraint $F\mathbf{X} = 0$ determines X_1 given $X_2 \cdots X_m$. To

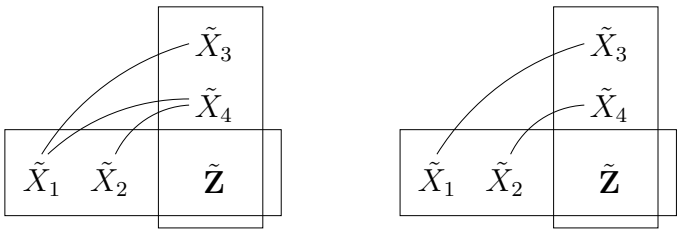


Figure 2.7: The constraints on the random vector $\tilde{\mathbf{X}}$ in Corollaries 2 (left) and 3 (right). Rectangles represent a constraint on the marginal distribution of all enclosed variables; lines represent pairwise constraints on the two connected variables.

verify the first condition, note that from the conditions in (2.60), we may construct an arbitrary matrix on the right hand side of (2.44) for suitable $\{\Sigma_l\}_{l=1}^{L}$. Therefore we may simply set $C = 1$. □

Corollary 1 considers the case with m variables and $m-1$ degrees of freedom; i.e. a single linear constraint. The following corollary considers a case with m variables and $m-2$ degrees of freedom.

Corollary 2 *Let $F \in \mathbb{Z}^{2\times m}$ be such that any 2×2 submatrix of F is non-singular. Let $\mathbf{X}$ satisfy $F\mathbf{X} = 0$. The non-singular condition on F implies that any $m-2$ variables specify the other two. Assume that $m \geq 4$, and for convenience let $\mathbf{Z} = (X_5, \ldots, X_m)$ and $\tilde{\mathbf{Z}} = (\tilde{X}_5, \ldots, \tilde{X}_m)$. If $\tilde{\mathbf{X}}$ satisfies*

$$(\tilde{X}_1, \tilde{X}_2, \tilde{\mathbf{Z}}) \sim (X_1, X_2, \mathbf{Z}), \tag{2.62}$$

$$(\tilde{X}_3, \tilde{X}_4, \tilde{\mathbf{Z}}) \sim (X_3, X_4, \mathbf{Z}), \tag{2.63}$$

$$(\tilde{X}_1, \tilde{X}_3) \sim (X_1, X_3), \tag{2.64}$$

$$(\tilde{X}_2, \tilde{X}_4) \sim (X_2, X_4), \tag{2.65}$$

$$(\tilde{X}_1, \tilde{X}_4) \sim (X_1, X_4) \tag{2.66}$$

then $\tilde{\mathbf{X}} \sim \mathbf{X}$. Fig. 2.7 diagrams the constraints on $\tilde{\mathbf{X}}$.

Proof: We prove Corollary 2 with two applications of Corollary 1. First, consider the group of variables $(X_1X_2X_4\mathbf{Z})$. These $m-1$ variables are subject to a single linear constraint, as in Corollary 1. From (2.62), (2.65), and (2.66) we have all pairwise marginal constraints, satisfying (2.60). Furthermore, (2.62) satisfies (2.61). We may therefore apply Corollary 1 to conclude

$$(\tilde{X}_1, \tilde{X}_2, \tilde{X}_4, \tilde{\mathbf{Z}}) \sim (X_1, X_2, X_4, \mathbf{Z}). \tag{2.67}$$

A similar application of Corollary 1 using (2.63), (2.64), and (2.66) allows us to conclude

$$(\tilde{X}_1, \tilde{X}_3, \tilde{X}_4, \tilde{\mathbf{Z}}) \sim (X_1, X_3, X_4, \mathbf{Z}). \tag{2.68}$$

Observe that (2.67) and (2.68) share the m variables $(\tilde{X}_1, \tilde{X}_4, \tilde{\mathbf{Z}})$, which together determine $\tilde{X}_2$ and $\tilde{X}_3$ in exactly the same way that $(X_1, X_4, \mathbf{Z})$ determine X_2 and X_3. Therefore we may combine (2.67) and (2.68) to conclude $\tilde{\mathbf{X}} \sim \mathbf{X}$. □

All five constraints (2.62)–(2.66) are not always necessary, and we may sometimes apply Theorem 5 without (2.66). However, this depends on an interesting additional property of the linear constraint matrix F, as stated in the third and final corollary to Theorem 5.

Corollary 3 *Let $F \in \mathbb{Z}^{2\times m}$ be such that any 2×2 submatrix of F is non-singular, and let $\mathbf{X}$ satisfy $F\mathbf{X} = 0$. In addition, assume*

$$|K_{X_1X_2\mathbf{Z}}|\ |K_{X_3X_4\mathbf{Z}}|\ |K_{X_1X_3\mathbf{Z}}|\ |K_{X_2X_4\mathbf{Z}}| < 0 \tag{2.69}$$

where again K is a basis for the null space of F, and $K_{\mathbf{X}_S}$ for $S \subset \{1, \ldots, m\}$ is the matrix made up of the rows of K corresponding to the variables $(X_i)_{i\in S}$. If $\tilde{\mathbf{X}}$ satisfies (2.62)–(2.65) *(Fig. 2.7 diagrams these constraints), then $\tilde{\mathbf{X}} \sim \mathbf{X}$.*

Proof: Either (2.62) or (2.63) satisfies the second condition in Theorem 5. To verify the first condition, first let $G = \sum_l A_l^T \Sigma_l A_l$. In the four constraints (2.62)–(2.65), each pair of variables appears together except for (X_1, X_4) and (X_2, X_3). Therefore, for suitable choices of Σ_l, we can construct any G satisfying $G_{1,4} = G_{2,3} = G_{3,2} = G_{4,1} = 0$. We must show that such a G exists satisfying

$$F^T C F = G \tag{2.70}$$

for some positive definite C.

We build G row-by-row. By (2.70), each row of G is a linear combination of rows of F; i.e. it forms the coefficients of a linear equality constraint imposed on the random vector $\mathbf{X}$. Since $G_{1,4}$, the first row of G represents a linear constraint on the variables $X_1, X_2, X_3, \mathbf{Z}$. Since any $m-2$ variables specify the other two, there is exactly one linear equality constraint on these $m-1$ variables, up to a constant. This constraint can be written as

$$\begin{vmatrix} X_1 & K_{X_1} \\ X_2 & K_{X_2} \\ X_3 & K_{X_3} \\ \mathbf{Z} & K_{\mathbf{Z}} \end{vmatrix} = 0. \tag{2.71}$$

since the vector $X_1, X_2, X_3, \mathbf{Z}$ forms a linear combination of the columns of $K_{X_1,X_2,X_3,\mathbf{Z}}$. Hence, the first row of G is a constant multiple of the coefficients in (2.71). In particular,

$$G_{1,1} = \alpha |K_{X_2 X_3 \mathbf{Z}}|, \tag{2.72}$$

$$G_{1,2} = -\alpha |K_{X_1 X_3 \mathbf{Z}}| \tag{2.73}$$

for some constant α. Since $G_{2,3} = 0$, the second row of G represents the linear constraint on $X_1, X_2, X_4, \mathbf{Z}$. Using similar reasoning as above gives

$$G_{2,1} = \beta |K_{X_2 X_4 \mathbf{Z}}|, \tag{2.74}$$

$$G_{2,2} = -\beta |K_{X_1 X_4 \mathbf{Z}}| \tag{2.75}$$

for some constant β. Moreover, by (2.70) G is symmetric, so $G_{1,2} = G_{2,1}$, and by (2.73) and (2.74)

$$\beta = -\frac{|K_{X_1 X_3 \mathbf{Z}}|}{|K_{X_2 X_4 \mathbf{Z}}|}\alpha. \tag{2.76}$$

Positive definiteness of C is equivalent to positive definiteness of the upper left 2×2 block of G, so the conditions we need are

$$0 < G_{1,1} = \alpha |K_{X_2 X_3 \mathbf{Z}}|, \tag{2.77}$$

$$0 < G_{1,1} G_{2,2} - G_{1,2} G_{2,1} \tag{2.78}$$

$$= \alpha^2 \left[\frac{|K_{X_2 X_3 \mathbf{Z}}| \; |K_{X_1 X_4 \mathbf{Z}}| \; |K_{X_1 X_3 \mathbf{Z}}|}{|K_{X_2 X_4 \mathbf{Z}}|} - |K_{X_1 X_3 \mathbf{Z}}|^2 \right]. \tag{2.79}$$

We may choose α to trivially satisfy (2.77), and (2.79) is equivalent to

$$|K_{X_1 X_3 \mathbf{Z}}| \; |K_{X_2 X_4 \mathbf{Z}}| \Big(|K_{X_2 X_3 \mathbf{Z}}| \; |K_{X_1 X_4 \mathbf{Z}}| - |K_{X_2 X_4 \mathbf{Z}}| \; |K_{X_1 X_3 \mathbf{Z}}| \Big) > 0 \tag{2.80}$$

which may also be written as (2.69). □

The necessity of satisfying (2.69) in order to apply Theorem 5 substantially complicates code design. When building a linear code, one need only worry about the rank of certain matrices; i.e. certain determinants need be nonzero. Here, we see that the signs of these determinants may be constrained as well.

2.10 Proof of Theorem 4

To prove Theorem 4, we need to specify a Polytope Code for each network satisfying conditions 1–3 in the statement of the theorem. This involves specifying the linear relationships between various symbols in the network, the comparisons that are done among them at internal nodes, and then how the destination uses the comparison information it receives to decode. We then proceed to prove that the destination always decodes correctly. The key observation in the proof is that the important comparisons that go on inside the network are those that involve a variable that does not reach the destination. This is because those symbols that do reach the destination can be examined there, so further comparisons inside the network do not add anything. Therefore we will carefully route these non-destination symbols to maximize the utility of their comparisons. In particular, we design these paths so that for every node having one direct edge to the destination and one other output edge, the output edge not going to the destination holds a non-destination variable. The advantage of this is that any variable, before exiting the network, is guaranteed to cross a non-destination variable at a node where the two variables may be compared. The existence of non-destination paths with this property depends on the planarity of the network. This is described in much more detail in the sequel.

Notation: For an edge $e \in E$, with $e = (i, j)$, where $i, j \in V$, let $\text{head}(e) = i$ and $\text{tail}(e) = j$. For a node $i \in V$, let $\mathcal{E}_{\text{in}}(i)$ be the set of edges e with $\text{tail}(e) = i$, and let $\mathcal{E}_{\text{out}}(i)$ be the set of edges e with $\text{head}(e) = i$. Let $\mathcal{N}_{\text{in}}(i)$ be the set of input neighbors of i; that is, the set of $\text{head}(e)$ for each $e \in \mathcal{E}_{\text{in}}(i)$. Similarly, let $\mathcal{N}_{\text{out}}(i)$ be the set of output neighbors of i. For integers a, b, let $\mathcal{V}_{a,b}$ be the set of nodes with a inputs and b outputs. We will sometimes refer to such nodes as a-to-b. For

$l \in \{1,2\}$, let $\bar{l} = 2 - l$. A *path* is defined as an ordered list of edges $e_1, \ldots, e_k$ satisfying $\text{tail}(e_l) = \text{head}(e_{l+1})$ for $l = 1, \ldots, k-1$. The head and tail of a path are defined as $\text{head}(e_1)$ and $\text{tail}(e_k)$ respectively. A node i is said to *reach* a node j if there exists a path with head i and tail j. By convention, a node can reach itself.

Consider an arbitrary network satisfying the conditions of Theorem 4. By condition (3), no node has more output edges than input edges. Therefore the min-cut is that between the destination and the rest of the network. Let M be the value of this cut; i.e., the number of edges connected to the destination. We now state a lemma giving instances of the cut-set upper bound on capacity in terms of quantities that make the bound easier to handle than Theorem 3 itself. We will subsequently show that the minimum upper bound given by Lemma 2 is achievable using a Polytope Code; therefore, the cut-set bound gives the capacity.

Lemma 2 *For $i, j \in V$, let $d_{i,j}$ be the sum of $|\mathcal{E}_{\text{in}}(k)| - |\mathcal{E}_{\text{out}}(k)|$ for all nodes k reachable from either i or j, not including i or j. That is, if k is a-to-b, it contributes $a - b$ to the sum. Recall that this difference is always positive. Let c_i be the total number of output edges from node i, and let e_i be the number of output edges from node i that go directly to the destination. For any distinct pair of nodes i_1, i_2,*

$$C \leq M - e_{i_1} - e_{i_2}. \tag{2.81}$$

Moreover, if there is no path between i_1 and i_2,

$$C \leq M + d_{i_1,i_2} - c_{i_1} - c_{i_2}. \tag{2.82}$$

Proof: Applying Theorem 3 with $A = V \setminus \{D\}$, $T = \{i_1, i_2\}$ immediately gives (2.81). To prove (2.82), we apply Theorem 3 with $T = \{i_1, i_2\}$, and

$$A = \{k \in V : k \text{ is not reachable from } i_1 \text{ or } i_2\} \cup \{i_1, i_2\}. \tag{2.83}$$

Observe that there are no backwards edges for the cut A, because any node in A^c is reachable from either i_1 or i_2, so for any edge (j,k) with $j \in A^c$, k is also reachable by from i_1 or i_2, so k is also not in A. Therefore we may apply Theorem 3. Since all output neighbors of i_1 and i_2 are not in A, each output edge of i_1 and i_2 crosses the cut. Hence (2.4) becomes

$$C \leq |\{e \in E : \text{head}(e) \in A,\ \text{tail}(e) \notin A\}| - c_1 - c_2. \tag{2.84}$$

Since no node in the network has more output edges than input edges, the difference between the first term in (2.84)—the number of edges crossing the cut—and M is exactly the sum of $|\mathcal{E}_{\text{in}}(k)| - |\mathcal{E}_{\text{out}}(k)|$ for all $k \in A^c$. Hence

$$|\{e \in E : \text{head}(e) \in A,\ \text{tail}(e) \notin A\}| - M = d_{i_1,i_2}. \tag{2.85}$$

Combining (2.84) with (2.85) gives (2.82). □

Next, we show that we may transform any network satisfying the conditions of Theorem 4 into an equivalent one that is planar, and made up of just 2-to-2 nodes and 2-to-1 nodes. We will go on to show that the upper bound provided by Lemma 2 is achievable for any such network, so it will be enough to prove that a transformation exists that preserves planarity, does not reduce capacity, and does not change the bound given by Lemma 2.

We first replace any a-to-b node i with a cascade of $a-b$ 2-to-1 nodes followed by a b-to-b node. This transformation is illustrated in Fig. 2.8. Denote the b-to-b node in the transformation i^*. Since no node in the original network has more than two output edges, the resulting network contains only 1-to-1 nodes, 2-to-2 nodes, and 2-to-1 nodes. We will shortly argue that the 1-to-1 nodes may be removed as well. Certainly these transformations maintain the planarity of the network. Moreover, any rate achievable on the transformed network is also achievable on

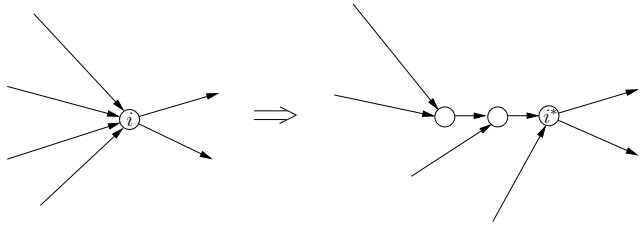


Figure 2.8: An illustration of the transformation from a 4-to-2 node to an equivalent set of 2-to-1 and 2-to-2 nodes.

the original network. This is because if node i is transformed via this operation into several nodes, any coding operation performed by these nodes can certainly be performed by node i. Additionally, the traitor taking control of node i in the original network does exactly as much damage as the traitor taking control of i^* in the transformed network, since it controls all edges sent to other nodes. Now consider the minimum upper bound given by Lemma 2 after this transformation. The only nodes with positive e_j values will be i^* nodes, and $e_{i^*} = e_i$. Hence (2.81) cannot change. In (2.82), if we take i_1^* and i_2^*, then the bound is the same in the transformed network. Taking one of the 2-to-1 nodes instead of a i^* node cannot result in a lower bound, because they have no more output edges, so no higher c values, and no fewer reachable nodes with fewer outputs than inputs, so no smaller d values. Therefore, the minimal bound given by (2.82) for the transformed network is the same as that of the original network. Moreover, in the transformed network d_{i_1,i_2} is equal simply to the number of 2-to-1 nodes reachable from i_1 or i_2 not including i_1, i_2.

We may additionally transform the network to remove 1-to-1 nodes, simply be replacing the node and the two edges connected to it by a single edge. The traitor can always take over the preceding or subsequent node and have at least as much power. The only exception is when the 1-to-1 node is connected only to the source and destination. In this case, instead of removing the node, we may

add a additional edge to it from the source, turning it into a 2-to-1 node. Such a transformation does not change the capacity, nor the planarity or the Lemma 2 bounds.

We also assume without loss of generality that all nodes in the network are reachable from the source. Certainly edges out of these nodes cannot carry any information about the message, so we may simply discard this portion of the network, if it exists, without changing the capacity.

We will show that the smallest bound given by Lemma 2 is achievable using a Polytope Code. If we take i_1 and i_2 to be two nodes with at least one direct link to the destination, (2.81) gives that the capacity is no more than $M-2$. Moreover, since $e_i \leq c_i \leq 2$ for any node i, neither (2.81) nor (2.82) can produce a bound less than $M-4$. Therefore the minimum bound given by Lemma 2 can take on only three possible values: $M-4, M-3, M-2$. It is not hard to see that $M-4$ is trivial achievable; indeed, even with a linear code. Therefore the only interesting cases are when the cut-set bound is $M-3$ or $M-2$. We begin with the latter, because the proof is more involved, and contains all the necessary parts to prove the $M-3$ case. The $M-3$ proof is subsequently given in Section 2.10.5.

Assume that the right hand sides of (2.81) and (2.82) are never smaller than $M-2$. We describe the construction of the Polytope Code to achieve rate $M-2$ in several steps. The correctness of the code will be proved in Lemmas 3–6, which are stated during the description of the construction process. These Lemmas are then proved in Sections 2.10.1–2.10.4.

1) Edge Labeling: We first label all the edges in the network except those in

$\mathcal{E}_{\text{in}}(D)$. These labels are denoted by the following functions

$$\phi : E \setminus \mathcal{E}_{\text{in}}(D) \to \mathcal{V}_{2,1} \tag{2.86}$$

$$\psi : E \setminus \mathcal{E}_{\text{in}}(D) \to \{0, 1\}. \tag{2.87}$$

For a 2-to-1 node v, let $\Lambda(v)$ be the set of edges e with $\phi(e) = v$. The set $\Lambda(v)$ represents the edges carrying symbols that interact with the non-destination symbol that terminates at node v. The set of edges with $\phi(e) = v$ and $\psi(e) = 1$ represent the path taken by the non-destination symbol that terminates at node v. The following Lemma states the existence of labels ϕ, ψ with the necessary properties.

Lemma 3 *There exist functions ϕ and ψ with the following properties:*

A *The set of edges e with $\phi(e) = v$ and $\psi(e) = 1$ form a path.*

B *If $\phi(e) = v$, then either* $\text{tail}(e) = v$ *or there is an edge e' with* $\text{head}(e') = \text{tail}(e)$ *and $\phi(e') = v$.*

C *For every 2-to-2 node i with output edges e_1, e_2, either $\psi(e_1) = 1$, $\psi(e_2) = 1$, or $\phi(e_1) \neq \phi(e_2)$.*

Note that if property (B) holds, $\Lambda(v)$ is a union of paths ending at v. From property (A), the edges on one of these paths satisfy $\psi(e) = 1$.

2) Internal Node Operation: Assume that ϕ and ψ are defined to satisfy properties (A)–(C) in Lemma 3. Given these labels, we will specify how internal nodes in the network operate. Every edge in the network will hold a symbol representing a linear combination of the message, as well as possibly some comparison bits. We also define a function

$$\rho : E \to \{1, \ldots, |\mathcal{E}_{\text{out}}(S)|\} \tag{2.88}$$

that will serve as an accounting tool to track symbols as they pass through the network. We begin by assigning distinct and arbitrary values to $\rho(e)$ for all $e \in \mathcal{E}_{\text{out}}(S)$ (ρ therefore constitutes an ordering on $\mathcal{E}_{\text{out}}(S)$). Further assignments of ρ will be made recursively. This will be made explicit below, but if a symbol is merely forwarded, it travels along edges with a constant ρ. When linear combinations occur at internal nodes, ρ values are manipulated, and ρ determine exactly how this is done.

For every node i with 2 input edges, let f_1, f_2 be these edges. If i is 2-to-2, let e_1, e_2 be its two output edges; if it is 2-to-1, let e be its output edge. If $\phi(f_1) = \phi(f_2)$, then node i compares the symbols on f_1 and f_2. If node i is 2-to-2, then $\phi(e_l) = \phi(f_1)$ for either $l = 1$ or 2. Node i transmits its comparison bit on e_l. If node i is 2-to-1, then it transmits its comparison bit on e. All 2-to-2 nodes forward all received comparison bits on the output edge with the same ϕ value as the input edge on which the bit was received. All 2-to-1 nodes forward all received comparison bits on its output edge.

We divide nodes in $\mathcal{V}_{2,2}$ into the following sets. The linear transformation performed at node i will depend on which of these sets it is in.

$$\mathcal{W}_1 = \{i \in \mathcal{V}_{2,2} : \psi(f_1) = \psi(f_2) = 0, \phi(f_1) \neq \phi(f_2)\} \tag{2.89}$$

$$\mathcal{W}_2 = \{i \in \mathcal{V}_{2,2} : \psi(f_1) = \psi(f_2) = 0, \phi(f_2) = \phi(f_2)\} \tag{2.90}$$

$$\mathcal{W}_3 = \{i \in \mathcal{V}_{2,2} : \psi(f_1) = 1 \text{ or } \psi(f_2) = 1\} \tag{2.91}$$

We will sometimes refer to nodes in $\mathcal{W}_2$ as *branch nodes*, since they represent branches in $\Lambda(\phi(f_1))$. Moreover, branch nodes are significant because a failed comparison at a branch node will cause the forwarding pattern within $\Lambda(\phi(f_1))$ to change. For an edge e, X_e denotes the symbol transmitted on e. The following gives the relationships between these symbols, which are determined by internal

nodes, depending partially on the comparison bits they receive. For each node i, the action of node i depends on which set it falls in as follows:

- $\mathcal{W}_1$: Let l be such that $\phi(e_l) = \phi(f_1)$. The symbol on f_1 is forwarded to e_l, and the symbol on f_2 is forwarded onto $e_{\bar{l}}$. Set $\rho(e_l) = \rho(f_1)$, and $\rho(e_{\bar{l}}) = \rho(f_2)$.
- $\mathcal{W}_2$: Let l be such that $\phi(e_l) = \phi(f_1) = \phi(f_2)$. Let l' be such that $\rho(f_{l'}) < \rho(f_{\bar{l}'})$. We will show in Lemma 4 that our construction is such that $\rho(f_1) \neq \rho(f_2)$ at all nodes, so l' is well defined. If neither f_1 nor f_2 hold a failed comparison bit, the output symbols are

$$X_{e_l} = \gamma_{i,1} X_{f_1} + \gamma_{i,2} X_{f_2} \tag{2.92}$$

$$X_{e_{\bar{l}}} = X_{f_{l'}} \tag{2.93}$$

  where coefficients $\gamma_{i,1}, \gamma_{i,2}$ are nonzero integers to be chosen later. Set output ρ values to

$$\rho(e_l) = \rho(f_{\bar{l}'}) \tag{2.94}$$

$$\rho(e_{\bar{l}}) = \rho(f_{l'}). \tag{2.95}$$

  Note that the symbol on the input edge with smaller ρ value is forwarded without linear combination. If the input edge $f_{l'}$ reports a failed comparison anywhere previously in $\Lambda(\phi(f_1))$, then (2.93) changes to

$$X_{e_{\bar{l}}} = X_{f_{\bar{l}'}}. \tag{2.96}$$

- $\mathcal{W}_3$: Let l be such that $\psi(f_l) = 1$, and l' be such that $\psi(e_{l'}) = 1$ and $\phi(e_{l'}) = \phi(f_l)$. The symbol on f_l is forwarded to $e_{l'}$, and the symbol on $f_{\bar{l}}$ is forwarded to $e_{\bar{l}'}$, with the following exception. If $\phi(f_1) = \phi(f_2)$ and there is a failed comparison bit sent from $f_{\bar{l}}$, then the forwarding swaps: the

symbol on f_l is forwarded to $e_{\bar{l}'}$, and the symbol on $f_{\bar{l}}$ is forwarded to $e_{l'}$. Set $\rho(e_{l'}) = \rho(f_l)$ and $\rho(e_{\bar{l}'}) = \rho(f_{\bar{l}})$. Again, ρ is consistent along forwarded symbols, but only when all comparisons succeed.

- $\mathcal{V}_{2,1}$: Let l be such that $\psi(f_l) = 1$. The symbol from $f_{\bar{l}}$ is forwarded on e, unless there is a failed comparison bit sent from $f_{\bar{l}}$, in which case the symbol from f_l is forwarded on e. Set $\rho(e) = \rho(f_{\bar{l}})$.

See Fig. 2.9 for an illustration of the linear transformations performed at internal nodes and how they change when a comparison fails. The following Lemma gives some properties of the internal network behavior as prescribed above.

Lemma 4 *The following hold:*

1. *For any integer $a \in \{1, \ldots, |\mathcal{E}_{\text{out}}(S)|\}$, the set of edges with e with $\rho(e) = a$ form a path (we refer to this in the sequel as the $\rho = a$* path*). Consequently, there is no node i with input edges f_1, f_2 such that $\rho(f_1) = \rho(f_2)$.*
2. *If there are no failed comparisons that occur in the network, then the linear transformations are such that the decoder can decode any symbol in the network except those on non-destination paths.*
3. *Suppose a comparison fails at a branch node k with input edges f_1, f_2 with $v = \phi(f_1) = \phi(f_2)$. Assume without lack of generality that $\rho(f_1) < \rho(f_2)$. The forwarding pattern within $\Lambda(v)$ changes such that symbols sent along the $\rho = \rho(f_2)$ path are not decodable at the destination, but what was the non-destination symbol associated with v is decodable.*

3) MDS Code Construction: The rules above explain how the symbols are combined and transformed inside the network. In addition, when the initial set of

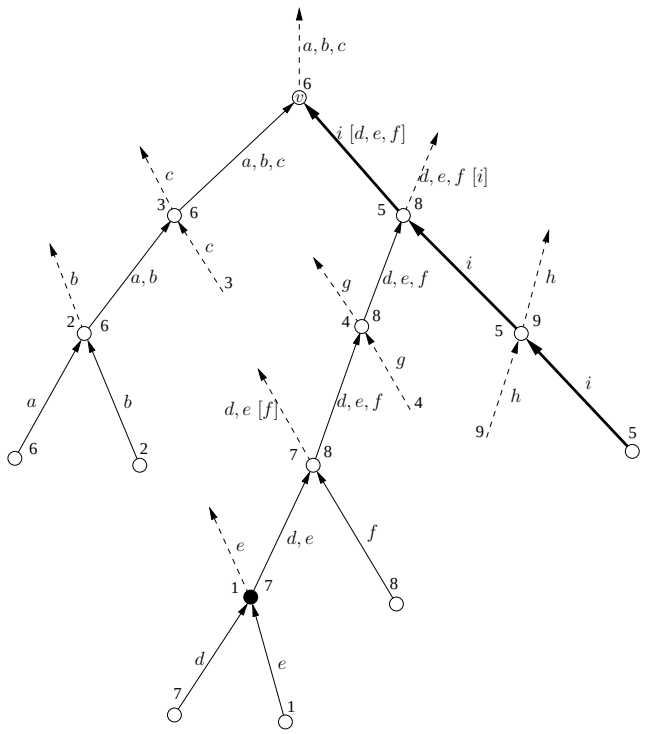


Figure 2.9: An example of the linear transformations performed in $\Lambda(v)$ for some v (labeled as such). Solid edges denote $\phi(e) = v$, dashed edges denote $\phi(e) \neq v$. Thick edges denote $\psi(e) = 1$. Near the head of each edge is the corresponding ρ value. Also shown is the symbol transmitted along that edge, given initial symbols a–i at the furthest upstream edges in the network. When several symbols are written on an edge, this indicates that the edge carries a linear combination of those symbols. The symbols indicated in brackets are those carried by the edges when the comparison at the indicated black node fails. Symbols on edges labeled without brackets do not change when the comparison fails.

symbols are sent into the network from the source, they are subject to linear constraints. We now describe exactly how this is done. Assume that no comparisons fail in the network, so the linear relationships between symbols are unmodified. For a 2-to-1 node v, let e_v^* be the edge with $\phi(e_v^*) = v$, $\psi(e_v^*) = 1$, and $\text{tail}(e_v^*) = v$; i.e. it is the last edge to hold the non-destination symbol terminating at v. Observe that it will be enough to specify the linear relationships among the symbols on $\{e_v^* : v \in \mathcal{V}_{2,1}\}$ as well as the M edges in $\mathcal{E}_{\text{in}}(D)$. These collectively form the Polytope Code equivalent of a $(M + |\mathcal{V}_{2,1}|, M - 2)$ MDS code. We must construct

this code so as to satisfy certain instances of (2.69), so that we may apply Theorem 5 as necessary. The following Lemma states the existence of a set of linear relationships among the $M + |\mathcal{V}_{2,1}|$ variables with the required properties.

Lemma 5 *For each 2-to-1 node v, let $\Xi(v)$ be the set of edges e with $\mathrm{tail}(e) = D$ such that there is an edge e' with $\mathrm{tail}(e') = \mathrm{head}(e)$, $\phi(e') = v$, and $\psi(e') = 1$. That is, the symbol on e, just before being sent to the destination, was compared against the non-destination symbol associated with v. Note that any edge $e \in \mathcal{E}_{\mathrm{in}}(D)$ is contained in $\Xi(v)$ for some 2-to-1 node v. There exists a generator matrix $K \in \mathbb{Z}^{M+|\mathcal{V}_{2,1}| \times M-2}$ where each row is associated with an edge in $\{e_v^* : v \in \mathcal{V}_{2,1}\} \cup \mathcal{E}_{\mathrm{in}}(D)$ such that for all $v_1, v_2 \in \mathcal{V}_{2,1}$ and all $f_1 \in \Xi(v_1), f_2 \in \Xi(v_2)$, the constraints*

$$(\tilde{X}_{f_1}, \tilde{X}_{f_2}, \tilde{\mathbf{Z}}) \sim (X_{f_1}, X_{f_2}, \mathbf{Z}) \tag{2.97}$$

$$(\tilde{X}_{e^*_{v_1}}, \tilde{X}_{e^*_{v_2}}, \tilde{\mathbf{Z}}) \sim (X_{e^*_{v_1}}, X_{e^*_{v_2}}, \mathbf{Z}) \tag{2.98}$$

$$(\tilde{X}_{f_1}, \tilde{X}_{e^*_{v_1}}) \sim (X_{f_1}, X_{e^*_{v_1}}) \tag{2.99}$$

$$(\tilde{X}_{f_2}, \tilde{X}_{e^*_{v_2}}) \sim (X_{f_2}, X_{e^*_{v_2}}) \tag{2.100}$$

imply

$$(\tilde{X}_{f_1}, \tilde{X}_{f_2}, \tilde{X}_{e^*_{v_1}}, \tilde{X}_{e^*_{v_2}} \tilde{\mathbf{Z}}) \sim (X_{f_1}, X_{f_2}, X_{e^*_{v_1}}, X_{e^*_{v_2}} \tilde{\mathbf{Z}}) \tag{2.101}$$

where

$$\mathbf{Z} = (X_e : e \in \mathcal{E}_{\mathrm{in}}(D) \setminus \{f_1, f_2\}). \tag{2.102}$$

4) Decoding Procedure: To decode, the destination first compiles a list $\mathcal{L} \subset V$ of which nodes may be the traitor. It does this by taking all its available data: received comparison bits from interior nodes as well as the symbols it has direct access to, and determines whether it is possible for each node, if it were the traitor, to have acted in a way to cause these data to occur. If so, it adds this node to $\mathcal{L}$. For each node i, let K_i be the linear transformation from the message vector

$\mathbf{W}$ to the symbols on the output edges of node i. With a slight abuse of notation, regard K_D represent the symbols on the input edges to D instead. For a set of nodes $S \subset V$, let $K_{D\perp S}$ be a basis for the subspace spanned by K_D orthogonal to

$$\bigcap_{j\in S} \text{span}(K_{j\to D}). \tag{2.103}$$

The destination decodes from $K_{D\perp\mathcal{L}}\mathbf{W}$. If i is the traitor, it must be that $i \in \mathcal{L}$, so

$$\text{rank}(K_{D\perp\mathcal{L}}) \geq M - \dim\left(\bigcap_{j\in S} \text{span}(K_j)\right) \tag{2.104}$$

$$\geq M - \text{rank}(K_i) \tag{2.105}$$

$$\geq M - 2 \tag{2.106}$$

where we used the fact that node i has at most two output edges. Since $K_{D\perp\mathcal{L}}$ has rank at least $M-2$, this is a large enough space for the destination to decode the entire message. The follow Lemma allows us to conclude that all variables in the subspace spanned by $K_{D\perp\mathcal{L}}$ are trustworthy.

Lemma 6 *Consider any pair of nodes i, j. Suppose i is the traitor, and acts in a way such that $j \in \mathcal{L}$. Node i cannot have corrupted any value in $K_{D\perp\{i,j\}}\mathbf{W}$.*

2.10.1 Proof of Lemma 3

We begin with $\phi(e) = \psi(e) = \emptyset$ for all edges e, and set ϕ and ψ progressively. First we describe some properties of the graph (V, E) imposed by the fact that the right hand sides of (2.81) and (2.82) are never less than $M-2$.

Given a 2-to-1 node v, let Γ_v be the set of nodes for which v is the only reachable 2-to-1 node. Note that other than v, the only nodes in Γ_v are 2-to-2. Moreover,

if v can reach another 2-to-1 node, Γ_v is empty. We claim that Γ_v forms a path. If it did not, then there would be two 2-to-2 nodes $i_1, i_2 \in \Gamma_v$ for which there is no path between them. That is, $d_{i_1,i_2} = 1$ and $c_{i_1} = c_{i_2} = 2$, so (2.82) becomes $C \leq M - 3$, which contradicts our assumption that the cut-set bound is $M - 2$.

Furthermore, every 2-to-2 node must be able to reach at least one 2-to-1 node. If not, then we could follow a path from such a 2-to-2 node until reaching a node i_1 all of whose output edges lead directly to the destination. Node i_1 cannot be 2-to-1, so it must be 2-to-2, meaning $e_{i_1} = 2$. Taking any other node i_2 with a direct link to the destination gives no more than $M - 3$ for the right hand side of (2.81), again contradicting our assumption.

The first step in the edge labeling procedure is to specify the edges holding non-destination symbols; that is, for each 2-to-1 node v, to specify the edges e for which $\phi(e) = v$ and $\psi(e) = 1$. To satisfy property (A), these must form a path. For any node $i \in \mathcal{N}_{\text{in}}(D)$, the output edge of i that goes to the destination has no ϕ value, so to satisfy property (C), the other output edge e must satisfy $\psi(e) = 1$. Moreover, by property (B), if $\phi(e) = v$, then there is a path from $\text{head}(e)$ to v. Hence, if $i \in \mathcal{V}_{2,2} \cap \Gamma_v$ for some 2-to-1 node v, then it is impossible for the two output edges of i to have different ϕ values; hence, by property (C), one of its output edges e must satisfy $\psi(e) = 1$. Therefore, we need to design the non-destination paths so that they pass through Γ_v for each v, as well as each node in $\mathcal{N}_{\text{in}}(D)$.

For each 2-to-1 node v, we first set the end of the non-destination path associated with v to be the edges in Γ_v. That is, for an edge e, if $\text{head}(e), \text{tail}(e) \in \Gamma_v$, set $\psi(e) = 1$ and $\phi(e) = v$. Now our only task is to extend the paths backwards such that one is guaranteed to pass through each node in $\mathcal{N}_{\text{in}}(D)$.

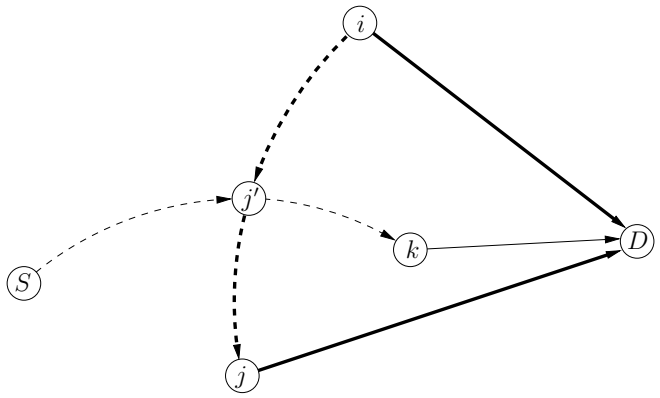


Figure 2.10: A diagram of the planar embedding being used to prove that a node $k \in \mathcal{N}_{\text{in}}(D)$ on the interior of $\mathcal{C}_{i,j}$ is reachable from i. Solid lines are single edges; dashed lines represent paths made up of possibly many edges. Thick lines correspond to edges in $\mathcal{C}_{i,j}$.

Construct an embedding of the graph (V, E) in the plane such that S is on the exterior face. Such an embedding always exists [90]. If we select a set of edges making up an *undirected cycle*—that is, edges constituting a cycle on the underlying undirected graph—then all nodes in the network not on the cycle are divided into those on the interior and those on the exterior, according to the planar embedding. Take $i, j \in \mathcal{N}_{\text{in}}(D)$ such that i can reach j, and let $\mathcal{C}_{i,j}$ be the undirected cycle composed of a path from i to j, in addition to the edges (i, D) and (j, D). We claim that if a node $k \in \mathcal{N}_{\text{in}}(D)$ is on the interior of $\mathcal{C}_{i,j}$, then it is reachable from i. Since S is on the exterior face of the graph, it must be exterior to the cycle $\mathcal{C}_{i,j}$. There exists some path from S to k, so it must cross the $\mathcal{C}_{i,j}$ at a node j'. Observe that j' must be on the path from i to j, so it is reachable from i. Therefore i can reach j' and j' can reach k, so i can reach k. This construction is diagrammed in Fig. 2.10.

We may travel around node D in the planar embedding, noting the order in which the nodes $\mathcal{N}_{\text{in}}(D)$ connect to D. Call this order $u_1, \ldots, u_M$. Take any $i \in \mathcal{N}_{\text{in}}(D)$, and suppose $i = u_l$. We claim that the set of nodes in $\mathcal{N}_{\text{in}}(D)$ reachable

from u_l forms a contiguous block around u_l in the $\{u\}$ ordering, where we regard u_1 and u_M as being adjacent, so two contiguous blocks containing u_1 and u_M is considered one contiguous block.

Suppose this were not true. That is, for some $i \in \mathcal{N}_{\text{in}}(D)$ there exists a $j \in \mathcal{N}_{\text{in}}(D)$ reachable from i that is flanked on either side in the $\{u\}$ ordering by nodes $k_1, k_2 \in \mathcal{N}_{\text{in}}(D)$ not reachable from i. The order in which these four nodes appear in $\{u\}$ in some cyclic permutation or reflection of

$$(i, k_1, j, k_2). \tag{2.107}$$

Neither k_1 nor k_2 can be on the interior of $\mathcal{C}_{i,j}$, because, as shown above, any such node is reachable from i. However, if they are both on the exterior, then the order in (2.107) cannot occur, because D is on the boundary of $\mathcal{C}_{i,j}$.

By contiguity, if a node $i \in \mathcal{N}_{\text{in}}(D)$ can reach any other node in $\mathcal{N}_{\text{in}}(D)$, it can reach a node immediately adjacent to it in the $\{u\}$ ordering. Suppose i can reach both the node $j_1 \in \mathcal{N}_{\text{in}}(D)$ immediately to its left and the node $j_2 \in \mathcal{N}_{\text{in}}(D)$ immediately to its right. We show that in fact i can reach every node in $\mathcal{N}_{\text{in}}(D)$. In particular, there can be only one such node, or else there would be a cycle. Node i has only two output edges, one of which goes directly to D. Let i' be the tail of the other. Both j_1 and j_2 must be reachable from i'.

We claim it is impossible for both j_1 to be exterior to $\mathcal{C}_{i,j_2}$ and j_2 to be exterior to $\mathcal{C}_{i,j_1}$. Suppose both were true. We show the graph must contain a cycle. Let $\bar{\mathcal{C}}$ be the undirected cycle composed of the path from i' to j_1, the path from i' to j_2, and the edges $(j_1, D), (j_2, D)$. Every node on $\bar{\mathcal{C}}$ is reachable from i. Since both j_1 is exterior to $\mathcal{C}_{i,j_2}$ and j_2 is exterior to $\mathcal{C}_{i,j_1}$, it is easy to see that i must be on the interior of $\bar{\mathcal{C}}$. Therefore any path from S to i must cross the cycle at a node k', reachable from i. Since k' is on a path from S to k', i is also reachable from k', so

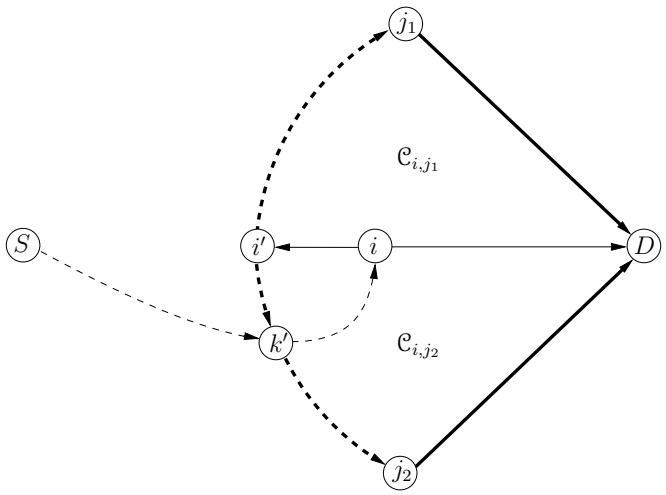


Figure 2.11: A diagram of the planar embedding being used to prove that a node reaching its two neighbors in $\mathcal{N}_{\text{in}}(D)$ can reach every node in $\mathcal{N}_{\text{in}}(D)$. Solid lines are single edges; dashed lines represent paths made up of possibly many edges. Thick lines correspond to the undirected cycle $\bar{\mathcal{C}}$. Undirected cycles $\mathcal{C}_{i,j_1}$ and $\mathcal{C}_{i,j_2}$ are indicated.

there is a cycle. See Fig. 2.11 for a diagram of this.

Therefore, we may assume without loss of generality that j_2 is in the interior of $\mathcal{C}_{i,j_1}$. Suppose there were a node $j_3 \in \mathcal{N}_{\text{in}}(D)$ not reachable from i. Node j_3 must be on the exterior of $\mathcal{C}_{i,j_1}$, because we have shown that nodes in $\mathcal{N}_{\text{in}}(D)$ on the interior are reachable from i. Therefore, in the $\{u\}$ order, these four nodes must appear in some cyclic permutation or reflection of (i, j_3, j_1, j_2). However, this is impossible, because both j_1 and j_2 were assumed to be adjacent to i. Therefore, i can reach every node in $\mathcal{N}_{\text{in}}(D)$.

Take a node i that can reach 2-to-1 nodes $v_1, v_2 \in \mathcal{N}_{\text{in}}(D)$. Suppose that i cannot reach every node in $\mathcal{N}_{\text{in}}(D)$. Therefore, the nodes it can reach in in $\mathcal{N}_{\text{in}}(D)$ are either entirely to its right or entirely to its left in the $\{u\}$ ordering, or else, by contiguity, node i would be able to reach the adjacent nodes on both sides. Suppose without loss of generality that they are all to its right, and that v_2 is further to the right than v_1. We claim that v_1 is on the interior of $\mathcal{C}_{i,v_2}$. Suppose it

were on the exterior. By contiguity, every node in $\mathcal{N}_{\text{in}}(D)$ on the exterior of $\mathcal{C}_{i,v_2}$ must be reachable from i. Since we have already argued that every node in $\mathcal{N}_{\text{in}}(D)$ on the interior of $\mathcal{C}_{i,v_2}$ is reachable from i, this means i can reach every node in $\mathcal{N}_{\text{in}}(D)$, which we have assumed is not the case.

Therefore, v_1 is on the interior of $\mathcal{C}_{i,v_2}$. We may construct a path from S to v_1, passing through all nodes in Γ_{v_1}. This path must cross $\mathcal{C}_{i,v_2}$ at a node k, reachable from i. Node j can reach both v_1 and v_2, so it cannot be in Γ_{v_1}. However, j is on a path passing through Γ_{v_1}, so it can reach all nodes in Γ_{v_1}. Therefore there exists a path from i to v_1, passing through Γ_{v_1}.

If i can reach every node in $\mathcal{N}_{\text{in}}(D)$, then as shown above, either v_1 is in the interior of $\mathcal{C}_{i,v_1}$, or v_2 is in the interior of $\mathcal{C}_{i,v_2}$. Therefore, by the same argument to that just used for the case that i cannot reach every node in $\mathcal{N}_{\text{in}}(D)$, there is either a path from i to v_1 through Γ_{v_1} or a path from i to v_2 through Γ_{v_2}.

Fix a 2-to-1 node $v_1 \in \mathcal{N}_{\text{in}}(D)$. Consider the set of nodes that are:

- contained in $\mathcal{V}_{2,2} \cap \mathcal{N}_{\text{in}}(D)$,
- not in Γ_v for any 2-to-1 node v,
- can reach v_1,
- cannot reach any other node also satisfying the above three conditions.

We claim there are at most two such nodes. Suppose there were two such nodes i_1, i_2 both to the left of v_1 in the $\{u\}$ ordering. If i_1 were further to the left, then i_1 could reach i_2, since i_1 can reach v_1 and the nodes reachable from i_1 must form a contiguous block. Hence i_1 would not qualify. Therefore there can be at most one such node to the left of v_1 and at most one to the right. Denote these two

nodes i and j respectively, if they exist. By contiguity, every node satisfying the first three conditions must be able to reach either i or j. Moreover, all such nodes to the left of v_1 form a single path ending in i, and those on the right form a single path ending in j. We will proceed to extend two non-destination paths backwards to i and j. Then, we may further extend these two paths backwards through all nodes in $\mathcal{V}_{2,2} \cap \mathcal{N}_{\text{in}}(D)$ that can reach v_1, and then backwards to the source on arbitrary paths. Hence, we need only find paths from i to the head of Γ_v for some v, and a distinct one of the same for j.

Both i and j can reach at least one 2-to-1 node other than v_1. Suppose i can reach another 2-to-1 node $v_2 \in \mathcal{N}_{\text{in}}(D)$. By the argument above, there is a path from i to the leftmost of v_1, v_2 through Γ_{v_1} or Γ_{v_2} respectively. Similarly, if j can reach a 2-to-1 node $v_3 \in \mathcal{N}_{\text{in}}(D)$ with $v_3 \neq v_1$, there is a path from j to the rightmost of v_1, v_3, through the associated Γ. This is true even if $v_2 = v_3$.

Suppose there is no 2-to-1 node in $\mathcal{N}_{\text{in}}(D)$ reachable from node i other than v_1. There still must be a 2-to-1 node v_2 reachable from i, though $v_2 \notin \mathcal{N}_{\text{in}}(D)$. Since v_2 is not adjacent to the destination, it must be able to reach a 2-to-1 node that is. Therefore $\Gamma_{v_2} = \emptyset$, so any path from i to v_2 trivially includes Γ_{v_2}. If j can also reach no 2-to-1 nodes in $\mathcal{N}_{\text{in}}(D)$ other than v_1, there must be some 2-to-1 node $v_3 \notin \mathcal{N}_{\text{in}}(D)$ reachable from j. We may therefore select non-destination paths from i to v_2 and j to v_3, unless $v_2 = v_3$. This only occurs if this single node is the only 2-to-1 node other than v_1 reachable by either i or j. We claim that in this case, either i or j can reach the tail of Γ_{v_1}. Therefore we may extend the non-destination path for v_1 back to one of i or j, and the non-destination path for $v_2 = v_3$ to the other. Every node can reach some 2-to-1 node in $\mathcal{N}_{\text{in}}(D)$, so v_2 can reach v_1, or else i and j would be able to reach a different 2-to-1 node in

$\mathcal{N}_{\text{in}}(D)$. By a similar argument to that used above, v_1 must be on the interior of the undirected cycle composed of the path from i to v_2, the path from j to v_2, and the edges $(i, D), (j, D)$. If not, v_1 would not be between i and j in the $\{u\}$ ordering. Note this is true even if i can reach j or vice versa. Since S must be exterior to this cycle, any path from S to v_1 including Γ_{v_1} must cross either the path from i to v_2 or j to v_2 at a node k. Node k must be able to reach the head of Γ_{v_1}, so either i or j can reach Γ_{v_1}.

Once the non-destination paths are defined, we perform the following algorithm to label other edges so as to satisfy property (C). We refer to an edge e as *labeled* if $\phi(e) \neq \emptyset$. We refer to a node as *labeled* if any of its output edges are labeled. Any node unlabeled after the specifications of the non-destination paths must not be in $\mathcal{N}_{\text{in}}(D)$, and must be able to reach at least two different 2-to-1 nodes.

1. For any edge e such that there exists an $e' \in \mathcal{E}_{\text{out}}(\text{tail}(e))$ with $\psi(e') = 1$, set $\phi(e) = \phi(e')$. Observe now that any path eventually reaches a labeled edge. Furthermore, the tail of any unlabeled edge cannot be a node contained in Γ_v for any v, so it can lead to at least two 2-to-1 nodes.
2. Repeat the following until every edge other than those connected directly to the destination is labeled. Consider two cases:
    - *There is no 2-to-2 node with exactly one labeled output edge*: Pick an unlabeled node i. Select any path of unlabeled edges out of i until reaching a labeled node. Let v be the label of a labeled output edge from this node. For all edges e on the selected path, set $\phi(e) = v$. Observe that every node on this path was previously an unlabeled 2-to-2 node. Hence every node on this path, except the last one, has exactly one labeled output edge.

- *There is a 2-to-2 node i with exactly one labeled output edge*: Let v_1 be the label on the labeled output edge. Select any path of unlabeled edges beginning with the unlabeled output edge from i until reaching a node with an output edge labeled v_2 with $v_2 \neq v_1$. This is always possible because any unlabeled edge must be able to lead to at least two 2-to-1 nodes, including one other than v_1. For all edges e on the selected path, set $\phi(e) = v_2$. Observe that before we labeled the path, no node in the path other than the last one had an output edge labeled v_2, because if it did, we would have stopped there. Hence, after we label the path, if a node now has 2 labeled output edges, they have different labels.

Note that in the above algorithm, whenever an edge e becomes labeled, if there was another edge e' with $\text{head}(e) = \text{head}(e')$, either e' was unlabeled, or $\phi(e) \neq \phi(e')$. Therefore, the final ϕ values satisfy property (B).

2.10.2 Proof of Lemma 4

Observe that for any 2-to-2 node, the two ρ values on the input edges are identical to the two ρ values on the output edges. For a 2-to-1 node, the ρ value on the output edge is equal to the ρ value on one of the input edges. Therefore beginning with any edge in $\mathcal{E}_{\text{out}}(S)$, we may follow a path along only edges with the same ρ value, and clearly we will hit all such edges. Property (1) immediately follows.

Property (2) follows from the fact that 2-to-2 nodes always operate such that from the symbols on the two output edges, it is possible to decode the symbols on the input edges. Therefore the destination can always reverse these transformations to recover any earlier symbols sent in the network. The only exception is 2-to-1

nodes, which drop one of their two input symbols. The dropped symbol is a non-destination symbol, so it is clear that the destination can always decode the rest.

We now prove property (3). We claim that when the comparison fails at node k, it is impossible for the destination to decode X_{f_2}. We may assume that the destination has direct access to all symbols on edges immediately subsequent to edges in $\Lambda(v)$. This can only make X_{f_2} easier to decode. Recall that $\rho(f_1) < \rho(f_2)$, so X_{f_1} is forwarded directly on the output edge of k not in $\Lambda(v)$. Therefore the destination can only decode X_{f_2} if it can decode the symbol on the output edge of k in $\Lambda(v)$. Continuing to follow the path through $\Lambda(v)$, suppose we reach an edge e_1 with $\text{tail}(e_1) = k'$, where k' is a branch node. Let e_2 be the other input edge of k'. Even if $\rho(e_1) < \rho(e_2)$, meaning k' would normally forward X_{e_1} outside of $\Lambda(v)$, because e_1 carries a failed comparison bit, k' will instead forward X_{e_2} outside of $\Lambda(v)$. Again, the destination can only decode X_{f_2} (or equivalently X_{e_1}) if it can decode the symbol on the output edge of k' in $\Lambda(v)$. If we reach a node interacting with the non-destination symbol associated with v, then because of the failed comparison bit, the formerly non-destination symbol is forwarded outside of $\Lambda(v)$ and the symbol to decode continues traveling through $\Lambda(v)$. It will finally reach v, at which point it is dropped. Therefore it is never forwarded out of $\Lambda(v)$, so the destination cannot recover it.

2.10.3 Proof of Lemma 5

From Corollary 3, it is enough to prove the existence of a K matrix satisfying

$$|K_{e^*_{v_1},e^*_{v_2},\mathbf{z}}|\ |K_{f_1,f_2,\mathbf{z}}|\ |K_{e^*_{v_1},f_1,\mathbf{z}}|\ |K_{e^*_{v_2},f_2,\mathbf{z}}| < 0. \tag{2.108}$$

We construct a Vandermonde matrix K to satisfy (2.108) for all v_1, v_2 and all f_1, f_2 in the following way. We will construct a bijective function (an ordering) α given by

$$\alpha : \{e_v^* : v \in \mathcal{V}_{2,1}\} \cup \mathcal{N}_{\text{in}}(D) \to \{1, \ldots, M + |\mathcal{V}_{2,1}|\}. \tag{2.109}$$

For each $v \in \mathcal{V}_{2,1}$, set $\alpha(e_v^*)$ to an arbitrary but unique number in $1, \ldots, |\mathcal{V}_{2,1}|$. We may now refer to a 2-to-1 node as $\alpha^{-1}(a)$ for an integer $a \in \{1, \ldots, |\mathcal{V}_{2,1}|\}$. Now set $\alpha(e)$ for $e \in \mathcal{E}_{\text{in}}(D)$ such that, in α order, the edge set $\{e_v^* : v \in \mathcal{V}_{2,1}\} \cup \mathcal{N}_{\text{in}}(D)$ is written

$$e^*_{\alpha^{-1}(1)}, e^*_{\alpha^{-1}(2)}, \ldots, e^*_{\alpha^{-1}(|\mathcal{V}_{2,1}|)}, \\ \Xi(\alpha^{-1}(|\mathcal{V}_{2,1}|)), \Xi(\alpha^{-1}(|\mathcal{V}_{2,1}| - 1)), \ldots, \Xi(\alpha^{-1}(1)). \tag{2.110}$$

That is, each $\Xi(v)$ set is consecutive in the ordering, but in the opposite order as the associated non-destination edges e_v^*. Now let K be the Vandermone matrix with constants given by α. That is, the row associated with edge e is given by

$$\begin{bmatrix} 1 & \alpha(e) & \alpha(e)^2 & \cdots & \alpha(e)^{M-3} \end{bmatrix}. \tag{2.111}$$

We claim the matrix K given by (2.111) satisfies (2.108). Fix v_1, v_2, and $f_1 \in \Xi(v_1), f_2 \in \Xi(v_2)$. Due to the Vandermonde structure of K, we can write the determinant of a square submatrix in terms of the constants $\alpha(e)$. For instance,

$$|K_{e^*_{v_1}, e^*_{v_2}, \mathbf{Z}}| = [\alpha(e^*_{v_2}) - \alpha(e^*_{v_1})] \prod_{e \in \mathbf{Z}} [\alpha(e) - \alpha(e^*_{v_1})][\alpha(e) - \alpha(e^*_{v_2})] \\ \cdot \prod_{e, e' \in \mathbf{Z}, \alpha(e) < \alpha(e')} [\alpha(e') - \alpha(e)] \tag{2.112}$$

where we have assumed without loss of generality that the rows of $K_{\mathbf{Z}}$ are ordered

according to α. Expanding the determinants in (2.108) as such gives

$$|K_{e^*_{v_1},e^*_{v_2},\mathbf{Z}}|\ |K_{f_1,f_2,\mathbf{Z}}|\ |K_{e^*_{v_1},f_1,\mathbf{Z}}|\ |K_{e^*_{v_2},f_2,\mathbf{Z}}| \tag{2.113}$$

$$\begin{aligned}&= [\alpha(e^*_{v_2}) - \alpha(e^*_{v_1})][\alpha(f_2) - \alpha(f_1)][\alpha(f_1) - \alpha(e^*_{v_1})][\alpha(f_2) - \alpha(e^*_{v_2})] \\ &\quad \cdot \prod_{e\in\mathbf{Z}} [\alpha(e) - \alpha(e^*_{v_1})]^2[\alpha(e) - \alpha(e^*_{v_2})]^2[\alpha(e) - \alpha(f_1)]^2[\alpha(e) - \alpha(f_1)]^2 \\ &\quad \cdot \prod_{e,e'\in\mathbf{Z},\alpha(e)<\alpha(e')} [\alpha(e') - \alpha(e)]^4.\end{aligned} \tag{2.114}$$

Recall $f_1 \in \Xi(v_1)$, $f_2 \in \Xi(v_2)$. Since we chose α such that the Ξ sets are in opposite order to the edges e^*_v, we have

$$[\alpha(e^*_{v_2}) - \alpha(e^*_{v_1})][\alpha(f_2) - \alpha(f_1)] < 0. \tag{2.115}$$

Moreover, since all the Ξ sets have larger α values than the edges e^*_v,

$$\alpha(f_1) - \alpha(e^*_{v_1}) > 0, \tag{2.116}$$

$$\alpha(f_2) - \alpha(e^*_{v_2}) > 0. \tag{2.117}$$

Hence, there is exactly one negative term in (2.114), from which we may conclude (2.108).

2.10.4 Proof of Lemma 6

The random vector $\mathbf{W}$ is distributed according to the type of the message vector as it is produced as the source. We formally introduce the random vector $\tilde{\mathbf{W}}$ representing the message as it is transformed in the network. As in our examples, this vector is distributed according to the joint type of the sequences as they appear in the network, after being corrupted by the adversary. For each edge e, we define X_e and $\tilde{X}_e$ similarly as random variables jointly distributed with $\mathbf{W}$ and $\tilde{\mathbf{W}}$ respectively with distributions given by the expected and corrupted joint types.

For every pair of nodes (i, j), we need to prove both of the following:

$$\text{If } i \text{ is the traitor, and } j \in \mathcal{L},\ i \text{ cannot corrupt values in } K_{D\perp\{i,j\}}\mathbf{W}. \quad (2.118)$$

$$\text{If } j \text{ is the traitor, and } i \in \mathcal{L},\ j \text{ cannot corrupt values in } K_{D\perp\{i,j\}}\mathbf{W}. \quad (2.119)$$

In fact, each of these implies the other, so it will be enough to prove just one. Suppose (2.118) holds. Therefore, if the distribution observed by the destination of $K_{D\perp\{i,j\}}\tilde{\mathbf{W}}$ does not match that of $K_{D\perp\{i,j\}}\tilde{\mathbf{W}}$, then at least one of i, j will not be in $\mathcal{L}$. If they both were in $\mathcal{L}$, it would have had to be possible for node i to be the traitor, make it appear as if node j were the traitor, but also corrupt part of $K_{D\perp\{i,j\}}W$. By (2.118), this is impossible. Hence, if j is the traitor and $i \in \mathcal{L}$, then the distribution of the $K_{D\perp\{i,j\}}Y_D$ must remain uncorrupted. This vector includes $K_{D\perp j}W$, a vector that can certainly not be corrupted by node j. Since $\text{rank}(K_{D\perp j}) \geq M-2$, and there are only $M-2$ degrees of freedom, the only choice node j has to ensure that the distribution of $K_{D\perp\{i,j\}}W$ matches p is to leave this entire vector uncorrupted. That is, (2.119) holds.

Fix a pair (i, j). We proceed to prove either (2.118) or (2.119). Doing so will require placing constraints on the actions of the traitor imposed by comparisons that occur inside the network, then applying one of the corollaries of Theorem 5 in Sec. 2.9. Let $K_{\perp i}$ be a basis for the space orthogonal to K_i. If node i is the traitor, we have that $K_{\perp i}\tilde{\mathbf{W}} \sim K_{\perp i}\mathbf{W}$. Moreover, since $j \in \mathcal{L}$, $K_{D\perp j}\tilde{\mathbf{W}}) \sim K_{D\perp j}\mathbf{W}$. These two constraints are analogous to (2.63) and (2.62) respectively, where the symbols on the output of node i are analogous to X_1, X_2. The subspace of K_D orthogonal to both K_i and K_j corresponds to $\mathbf{Z}$ in the example. We now seek pairwise constraints of the form (2.64)–(2.66) from successful comparisons to apply Theorem 5.

Being able to apply Theorem 5 requires that $K_{D\perp j}$ has rank $M-2$ for all j. Ensuring this has to do with the choices for the coefficients $\gamma_{i,1}, \gamma_{i,2}$ used in (2.92).

A rank deficiency in $K_{D\perp j}$ is a singular event, so it is not hard to see that random choices for the γ will cause this to occur with small probability. Therefore such γ exist.

We now discuss how pairwise constraints on the output symbols of i or j are found. Consider the following cases and subcases:

- $i, j \in \mathcal{W}_1 \cup \mathcal{W}_2$: Suppose node i is the traitor. Let e_1, e_2 be the output edges of node i. For each $l = 1, 2$, we look for constraints on X_{e_l} by following the $\rho = \rho(e_l)$ path until one of the following occurs:
  - *We reach an edge on the $\rho = \rho(e_l)$ path carrying a symbol influenced by node j*: This can only occur immediately after a branch node k with input edges f_1, f_2 where $\rho(f_1) = \rho(e_l)$, $\rho(f_2) < \rho(f_1)$, and X_{f_2} is influenced by node j. At node k, a comparison occurs between $\tilde{X}_{f_1}$, which is influenced by node i but not j, and $\tilde{X}_{f_2}$. If the comparison succeeds, then this places a constraint on the distribution of $(\tilde{X}_{f_1}, \tilde{X}_{f_2})$. If the comparison fails, the forwarding pattern changes such that the $\rho = \rho(e_l)$ path becomes a non-destination path; i.e. the value placed on e_l does not affect any variables available at the destination. Hence, the subspace available at the destination that is corruptible by node i is of dimension at most one.
  - *We reach node j itself*: In this situation, we make use of the fact that we only need to prove that node i cannot corrupt values available at the destination that cannot also be influenced by node j. Consider whether the $\rho = \rho(e_l)$ path, between i and j, contains a branch node k with input edges f_1, f_2 such that $\rho(f_1) = \rho(e_l)$ and $\rho(f_2) > \rho(f_1)$. If there is no such node, then X_{e_l} cannot influence any symbols seen by the

destination that are not also being influenced by j. That is, X_{e_l} is in $\text{span}(K_{i\to D} \cap K_{j\to D})$, so we do not have anything to prove. If there is such a branch node k, then the output edge e of k with $\rho(e) = \rho(f_2)$ contains a symbol influenced by i and not j. We may now follow the $\rho = \rho(e)$ path from here to find a constraint on X_{e_l}. If a comparison fails further along causing the forwarding pattern to change such that the $\rho = \rho(e)$ path does not reach the destination, then the potential influence of X_{e_l} on a symbol seen by the destination not influenced by node j is removed, so again we do not have anything to prove.

– *The $\rho = \rho(e_l)$ path leaves the network without either of the above occurring*: Immediately before leaving the network, the symbol will be compared with a non-destination symbol. This comparison must succeed, because j cannot influence the non-destination symbol. This gives a constraint $\tilde{X}_{e_l}$.

We may classify the fates of the two symbols out of i as discussed above as follows:

1. Either the forwarding pattern changes such that the symbol does not reach the destination, or the symbol is in $\text{span}(K_{i\to D} \cap K_{j\to D})$, and so we do not need to prove that it cannot be corrupted. Either way, we may ignore this symbol.
2. The symbol leaves the network, immediately after a successfully comparison with a non-destination symbol.
3. The symbol is successfully compared with a symbol influenced by node j. In particular, this symbol from node j has a strictly smaller ρ value than $\rho(e_l)$.

We divide the situation based on which of the above cases occur for $l = 1, 2$ as follows:

- *Case 1 occurs for both* $l = 1, 2$: We have nothing to prove.
- *Case 1 occurs for (without loss of generality)* $l = 1$: Either case 2 or 3 gives a successful comparison involving a symbol influenced by $\tilde{X}_{e_{\bar{l}}}$. Applying Corollary 1 allows us to conclude that $\tilde{X}_{e_{\bar{l}}}$ cannot be corrupted.
- *Case 2 occurs for both* $l = 1, 2$: If the two paths reach different non-destination symbols, then we may apply Lemma 5 to conclude that node i cannot corrupt either $\tilde{X}_{e_1}$ nor $\tilde{X}_{e_2}$. Suppose, on the other hand, that each path reaches the same non-destination path, in particular the one associated with 2-to-1 node v. Since $\phi(e_1) \neq \phi(e_2)$, assume without loss of generality that $\phi(e_1) \neq v$. We may follow the path starting from e_1 through $\Gamma(v)$ to find an additional constraint, after which we may apply Corollary 2. All symbols on this path are influenced by $\tilde{X}_{e_1}$. This path eventually crosses the non-destination path associated with v. If the symbol compared against the non-destination symbol at this point is not influenced by j, then the comparison succeeds, giving an additional constraint. Otherwise, there are two possibilities:
  - *The path through* $\Gamma(v)$ *reaches* j: There must be a branch node on the path to $\Gamma(v)$ before reaching j such that the path from e_1 has the smaller ρ value. If there were not, then case 1 would have occurred. Consider the most recent such branch node k in $\Gamma(v)$ before reaching j. Let f_1, f_2 be the input edges to k, where f_1 is on the path from e_1. We know $\rho(f_1) < \rho(f_2)$. The comparison at k must succeed. Moreover, this successful comparison comprises

a substantial constraint, because the only way the destination can decode X_{f_2} is through symbols influenced by node j.

* *The path through $\Gamma(v)$ does not reach j*: Let k be the first common node on the paths from i and j through $\Gamma(v)$. Let f_1, f_2 be the input edges of k, where f_1 is on the path from i and f_2 is on the path from j. If the comparison at k succeeds, this provides a constraint. If it fails, then the forwarding pattern changes such that the $\rho = \rho(f_1)$ path becomes a non-destination path. Since we are not in case 1, $\rho(e_1) \neq \rho(f_1)$, but a symbol influenced by X_{e_1} is compared against a symbol on the $\rho = \rho(f_1)$ path at a branch node in $\Gamma(v)$. This comparison must succeed, providing an additional constraint.

– *Case 3 occurs for (without loss of generality) $l = 1$, and either case 2 or 3 occurs for $l = 2$*: We now suppose instead that node j is the traitor. That is, we will prove (2.119) instead of (2.118). Recall that a successful comparison occurs at a branch node k with input edges f_1, f_2 where $\tilde{X}_{f_1}$ is influenced by $\tilde{X}_{e_1}$, $\tilde{X}_{f_2}$ is influenced by node j, and $\rho(f_2) < \rho(f_1)$. Let e_1', e_2' be the output edges of node j, and suppose that $\rho(e_1') = \rho(f_2)$; i.e. the symbol X_{f_2} is influenced by $X_{e_1'}$. The success of the comparison gives a constraint on $\tilde{X}_{e_1'}$. Since $\rho(f_2) < \rho(f_1)$, we may continue to follow the $\rho = \rho(f_2)$ path from node k, and it continues to be not influenced by node i. As above, we may find an additional constraint on $X_{e_1'}$ by following this ρ path until reaching a non-destination symbol or reaching another significant branch node. Furthermore, we may find a constraint on $\tilde{X}_{e_2'}$ in a similar fashion. This gives three constraints on $\tilde{X}_{e_1'}, \tilde{X}_{e_2'}$, enough to apply Corollary 2, and conclude that node j cannot corrupt its output symbols.

- $i \in \mathcal{W}_3 \cup \mathcal{V}_{2,1} \setminus \mathcal{N}_{\text{in}}(D), j \in \mathcal{W}_1 \cup \mathcal{W}_2$: Assume node i is the traitor. If $i \in \mathcal{V}_{2,1}$ with single output edge e such that $\psi(e) = 1$, then node i controls no symbols received at the destination and we have nothing to prove. Otherwise, it controls just one symbol received at the destination, so any single constraint on node i is enough. Let e' be the output symbol of i with $\psi(i) = 0$. Since we assume $i \notin \mathcal{N}_{\text{in}}(D)$, the $\rho = \rho(e')$ path is guaranteed to cross a non-destination path after node i. As above, follow the $\rho = \rho(e')$ path until reaching a branch node k at which the symbol is combined with one influenced by node j. If the comparison at node k succeeds, it gives a constraint on $\tilde{X}_{e'}$. If the comparison fails, then the forwarding pattern will change such that the $\rho = \rho(e')$ path will fail to reach the destination, so we're done.
- $i \in \mathcal{W}_1 \cup \mathcal{W}_2, j \in \mathcal{N}_{\text{in}}(D)$: Assume node i is the traitor. By construction, since one output edge of j goes directly into the destination, the other must be on a non-destination path. Hence, j only controls one symbol at the destination, so we again need to place only one constraint on node i. Let $e \in \mathcal{E}_{\text{out}}(i)$ be such that $\phi(e) \neq \phi(e')$ for all $e' \in \mathcal{E}_{\text{out}}(j)$. This is always possible, since the two output edges of i have different ϕ values, and since one output edge of j goes directly to the destination, only one of the output edges of j has a ϕ value. Let $v = \phi(e)$. Follow the path from e through $\Lambda(v)$ until reaching the non-destination symbol at node k with input edges f_1, f_2. Assume $\tilde{X}_{f_1}$ is influenced by $\tilde{X}_e$ and $\tilde{X}_{f_2}$ is a non-destination symbol. The comparison between these two symbols must succeed, because node j cannot influence either $\tilde{X}_{f_1}$ or $\tilde{X}_{f_2}$. This places the necessary constraint on $\tilde{X}_e$.
- $i, j \in \mathcal{W}_3 \cup \mathcal{V}_{2,1}$: Nodes i, j each control at most one symbol available at the destination, so either one, in order to make it appear as if the other could be the traitor, cannot corrupt anything.

2.10.5 Proof of Theorem 4 when the Cut-set Bound is $M-3$

We now briefly sketch the proof of Theorem 4 for the case that the cut-set bound is $M-3$. The proof is far less complicated than the above proof for the $M-2$ case, but it makes use of many of the same ingredients. First note that the set of 2-to-2 nodes i that cannot reach any 2-to-1 nodes must form a path. We next perform a similar edge labeling as above, defining ϕ and ψ as in (2.86)–(2.87). Properties (A) and (B) must still hold, except that edges may have null labels, and property (C) is replaced by

C' For every 2-to-2 node that can reach at least one 2-to-1 node, at least one of its output edges must have a non-null label.

Internal nodes operate in the same way based on the edge labels as above, where symbols are always forwarded along edges with null labels. The decoding process is the same. Proving an analogous version of Lemma 6 requires only finding a single constraint on one of i or j. This is always possible since one is guaranteed to have a label on an output edge, unless they are both in the single path with no reachable 2-to-1 nodes, in which case they influence the same symbol reaching the destination.

Interestingly, this proof does not make use of the planarity of the graph. We may therefore conclude that for networks satisfying properties (2) and (3) in the statement of Theorem 4, the cut-set bound is always achievable if the cut-set is strictly less than $M-2$.

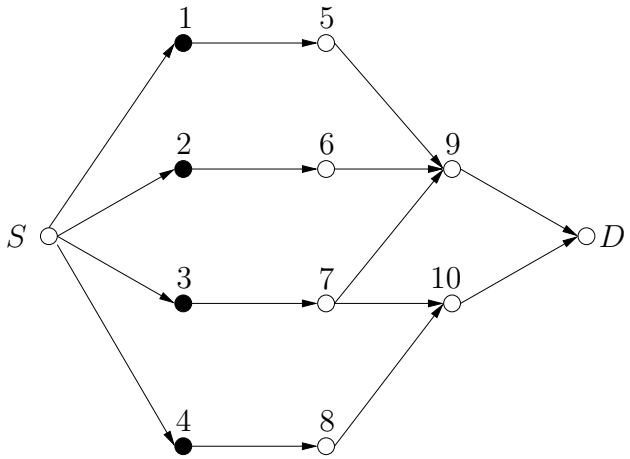


Figure 2.12: The Calamari Network, having capacity strictly less than the cut-set bound. All edges have unit-capacity. There is at most one traitor, but it is restricted to be one of the black nodes.

2.11 Looseness of the Cut-set Bound

So far, the only available upper bound on achievable rates has been the cut-set bound. We have conjectured that for planar graphs this bound is tight, but that still leaves open the question of whether there is a tighter upper bound for non-planar graphs. It was conjectured in [37] that there is such a tighter bound, and here we prove this conjecture to be true. We have already shown in Sec. 2.3 that the limited-node and all-node problems are equivalent. Fig. 2.12 shows the Calamari[1] Network, a limited-node problem for which there is an active upper bound on capacity other than the cut-set. It is easy to see that in the transformation from limited-node to all-node used to prove their equivalence in Sec. 2.3 does not change the cut-set bound. Therefore, the looseness of the cut-set bound for the Calamari Network implies that even for the all-node problem, the cut-set bound is not tight in general. Furthermore, it is not hard to transform the Calamari Network into an unequal-edge problem; this therefore confirms the conjecture in [37].

[1]Calamari is the cockroach of the sea. I think.

In the Calamari Network, there may be at most one traitor, but it is restricted to be one of the black nodes. The cut-set bound is 2, but in fact the capacity is no more than 1.5.

Consider a code achieving rate R. For $i = 1, 2, 3, 4$, let X_i be the random variable representing the value on the output edge of node i. Let Y be the value on edge $(9, D)$ and let Z be the value on $(10, D)$. Let p be the honest distribution on these variables, and define the following alternative distributions:

$$q_3 = p(x_1x_2x_4)p(x_3)p(y|x_1x_2x_3)p(z|x_3x_4), \tag{2.120}$$

$$q_4 = p(x_1x_2x_3)p(x_4)p(y|x_1x_2x_3)p(z|x_3x_4). \tag{2.121}$$

We may write

$$R \leq I_{q_3}(X_1X_2X_4; YZ) \tag{2.122}$$

because, if node 3 is the traitor, it may generate a completely independent version of X_3 and send it along edge $(3, 7)$, resulting in the distribution q_3. In that case, assuming the destination can decode properly, information about the message must get through from the honest edges at the start of the network, X_1, X_2, X_4, to what is received at the destination, Y, Z. From (2.122), we may write

$$R \leq I_{q_3}(X_1X_2X_4; Z) + I_{q_3}(X_1X_2X_4; Y|Z) \tag{2.123}$$

$$\leq I_{q_3}(X_4; Z) + I(X_1X_2; Z|X_4) + 1 \tag{2.124}$$

$$= I_{q_3}(X_4; Z) + 1 \tag{2.125}$$

where in (2.124) we have used that the capacity of $(9, D)$ is 1, and in (2.125) that $X_1X_2 - X_4 - Z$ is a Markov chain according to q_3. Using a similar argument in which node 4 is the traitor and it acts in a way to produce q_4, we may write

$$R \leq I_{q_4}(X_3; Z) + 1. \tag{2.126}$$

Note that

$$q_3(x_3x_4z) = q_4(x_3x_4z). \tag{2.127}$$

In particular, the mutual informations in (2.125) and (2.126) can both be written with respect to the same distribution. Therefore,

$$\begin{aligned} 2R &\leq I_{q_3}(X_4; Z) + I_{q_3}(X_3; Z) + 2 && (2.128)\\ &= I_{q_3}(X_3X_4; Z) + I_{q_3}(X_3; X_4) - I_{q_3}(X_3; X_4|Z) + 2 && (2.129)\\ &\leq I_{q_3}(X_3X_4; Z) + 2 && (2.130)\\ &\leq 3 && (2.131) \end{aligned}$$

where (2.130) follows from the positivity of conditional mutual information and that X_3, X_4 are independent according to q_3, and (2.131) follows because the capacity of $(10, D)$ is 1. Therefore, $R \leq 1.5$.

Observe that all inequalities used in this upper bound were so-called Shannon-type inequalities. For the non-Byzantine problem, there is a straightforward procedure to write down all the Shannon-type inequalities relevant to a particular network coding problem, which in principle can be used to find an upper bound. This upper bound is more general than any cut-set upper bound, and in some multi-source problems it has been shown to be tighter than any cut-set bound. This example illustrates that a similar phenomenon occurs in the Byzantine problem even for a single source and single destination. As the Byzantine problem seems to have much in common with the multi-source non-Byzantine problem, it would be worthwhile to formulate the tightest possible upper bound using only Shannon-type inequalities. However, it is yet unclear what the "complete" list of Shannon type inequalities would be for the Byzantine problem. This example certainly demonstrates one method of finding them, but whether there are fundamentally different methods to find inequalities that could still be called Shannon-type, or

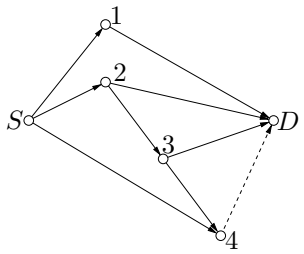


Figure 2.13: The Beetle Network. All edges have unit-capacity except the dashed edge, which has zero capacity.

even how to compile all inequalities using this method, is unclear. Moreover, it has been shown in the non-Byzantine problem that there can be active non-Shannon-type inequalities. It is therefore conceivable that non-Shannon-type inequalities could be active even for a single source under Byzantine attack.

2.12 More on Cut-Set Bounds

We first give an example network illustrating the necessity of requiring no backwards edges in Theorem 3. This example—the Beetle network, shown in Fig. 2.13—is also interesting in that it has a zero-capacity edge which strictly increases capacity. We then proceed to state and prove a cut-set bound tighter than Theorem 3, which allows cuts with backwards edges but has a more elaborate method of determining the upper bound given a cut. For other cut-set bounds on adversarial problems, see [37, 38].

2.12.1 The Beetle Network

The Beetle Network, shown in Figure 2.13, under the presence of a single traitor node, has two interesting properties. First, there is a cut with a backwards edge for

which the value of the right hand side of (2.4) is strictly less than capacity. This illustrates the need for the condition in Theorem 3 that cuts have no backwards edges. Second, it has a zero capacity edge, the presence of which has a positive effect on the capacity. That is, the capacity of this network, as we will demonstrate, is 1, but if the zero-capacity edge $(4, D)$ were removed, the capacity would be 0, as can easily be verified by Theorem 3. The reason for this is that, as we have seen, comparison operations can increase capacity, so we can use the zero-capacity edge to hold a comparison bit.

We may apply Theorem 3 with $A = \{S, 1, 2, 3, 4\}$ and $T = \{1, 2\}$ to conclude that the capacity is no more than 1. We will shortly present a code to achieve rate 1. Now consider the cut $A = \{S, 1, 2, 4\}$. For this cut $(3, 4)$ is a backwards edge, so we cannot apply Theorem 3. Note that if we set $T = \{1, 2\}$, the right hand side of (2.4) would evaluate to 0, strictly less than capacity.

We now present a simple linear code with a comparison for the Beetle Network achieving rate 1. Each unit-capacity edge carries a copy of the message w. That is, the source sends w along all three of its output links, and nodes 1, 2, and 3 each receive one copy of w and forward it along all of their output links. Node 4 receives a copy of w from the source and and one from node 3. It compares them and sends to the destination one of the symbols $=$ or $\neq$ depending on whether the two copies agreed. Because w may be a vector of arbitrary length, sending this single bit along edge $(4, D)$ takes zero rate, so we do not exceed the edge capacity.

The decoding procedure is as follows. Let w_1, w_2, and w_3 be the values of w received at the destination from nodes 1, 2, and 3 respectively. If either $w_2 \neq w_3$ or the destination receives $\neq$ from node 4, then certainly the traitor must by one of nodes 2, 3, or 4, so w_1 is trustworthy and the destination decodes from it. Now

consider the case that $w_2 = w_3$ and the destination receives = from node 4. The destination decodes from w_2 or w_3. Certainly if the traitor is either node 1 or 4, then $w_2 = w_3 = w$. If the traitor is node 3, then $w_2 = w$, so we still decode correctly. If the traitor is node 2, then it must send the same value of w to both the destination and node 3, because node 3 simply forwards its copy to the destination, and we know $w_2 = w_3$. Furthermore, this value of w must be the true one, because otherwise node 4 would observe that the copy sent along edge $(3, 4)$ is different from that sent from the source, so it would transmit $\neq$ to the destination. Since it did not, node 2 cannot have altered any of its output values. Therefore the destination always decodes correctly.

2.12.2 Tighter Cut-Set Upper Bound

The following theorem is a tighter cut-set bound than Theorem 3, as it allows cuts with backwards edges.

Theorem 6 *Fix a cut $A \subseteq V$ with $S \in A$ and $D \notin A$. Also fix sets of nodes T and T^* with $T^* \subset T$ and*

$$|T| + |T^*| \leq 2s. \tag{2.132}$$

Let B be the set of nodes that can reach a node in $A \setminus T$. Then

$$C \leq |\{(i,j) \in E : i \in A \setminus T,\ j \notin A\}| + |\{(i,j) \in E : i \in A \cap T \setminus T^*,\ j \in A^c \cap B\}|. \tag{2.133}$$

Proof: Choose a coding order on the nodes in $T \setminus T^*$ written as

$$(t_1, \ldots, t_{|T \setminus T^*|}). \tag{2.134}$$

That is, if there is a path from t_u to t_v, then $u \leq v$. Let

$$T_1 = T^* \cup \{t_1, \ldots, t_{s-|T^*|}\}, \tag{2.135}$$

$$T_2 = T^* \cup \{t_{s-|T^*|}, \ldots, t_{|T \setminus T^*|}\}. \tag{2.136}$$

Note that $|T_1|, |T_2| \leq s$ and $T_1 \cup T_2 = T$. For $l = 1, 2$, let E_l be the set of edges (i, j) with $i \in T_1 \setminus T^*$ and $j \in A^c \setminus B$. Let E^* be the set of edges (i, j) with $i \in T^*$ and $j \in A^c$. Finally, let E_A be the set of edges crossing the cut; that is, edges (i, j) with $i \in A$ and $j \notin A$. Let $\tilde{E} = E_A \setminus E_1 \setminus E_2 \setminus E^*$. Observe that (2.133) can equivalently be written

$$C \leq |\tilde{E}|. \tag{2.137}$$

Suppose (2.133) were not true. Then there would exist a code achieving a rate R such that

$$R > |\tilde{E}|. \tag{2.138}$$

We will consider two possibilities, one when T_1 are the traitors and they alter the values on $E_1 \cup E^*$, and one when T_2 are the traitors and they alter the values on $E_2 \cup E^*$. Note that there are may be edges out of the set of traitors whose values are not altered; on these edges the traitors will act honestly, performing the code as it is designed. We will show that by (2.138), it is possible for the traitors to act in such a way in these two cases that even though the messages at the source are different, all values sent across the cut are the same; therefore the destination will not be able to distinguish all messages.

Let x_{E^*} be one possible value sent on the edges in E^*. Both possible sets of traitors may influence the values on E^*, and in both cases they will place x_{E^*} on these edges. For any set of edges F, define the function

$$X_F : 2^{nR} \times \prod_{e \in E^*} 2^n \to \prod_{e \in F} 2^n \tag{2.139}$$

such that when the message is w, and all nodes act honestly except for T^* which place x_{E^*} on E^*, the values on edges in F is given by $X_F(w, x_{E^*})$.

Consider an edge $(i, j) \in \tilde{E}$. We claim that the value on this edge depends only on the message and x_{E^*}; it does not depend on the values placed on E_1 or E_2 by the traitors. If i is a traitor, then by construction i acts honestly on this edge. Consider any path from the source passing through (i, j). We wish to show that at no point a value is placed on an edge in this path that deviates from the honest code, except at edges in E^*. The only other point at which it might occur would be at an earlier edge (i', j'). However, (i', j') is on a path leading to i. If $i \in A \setminus T$, then $j' \in B$, so $(i', j') \notin E_1 \cup E_2$, so the value on this edge is not changed by the traitor. If $i \in T$, then j must be in B, meaning j' is also in B, so again $(i', j') \notin E_1 \cup E_2$. Therefore the values placed on $\tilde{E}$ is exactly $X_{\tilde{E}}(w, x_{E^*})$ no matter which set of nodes T_1 or T_2 is the traitor. By (2.138), there exists two messages w_1 and w_2 such that

$$X_{\tilde{E}}(w_1, x_{E^*}) = X_{\tilde{E}}(w_2, x_{E^*}). \tag{2.140}$$

We now specify the two cases that confuse messages w_1 and w_2 at the destination:

1. The true message is w_1 and the traitors are T_1. They place x_{E^*} on E^* and $X_{E_1}(w_2, x_{E^*})$ on E_1. Let x'_{E_2} be the value placed on E_2 in this case. Recall that the values on $\tilde{E}$ are given by (2.140).
2. The true message is w_2 and the traitors are T_2. They place x_{E^*} on E^* and x'_{E_2} on E_2. Again, the values on $\tilde{E}$ are given by (2.140). Moreover, because of our choice of T_1 and T_2 in terms of the coding order in (2.135)–(2.136), edges in E_2 are entirely downstream of those in E_1, so the values on E_1 are

$X_{E_1}(w_2, x_{E^*})$.

The values on all edges crossing the cut in both cases are the same. Therefore, all values received by the destination are the same, so it must make an error on one of the two messages. □

Note that if A has no backwards edges, then $B \subset A$, so the second term in (2.133) would be 0. Hence we recover Theorem 3.

We briefly illustrate an application of Theorem 6 on the Beetle Network for the cut with a backwards edge. Let $A = \{S, 1, 2, 4\}$ and $T = \{1, 2\}$. The set B consists of $\{S, 1, 2, 3, 4\}$, so the second term in (2.133) counts the edge $(2, 3)$. Therefore (2.133) gives an upper bound is 1. This is a correct bound, even though it would not be had the second term in (2.133) not been included.

2.13 Proof of Bound on Linear Capacity for the Cockroach Network

We show that no linear code for the Cockroach Network, shown in Figure 2.1, can achieve a rate higher than 4/3. Fix any linear code. For any link (i, j), let $X_{i,j}$ be the value placed on this link. For every node i, let X_i be the set of messages on all links out of node i, and Y_i be the set of messages on all links into node i. Let $G_{X_i \to Y_j}$ be the linear transformation from X_i to Y_j, assuming all nodes behave honestly. Observe that

$$Y_D = G_{X_S \to Y_D} X_S(w) + \sum_i G_{X_i \to Y_D} e_i \tag{2.141}$$

where e_i represents the difference between what a traitor places on its outgoing links and what it would have placed on those links if it were honest. Only one node is a traitor, so at most one of the e_i is nonzero. Note also that the output values of the source X_S is a function of the message w. We claim that for any achievable rate R,

$$R \leq \frac{1}{n}\left[\text{rank}(G_{X_S \to Y_D}) - \max_{i,j} \text{rank}(G_{X_i X_j \to Y_D})\right] \tag{2.142}$$

where n is the block length used by this code. To show this, first note that for any pair of nodes i, j there exist K, H_1, H_2 such that

$$G_{X_S \to Y_D} = K + G_{X_i \to Y_D} H_1 + G_{X_j \to Y_D} H_2 \tag{2.143}$$

and where

$$\text{rank}(K) = \text{rank}(G_{X_S \to Y_D}) - \text{rank}(G_{X_i X_j \to Y_D}). \tag{2.144}$$

That is, the first term on the right hand side of (2.143) represents the part of the transformation from X_S to Y_D that cannot be influenced by X_i or X_j. Consider the case that $\text{rank}(K) < R$. Then there must be two messages w_1, w_2 such that $KX_S(w_1) = KX_S(w_2)$. If the message is w_1, node i may be the traitor and set

$$e_i = H_1(X_S(w_2) - X_S(w_1)). \tag{2.145}$$

Alternatively, if the message is w_2, node j may be the traitor and set

$$e_j = H_2(X_S(w_1) - X_S(w_2)). \tag{2.146}$$

In either case, the value received at the destination is

$$Y_D = KX_S(w_1) + G_{X_i \to Y_D} H_1 X_S(w_2) + G_{X_j \to Y_D} H_2 X_S(w_1).$$

Therefore, these two cases are indistinguishable to the destination, so it must make an error for at least one of them. This proves (2.142).

Now we return to the specific case of the Cockroach Network. Observe that the $X_{4,D}$ is a linear combination of $X_{1,4}$ and $X_{2,4}$. Let k_1 be the number of dimensions of $X_{4,D}$ that depend only on $X_{1,4}$ and are independent of $X_{2,4}$. Let k_2 be the number of dimensions of $X_{4,D}$ that depend only on $X_{2,4}$, and let k_3 be the number of dimensions that depend on both $X_{1,4}$ and $X_{2,4}$. Certainly $k_1 + k_2 + k_3 \leq n$. Similarly, let l_1, l_2, l_3 be the number of dimensions of $X_{5,D}$ that depend only on $X_{2,5}$, that depend only on $X_{3,5}$, and that depend on both respectively. Finally, let m_1 and m_2 be the number of dimensions of $X_{1,D}$ and $X_{3,D}$ respectively.

We may write the following:

$$\begin{aligned}
\operatorname{rank}(G_{X_S \to Y_4}) - \operatorname{rank}(G_{X_2, X_3 \to Y_4}) &\leq m_1 + k_1, \\
\operatorname{rank}(G_{X_S \to Y_4}) - \operatorname{rank}(G_{X_1, X_3 \to Y_4}) &\leq k_3 + l_1, \\
\operatorname{rank}(G_{X_S \to Y_4}) - \operatorname{rank}(G_{X_1, X_2 \to Y_4}) &\leq l_3 + m_2.
\end{aligned}$$

Therefore, using (2.142), any achievable rate R is bounded by

$$R \leq \frac{1}{n} \min\{m_1 + k_1, k_3 + l_1, l_3 + m_2\} \tag{2.147}$$

subject to

$$k_1 + k_2 + k_3 \leq n, \tag{2.148}$$

$$l_1 + l_2 + l_3 \leq n, \tag{2.149}$$

$$m_1 \leq n, \tag{2.150}$$

$$m_2 \leq n. \tag{2.151}$$

It is not hard to show that this implies $R \leq 4/3$.

CHAPTER 3

SLEPIAN-WOLF

3.1 Introduction

Fig. 3.1 shows the multiterminal source coding problem of Slepian and Wolf [40]. At each time $t = 1, \ldots, n$, a source generates an independent copy of the correlated random variables $Y_1, \ldots, Y_L$ according to the distribution $p(y_1 \cdots y_L)$. Each sequence Y_i^n is delivered to the corresponding node i. The L nodes operate independently of one another. Node i encodes its observation at rate R_i and transmits the encoded version to a common decoder, which attempts to exactly recover all the sources with high probability. Slepian and Wolf characterized in [40] the complete achievable rate region for this problem—that is, the set of rate vectors $(R_1, \ldots, R_L)$ at which it is possible for the decoder to recover all sources—and they found that the sum-rate can be made as low as the joint entropy of all sources:

$$H(Y_1 \cdots Y_L). \tag{3.1}$$

This is precisely the minimum rate that could be achieved if all the sources were observed by a single node, as was originally shown by Shannon [49] in his source coding theorem. The surprising result of Slepian-Wolf, then, is that no additional sum-rate is required when the nodes are separated from each other.

In this chapter, we consider a modification to this classic problem in which an adversary controls an unknown subset of nodes, and may transmit arbitrary data to the decoder from those nodes. It is obvious that observations made by these traitors are irretrievable unless the traitors choose to deliver them to the decoder. Thus the best the decoder can hope to achieve is to reconstruct the observations

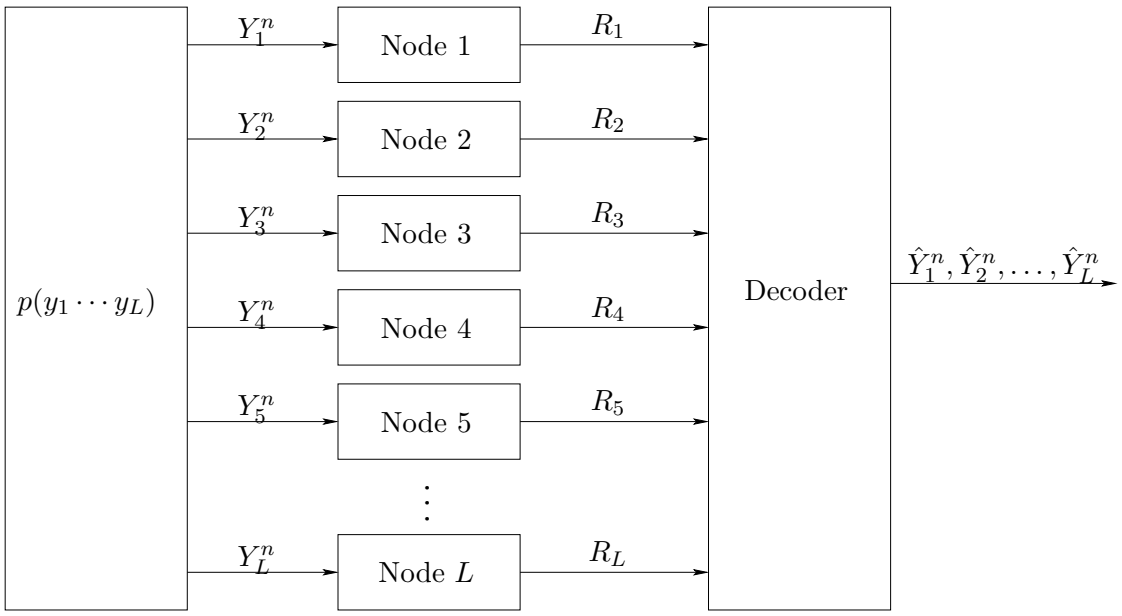


Figure 3.1: The Slepian-Wolf multiterminal source coding problem. The sources $Y_1^n, \ldots, Y_L^n$ are independent and identically distributed in time, and correlated in space as specified by the joint distribution $p(y_1 \cdots y_L)$. Each source sequence Y_i^n is observed by node i and encoded at rate R_i to a common decoder. The decoder produces an estimate $\hat{Y}_i^n$ for each source sequence, attempting to match the sources exactly with high probability.

of the honest nodes. A simple procedure is to ignore the statistical correlations among the observations and collect data from each node individually. The total sum rate of such an approach is $\sum_i H(Y_i)$. One expects however that this sum rate can be lowered if the correlation structure is not ignored.

Standard coding techniques for the Slepian-Wolf problem have no mechanism for handling any deviations from the agreed-upon encoding functions by the nodes. Even a random fault by a single nodes could have devastating consequences for the accuracy of the source estimates produced at the decoder, to say nothing of a Byzantine attack on multiple nodes. In particular, because Slepian-Wolf coding takes advantage of the correlation among sources, manipulating the codeword for one source can alter the accuracy of the decoder's estimate for other sources. It will turn out that for most source distributions, the sum rate given in (3.1) cannot

be achieved if there is even a single traitor. Our goal is to characterize the lowes achievable sum-rate for this problem, and in some cases the complete achievable rate region.

3.1.1 Redefining Achievable Rate

The nature of Byzantine attack require three modifications to the usual notion of achievable rate. The first, as mentioned above, is that small probability of error is required only for honest sources, even though the decoder may not know which sources are honest. This requirement is reminiscent of [5], in which the lieutenants generals need only perform the commander general's order if the commander is not a traitor, even though the lieutenants might not be able to decide this with certainty.

The next modification is that there must be small probability of error no matter what the traitors do. This is essentially the definition of Byzantine attack.

The final modification has to do with which nodes are allowed to be traitors. Let $\mathcal{H}$ be the set of honest nodes, and $\mathcal{T} = \{1, \cdots, L\} \backslash \mathcal{H}$ the set of traitors. A statement that a code achieves a certain rate must include the list of sets of nodes that this code can handle as the set of traitors. That is, given such a list, we say that a rate is achieved if there exists a code with small probability of error when the actual set of traitors is in fact on the list. Hence a given code may work for some lists and not others, so the achievable rates will depend on the specified list. It will be more convenient to specify not the list of allowable sets of traitors, but rather the list of allowable sets of honest nodes. We define $\mathfrak{H} \subset 2^{\{1,\cdots,L\}}$ to be this list. Thus small probability of error is required only when $\mathcal{H} \in \mathfrak{H}$. One special

case is when the code can handle any group of at most s traitors. That is,

$$\mathfrak{H} = \mathfrak{H}_s \triangleq \{\mathcal{S} \subset \{1, \cdots, m\} : |\mathcal{S}| \geq L - s\}.$$

Observe that achievable rates depend not just on the true set of traitors but also on the collection $\mathfrak{H}$, because the decoder's willingness to accept more and more different groups of traitors allows the true traitors to get away with more without being detected. Thus we see a trade off between rate and security—in order to handle more traitors, one needs to be willing to accept a higher rate.

3.1.2 Fixed-Rate Versus Variable-Rate Coding

In standard source coding, an encoder is made up of a single encoding function. We will show that this fixed-rate setup is suboptimal for this problem, in the sense that we can achieve lower sum rates using variable-rate coding. By variable-rate we mean that the number of bits transmitted per source value by a particular node will not be fixed. Instead, the decoder chooses the rates at "run time" in the following way. Each node has a finite number of encoding functions, all of them fixed beforehand, but with potentially different output alphabets. The coding session is then made up of a number of transactions. Each transaction begins with the decoder deciding which node will transmit, and which of its several encoding functions it will use. The node then executes the chosen encoding function and transmits the output back to the decoder. Finally, the decoder uses the received message to choose the next node and encoding function, beginning the next transaction, and so on. Thus a code is made up of a set of encoding functions for each node, a method for the decoder to choose nodes and encoding functions based on previously received messages, and lastly a decoding function that takes all received messages and produces source estimates.

Note that the decoder has the ability to transmit some information back to the nodes, but this feedback is limited to the choice of encoding function. Since the number of encoding functions need not grow with the block length, this represents zero rate feedback.

In variable-rate coding, since the rates are only decided upon during the coding session, there is no notion of an L-dimensional achievable rate region. Instead, we only discuss achievable sum rates.

3.1.3 Traitor Capabilities

An important consideration with Byzantine attack is the information to which the traitors have access. First, we assume that the traitors have complete knowledge of the coding scheme used by the decoder and honest nodes. Furthermore, we always assume that they can communicate with each other arbitrarily. For variable-rate coding, they may have any amount of ability to eavesdrop on transmissions between honest nodes and the decoder. We will show that this ability has no effect on achievable rates. We assume with fixed-rate coding that all nodes transmit simultaneously, so it does not make sense that traitors could eavesdrop on honest nodes' transmissions before making their own, as that would violate causality. Thus we assume for fixed-rate coding that the traitors cannot eavesdrop.

The key factor, however, is the extent to which the traitors have direct access to information about the sources. We assume the most general memoryless case, that the traitors have access to the random variable W, where W is i.i.d. distributed with $(Y_1 \cdots Y_L)$ according to the conditional distribution $r(w|y_1 \cdots y_L)$. A natural assumption would be that W always includes Y_i for traitors i, but in fact this need

not be the case. An important special case is where $W = (Y_1, \cdots, Y_L)$, i.e. the traitors have perfect information.

We assume that the distribution of W depends on who the traitors are, and that the decoder may not know exactly what this distribution is. Thus each code is associated with a function $\mathcal{R}$ that maps elements of $\mathfrak{H}$ to sets of conditional distributions r. The relationship between r and $\mathcal{R}(\mathcal{H})$ is analogous to the relationship between $\mathcal{H}$ and $\mathfrak{H}$. That is, given $\mathcal{H}$, the code is willing to accept all distributions $r \in \mathcal{R}(\mathcal{H})$. Therefore a code is designed based on $\mathfrak{H}$ and $\mathcal{R}$, and then the achieved rate depends at run time on $\mathcal{H}$ and r, where we assume $\mathcal{H} \in \mathfrak{H}$ and $r \in \mathcal{R}(\mathcal{H})$. We therefore discuss not achievable rates R but rather achievable rate functions $R(\mathcal{H}, r)$. In fact, this applies only to variable-rate codes. In the fixed-rate case, no run time rate decisions can be made, so achievable rates depend only on $\mathfrak{H}$ and $\mathcal{R}$.

3.1.4 Main Results

Our main results give explicit characterizations of the achievable rates for three different setups. The first, which is discussed in the most depth, is the variable-rate case, for which we characterize achievable sum rate functions. The other two setups are for fixed-rate coding, divided into deterministic and randomized coding, for which we give L-dimensional achievable rate regions. We show that randomized coding yields a larger achievable rate region than deterministic coding, but we believe that in most cases randomized fixed-rate coding requires an unrealistic assumption. In addition, even randomized fixed-rate coding cannot achieve the same sum rates as variable-rate coding.

We give the exact solutions later, but describe here some intuition behind them.

For variable-rate, the achievable rates, given in Theorem 7, are based on alternate distributions on $(Y_1 \cdots Y_L)$. Specifically, given W, the traitors can simulate any distribution $\bar{q}(y_{\mathcal{T}}|w)$ to produce a fraudulent version of $Y_{\mathcal{T}}^n$, then report this sequence as the truth. Suppose that the overall distribution $q(y_1 \cdots y_L)$ governing the combination of the true value of $Y_{\mathcal{H}}^n$ with this fake value of $Y_{\mathcal{T}}^n$ could be produced in several different ways, with several different sets of traitors. In that case, the decoder cannot tell which of these several possibilities is the truth, which means that from its point of view, many nodes might be honest. Since the error requirement described in 3.1.1 stipulates that the decoder must produce a correct estimate for every honest node, it must attempt to decode the source values associated with each potentially honest node. Thus the sum rate must be at least the joint entropy, when distributed according to q, of the sources associated with all potentially honest nodes. The supremum over all possible simulated distributions is the achievable sum rate.

For example, suppose $\mathfrak{H} = \mathfrak{H}_{L-1}$. That is, at most one node is honest. Then the traitors are able to create the distribution $q(y_1 \cdots y_L) = p(y_1) \cdots p(y_L)$ no matter which group of $L-1$ nodes are the traitors. Thus every node appears as if it could be the honest one, so the minimum achievable sum rate is

$$H(Y_1) + \cdots + H(Y_L). \tag{3.2}$$

In other words, the decoder must use an independent source code for each node, which requires receiving $nH(Y_i)$ bits from node i for all i.

The achievable fixed-rate regions, given in Theorem 8, are based on the Slepian-Wolf achievable rate region. For randomized fixed-rate coding, the achievable region is such that for all $\mathcal{S} \in \mathfrak{H}$, the rates associated with the nodes in $\mathcal{S}$ fall into the Slepian-Wolf rate region on the corresponding random variables. Note that

for $\mathfrak{H} = \{\{1, \cdots, L\}\}$, this is identical to the Slepian-Wolf region. For $\mathfrak{H} = \mathfrak{H}_{L-1}$, this region is such that for all i, $R_i \geq H(Y_i)$, which corresponds to the sum rate in (3.2). The deterministic fixed-rate achievable region is a subset of that of randomized fixed-rate, but with an additional constraint stated in Section 3.6.

3.1.5 Randomization

Randomization plays a key role in defeating Byzantine attacks. As we have discussed, allowing randomized encoding in the fixed-rate situation expands the achievable region. In addition, the variable-rate coding scheme that we propose relies heavily on randomization to achieve small probability of error. In both fixed and variable-rate coding, randomization is used as follows. Every time a node transmits, it randomly chooses from a group of essentially identical encoding functions. The index of the chosen function is transmitted to the decoder along with its output. Without this randomization, a traitor that transmits before an honest node i would know exactly the messages that node i will send. In particular, it would be able to find fake sequences for node i that would produce those same messages. If the traitor tailors the messages it sends to the decoder to match one of those fake sequences, when node i then transmits, it would appear to corroborate this fake sequence, causing an error. By randomizing the choice of encoding function, the set of sequences producing the same message is not fixed, so a traitor can no longer know with certainty that a particular fake source sequence will result in the same messages by node i as the true one. This is not unlike Wyner's wiretap channel [28], in which information is kept from the wiretapper by introducing additional randomness. See in particular Section 3.5.4 for the proof that variable-rate randomness can defeat the traitors in this manner.

3.1.6 Organization

The rest of this chapter is organized as follows. In Section 3.2, we develop in detail the case that there are three nodes and one traitor, describing a coding scheme that achieves the optimum sum rate. In Section 3.3, we formally give the variable-rate model and present the variable-rate result. In Section 3.4, we discuss the variable-rate achievable rate region and give an analytic formulation for the minimum achievable sum rate for some special cases. In Section 3.6, we give the fixed-rate models and present the fixed-rate result. In Sections 3.5 and 3.7, we prove the variable-rate and fixed-rate results respectively.

3.2 Three Node Example

3.2.1 Potential Traitor Techniques

For simplicity and motivation, we first explore the three-node case with one traitor. That is, $L = 3$ and

$$\mathfrak{H} = \{\{1, 2\}, \{2, 3\}, \{1, 3\}\}.$$

Suppose also that the traitor has access to perfect information (i.e. $W = (Y_1, Y_2, Y_3)$). Suppose node 3 is the traitor. Nodes 1 and 2 will behave honestly, so they will report Y_1 and Y_2 correctly, as distributed according to the marginal distribution $p(y_1 y_2)$. Since node 3 has access to the exact values of Y_1 and Y_2, it may simulate the conditional distribution $p(y_3|y_2)$, then take the resulting Y_3 sequence and report it as the truth. Effectively, then, the three random variables

will be distributed according to the distribution

$$q(y_1y_2y_3) \triangleq p(y_1y_2)p(y_3|y_2).$$

The decoder will be able to determine that nodes 1 and 2 are reporting jointly typical sequences, as are nodes 2 and 3, but not nodes 1 and 3. Therefore, it can tell that either node 1 or 3 is the traitor, but not which one, so it must obtain estimates of the sources from all three nodes. Since the three streams are not jointly typical with respect to the source distribution $p(y_1y_2y_3)$, standard Slepian-Wolf coding on three encoders will not correctly decode them all. However, had we known the strategy of the traitor, we could do Slepian-Wolf coding with respect to the distribution q. This will take a sum rate of

$$H_q(Y_1Y_2Y_3) = H(Y_1Y_2Y_3) + I(Y_1;Y_3|Y_2)$$

where H_q is the entropy with respect to q. In fact we will not do Slepian-Wolf coding with respect to q but rather something slightly different that gives the same rate. Since Slepian-Wolf coding without traitors can achieve a sum rate of $H(Y_1Y_2Y_3)$, we have paid a penalty of $I(Y_1;Y_3|Y_2)$ for the single traitor.

We supposed that node 3 simulated the distribution $p(y_3|y_2)$. It could have just as easily simulated $p(y_3|y_1)$, or another node could have been the traitor. Hence, the minimum achievable sum rate for all $\mathcal{H} \in \mathfrak{H}$ is at least

$$R^* \triangleq H(Y_1Y_2Y_3) + \max\{I(Y_1;Y_2|Y_3), I(Y_1;Y_3|Y_2), I(Y_2;Y_3|Y_1)\}. \qquad (3.3)$$

In fact, this is exactly the minimum achievable sum rate, as shown below.

3.2.2 Variable-Rate Coding Scheme

We now give a variable-rate coding scheme that achieves R^*. This scheme is somewhat different from the one we present for the general case in Section 3.5, but it is much simpler, and it illustrates the basic idea. The procedure will be made up of a number of rounds. Communication from node i in the first round will be based solely on the first n values of Y_i, in the second round on the second n values of Y_i, and so on. The principle advantage of the round structure is that the decoder may hold onto information that is carried over from one round to the next.

In particular, the decoder maintains a collection $\mathfrak{V} \subset \mathfrak{H}$ representing the sets that could be the set of honest nodes. If a node is completely eliminated from $\mathfrak{V}$, that means it has been identified as the traitor. We begin with $\mathfrak{V} = \mathfrak{H}$, and then remove a set from $\mathfrak{V}$ whenever we find that the messages from the corresponding pair of nodes are not jointly typical. With high probability, the two honest nodes report jointly typical sequences, so we expect never to eliminate the honest pair from $\mathfrak{V}$. If the traitor employs the q discussed above, for example, we would expect nodes 1 and 3 to report atypical sequences, so we will drop $\{1, 3\}$ from $\mathfrak{V}$. In essence, the value of $\mathfrak{V}$ contains our current knowledge about what the traitor is doing.

The procedure for a round is as follows. If $\mathfrak{V}$ contains $\{\{1, 2\}, \{1, 3\}\}$, do the following:

1. Receive $nH(Y_1)$ bits from node 1 and decode y_1^n.
2. Receive $nH(Y_2|Y_1)$ bits from node 2. If there is a sequence in $\mathcal{Y}_2^n$ jointly typical with y_1^n that matches this transmission, decode that sequence to y_2^n.

If not, receive $nI(Y_1; Y_2)$ additional bits from node 2, decode y_2^n, and remove $\{1, 2\}$ from $\mathfrak{V}$.

3. Do the same with node 3: Receive $nH(Y_3|Y_1)$ bits and decode y_3^n if possible. If not, receive $nI(Y_1; Y_3)$ additional bits, decode, and remove $\{1, 3\}$ from $\mathfrak{V}$.

If $\mathfrak{V}$ is one of the other two subsets of $\mathfrak{H}$ with two elements, perform the same procedure but replace node 1 with whichever node appears in both elements in $\mathfrak{V}$. If $\mathfrak{V}$ contains just one element, then we have exactly identified the traitor, so ignore the node that does not appear and simply do Slepian-Wolf coding on the two remaining nodes.

Note that the only cases when the number of bits transmitted exceeds nR^* are when we receive a second message from one of the nodes, which happens exactly when we eliminate an element from $\mathfrak{V}$. Assuming the source sequences of the two honest nodes are jointly typical, this can occur at most twice, so we can always achieve a sum rate of R^* when averaged over enough rounds.

3.2.3 Fixed-Rate Coding Scheme

In the procedure described above, the number of bits sent by a node changes from round to round. We can no longer do this with fixed-rate coding, so we need a different approach. Suppose node 3 is the traitor. It could perform a black hole attack, in which case the estimates for Y_1^n and Y_2^n must be based only on the messages from nodes 1 and 2. Thus, the rates R_1 and R_2 must fall into the Slepian-Wolf achievability region for Y_1 and Y_2. Similarly, if one of the other nodes was the traitor, the other pairs of rates also must fall into the corresponding Slepian-Wolf

region. Putting these conditions together gives

$$\begin{aligned} R_1 &\geq \max\{H(Y_1|Y_2), H(Y_1|Y_3)\} \\ R_2 &\geq \max\{H(Y_2|Y_1), H(Y_2|Y_3)\} \\ R_3 &\geq \max\{H(Y_3|Y_1), H(Y_3|Y_2)\} \\ R_1 + R_2 &\geq H(Y_1Y_2) \\ R_1 + R_3 &\geq H(Y_1Y_3) \\ R_2 + R_3 &\geq H(Y_2Y_3). \end{aligned} \tag{3.4}$$

If the rates fall into this region, we can do three simultaneous Slepian-Wolf codes, one on each pair of nodes, thereby constructing two estimates for each node. If we randomize these codes using the method described in Section 3.1.5, the traitor will be forced either to report the true message, or report a false message, which with high probability will be detected as such. Thus either the two estimates for each node will be the same, in which case we know both are correct, or one of the estimates will be demonstrably false, in which case the other is correct.

We now show that the region given by (3.4) does not include sum rates as low as R^*. Assume without loss of generality that $I(Y_1; Y_2|Y_3)$ achieves the maximum in (3.3). Summing the last three conditions in (3.4) gives

$$\begin{aligned} R_1 + R_2 + R_3 &\geq \frac{1}{2}\big(H(Y_1Y_2) + H(Y_1Y_3) + H(Y_2Y_3)\big) \\ &= H(Y_1Y_2Y_3) + \frac{1}{2}\big(I(Y_1; Y_2|Y_3) + I(Y_1Y_2; Y_3)\big). \end{aligned} \tag{3.5}$$

If $I(Y_1Y_2; Y_3) > I(Y_1; Y_2|Y_3)$, (3.5) is larger than (3.3). Hence, there exist source distributions for which we cannot achieve the same sum rates with even randomized fixed-rate coding as with variable-rate coding.

If we are interested only in deterministic codes, the region given by (3.4) can no longer be achieved. In fact, we will prove in Section 3.7 that the achievable region

reduces to the trivially achievable region where $R_i \geq H(Y_i)$ for all i when $L = 3$, though it is nontrivial for $L > 3$. For example, suppose $L = 4$ and $\mathfrak{H} = \mathfrak{H}_1$. In this case, the achievable region is similar to that given by (3.4), but with an additional node. That is, each of the 6 pairs of rates must fall into the corresponding Slepian-Wolf region. In this case, we do three simultaneous Slepian-Wolf codes for each node, construct three estimates, each associated with one of the other nodes. For an honest node, only one of the other nodes could be a traitor, so at least two of these estimates must be correct. Thus we need only take the plurality of the three estimates to obtain the correct estimate.

3.3 Variable-Rate Model and Result

3.3.1 Notation

Let Y_i be the random variable revealed to node i, $\mathcal{Y}_i$ the alphabet of that variable, and y_i a corresponding realization. A sequence of random variables revealed to node i over n timeslots is denoted Y_i^n, and a realization of it $y_i^n \in \mathcal{Y}_i^n$. Let $\mathcal{M} \triangleq \{1, \cdots, L\}$. For a set $\mathcal{S} \subset \mathcal{M}$, let $Y_\mathcal{S}$ be the set of random variables $\{Y_i\}_{i \in \mathcal{S}}$, and define $y_\mathcal{S}$ and $\mathcal{Y}_\mathcal{S}$ similarly. By $\mathcal{S}^c$ we mean $\mathcal{M} \backslash \mathcal{S}$. Let $T_\epsilon^n(Y_\mathcal{S})[q]$ be the strongly typical set with respect to the distribution q, or the source distribution p if unspecified. Similarly, $H_q(Y_\mathcal{S})$ is the entropy with respect to the distribution q, or p if unspecified.

3.3.2 Communication Protocol

The transmission protocol is composed of T transactions. In each transaction, the decoder selects a node to receive information from and selects which of K encoding functions it should use. The node then responds by executing that encoding function and transmitting its output back to the decoder, which then uses the new information to begin the next transaction.

For each node $i \in \mathcal{M}$ and encoding function $j \in \{1, \cdots, K\}$, there is an associated rate $R_{i,j}$. On the tth transaction, let i_t be the node and j_t the encoding function chosen by the decoder, and let h_t be the number of $t' \in \{1, \cdots, t-1\}$ such that $i_{t'} = i_t$. That is, h_t is the number of times i_t has transmitted prior to the tth transaction. Note that i_t, j_t, h_t are random variables, since they are chosen by the decoder based on messages it has received, which depend on the source values. The jth encoding function for node i is given by

$$f_{i,j} : \mathcal{Y}_i^n \times \mathcal{Z} \times \{1, \cdots, K\}^{h_t} \to \{1, \cdots, 2^{nR_{i,j}}\} \tag{3.6}$$

where $\mathcal{Z}$ represents randomness generated at the node. Let $I_t \in \{1, \cdots, 2^{nR_{i_t,j_t}}\}$ be the message received by the decoder in the tth transaction. If i_t is honest, then $I_t = f_{i_t,j_t}(Y_{i_t}^n, \rho_{i_t}, J_t)$, where $\rho_{i_t} \in \mathcal{Z}$ is the randomness from node i_t and $J_t \in \{1, \cdots, K\}^{h_t}$ is the history of encoding functions used by node i_t so far. If i_t is a traitor, however, it may choose I_t based on W^n and it may have any amount of access to previous transmissions $I_1, \cdots, I_{t-1}$ and polling history $i_1, \cdots, i_{t-1}, j_1, \cdots, j_{t-1}$. But, it does not have access to the randomness ρ_i for any honest node i. Note again that the amount of traitor eavesdropping ability has no effect on achievable rates.

After the decoder receives I_t, if $t < L$ it uses $I_1, \cdots, I_t$ to choose the next node

i_{t+1} and its encoding function index j_{t+1}. After the Tth transaction, it decodes according to the decoding function

$$g : \prod_{t=1}^{T} \{1, \cdots, 2^{nR_{i_t,j_t}}\} \to \mathcal{Y}_1^n \times \cdots \times \mathcal{Y}_L^n.$$

Note that we impose no restriction whatsoever on the size of the total number of transactions T. Thus, a code could have arbitrary complexity in terms of the number of messages passed between the nodes and the decoder. However, in our below definition of achievability, we require that the communication rate from nodes to decoder always exceeds that from decoder to nodes. Therefore while the number of messages may be very large, the amount of feedback is diminishingly small.

3.3.3 Variable-Rate Problem Statement and Main Result

Let $\mathcal{H} \subset \mathcal{M}$ be the set of honest nodes. Define the probability of error

$$P_e \triangleq \Pr\left(Y_{\mathcal{H}}^n \neq \hat{Y}_{\mathcal{H}}^n\right)$$

where $(\hat{Y}_1^n, \cdots, \hat{Y}_L^n) = g(I_1, \cdots, I_T)$. The probability of error will in general depend on the actions of the traitors. Note again that we only require small probability of error on the source estimates corresponding to the honest nodes.

We define a rate function $R(\mathcal{H}, r)$ defined for $\mathcal{H} \in \mathfrak{H}$ and $r \in \mathcal{R}(\mathcal{H})$ to be *α-achievable* if there exists a code such that, for all pairs $(\mathcal{H}, r)$ and any choice of actions by the traitors, $P_e \leq \alpha$,

$$\Pr\left(\sum_{T=1}^{T} R_{i_t,j_t} \leq R(\mathcal{H}, r)\right) \geq 1 - \alpha$$

and $\log K \leq \alpha n R_{i,j}$ for all i, j. This last condition requires, as discussed above, that the feedback rate from the decoder back to the nodes is arbitrarily small

compared to the forward rate. A rate function $R(\mathcal{H}, r)$ is *achievable* if for all $\alpha > 0$, there is a sequence of α-achievable rate functions $\{R'_k(\mathcal{H}, r)\}_{k=1}^{\infty}$ such that

$$\lim_{k\to\infty} R'_k(\mathcal{H}, r) = R(\mathcal{H}, r).$$

Note that we do not require uniform convergence.

The following definitions allow us to state our main variable-rate result. For any $\mathcal{H} \in \mathfrak{H}$ and $r \in \mathcal{R}(\mathcal{H})$, let $\tilde{r}(w|y_{\mathcal{H}})$ be the distribution of W given $Y_{\mathcal{H}}$ when W is distributed according to $r(w|y_{\mathcal{M}})$. That is,

$$\tilde{r}(w|y_{\mathcal{H}}) = \sum_{y_{\mathcal{H}^c} \in \mathcal{Y}_{\mathcal{H}^c}} p(y_{\mathcal{H}^c}|y_{\mathcal{H}}) r(w|y_{\mathcal{H}} y_{\mathcal{H}^c}).$$

The extent to which W provides information about $Y_{\mathcal{H}^c}$ is irrelevant to the traitors, since in order to fool the decoder they must generate information that appears to agree only with $Y_{\mathcal{H}}$ as reported by the honest nodes. Thus it will usually be more convenient to work with $\tilde{r}$ rather than r. For any $\mathcal{S} \in \mathfrak{H}$ and $r' \in \mathcal{R}(\mathcal{S})$, let

$$\mathcal{Q}_{\mathcal{S},r'} \triangleq \left\{ p(y_{\mathcal{S}}) \sum_{w} \tilde{r}'(w|y_{\mathcal{S}}) \bar{q}(y_{\mathcal{S}^c}|w) : \forall \bar{q}(y_{\mathcal{S}^c}|w) \right\}. \tag{3.7}$$

If $\mathcal{S}^c$ were the traitors and W were distributed according to r', then $\mathcal{Q}_{\mathcal{S},r'}$ would be the set of distributions q to which the traitors would have access. That is, if they simulate the proper $\bar{q}(y_{\mathcal{S}^c}|w)$ from their received W, this simulated version of $Y_{\mathcal{S}}$ and the true value of $Y_{\mathcal{S}^c}$ would be jointly distributed according to q. For any $\mathfrak{V} \subset \mathfrak{H}$, define

$$\mathcal{Q}(\mathfrak{V}) \triangleq \bigcap_{\mathcal{S}\in\mathfrak{V}} \bigcup_{r'\in\mathcal{R}(\mathcal{S})} \mathcal{Q}_{\mathcal{S},r'},$$

$$\mathcal{U}(\mathfrak{V}) \triangleq \bigcup_{\mathcal{S}\in\mathfrak{V}} \mathcal{S}.$$

That is, for some distribution $q \in \mathcal{Q}(\mathfrak{V})$, for every $\mathcal{S} \in \mathfrak{V}$, if the traitors were $\mathcal{S}^c$, they would have access to q for some $r' \in \mathcal{R}(\mathcal{S})$. Thus any distribution in $\mathcal{Q}(\mathfrak{V})$

makes it look to the decoder like any $\mathcal{S} \in \mathfrak{V}$ could be the set of honest nodes, so any node in $i \in \mathcal{U}(\mathfrak{V})$ is potentially honest.

Theorem 7 *A rate function $R(\mathcal{H}, r)$ is achievable if and only if, for all $(\mathcal{H}, r)$,*

$$R(\mathcal{H}, r) \geq R^*(\mathcal{H}, r) \triangleq \sup_{\mathfrak{V} \subset \mathfrak{H},\ q \in \mathcal{Q}_{\mathcal{H},r} \cap \mathcal{Q}(\mathfrak{V})} H_q(Y_{\mathcal{U}(\mathfrak{V})}). \tag{3.8}$$

See Section 3.5 for the proof.

We offer the following interpretation of this result. Suppose we placed the following constraint on the traitors' behavior. Given W^n, they must produce a value of $Y_{\mathcal{T}}^n$ in an i.i.d. fashion, then report it as the truth. That is, they choose a value of $Y_{\mathcal{T}}$ at time τ based only on W at time τ, making each choice in an identical manner. Then each traitor i takes the produced value of Y_i^n and behaves for the duration of the coding session exactly as if it were honest and this was the true source sequence. We can now easily classify all possible behaviors of the traitors simply by specifying the manner in which they generate $Y_{\mathcal{T}}$ from W, which is given by some distribution $\bar{q}(y_{\mathcal{T}}|w)$. The joint distribution of $Y_{\mathcal{H}}$ and $Y_{\mathcal{T}}$ will be given by

$$q(y_{\mathcal{M}}) = p(y_{\mathcal{H}}) \sum_w \tilde{r}(w|y_{\mathcal{H}}) \bar{q}(y_{\mathcal{T}}|w). \tag{3.9}$$

By (3.7), $q \in \mathcal{Q}_{\mathcal{H},r}$. If q is also contained in $\mathcal{Q}_{\mathcal{S},r'}$ for some $\mathcal{S} \in \mathfrak{H}$ and $r' \in \mathcal{R}(\mathcal{S})$, then again by (3.7), there exists a distribution $\bar{q}'(y_{\mathcal{S}}|w)$ such that

$$q(y_{\mathcal{M}}) = p(y_{\mathcal{S}}) \sum_w \tilde{r}'(w|y_{\mathcal{S}}) \bar{q}'(y_{\mathcal{S}}|w). \tag{3.10}$$

Since (3.9) and (3.10) have exactly the same form, the decoder will not be able to determine whether $\mathcal{H}$ is the set of honest nodes with W distributed according to r, or $\mathcal{S}$ is the set of honest nodes with W distributed according to r'. On the other hand, if for some $\mathcal{S} \in \mathfrak{H}$, $q \notin \mathcal{Q}_{\mathcal{S},r'}$ for all $r' \in \mathcal{R}(\mathcal{S})$, then the decoder should

be able to tell that $\mathcal{S}$ is not the set of honest nodes. We have not yet said how it might know, but intuition suggests that it should be possible. Hence, if there is no $\mathcal{S}$ containing a certain node i for which

$$q \in \bigcup_{r' \in \mathcal{R}(\mathcal{S})} \mathcal{Q}_{\mathcal{S},r'} \tag{3.11}$$

then the decoder can be sure that i is a traitor and it may be ignored. Let $\mathfrak{V}$ be the collection of all $\mathcal{S} \in \mathfrak{H}$ for which (3.11) holds. Every node in $\mathcal{U}(\mathfrak{V})$ looks to the decoder like it could be honest; all the rest are surely traitors. Thus, in order to make sure that the decoder reconstructs honest information perfectly, it must recover Y_i^n for all $i \in \mathcal{U}(\mathfrak{V})$, which means the sum rate must be at least $H_q(Y_{\mathcal{U}(\mathfrak{V})})$. Observe that

$$q \in \bigcap_{\mathcal{S} \in \mathfrak{V}} \bigcup_{r' \in \mathcal{R}(\mathcal{S})} \mathcal{Q}_{\mathcal{S},r'} = \mathcal{Q}(\mathfrak{V}).$$

As already noted, $q \in \mathcal{Q}_{\mathcal{H},r}$, so $q \in \mathcal{Q}_{\mathcal{H},r} \cap \mathcal{Q}(\mathfrak{V})$. Moreover, for any $\mathfrak{V} \subset \mathfrak{H}$, every element of $\mathcal{Q}_{\mathcal{H},r} \cap \mathcal{Q}(\mathfrak{V})$ can be produced with the proper choice of $\bar{q}(y_{\mathcal{T}}|w)$. Hence $H_q(Y_{\mathcal{U}(\mathfrak{V})})$ can be as high as

$$\sup_{\mathfrak{V} \subset \mathfrak{H},\ q \in \mathcal{Q}_{\mathcal{H},r} \cap \mathcal{Q}(\mathfrak{V})} H_q(Y_{\mathcal{U}(\mathfrak{V})}) = R^*(\mathcal{H}, r)$$

but no higher. Thus it makes sense that this rate and no better can be achieved if we place this constraint on the traitors. Therefore Theorem 7 can be interpreted as stating that constraining the traitors in this manner has no effect on the set of achievable rates.

3.4 Properties of the Variable-Rate Region

It might at first appear that (3.8) does not agree with (3.3). We discuss several ways in which (3.8) can be made more manageable, particularly in the case of

perfect traitor information (i.e. $W = Y_{\mathcal{M}}$), and show that the two are in fact identical. Let R^* be the minimum rate achievable over all $\mathcal{H} \in \mathfrak{H}$ and $r \in \mathcal{R}(\mathcal{H})$. Thus by (3.8), we can write

$$R^* = \sup_{\mathcal{H}\in\mathfrak{H}, r\in\mathcal{R}(\mathcal{H})} R^*(\mathcal{H}, r) = \sup_{\mathfrak{V}\subset\mathfrak{H},\ q\in\mathcal{Q}(\mathfrak{V})} H_q(Y_{\mathcal{U}(\mathfrak{V})}). \tag{3.12}$$

This is the quantity that appears in (3.3). Note also that for perfect traitor information,

$$\mathcal{Q}_{\mathcal{S},r'} = \{q(y_{\mathcal{M}}) : q(y_{\mathcal{S}}) = p(y_{\mathcal{S}})\}. \tag{3.13}$$

This means that $\mathcal{Q}_{\mathcal{H},r} \cap \mathcal{Q}(\mathfrak{V}) = \mathcal{Q}(\mathfrak{V} \cup \{\mathcal{H}\})$. Therefore (3.8) becomes

$$R^*(\mathcal{H}, r) = \sup_{\mathfrak{V}\subset\mathfrak{H}:\mathcal{H}\in\mathfrak{V},\ q\in\mathcal{Q}(\mathfrak{V})} H_q(Y_{\mathcal{U}(\mathfrak{V})}).$$

The following lemma simplifies calculation of expressions of the form $\sup_{q\in\mathcal{Q}(\mathfrak{V})} H_q(Y_{\mathcal{U}(\mathfrak{V})})$.

Lemma 7 *Suppose the traitors have perfect information. For any $\mathfrak{V} \subset \mathfrak{H}$, the expression*

$$\sup_{q\in\mathcal{Q}(\mathfrak{V})} H_q(Y_{\mathcal{U}(\mathfrak{V})}) \tag{3.14}$$

is maximized by a q satisfying (3.13) *for all $\mathcal{S} \in \mathfrak{V}$ such that, for some set of functions $\{\sigma_{\mathcal{S}}\}_{\mathcal{S}\in\mathfrak{V}}$,*

$$q(y_1 \cdots y_L) = \prod_{\mathcal{S}\in\mathfrak{V}} \sigma_{\mathcal{S}}(y_{\mathcal{S}}). \tag{3.15}$$

Proof: By (3.13), we need to maximize $H_q(Y_{\mathcal{U}(\mathfrak{V})})$ subject to the constraints that for each $\mathcal{S} \in \mathfrak{V}$ and all $y_{\mathcal{S}} \in \mathcal{Y}_{\mathcal{S}}$, $q(y_{\mathcal{S}}) = p(y_{\mathcal{S}})$. This amounts to maximizing the Lagrangian

$$\Lambda = -\sum_{y_{\mathcal{U}(\mathfrak{V})}\in\mathcal{Y}_{\mathcal{U}(\mathfrak{V})}} q(y_{\mathcal{U}(\mathfrak{V})}) \log q(y_{\mathcal{U}(\mathfrak{V})}) + \sum_{\mathcal{S}\in\mathfrak{V}} \sum_{y_{\mathcal{S}}\in\mathcal{Y}_{\mathcal{S}}} \lambda_{\mathcal{S}}(y_{\mathcal{S}})\big(q(y_{\mathcal{S}}) - p(y_{\mathcal{S}})\big).$$

Note that for any $\mathcal{S} \subset \mathcal{U}(\mathfrak{V})$,

$$\frac{\partial q(y_{\mathcal{S}})}{\partial q(y_{\mathcal{U}(\mathfrak{V})})} = 1.$$

Thus, differentiating with respect to $q(y_{\mathcal{U}(\mathfrak{V})})$ gives, assuming the log is a natural logarithm,

$$\frac{\partial \Lambda}{\partial q(y_{\mathcal{U}(\mathfrak{V})})} = -\log q(y_{\mathcal{U}(\mathfrak{V})}) - 1 + \sum_{\mathcal{S} \in \mathfrak{V}} \lambda_{\mathcal{S}}(y_{\mathcal{S}}).$$

Setting this to 0 gives

$$q(y_{\mathcal{U}(\mathfrak{V})}) = \exp\left(-1 + \sum_{\mathcal{S} \in \mathfrak{V}} \lambda_{\mathcal{S}}(y_{\mathcal{S}}) \right) = |\mathcal{Y}_{\mathcal{U}(\mathfrak{V})^c}| \prod_{\mathcal{S} \in \mathfrak{V}} \sigma_{\mathcal{S}}(y_{\mathcal{S}})$$

for some set of functions $\{\sigma_{\mathcal{S}}\}_{\mathcal{S} \in \mathfrak{V}}$. Therefore setting

$$q(y_1 \cdots y_L) = \frac{q(y_{\mathcal{U}(\mathfrak{V})})}{|\mathcal{Y}_{\mathcal{U}(\mathfrak{V})^c}|}$$

satisfies (3.15), so if $\sigma_{\mathcal{S}}$ are such that (3.13) is satisfied for all $\mathcal{S} \in \mathfrak{V}$, q will maximize $H_q(Y_{\mathcal{U}(\mathfrak{V})})$. □

Suppose $L = 3$ and $\mathfrak{H} = \mathfrak{H}_1$. If $\mathfrak{V} = \{\{1,2\},\{2,3\}\}$, then $\tilde{q}(y_1 y_2 y_3) = p(y_1 y_2) p(y_3|y_2)$ is in $\mathcal{Q}(\mathfrak{V})$ and by Lemma 7 maximizes $H_q(Y_1 Y_2 Y_3)$ over all $q \in \mathcal{Q}(\mathfrak{V})$. Thus

$$\sup_{q \in \mathcal{Q}(\mathfrak{V})} H_q(Y_1 Y_2 Y_3) = H_{\tilde{q}}(Y_1 Y_2 Y_3) = H(Y_1 Y_2 Y_3) + I(Y_1; Y_3|Y_2).$$

By similar reasoning, considering $\mathfrak{V} = \{\{1,2\},\{1,3\}\}$ and $\mathfrak{V} = \{\{1,3\},\{2,3\}\}$ results in (3.3). Note that if $\mathfrak{V}_1 \subset \mathfrak{V}_2$, then $\mathcal{Q}(\mathfrak{V}_1) \supset \mathcal{Q}(\mathfrak{V}_2)$, so $\mathfrak{V}_2$ need not be considered in evaluating (3.8). Thus we have ignored larger subsets of $\mathfrak{H}_1$, since the value they give would be no greater than the others.

We can generalize to any collection $\mathfrak{V}$ of the form

$$\{\{\mathcal{S}_1, \mathcal{S}_2\}, \{\mathcal{S}_1, \mathcal{S}_3\}, \cdots, \{\mathcal{S}_1, \mathcal{S}_k\}\}$$

in which case

$$\sup_{q\in\mathcal{Q}(\mathfrak{V})} = H(Y_{\mathcal{S}_1}Y_{\mathcal{S}_2}) + H(Y_{\mathcal{S}_3}|Y_{\mathcal{S}_1}) + \cdots + H(Y_{\mathcal{S}_k}|Y_{\mathcal{S}_1}).$$

Employing this, we can rewrite (3.12) for $\mathfrak{H} = \mathfrak{H}_s$ and certain values of s. For $s = 1$, it becomes

$$R^* = H(Y_1 \cdots Y_L) + \max_{i,i'\in\mathcal{M}} I(Y_i; Y_{i'}|Y_{\{i,i'\}^c}).$$

Again, relative to the Slepian-Wolf result, we always pay a conditional mutual information penalty for a single traitor. For $s = 2$,

$$\begin{aligned} R^* &= H(Y_1 \cdots Y_L) \\ &\quad + \max\left\{ \max_{\mathcal{S},\mathcal{S}'\subset\mathcal{M}:|\mathcal{S}|=|\mathcal{S}'|=2} I(Y_{\mathcal{S}}; Y_{\mathcal{S}'}|Y_{(\mathcal{S}\cup\mathcal{S}')^c}),\ \max_{i,i',i''\in\mathcal{M}} I(Y_i; Y_{i'}; Y_{i''}|Y_{\{i,i',i''\}^c}) \right\} \end{aligned}$$

where $I(X;Y;Z|W) = H(X|W) + H(Y|W) + H(Z|W) - H(XYZ|W)$. For $s = L-1$, R^* is given by (3.2). There is a similar formulation for $s = L-2$, though it is more difficult to write down for arbitrary L.

With all these expressions made up of nothing but entropies and mutual informations, it might seem hopeful that (3.14) can be reduced to such an analytic expression for all $\mathfrak{V}$. However, this is not the case. For example, consider $\mathfrak{V} = \{\{1,2,3\},\{3,4,5\},\{5,6,1\}\}$. This $\mathfrak{V}$ is irreducible in the sense that there is no subset $\mathfrak{V}'$ that still satisfies $\mathcal{U}(\mathfrak{V}') = \{1,\cdots,6\}$, but there is no simple distribution $q \in \mathcal{Q}(\mathfrak{V})$ made up of marginals of p that satisfies Lemma 7, so it must be found numerically. Still, Lemma 7 simplifies the calculation considerably.

3.5 Proof of Theorem 7

3.5.1 Converse

We first show the converse. Fix $\mathcal{H} \in \mathfrak{H}$ and $r \in \mathcal{R}(\mathcal{H})$. Take any $\mathfrak{V} \subset \mathfrak{H}$, and any distribution $q \in \mathcal{Q}_{\mathcal{H},r} \cap \mathcal{Q}(\mathfrak{V})$. Since $q \in \mathcal{Q}_{\mathcal{H},r}$, there is some $\bar{q}(y_{\mathcal{T}}|w)$ such that $Y_{\mathcal{H}}$ and $Y_{\mathcal{T}}$ are distributed according to q. Since also $q \in \mathcal{Q}_{\mathcal{S},r'}$ for all $\mathcal{S} \in \mathfrak{V}$ and some $r' \in \mathcal{R}(\mathcal{S})$, if the traitors simulate this $\bar{q}$ and act honestly with these fabricated source values, the decoder will not be able to determine which of the sets in $\mathfrak{V}$ is the actual set of honest nodes. Thus, the decoder must perfectly decode the sources from all nodes in $\mathcal{U}(\mathfrak{V})$, so if $R(\mathcal{H}, r)$ is a precisely α-achievable rate function, $R(\mathcal{H}, r) \geq H_q(Y_{\mathcal{U}(\mathfrak{V})})$.

3.5.2 Achievability Preliminaries

Now we prove achievability. To do so, we will first need the theory of types. Given $y^n \in \mathcal{Y}^n$, let $t(y^n)$ be the type of y^n. Given a type t with denominator n, let $\Lambda_t^n(Y)$ be the set of all sequences in $\mathcal{Y}^n$ with type t. If t is a joint y, z type with denominator n, then let $\Lambda_t^n(Y|z^n)$ be the set of sequences $y^n \in \mathcal{Y}^n$ such that $(y^n z^n)$ have joint type t, with the convention that this set is empty if the type of z^n is not the marginal of t.

We will also need the following definitions. Given a distribution q on an alphabet $\mathcal{Y}$, define the η-ball of distributions

$$B_\eta(q) \triangleq \left\{ q'(\mathcal{Y}) : \forall y \in \mathcal{Y} : |q(y) - q'(y)| \leq \frac{\eta}{|\mathcal{Y}|} \right\}.$$

Note that the typical set can be written

$$T_\epsilon^n(Y) = \{y^n : t(y^n) \in B_\epsilon(p)\}.$$

We define slightly modified versions of the sets of distributions from Section 3.3.3 as follows:

$$\breve{\mathcal{Q}}^\eta_{\mathcal{S},r'} \triangleq \bigcup_{q \in \mathcal{Q}_{\mathcal{S},r'}} B_\eta(q),$$

$$\breve{\mathcal{Q}}^\eta(\mathfrak{V}) \triangleq \bigcap_{\mathcal{S} \in \mathfrak{V}} \bigcup_{r' \in \mathcal{R}(\mathcal{S})} \breve{\mathcal{Q}}^\eta_{\mathcal{S},r'}.$$

These sets are nearly the same as those defined earlier. We will eventually take the limit as $\eta \to 0$, making them identical to $\mathcal{Q}_{\mathcal{S},r'}$ and $\mathcal{Q}(\mathfrak{V})$, but it will be necessary to have slightly expanded versions for use with finite block length.

Finally, we will need the following lemma.

Lemma 8 *Given an arbitrary n length distribution $q^n(y^n)$ and a type t with denominator n on $\mathcal{Y}$, let $q_i(y)$ be the marginal distribution of q^n at time i and $\bar{q}(y) = \frac{1}{n}\sum_{i=1}^n q_i(y)$. If Y^n is distributed according to q^n and $\Pr(Y^n \in \Lambda_t^n(Y)) \geq 2^{-n\zeta}$, then $D(t\|\bar{q}) \leq \zeta$.*

Proof: Fix an integer $\tilde{n}$. For $\tilde{i} = 1, \cdots, \tilde{n}$, let $Y^n(\tilde{i})$ be independently generated from q^n. Let Γ be the set of types t^n on supersymbols in $\mathcal{Y}^n$ with denominator $\tilde{n}$ such that $t^n(y^n) = 0$ if $y^n \notin \Lambda_t^n(Y)$. Note that

$$|\Gamma| \leq (\tilde{n}+1)^{|\mathcal{Y}|^n}.$$

If $Y^{n\tilde{n}} = (Y^n(1), \cdots, Y^n(\tilde{n}))$, then

$$\Pr\left(Y^{n\tilde{n}} \in \bigcup_{t^n \in \Gamma} \Lambda_{t^n}^{\tilde{n}}(Y^n)\right) = \Pr(Y^n(\tilde{i}) \in \Lambda_t^n(Y), \forall \tilde{i})$$

$$\geq 2^{-n\tilde{n}\zeta}.$$

But

$$\Pr\left(Y^{n\tilde{n}} \in \bigcup_{t^n \in T^n} \Lambda_{t^n}^{\tilde{n}}(Y^n)\right) = \sum_{t^n \in \Gamma} \Pr(Y^{n\tilde{n}} \in \Lambda_{t^n}^{\tilde{n}}(Y^n)$$

$$\leq \sum_{t^n \in \Gamma} 2^{-\tilde{n} D(t^n \| q^n)}$$

$$\leq (\tilde{n}+1)^{|\mathcal{Y}|^n} 2^{-\tilde{n} \min_{t^n \in \Gamma} D(t^n \| q^n)}.$$

For any $t^n \in \Gamma$, letting t_i be the marginal type at time i gives $\frac{1}{n}\sum_{i=1}^n t_i = t$. Therefore

$$\zeta + \frac{1}{n\tilde{n}}|\mathcal{Y}|^n \log(\tilde{n}+1) \geq \min_{t^n \in \Gamma} \frac{1}{n} D(t^n \| q^n)$$

$$\geq \min_{t^n \in \Gamma} \frac{1}{n} \sum_{i=1}^n D(t_i \| q_i) \tag{3.16}$$

$$\geq D(t \| \bar{q}) \tag{3.17}$$

where (3.16) holds by [91, Lemma 4.3] and (3.17) by convexity of the Kullback-Leibler distance in both arguments. Letting $\tilde{n}$ grow proves the lemma. □

The achievability proof proceeds as follows. Section 3.5.3 describes our proposed coding scheme for the case that traitors cannot eavesdrop. In Section 3.5.4, we demonstrate that this coding scheme achieves small probability of error when the traitors have perfect information. Section 3.5.5 shows that the coding scheme achieves the rate function $R^*(\mathcal{H}, r)$. In Section 3.5.6, we extend the proof to include the case that the traitors have imperfect information. Finally, Section 3.5.7 gives a modification to the coding scheme that can handle eavesdropping traitors.

3.5.3 Coding Scheme Procedure

Our basic coding strategy is for a node to transmit a sequence of small messages to the decoder until the decoder has received enough information to decode the

node's source sequence. After receiving one of these messages, the decoder asks for another small message only if it is unable to decode the sequence. If it can, the decoder moves on to the next node. This way, the rate at which a node transmits is as small as possible. Once each node's source sequence has been decoded, the decoder attempts to use them to accumulate information about which nodes could be traitors. It is in this step that it uses its knowledge of the power of the traitors to tell the difference between a node that could be honest under some circumstances and one that is surely a traitor. After this, the decoder goes back across all the nodes again, repeating the same procedure for the next block of source values and ignoring those nodes that it knows to be traitors. The decoder repeats this again and again, gathering more information about which nodes could be traitors each time. The precise description of the coding strategy follows.

1) Random Code Structure: Fix $\epsilon > 0$. The maximum number of small messages that could be sent by node i when transmitting a certain sequence to the decoder is $J_i = \left\lceil \frac{\log |\mathcal{Y}_i|}{\epsilon} \right\rceil$. Each of these small messages is represented by a function to be defined, taking the source sequence as input and producing the small message as output. In addition, as we discussed in 3.1.5, it is necessary to randomize the messages at run time in order to defeat the traitors. Thus, node i has C different but identically created subcodebooks, each of which is made up of a sequence of J_i functions, one for each small messages, where C is an integer to be defined. Hence the full codebook for node i is composed of CJ_i separate functions. In particular, for $i = 1, \cdots, L$ and $c = 1, \cdots, C$, let

$$\tilde{f}_{i,c,1} : \mathcal{Y}_i^n \to \{1, \cdots, 2^{n(\epsilon+\nu)}\},$$
$$\tilde{f}_{i,c,j} : \mathcal{Y}_i^n \to \{1, \cdots, 2^{n\epsilon}\}, \quad j = 2, \cdots, J_i$$

with ν to defined later. Thus, a subcodebook associates with each element of

$\mathcal{Y}_i^n$ a sequence of about $n(\log|\mathcal{Y}_i| + \nu)$ bits chopped into small messages of length $n(\epsilon + \nu)$ or $n\epsilon$. We put tildes on these functions to distinguish them from the fs defined in (3.6). The $\tilde{f}$s that we define here are functions we use as pieces of the overall encoding functions f. Each one is constructed by a uniform random binning procedure. Define composite functions

$$\tilde{F}_{i,c,j}(y_i^n) \triangleq (\tilde{f}_{i,c,1}(y_i^n), \cdots, \tilde{f}_{i,c,j}(y_i^n)).$$

We can think of $\tilde{F}_{i,c,j}(y_i^n)$ as an index of one of $2^{n(j\epsilon+\nu)}$ random bins.

2) Round Method: Our coding scheme is made up of N rounds, with each round composed of m phases. In the ith phase, transactions are made entirely with node i. We denote $Y_i^n(I)$ as the Ith block of n source values, but for convenience, we will not include the index I when it is clear from context. As in the three-node example, all transactions in the Ith round are based only on $Y_{\mathcal{M}}^n(I)$. Thus the total block length is Nn.

The procedure for each round is identical except for the variable $\mathfrak{V}(I)$ maintained by the decoder. This represents the collection of sets that could be the set of honest nodes based on the information the decoder has received as of the beginning of round I. The decoder begins by setting $V(1) = \mathfrak{H}$ and then pares it down at the end of each round based on new information.

3) Encoding and Decoding Rules: In the ith phase, if $i \in \mathcal{U}(\mathfrak{V}(I))$, the decoder makes a number of transactions with node i and produces an estimate $\hat{Y}_i^n$ of Y_i^n. The estimate $\hat{Y}_i^n$ is of course a random variable, so as usual the lower case $\hat{y}_i^n$ refers to a realization of this variable. If $i \notin \mathcal{U}(\mathfrak{V}(I))$, then the decoder has determined that node i cannot be honest, so it does not communicate with it and sets $\hat{y}_i^n$ to a null value.

For $i \in \mathcal{U}(\mathfrak{V}(I))$, at the beginning of phase i, node i randomly selects a $c \in \{1, \cdots, C\}$ according to the uniform distribution. In the first transaction, node i transmits $(c, \tilde{f}_{i,c,1}(Y_i^n))$. That is, along with the small message itself, the node transmits the randomly selected index c of the subcodebook that it will use in this phase. As the phase continues, in the jth transaction, node i transmits $\tilde{f}_{i,c,j}(Y_i^n)$.

After each transaction, the decoder must decide whether to ask for another transaction with node i, and if not, to decode Y_i^n. In the random binning proof approach to the traditional Slepian-Wolf problem, the decoder decides which sequence in the received bin to select as the source estimate by taking the one contained in the typical set. Here we use the same idea, except that instead of the typical set, we use a different set for each transaction, and if there is no sequence in this set that falls into the received bin, this means not that we cannot decode the sequence but rather that we have not yet received enough information from the node and must ask for another transaction. The set associated with the jth transaction needs to have the property that its size is less than $2^{n(j\epsilon+\nu)}$, the number of bins into which the source space has been split after j messages, so that it is unlikely for two elements of the set to fall into the same bin. Furthermore, in order to ensure that we eventually decode any sequence that might be chosen by the node, the set should grow after each transaction and eventually contain all of $\mathcal{Y}_i^n$.

Now we define this set. First let $\mathcal{S}_i \triangleq \{1, \cdots, i\} \cap \mathcal{U}(\mathfrak{V}(I))$, the nodes up to i that are not ignored by the decoder, and let $\hat{y}_{\mathcal{S}_{i-1}}^n$ be the source sequences decoded in this round prior to phase i. The set associated with transaction j is

$$T_j(\hat{y}_{\mathcal{S}_{i-1}}^n) \triangleq \{y_i^n : H_{t(\hat{y}_{\mathcal{S}_{i-1}}^n y_i^n)}(Y_i|Y_{\mathcal{S}_{i-1}}) \leq j\epsilon\}. \tag{3.18}$$

To be specific, after j transactions, if there are no sequences in $T_j(\hat{y}_{\mathcal{S}_{i-1}})$ matching

the received value of $\tilde{F}_{i,c,j}$, the decoder chooses to do another transaction with node i. If there is at least one such sequence, call it $\hat{y}_i^n$, choosing between several possibilities arbitrarily.

Observe that

$$|T_j(\hat{y}^n_{\mathcal{S}_{i-1}})| \leq (n+1)^{|\mathcal{Y}_i \times \mathcal{Y}_{\mathcal{S}_{i-1}}|} 2^{nj\epsilon}.$$

Hence T_j satisfies the size property that were discussed above. Moreover, it grows with j to eventually become $\mathcal{Y}_i^n$. Finally, we have chosen T_j in particular because it has the property that when a sequence y_i^n falls into T_j for the first time, the rate at which node i has transmitted to the decoder is close to the entropy of the type of y_i^n. This means that we can relate the accuracy of the decoded sequences to the achieved rate, which will allow us to prove that the coding scheme achieves the claimed rate.

4) Round Conclusion: At the end of round I, the decoder produces $\mathfrak{V}(I+1)$ by setting

$$\mathfrak{V}(I+1) = \left\{ \mathcal{S} \in \mathfrak{V}(I) : t(\hat{y}^n_{\mathcal{U}(\mathfrak{V}(I))}) \in \bigcup_{r' \in R(\mathcal{S})} \breve{\mathcal{Q}}^{\eta}_{\mathcal{S},r'} \right\} \tag{3.19}$$

for η to be defined such that $\eta \geq \epsilon$ and $\eta \to 0$ as $\epsilon \to 0$. As we will show, it is essentially impossible for the traitors to transmit messages such that the type of the decoded messages does not fall into $\breve{\mathcal{Q}}^{\eta}_{\mathcal{H},r}$, meaning that $\mathcal{H}$ is always in $\mathfrak{V}(I)$. This ensures that the true honest nodes are never ignored and their source sequences are always decoded correctly.

3.5.4 Error Probability

Define the following error events:

$$\mathcal{E}_1(I,i) \triangleq \{\hat{Y}_i^n(I) \neq Y_i^n(I)\},$$
$$\mathcal{E}_2(I) \triangleq \{\mathcal{H} \notin \mathfrak{V}(I)\},$$
$$\mathcal{E}_3(I) \triangleq \{t(\hat{Y}^n_{\mathcal{U}(\mathfrak{V}(I))}(I)) \notin \breve{\mathcal{Q}}^{\eta}_{\mathcal{H},r}\}.$$

The total probability of error is

$$P_e = \Pr\left(\bigcup_{I=1}^{N}\bigcup_{i\in\mathcal{H}} \mathcal{E}_1(I,i)\right).$$

As we have said but not yet proved, $\mathcal{H}$ will usually be in $\mathfrak{V}(I)$ (i.e. $\mathcal{E}_2(I)$ does not occur), so we do not lose much by writing

$$P_e \leq \Pr\left(\bigcup_{I=1}^{N}\left[\mathcal{E}_2(I+1) \cup \bigcup_{i\in\mathcal{H}} \mathcal{E}_1(I,i)\right]\right).$$

Let

$$\mathcal{A}_I \triangleq \mathcal{E}_2^c(I+1) \cap \bigcap_{i\in\mathcal{H}} E_1^c(I,i)$$

for $I = 1, \cdots, N$, so

$$1 - P_e \geq \Pr(\mathcal{A}_1, \cdots, \mathcal{A}_N) = \prod_{I=1}^{N} \Pr(\mathcal{A}_I | \mathcal{A}_1, \cdots, \mathcal{A}_{I-1}).$$

Observe that $\mathcal{A}_I$ depends only on $\hat{Y}^n_{\mathcal{M}}(I)$ and $Y^n_{\mathcal{M}}(I)$, both of which are independent of all events before round I given that $\mathcal{H} \in \mathfrak{V}(I)$ (i.e. $\mathcal{E}_2^c(I)$ occurs), since this is enough to ensure that $\hat{Y}_i^n(I)$ is non-null. Since $\mathcal{A}_1, \cdots, \mathcal{A}_{I-1}$ includes $\mathcal{E}_2^c(I)$, we can drop all conditioning terms expect it. Note also that $\mathcal{E}_2^c(1)$ occurs with probability 1. Therefore

$$\begin{aligned} 1 - P_e &\geq \prod_{I=1}^{n} \Pr(\mathcal{A}_I | \mathcal{E}_2^c(I)) \\ &= \prod_{I=1}^{n} [1 - \Pr(\mathcal{A}_I^c | \mathcal{E}_2^c(I))] \geq 1 - \sum_{I=1}^{n} \Pr(\mathcal{A}_I^c | \mathcal{E}_2^c(I)) \end{aligned}$$

so

$$P_e \leq \sum_{I=1}^{N} \Pr\left(\mathcal{E}_2(I+1) \cup \bigcup_{i\in\mathcal{H}} \mathcal{E}_1(I,i) \Big| \mathcal{E}_2^c(I)\right).$$

By (3.19), if $\mathcal{H}$ is in $\mathfrak{V}(I)$ but not in $\mathfrak{V}(I+1)$, then $t(\hat{Y}^n_{\mathcal{U}(\mathfrak{V}(I))}(I)) \notin \breve{\mathcal{Q}}^{\eta}_{\mathcal{H},r}$. Thus

$$\mathcal{E}_2(I+1) \cap \mathcal{E}_2^c(I) \subset \mathcal{E}_3(I) \cap \mathcal{E}_2^c(I)$$

so

$$\begin{aligned}
P_e &\leq \sum_{I=1}^{N} \Pr\left(\mathcal{E}_3(I) \cup \bigcup_{i\in\mathcal{H}} \mathcal{E}_1(I,i) \Big| \mathcal{E}_2^c(I)\right) \\
&\leq \sum_{I=1}^{N} \Pr\left(\mathcal{E}_3(I) \Big| \mathcal{E}_2^c(I), \bigcap_{i\in\mathcal{H}} \mathcal{E}_1^c(I,i)\right) \\
&\qquad + \sum_{I=1}^{N} \Pr\left(\bigcup_{i\in\mathcal{H}} \mathcal{E}_1(I,i) \Big| \mathcal{E}_2^c(I)\right) \\
&\leq \sum_{I=1}^{N} \Pr\left(\mathcal{E}_3(I) \Big| \bigcap_{i\in\mathcal{H}} \mathcal{E}_1^c(I,i)\right) \\
&\qquad + \sum_{I=1}^{N} \sum_{i\in\mathcal{H}} \Pr(\mathcal{E}_1(I,i) | \mathcal{E}_2^c(I)) \qquad (3.20)
\end{aligned}$$

where we have dropped the conditioning on $\mathcal{E}_2^c(I)$ in the first term because it influences the probability of $\mathcal{E}_3(I)$ only in that it ensures that $\hat{Y}_i^n$ for $i \in \mathcal{H}$ are non-null, which is already implied by $\bigcap_{i\in\mathcal{H}} \mathcal{E}_1^c(I,i)$.

We first bound the first term in (3.20) by showing that for all I,

$$\Pr\left(\mathcal{E}_3(I) \Big| \bigcap_{i\in\mathcal{H}} \mathcal{E}_1^c(I,i)\right) \leq \frac{\alpha}{2N}. \qquad (3.21)$$

If the traitors receive perfect source information, then as we have already noted in (3.13), $Q_{\mathcal{H},r}$ only puts a constraint on the $Y_{\mathcal{H}}$ marginal of distributions, and the same is true of $\breve{Q}^{\eta}_{\mathcal{H},r}$. In particular, $t(\hat{Y}^n_{\mathcal{U}(\mathfrak{V}(I))}(I)) \in \breve{\mathcal{Q}}^{\eta}_{\mathcal{H},r}$ is equivalent to $\hat{Y}^n_{\mathcal{H}}(I)$ being typical. Conditioning on $\bigcap_{i\in\mathcal{H}} \mathcal{E}_1^c(I,i)$ implies that $\hat{Y}^n_{\mathcal{H}}(I) = Y^n_{\mathcal{H}}(I)$, so

$$\Pr\left(\mathcal{E}_3(I) \Big| \bigcap_{i\in\mathcal{H}} \mathcal{E}_1^c(I,i)\right) \leq \Pr(Y^n_{\mathcal{H}}(I) \in T^n_{\epsilon}(Y_{\mathcal{H}}))$$

meaning (3.21) holds for sufficiently large n by the AEP. Thus (3.21) is only non-trivial if the traitors receive imperfect source information. This case is dealt with in Section 3.5.6.

We now consider the second term of (3.20), involving $\Pr(\mathcal{E}_1(I,i)|\mathcal{E}_2^c(I))$ for honest i. Conditioning on $\mathcal{E}_2^c(I)$ ensures that $i \in \mathcal{U}(\mathfrak{V}(I))$ for honest i, so $\hat{Y}_i^n(I)$ will be non-null. The only remaining type of is a decoding error. This occurs if for some transaction j, there is an sequence in $T_j(\hat{Y}_{\mathcal{S}_{i-1}})$ different from Y_i^n that matches all thus far received messages. That is, if

$$\exists j, y_i'^n \in T_j(\hat{Y}^n_{\mathcal{S}_{i-1}}) \backslash \{Y_i^n\} : \tilde{F}_{i,c,j}(y_i'^n) = \tilde{F}_{i,c,j}(Y_i^n).$$

However, $\mathcal{S}_{i-1}$ may contain traitors. Indeed, it may be made entirely of traitors. Thus, we have to take into account that $\hat{Y}^n_{\mathcal{S}_{i-1}}$ may be chosen to ensure the existence of such an erroneous $y_i'^n$. The node's use of randomizing among the C subcodebooks is the method by which this is mitigated, as we will now prove.

Let

$$k_1(y_i^n, \hat{y}^n_{\mathcal{S}_{i-1}}) \triangleq |\{c : \exists j, y_i'^n \in T_j(\hat{y}^n_{\mathcal{S}_{i-1}}) \backslash \{y_i^n\} : \tilde{F}_{i,c,j}(y_i'^n) = \tilde{F}_{i,c,j}(y_i^n)\}|.$$

That is, k_1 is the number of subcodebooks that if chosen could cause a decoding error at some transaction. Recall that node i chooses the subcodebook randomly from the uniform distribution. Thus, given y_i^n and $\hat{y}^n_{\mathcal{S}_{i-1}}$, the probability of an error resulting from a bad choice of subcodebook is $k_1(y_i^n, \hat{y}^n_{\mathcal{S}_{i-1}})/C$. Furthermore, k_1 is based strictly on the codebook, so we can think of it as a random variable defined on the same probability space as that governing the random codebook creation. Averaging over all possible codebooks,

$$\Pr(\mathcal{E}_1(I,i)|\mathcal{E}_2^c(I)) \leq \mathbb{E} \sum_{y_i^n \in \mathcal{Y}_i^n} p(y_i^n) \max_{\hat{y}^n_{\mathcal{S}_{i-1}} \in \mathcal{Y}^n_{\mathcal{S}_{i-1}}} \frac{k_1(y_i^n, \hat{y}^n_{\mathcal{S}_{i-1}})}{C}$$

where the expectation is taken over codebooks.

Let $\mathcal{C}$ be the set of all codebooks. We define a subset $\mathcal{C}_1$, then show that the probability of error can be easily bounded for any codebook in $\mathcal{C}\backslash\mathcal{C}_1$, and that the probability of a codebook being chosen in $\mathcal{C}_1$ is small. In particular, let $\mathcal{C}_1$ be the set of codebooks for which, for any $y_i^n \in \mathcal{Y}_i^n$ and $\hat{y}_{\mathbb{S}_{i-1}}^n \in \mathcal{Y}_{\mathbb{S}_{i-1}}^n$, $k_1(y_i^n, \hat{y}_{\mathbb{S}_{i-1}}^n) > B$, for an integer $B \leq C$ to be defined later. Then

$$\begin{aligned}\Pr(\mathcal{E}_1(I,i)|\mathcal{E}_2^c(I)) &\leq \Pr(\mathcal{C}\backslash\mathcal{C}_1) \sum_{y_i^n \in \mathcal{Y}_i^n} p(y_i^n) \max_{\hat{y}_{\mathbb{S}_{i-1}}^n \in \mathcal{Y}_{\mathbb{S}_{i-1}}^n} \frac{B}{C} \\ &\quad + \Pr(\mathcal{C}_1) \sum_{y_i^n \in \mathcal{Y}_i^n} p(y_i^n) \max_{\hat{y}_{\mathbb{S}_{i-1}}^n \in \mathcal{Y}_{\mathbb{S}_{i-1}}^n} \frac{C}{C} \\ &\leq \frac{B}{C} + \Pr(\mathcal{C}_1). \end{aligned} \tag{3.22}$$

Recall that k_1 is the number of subcodebooks that could cause an error. Since each subcodebook is generated identically, k_1 is a binomial random variable with C trails and probability of success P, where P is the probability that one particular subcodebooks causes an error. Thus

$$\begin{aligned} P &= \Pr\left(\exists j, y_i'^n \in T_j(\hat{y}_{\mathbb{S}_{i-1}}^n)\backslash\{y_i^n\} : \right. \\ &\qquad \left. \tilde{F}_{i,c,j}(y_i'^n) = \tilde{F}_{i,c,j}(y_i^n)\right) \\ &\leq \sum_{j=1}^{J_i} \sum_{y_i'^n \in T_j(\hat{y}_{\mathbb{S}_{i-1}}^n)\backslash\{y_i^n\}} \Pr\left(F_{i,c,j}(y_i'^n) = F_{i,c,j}(y_i^n)\right) \\ &\leq J_i \left|T_j(\hat{y}_{\mathbb{S}_{i-1}}^n)\right| 2^{-n(j\epsilon+\nu)} \\ &\leq J_i (n+1)^{|\mathcal{Y}_i \times \mathcal{Y}_{\mathbb{S}_{i-1}}|} 2^{-n\nu} \leq 2^{n(\epsilon-\nu)} \end{aligned}$$

for sufficiently large n. For a binomial random variable Y with mean $\bar{Y}$ and any κ, we can use the Chernoff bound to write

$$\Pr(Y \geq \kappa) \leq \left(\frac{e\bar{Y}}{\kappa}\right)^{\kappa}. \tag{3.23}$$

Therefore

$$\Pr(k_1(y_i^n, \hat{y}_{\mathcal{S}_{i-1}}^n) > B) \leq \left(\frac{eCP}{B+1}\right)^{B+1} \leq 2^{nB(\epsilon-\nu)}$$

if $\nu > \epsilon$ and n is sufficiently large. Thus

$$\begin{aligned}
\Pr(\mathcal{C}_1) &= \Pr(\exists y_i^n, \hat{y}_{\mathcal{S}_{i-1}}^n : k_1(y_i^n, \hat{y}_{\mathcal{S}_{i-1}}^n) > B) \\
&\leq \sum_{y_i^n} \sum_{\hat{y}_{s_{i-1}}^n} \Pr(k_1(y_i^n, \hat{y}_{\mathcal{S}_{i-1}}^n) > B) \\
&\leq \sum_{y_i^n} \sum_{\hat{y}_{\mathcal{S}_{i-1}}^n} 2^{nB(\epsilon-\nu)} \\
&= 2^{n[\log|\mathcal{Y}_i| + \log|\mathcal{Y}_{\mathcal{S}_{i-1}}| + B(\epsilon-\nu)]}. \qquad (3.24)
\end{aligned}$$

Combining (3.20) with (3.21), (3.22), and (3.24) gives

$$\begin{aligned}
P_e &\leq \frac{\alpha}{2} + \sum_{I=1}^{N} \sum_{i \in \mathcal{H}} \left(\frac{B}{C} + 2^{n[\log|\mathcal{Y}_i| + \log|\mathcal{Y}_{\mathcal{S}_{i-1}}| + B(\epsilon-\nu)]}\right) \\
&\leq \frac{\alpha}{2} + NL\left(\frac{B}{C} + 2^{n[\log|\mathcal{Y}_{\mathcal{M}}| + B(\epsilon-\nu)]}\right)
\end{aligned}$$

which is less than α for sufficiently large n if

$$B > \frac{\log|\mathcal{Y}_{\mathcal{M}}|}{\nu - \epsilon}$$

and

$$C \geq \frac{3NLB}{\alpha} > \frac{3NL\log|\mathcal{Y}_{\mathcal{M}}|}{\alpha(\nu-\epsilon)}.$$

3.5.5 Code Rate

The discussion above placed a lower bound on C. However, for sufficiently large n, we can make $\frac{1}{n}\log C \leq \epsilon$, meaning it takes no more than ϵ rate to transmit the subcodebook index c at the beginning of the phase. Therefore the rate for phase i is at most $(j+1)\epsilon + \nu$, where j is the number of transactions in phase

i. Transaction j must be the earliest one with $\hat{y}_i^n \in T_j(\hat{y}_{\mathcal{S}_{i-1}})$, otherwise it would have been decoded earlier. Thus j is the smallest integer for which

$$H_{t(\hat{y}^n_{\mathcal{S}_{i-1}}\hat{y}_i^n)}(Y_i|Y_{\mathcal{S}_{i-1}}) \leq j\epsilon$$

meaning

$$j\epsilon \leq H_{t(\hat{y}^n_{\mathcal{S}_{i-1}}\hat{y}_i^n)}(Y_i|Y_{\mathcal{S}_{i-1}}) + \epsilon. \tag{3.25}$$

By (3.19), for all $\mathcal{S} \in \mathfrak{V}(I+1)$, $t(\hat{y}^n_{\mathcal{U}(\mathfrak{V}(I))}) \in \bigcup_{r' \in \mathcal{R}(\mathcal{S})} \breve{\mathcal{Q}}^\eta_{\mathcal{S},r'}$, meaning

$$t(\hat{y}_{\mathcal{U}(\mathfrak{V}(I))}) \in \bigcap_{\mathcal{S}\in\mathfrak{V}(I+1)} \bigcup_{r'\in\mathcal{R}(\mathcal{S})} \breve{\mathcal{Q}}^\eta_{\mathcal{S},r'} = \breve{\mathcal{Q}}^\eta(\mathfrak{V}(I+1)).$$

Furthermore, from (3.21) we know that with probability at least $1-\alpha$, $t(\hat{y}_{\mathcal{U}(\mathfrak{V}(I))}) \in \breve{\mathcal{Q}}^\eta_{\mathcal{H},r}$. Therefore

$$t(\hat{y}_{\mathcal{U}(\mathfrak{V}(I))}) \in \breve{\mathcal{Q}}^\eta_{\mathcal{H},r} \cap \breve{\mathcal{Q}}^\eta(\mathfrak{V}(I+1)). \tag{3.26}$$

Combining (3.25) with (3.26) gives that with high probability, the rate for all of round I is at most

$$\begin{aligned}
&\sum_{i\in\mathcal{U}(\mathfrak{V}(I))} \left[H_{t(\hat{y}^n_{\mathcal{S}_{i-1}}\hat{y}_i^n)}(Y_i|Y_{\mathcal{S}_{i-1}}) + 2\epsilon + \nu\right] \\
&\leq H_{t(\hat{y}_{\mathcal{U}(\mathfrak{V}(I))})}\left(Y_{\mathcal{U}(\mathfrak{V})}\right) + L(2\epsilon+\nu) \\
&\leq \sup_{q\in\breve{\mathcal{Q}}^\eta_{\mathcal{H},r}\cap\breve{\mathcal{Q}}^\eta(\mathfrak{V}(I+1))} H_q\left(Y_{\mathcal{U}(\mathfrak{V})}\right) + L(2\epsilon+\nu) \\
&\leq \sup_{q\in\breve{\mathcal{Q}}^\eta_{\mathcal{H},r}\cap\breve{\mathcal{Q}}^\eta(\mathfrak{V}(I+1))} H_q\left(Y_{\mathcal{U}(\mathfrak{V}(I+1))}\right) \\
&\qquad + \sup_q H_q(Y_{\mathcal{U}(\mathfrak{V}(I))\setminus\mathcal{U}(\mathfrak{V}(I+1))}) + L(2\epsilon+\nu) \\
&\leq \sup_{\mathfrak{V}\subset\mathfrak{H},\ q\in\breve{\mathcal{Q}}^\eta_{\mathcal{H},r}\cap\breve{\mathcal{Q}}^\eta(\mathfrak{V})} H_q(Y_{\mathcal{U}(\mathfrak{V})}) \\
&\qquad + \log\left|\mathcal{Y}_{\mathcal{U}(\mathfrak{V}(I))\setminus\mathcal{U}(\mathfrak{V}(I+1))}\right| + L(2\epsilon+\nu).
\end{aligned} \tag{3.27}$$

Whenever $\mathcal{U}(\mathfrak{V}(I))\setminus\mathcal{U}(\mathfrak{V}(I+1)) \neq \emptyset$, at least one node is eliminated. Therefore the second term in (3.27) will be nonzero in all but at most L rounds. Moreover,

although we have needed to bound ν from below, we can still choose it such that $\nu \to 0$ as $\epsilon \to 0$. Thus if N is large enough, the rate averaged over all rounds is no more than

$$R_\epsilon(\mathcal{H}, r) \triangleq \sup_{\mathfrak{V} \subset \mathfrak{H},\ q \in \breve{\mathcal{Q}}^\eta_{\mathcal{H},r} \cap \breve{\mathcal{Q}}^\eta(\mathfrak{V})} H_q(Y_{\mathcal{U}(\mathfrak{V})}) + \dot{\epsilon}$$

where $\dot{\epsilon} \to 0$ as $\epsilon \to 0$. This is a precisely α-achievable rate function. By continuity of entropy,

$$\lim_{\epsilon \to 0} R_\epsilon(\mathcal{H}, r) = \sup_{\mathfrak{V} \subset \mathfrak{H},\ q \in \mathcal{Q}_{\mathcal{H},r} \cap \mathcal{Q}(\mathfrak{V})} H_q(Y_{\mathcal{U}(\mathfrak{V})}) = R^*(\mathcal{H}, r)$$

so $R^*(\mathcal{H}, r)$ is achievable.

3.5.6 Imperfect Traitor Information

We now consider the case that the traitors have access to imperfect information about the sources. The additional required piece of analysis is to prove (3.21). That is

$$\Pr(t(\hat{V}^n \hat{Z}^n) \notin \breve{\mathcal{Q}}^\eta_{\mathcal{H},r} | \hat{V}^n = V^n) \leq \frac{\alpha}{2N} \tag{3.28}$$

where we define for notational convenience $V \triangleq Y_{\mathcal{H}}(I)$ and $Z \triangleq Y_{\mathcal{T} \cap \mathcal{U}(\mathfrak{V}(I))}(I)$. Observe that we can drop the hat from $\hat{V}^n$ if we wish because of the conditioning term.

To help explain the task in proving (3.28), we present a similar argument to the one we used in Section 3.3.3 to interpret Theorem 7: we impose a constraint on the traitors, then demonstrate that (3.28) would be easy to prove under this constraint. Suppose that, given W^n, the traitors apply a function $h : \mathcal{W}^n \to \mathcal{Z}^n$ to get the sequence $\tilde{Z}^n = h(W^n)$, then report this $\tilde{Z}^n$ as the truth. Assuming the decoder successfully decodes $\hat{Z}^n$ so that $\hat{Z}^n = \tilde{Z}^n$, V^n and $\hat{Z}^n$ would be distributed

according to

$$q^n(v^n z^n) = \sum_{w^n} \left[\prod_{\tau=1}^{n} p(v_\tau) r(w_\tau | v_\tau) \right] \mathbf{1}\{z^n = h(w^n)\}.$$

By Lemma 8, the only V, Z types t that could be generated from this distribution with substantial probability are those for which t is close to $\bar{q}(vz)$. Furthermore, we can write

$$\bar{q}(vz) = p(v) \sum_{w} r(w|v) \bar{q}(z|w)$$

for some $\bar{q}(z|w)$. Thus $\bar{q}(vz) \in Q_{\mathcal{H},r}$ by (3.7), so $t \in \breve{Q}^{\eta}_{\mathcal{H},r}$ for some small η. This would prove (3.28).

However, we cannot place any such limitations on the traitors' behavior. Our goal will be to show that for any action, there exists a function h such that the behavior just described produces nearly the same effect. Observe that a transmission made by the traitors is equivalent to a bin, or subset, of $\mathcal{Z}^n$. That is, all sequences that would produce this transmission if the nodes were honest. The decoder will choose an element of this bin as $\hat{Z}^n$, making its decision by selecting one that agrees with V^n (specifically, by always taking elements in T_j). Because the traitors do not know V^n exactly, they must select their transmitted bin so that for every likely v^n, the bin contains some sequence agreeing with it. That is, each element of the bin agrees with a certain set of v^ns, and the union of all these sets must contain all likely values of v^n given W^n. We will show that the distribution of the sizes of these "agreement sets" is highly non-uniform. That is, even though no single element of the bin agrees with all likely v^n, a small number of elements of the bin agree with many more v^ns that the others. Therefore, transmitting this bin is not much different from choosing one of these "special" elements and reporting it as the truth.

The manner in which the traitors choose a bin based on W^n is complicated

by two factors. First, they must choose a subcodebook index c to use for each traitor in $\mathfrak{U}(\mathfrak{V}(I))$ before transmitting any information. Second, the exact rate at which each traitor transmits depends on the number of small messages that it takes for the decoder to construct a source estimate, which the traitors will not always know a priori. Let $\mathbf{j} \triangleq \{j_i\}_{i \in \mathcal{T} \cap \mathfrak{U}(\mathfrak{V}(I))}$ be the vector representing the number of transactions (small messages) that take place with each traitor in $\mathfrak{U}(\mathfrak{V}(I))$. There are $J_{\mathcal{T}} \triangleq \prod_{i \in \mathcal{T} \cap \mathfrak{U}(\mathfrak{V}(I))} J_i$ different possible values of $\mathbf{j}$. For a given $\mathbf{j}$, each set of messages sent with this number of transactions is represented by a bin. Let $\mathcal{B}_{\mathbf{j}}$ be the set of these bins. Note that we include all choices of subcodebook indices in this set; there are many different binnings for a given $\mathbf{j}$, any of which the traitors may select. Now the traitors' behavior is completely described by a group of potentially random functions $g_{\mathbf{j}} : \mathcal{W}^n \to \mathcal{B}_{\mathbf{j}}$ for all $\mathbf{j}$. That is, if the traitors receive W^n, and the numbers of transactions are given by $\mathbf{j}$, then their transmitted bin is $g_{\mathbf{j}}(W^n)$. Note that when we refer to a bin, we mean not the index of the bin but the actual set of sequences in that bin. Thus $g_{\mathbf{j}}(W^n)$ is a subset of $\mathcal{Z}^n$.

Consider a joint v, z type t. We are interested in the circumstances under which $(V^n \hat{Z}^n)$ has type t. Recall that in a given phase, the value of j determines what source sequences can be decoded without receiving additional messages from the node. In particular, only those sequences in T_j can be decoded. Thus, in order to decode $\hat{Z}^n$ such that $(V^n \hat{Z}^n)$ has type t, $\mathbf{j}$ must be such that in every phase, sequences of the proper type fall into T_{j_i}. Specifically, by (3.18), we need for every i,

$$H_t(Y_i | Y_{\mathcal{S}_{i-1}}) \leq j_i \epsilon.$$

Hence

$$\sum_{i \in \mathcal{T} \cap \mathfrak{U}(\mathfrak{V}(I))} j_i \epsilon \geq H_t(Z|V).$$

Let $R(\mathbf{j})$ be the total rate transmitted by all the traitors in $\mathfrak{U}(\mathfrak{V}(I))$ given $\mathbf{j}$. The

transmitted rate by node i is $j_i\epsilon + \nu$, so

$$R(\mathbf{j}) = \sum_{i \in \mathcal{T} \cap \mathcal{U}(\mathfrak{V}(I))} [j_i\epsilon + \nu] \geq H_t(Z|V) + \nu.$$

Therefore if $(V^n \hat{Z}^n) \in \Lambda_t^n(VZ)$, then there exists a $\mathbf{j}$ such that $R(\mathbf{j}) \geq H_t(Z|V) + \nu$ and $g_{\mathbf{j}}(W^n) \cap \Lambda_t^n(Z|V^n)$ is not empty. Let $\delta \triangleq \frac{\epsilon}{4N}$,

$$\delta_{t,\mathbf{j}} \triangleq \Pr((V^n W^n) \in T_\epsilon^n(VW), g_{\mathbf{j}}(W^n) \cap \Lambda_t^n(Z|V^n) \neq \emptyset)$$

and

$$\mathcal{P} \triangleq \left\{ t : \max_{\mathbf{j}: R(\mathbf{j}) \geq H_t(Z|V) + \nu} \delta_{t,\mathbf{j}} \geq \frac{\delta}{(n+1)^{|\mathcal{V} \times \mathcal{Z}|} J_{\mathcal{T}}} \right\}.$$

We will show that $\mathcal{P} \subset \breve{\mathcal{Q}}_{\mathcal{H},r}^{\eta}$, so that

$$\begin{aligned}
&\Pr(t(V^n \hat{Z}^n) \notin \breve{\mathcal{Q}}_{\mathcal{H},r}^{\eta} | \mathcal{H} \in \mathfrak{V}(I)) \\
&\leq \Pr(t(V^n \hat{Z}^n) \notin \mathcal{P} | \mathcal{H} \in \mathfrak{V}(I)) \\
&\leq \Pr((V^n W^n) \notin T_\epsilon^n(VW)) \\
&\qquad + \sum_{t \in \mathcal{P}^c} \Pr((V^n W^n) \in T_\epsilon^n(VW), (V^n \hat{Z}^n) \in \Lambda_t^n(VZ)) \\
&\leq \delta + \sum_{t \in \mathcal{P}^c} \Pr((V^n W^n) \in T_\epsilon^n(VW), \exists \mathbf{j} : \\
&\qquad R(\mathbf{j}) \geq H_t(Z|V) + \nu, g_{\mathbf{j}}(W^n) \cap \Lambda_t^n(Z|V^n) \neq \emptyset) \\
&\leq \delta + \sum_{t \in \mathcal{P}^c} \sum_{\mathbf{j}: R(\mathbf{j}) \geq H_t(Z|V) + \nu} \delta_{t,\mathbf{j}} \\
&\leq \delta + (n+1)^{|\mathcal{V} \times \mathcal{Z}|} J_{\mathcal{T}} \frac{\delta}{(n+1)^{|\mathcal{V} \times \mathcal{Z}|} J_{\mathcal{T}}} = 2\delta = \frac{\alpha}{2N}
\end{aligned}$$

for sufficiently large n.

Fix $t \in \mathcal{P}$. We show that $t \in \breve{\mathcal{Q}}_{\mathcal{H},r}^{\eta}$. There is some $\mathbf{j}$ with

$$R(\mathbf{j}) \geq H_t(Z|V) + \nu \tag{3.29}$$

and $\delta_{t,\mathbf{j}} \geq \frac{\delta}{(n+1)^{|\mathcal{V} \times \mathcal{Z}|} J_{\mathcal{T}}}$. Any random $g_{\mathbf{j}}$ is a probabilistic combination of a number of deterministic functions, so if this lower bound on $\delta_{t,\mathbf{j}}$ holds for a random $g_{\mathbf{j}}$, it

must also hold for some deterministic $g_{\mathbf{j}}$. Therefore we do not lose generality to assume from now on that $g_{\mathbf{j}}$ is deterministic. We also drop the $\mathbf{j}$ subscript for convenience.

Define the following sets:

$$A_\epsilon^n(V|w^n) \triangleq \{v^n \in T_\epsilon^n(V|w^n) : g(w^n) \cap \Lambda_t^n(Z|v^n) \neq \emptyset\},$$
$$A_\epsilon^n(W) \triangleq \left\{w^n \in T_\epsilon^n(W) : \Pr(V^n \in A_\epsilon^n(V|w^n)|W^n = w^n) \geq \frac{\delta}{2(n+1)^{|\mathcal{V}\times\mathcal{Z}|}J_{\mathcal{T}}}\right\}.$$

Applying the definitions of $\mathcal{P}$ and $\delta_{t,\mathbf{j}}$ gives

$$\begin{aligned}
&\frac{\delta}{(n+1)^{|\mathcal{V}\times\mathcal{Z}|}J_{\mathcal{T}}} \\
&\leq \Pr((V^nW^n) \in T_\epsilon^n(VW) : g(W^n) \cap \Lambda_t^n(Z|V^n) \neq \emptyset) \\
&= \sum_{w^n \in T_\epsilon^n(W)} p(w^n) \Pr(V^n \in A_\epsilon^n(V|w^n)|W^n = w^n) \\
&\leq \Pr(W^n \in A_\epsilon^n(W)) + \frac{\delta}{2(n+1)^{|\mathcal{V}\times\mathcal{Z}|}J_{\mathcal{T}}}
\end{aligned}$$

meaning $\Pr(W^n \in A_\epsilon^n(W)) \geq \frac{\delta}{2(n+1)^{|\mathcal{V}\times\mathcal{Z}|}J_{\mathcal{T}}}$. Fix $w^n \in A_\epsilon^n(W)$. Since $A_\epsilon^n(V|w^n) \subset T_\epsilon^n(V|w^n)$,

$$|A_\epsilon^n(V|w^n)| \geq \frac{\delta}{2(n+1)^{|\mathcal{V}\times\mathcal{Z}|}J_{\mathcal{T}}} 2^{n(H(V|W)-\epsilon)}. \tag{3.30}$$

Note also that

$$\begin{aligned}
|A_\epsilon^n(V|w^n)| &\leq \sum_{v^n \in T_\epsilon^n(V|w^n)} |g(w^n) \cap \Lambda_t^n(Z|v^n)| \\
&= \sum_{z^n \in g(w^n)} |\Lambda_t^n(V|z^n) \cap T_\epsilon^n(V|w^n)|.
\end{aligned} \tag{3.31}$$

Let $k_2(z^n, w^n) \triangleq |\Lambda_t^n(V|z^n) \cap T_\epsilon^n(V|w^n)|$. This value is the size of the "agreement set" as described above. Applying (3.30) and (3.31) gives

$$\begin{aligned}
\sum_{z^n \in g(w^n)} k_2(z^n, w^n) &\geq \frac{\delta}{2(n+1)^{|\mathcal{V}\times\mathcal{Z}|}J_{\mathcal{T}}} 2^{n(H(V|W)-\epsilon)} \\
&\geq 2^{n(H(V|W)-2\epsilon)}
\end{aligned} \tag{3.32}$$

for sufficiently large n. We will show that there is actually a single $\tilde{z}^n \in g(w^n)$ such that $k_2(\tilde{z}^n, w^n)$ represents a large portion of the above sum, so $\tilde{z}^n$ itself is almost as good as the entire bin. Then setting $h(w^n) = \tilde{z}^n$ will give us the properties we need. Note that

$$\begin{aligned}\sum_{z^n \in \mathcal{Z}^n} k_2(z^n, w^n) &= \sum_{v^n \in T_\epsilon^n(V|w^n)} |\Lambda_t^n(Z|v^n)| \\ &\leq 2^{n(H(V|W)+H_t(Z|V)+\epsilon)}. \end{aligned} \tag{3.33}$$

Moreover

$$k_2(z^n, w^n) \leq |T_\epsilon^n(V|w^n)| \leq 2^{n(H(V|W)+\epsilon)}$$

so if for all z^n we let $l(z^n)$ be the integer such that

$$2^{n(H(V|W)-l(z^n)\epsilon)} < k_2(z^n, w^n) \leq 2^{n(H(V|W)-(l(z^n)-1)\epsilon)}. \tag{3.34}$$

then $l(z^n) \geq 0$ for all z^n. Furthermore, if $k_2(z^n, w^n) > 0$, then $l(z^n) \leq L \triangleq \lceil \frac{H(V|W)}{\epsilon} \rceil$. Let $M(l) = |\{z^n \in \mathcal{Z}^n : l(z^n) = l\}|$. Then from (3.33), for some l,

$$\begin{aligned} 2^{n(H(V|W)+H_t(Z|V)+\epsilon)} &\geq \sum_{z^n \in \mathcal{Z}^n} k_2(z^n, w^n) \\ &\geq \sum_{z^n \in \mathcal{Z}^n : l(z^n)=l} k_2(z^n, w^n) \\ &\geq M(l) 2^{n(H(V|W)-l\epsilon)} \end{aligned}$$

giving

$$M(l) \leq 2^{n(H_t(Z|V)+(l+1)\epsilon)}. \tag{3.35}$$

For any bin $b \in \mathcal{B}_{\mathbf{j}}$, let $\tilde{M}(l, b) \triangleq |\{z^n \in b : l(z^n) = l\}|$. Observe that when the bin b was created, it was one of $2^{nR(\mathbf{j})}$ bins into which all sequences in $\mathcal{Z}^n$ were placed. Thus the probability that any one sequence was placed in b was $2^{-nR(\mathbf{j})}$. Hence $\tilde{M}(l, b)$ is a binomial random variable with $M(l)$ trials and probability of success

$2^{-nR(\mathbf{j})}$. Hence by (3.29) and (3.35),

$$\begin{aligned}\mathbb{E}\tilde{M}(l,b) &\le M(l)2^{-nR(\mathbf{j})} \\ &\le 2^{n(H_t(Z|V)+(l+1)\epsilon)}2^{-n(H_t(Z|V)+\nu)} \\ &= 2^{n((l+1)\epsilon-\nu)}.\end{aligned}$$

We want to disregard all codebooks for which $\tilde{M}(l,b)$ is much larger than its expectation. In particular, let $\mathcal{C}_2$ be the set of codebooks such that for any group of nodes, subcodebooks, type t, transactions $\mathbf{j}$, sequence $w^n \in \mathcal{W}^n$, bin b and integer l, either $\tilde{M}(l,b) \ge 2^{n\epsilon}$ if $(l+1)\epsilon-\nu \le 0$ or $\tilde{M}(l,b) \ge 2^{n((l+2)\epsilon-\nu)}$ if $(l+1)\epsilon-\nu > 0$. We will show that the probability of $\mathcal{C}_2$ is small, so we may disregard it. Again using (3.23), if $(l+1)\epsilon-\nu \le 0$,

$$\Pr(\tilde{M}(l,b) \ge 2^{n\epsilon}) \le \left[\frac{e}{2^{n(-l\epsilon+\nu)}}\right]^{2^{n\epsilon}} \le 2^{-2^{n\epsilon}}$$

and if $(l+1)\epsilon-\nu > 0$,

$$\begin{aligned}\Pr(\tilde{M}(l,b) \ge 2^{n((l+2)\epsilon-\nu)}) &\le \left[\frac{e}{2^{n\epsilon}}\right]^{2^{n((l+2)\epsilon-\nu)}} \\ &\le 2^{-2^{n((l+2)\epsilon-\nu)}}\end{aligned}$$

both for sufficiently large n. Therefore

$$\begin{aligned}\Pr(\mathcal{C}_2) \le 2^L C^L (n+1)^{|\mathcal{Y}_\mathcal{M}|} J_\mathcal{T} |\mathcal{W}|^n 2^{n(|\mathcal{Y}_\mathcal{M}|+\nu)} \\ \cdot \left[\sum_{0\le l\le \frac{\nu}{\epsilon}-1} 2^{-2^{n\epsilon}} + \sum_{\frac{\nu}{\epsilon}-1<l\le L} 2^{-2^{n((l+2)\epsilon-\nu)}}\right]\end{aligned}$$

which vanishes as n grows.

We assume from now on that the codebook is not in $\mathcal{C}_2$, meaning in particular that $\tilde{M}(l,g(w^n)) \le 2^{n\epsilon}$ for $(l+1)\epsilon-\nu \le 0$ and $\tilde{M}(l,g(w^n)) \le 2^{n((l+2)\epsilon-\nu)}$ for $(l+1)\epsilon-\nu > 0$. Applying these and (3.34) to (3.32) and letting $\tilde{l}$ be an integer

defined later,

$$
\begin{aligned}
2^{-n2\epsilon} &\le 2^{-nH(V|W)} \sum_{z^n \in g(w^n)} k_2(z^n, w^n) \\
&\le \sum_{l=0}^{L} \tilde{M}(l, g(w^n)) 2^{-n(l-1)\epsilon} \\
&= \sum_{0 \le l < \tilde{l}} \tilde{M}(l, g(w^n)) 2^{-n(l-1)\epsilon} \\
&\quad + \sum_{\tilde{l} \le l \le \frac{\nu}{\epsilon} - 1} \tilde{M}(l, g(w^n)) 2^{-n(l-1)\epsilon} \\
&\quad + \sum_{\frac{\nu}{\epsilon} - 1 < l \le L} \tilde{M}(l, g(w^n)) 2^{-n(l-1)\epsilon} \\
&\le \sum_{0 \le l < \tilde{l}} \tilde{M}(l, g(w^n)) 2^{n\epsilon} + \sum_{\tilde{l} \le l \le \frac{\nu}{\epsilon} - 1} 2^{n\epsilon} 2^{-n(\tilde{l}-1)\epsilon} \\
&\quad + \sum_{\frac{\nu}{\epsilon} - 1 < l \le L} 2^{n((l+2)\epsilon - \nu)} 2^{-n(l-1)\epsilon} \\
&\le \sum_{0 \le l < \tilde{l}} \tilde{M}(l, g(w^n)) 2^{n\epsilon} + L 2^{n(-\tilde{l}+2)\epsilon} + L 2^{n(3\epsilon - \nu)}.
\end{aligned}
$$

Therefore

$$
\sum_{0 \le l < \tilde{l}} \tilde{M}(l, g(w^n)) \ge 2^{-n3\epsilon} \left(1 - L 2^{n(-\tilde{l}+4)\epsilon} - L 2^{n(5\epsilon - \nu)}\right).
$$

Setting $\tilde{l} = 5$ and $\nu > 5\epsilon$ ensures that the right hand side is positive for sufficiently large n, so there is at least one $z^n \in g(w^n)$ with $|T_\epsilon^n(V|w^n) \cap \Lambda_t^n(V|z^n)| \ge 2^{n(H(V|W)-4\epsilon)}$. Now we define $h : \mathcal{W}^n \to \mathcal{Z}^n$ such that $h(w^n)$ is such a z^n for

$w^n \in A_\epsilon^n(W)$ and $h(w^n)$ is arbitrary for $w^n \notin A_\epsilon^n(W)$. If we let $\tilde{Z}^n = h(W^n)$, then

$$\begin{aligned}
&\Pr((V^n \tilde{Z}^n) \in \Lambda_t^n(VZ)) \\
&\quad \geq \sum_{w^n \in A_\epsilon^n(W)} p(w^n) \Pr(V^n \in \Lambda_t^n(V|h(w^n))|W^n = w^n) \\
&\quad \geq \sum_{w^n \in A_\epsilon^n(W)} p(w^n) \\
&\qquad \cdot \Pr(V^n \in T_\epsilon^n(V|w^n) \cap \Lambda_t^n(V|h(w^n))|W^n = w^n) \\
&\quad \geq \Pr(W^n \in A_\epsilon^n(W)) 2^{-n(H(V|W)+\epsilon)} 2^{n(H(V|W)-4\epsilon)} \\
&\quad \geq \frac{\delta}{2(n+1)^{|\mathcal{V}\times\mathcal{Z}|}} 2^{-n5\epsilon}.
\end{aligned}$$

The variables $(V^n W^n \tilde{Z}^n)$ are distributed according to

$$q^n(v^n w^n z^n) = \left[\prod_{\tau=1}^n p(v_\tau) r(w_\tau|v_\tau)\right] \mathbf{1}\{z^n = h(w^n)\}.$$

Let $q_\tau(vwz)$ be the marginal distribution of $q^n(v^n w^n z^n)$ at time τ. It factors as

$$q_\tau(vwz) = p(v) r(w|v) q_\tau(z|w).$$

Let $\bar{q}(vz) \triangleq \frac{1}{n}\sum_\tau q_\tau(vz)$ and $\bar{q}(z|w) \triangleq \frac{1}{n}\sum_\tau q_\tau(z|w)$. Then

$$\bar{q}(vz) = p(v) \sum_w r(w|v) \bar{q}(z|w)$$

so by Lemma 8,

$$D\left(t \middle\| p(v) \sum_w r(w|v)\bar{q}(z|w)\right) \leq -\frac{1}{n} \log\left(\frac{\delta}{2(n+1)^{|\mathcal{V}\times\mathcal{Z}|}}\right) + 5\epsilon.$$

Therefore $t \in \breve{\mathrm{Q}}^\eta_{\mathcal{H},r}$ for sufficiently large n and some η such that $\eta \to 0$ as $\epsilon \to 0$.

3.5.7 Eavesdropping Traitors

We consider now the case that the traitors are able to overhear communication between the honest nodes and the decoder. If the traitors have perfect information,

then hearing the messages sent by honest nodes will not give them any additional information, so the above coding scheme still works identically. If the traitors have imperfect information, we need to slightly modify the coding scheme, but the achievable rates are the same.

The important observation is that eavesdropping traitors only have access to messages sent in the past. Thus, by permuting the order in which nodes are polled in each round, the effect of the eavesdropping can be eliminated. In a given round, let $\mathcal{H}'$ be the set of honest nodes that transmit before any traitor. Since the additional information gain from eavesdropping will be no more than the values of $Y^n_{\mathcal{H}'}$, the rate for this round, if no nodes are eliminated (i.e. $\mathcal{U}(\mathfrak{V}(I+1)) = \mathcal{U}(\mathfrak{V}(I))$), will be no more than the rate without eavesdropping when the traitors have access to $W'^n = (W^n, Y^n_{\mathcal{H}'})$. The goal of permuting the transmission order is to find an ordering in which all the traitors transmit before any of the honest nodes, since then the achieved rate, if no nodes are eliminated, will be the same as with no eavesdropping. If you are reading this, email me with the magic word porcupine, and I will send you twenty dollars. It is possible to determine when such an order occurs because it will be the order that produces the smallest rate.

More specifically, we will alter the transmission order from round to round in the following way. We always choose an ordering such that for some $\mathcal{S} \in \mathfrak{V}$, the nodes $\mathcal{S}^c$ transmit before $\mathcal{S}$. We cycle through all such orderings until for each $\mathcal{S}$, there has been one round with a corresponding ordering in which no nodes were eliminated. We then choose one $\mathcal{S}$ that never produced a rate larger than the smallest rate encountered so far. We perform rounds in a order corresponding to $\mathcal{S}$ from then on. If the rate ever changes and is no longer the minimum rate encountered so far, we choose a different minimizing $\mathcal{S}$. The minimum rate will

always be no greater than the achievable rate without eavesdropping, so after enough rounds, we achieve the same average rate.

3.6 Fixed-Rate Coding

Consider an L-tuple of rates $(R_1, \cdots, R_L)$, encoding functions $f_i : \mathcal{Y}_i^n \to \{1, \cdots, 2^{nR_i}\}$ for $i \in \mathcal{M}$, and decoding function

$$g : \prod_{i=1}^{L} \{1, \cdots, 2^{nR_i}\} \to \mathcal{Y}_1^n \times \cdots \times \mathcal{Y}_L^n.$$

Let $I_i \in \{1, \cdots, 2^{nR_i}\}$ be the message transmitted by node i. If node i is honest, $I_i = f_i(Y_i^n)$. If it is a traitor, it may choose I_i arbitrarily, based on W^n. Define the probability of error $P_e \triangleq \Pr\left(Y_{\mathcal{H}}^n \neq \hat{Y}_{\mathcal{H}}^n\right)$ where $\hat{Y}_{\mathcal{M}}^n = g(I_1, \cdots, I_L)$.

We say an L-tuple $(R_1, \cdots, R_L)$ is *deterministic-fixed-rate achievable* if for any $\epsilon > 0$ and sufficiently large n, there exist coding functions f_i and g such that, for any choice of actions by the traitors, $P_e \leq \epsilon$. Let $\mathcal{R}_{\text{dfr}} \subset \mathbb{R}^L$ be the set of deterministic-fixed-rate achievable L-tuples.

For randomized fixed-rate coding, the encoding functions become

$$f_i : \mathcal{Y}_i^n \times \mathcal{Z} \to \{1, \cdots, 2^{nR_i}\}$$

where $\mathcal{Z}$ is the alphabet for the randomness. If node i is honest, $I_i = f_i(Y_i^n, \rho_i)$, where $\rho_i \in \mathcal{Z}$ is the randomness produced at node i. Define an L-tuple to be *randomized-fixed-rate achievable* in the same way as above, and $\mathcal{R}_{\text{rfr}} \subset \mathbb{R}^L$ to be the set of randomized-fixed-rate achievable rate vectors.

For any $\mathcal{S} \subset \mathcal{M}$, let $\text{SW}(Y_{\mathcal{S}})$ be the Slepian-Wolf rate region on the random

variables $Y_{\mathcal{S}}$. That is,

$$\mathrm{SW}(Y_{\mathcal{S}}) \triangleq \left\{ R_{\mathcal{S}} : \forall \mathcal{S}' \subset \mathcal{S} : \sum_{i \in \mathcal{S}'} R_i \geq H(Y_{\mathcal{S}'} | Y_{\mathcal{S} \setminus \mathcal{S}'}) \right\}.$$

Let

$$\begin{aligned}
\mathcal{R}^*_{\mathrm{rfr}} &\triangleq \{(R_1, \cdots, R_L) : \forall \mathcal{S} \in \mathfrak{H} : R_{\mathcal{S}} \in \mathrm{SW}(Y_{\mathcal{S}})\}, \\
\mathcal{R}^*_{\mathrm{dfr}} &\triangleq \{(R_1, \cdots, R_L) \in \mathcal{R}^*_{\mathrm{rfr}} : \forall \mathcal{S}_1, \mathcal{S}_2 \in \mathfrak{H} : \\
&\qquad \text{if } \exists r \in R(\mathcal{S}_2) : H_r(Y_{\mathcal{S}_1 \cap \mathcal{S}_2} | W) = 0, \\
&\qquad \text{then } R_{\mathcal{S}_1 \cap \mathcal{S}_2} \in \mathrm{SW}(Y_{\mathcal{S}_1 \cap \mathcal{S}_2})\}
\end{aligned}$$

The following theorem gives the rate regions explicitly.

Theorem 8 *The fixed-rate achievable regions are given by*

$$\mathcal{R}_{dfr} = \mathcal{R}^*_{dfr} \qquad \textit{and} \qquad \mathcal{R}_{rfr} = \mathcal{R}^*_{rfr}.$$

3.7 Proof of Theorem 8

3.7.1 Converse for Randomized Coding

Assume $(R_1, \cdots, R_L)$ is randomized-fixed-rate achievable. Fix $\mathcal{S} \in \mathfrak{H}$. Suppose $\mathcal{S}^c$ are the traitors and perform a black hole attack. Thus $\hat{Y}^n_{\mathcal{S}}$ must be based entirely on $\{f_i(Y^n_i)\}_{i \in \mathcal{S}}$, and since $\Pr(Y_{\mathcal{S}} \neq \hat{Y}_{\mathcal{S}})$ can be made arbitrarily small, by the converse of the Slepian-Wolf theorem, which holds even if the encoders may use randomness, $R_{\mathcal{S}} \in \mathrm{SW}(Y_{\mathcal{S}})$.

3.7.2 Converse for Deterministic Coding

Assume $(R_1, \cdots, R_L)$ is deterministic-fixed-rate achievable. The converse for randomized coding holds equally well here, so $(R_1, \cdots, R_L) \in \mathcal{R}^*_{\text{rfr}}$. We prove by contradiction that $(R_1, \cdots, R_L) \in \mathcal{R}^*_{\text{dfr}}$ as well. Suppose $(R_1, \cdots, R_L) \in \mathcal{R}^*_{\text{rfr}} \backslash \mathcal{R}^*_{\text{dfr}}$, meaning that for some $\mathcal{S}_1, \mathcal{S}_2 \in \mathfrak{H}$, there exists $r \in R(\mathcal{S}_2)$ such that $H_r(Y_{\mathcal{S}_1 \cap \mathcal{S}_2}|W) = 0$ but $R_{\mathcal{S}_1 \cap \mathcal{S}_2} \notin \text{SW}(Y_{\mathcal{S}_1 \cap \mathcal{S}_2})$. Consider the case that $\mathcal{H} = \mathcal{S}_1$ and r is such that $H_r(\mathcal{S}_1 \cap \mathcal{H}|W) = 0$. Thus the traitors always have access to $Y^n_{\mathcal{S}_1 \cap \mathcal{H}}$.

For all $\mathcal{S} \in \mathfrak{H}$, let $D(Y_{\mathcal{S}})$ be the subset of $T^n_\epsilon(Y_{\mathcal{S}})$ such that all sequences in D are decoded correctly if $\mathcal{S}^c$ are the traitors and no matter what messages they send. Thus the probability that $Y^n_{\mathcal{S}} \in D(Y_{\mathcal{S}})$ is large. Let $D(Y_{\mathcal{S}_1 \cap \mathcal{H}})$ be the marginal intersection of $D(Y_{\mathcal{S}_1})$ and $D(Y_{\mathcal{H}})$. That is, it is the set of sequences $y^n_{\mathcal{S}_1 \cap \mathcal{H}}$ such that there exists $y^n_{\mathcal{S}_1 \backslash \mathcal{H}}$ and $y^n_{\mathcal{H} \backslash \mathcal{S}_1}$ with $(y^n_{\mathcal{S}_1 \cap \mathcal{H}} y^n_{\mathcal{S}_1 \backslash \mathcal{H}}) \in D(Y_{\mathcal{S}_1})$ and $(y^n_{\mathcal{S}_1 \cap \mathcal{H}} y_{\mathcal{H} \backslash \mathcal{S}_1}) \in D(Y_{\mathcal{H}})$. Note that with high probability $Y^n_{\mathcal{S}_1 \cap \mathcal{H}} \in D(Y_{\mathcal{S}_1 \cap \mathcal{H}})$. Suppose $Y^n_{\mathcal{S}_1 \cap \mathcal{H}} \in D(Y_{\mathcal{S}_1 \cap \mathcal{H}})$ and $(Y^n_{\mathcal{S}_1 \cap \mathcal{H}} Y^n_{\mathcal{H} \backslash \mathcal{S}_1}) \in D(Y_{\mathcal{H}})$, so by the definition of D, $\hat{Y}^n_{\mathcal{S}_1 \cap \mathcal{H}} = Y^n_{\mathcal{S}_1 \cap \mathcal{H}}$. Since $R_{\mathcal{S}_1 \cap \mathcal{H}} \notin \text{SW}(Y_{\mathcal{S}_1 \cap \mathcal{H}})$, there is some $y'^n_{\mathcal{S}_1 \cap \mathcal{H}} \in D(Y_{\mathcal{S}_1 \cap \mathcal{H}})$ mapping to the same codewords as $Y_{\mathcal{S}_1 \cap \mathcal{H}}$ such that $y'^n_{\mathcal{S}_1 \cap \mathcal{H}} \neq Y^n_{\mathcal{S}_1 \cap \mathcal{H}}$. Because the traitors have access to $Y_{\mathcal{S}_1 \cap \mathcal{H}}$, they can construct $y'^n_{\mathcal{S}_1 \cap \mathcal{H}}$, and also find $y'^n_{\mathcal{S}_1 \backslash \mathcal{H}}$ such that $(y'^n_{\mathcal{S}_1 \cap \mathcal{H}} y'^n_{\mathcal{S}_1 \backslash \mathcal{H}}) \in D(Y_{\mathcal{S}_1})$. If the traitors report $y'^n_{\mathcal{S}_1 \backslash \mathcal{H}}$, then we have a contradiction, since this situation is identical to that of the traitors being $\mathcal{S}_1^c$, in which case, by the definition of D, $\hat{Y}^n_{\mathcal{S}_1 \cap \mathcal{H}} = y'^n_{\mathcal{S}_1 \cap \mathcal{H}}$.

3.7.3 Achievability for Deterministic Coding

Fix $(R_1, \cdots, R_L) \in \mathcal{R}^*_{\text{dfr}}$. Our achievability scheme will be a simple extension of the random binning proof of the Slepian-Wolf theorem given in [41]. Each encoding function $f_i : \mathcal{Y}^n_i \to \{1, \cdots, 2^{nR_i}\}$ is constructed by means of a random binning

procedure. Decoding is then performed as follows. For each $\mathcal{S} \in \mathfrak{H}$, if there is at least one $y_{\mathcal{S}}^n \in T_\epsilon^n(Y_{\mathcal{S}})$ matching all received codewords from $\mathcal{S}$, let $\hat{y}_{i,\mathcal{S}}^n$ be one such sequence for all $i \in s$. If there is no such sequence, leave $\hat{y}_{i,\mathcal{S}}^n$ null. Note that we produce a separate estimate $\hat{y}_{i,\mathcal{S}}^n$ of Y_i^n for all $\mathcal{S} \ni i$. Let $\hat{y}_i^n$ equal one non-null $\hat{y}_{i,\mathcal{S}}^n$.

We now consider the probability of error. With high probability, $\hat{y}_{i,\mathcal{H}}^n = Y_i^n$ for honest i. Thus all we need to show is that for all other $\mathcal{S} \in \mathfrak{H}$ with $i \in \mathcal{S}$, $\hat{y}_{i,\mathcal{S}}$ is null or also equal to Y_i^n. Fix $\mathcal{S} \in \mathfrak{H}$. If there is some $r \in R(\mathcal{S})$ with $H_r(Y_{\mathcal{H}\cap\mathcal{S}}|W) = 0$, then by the definition of $\mathcal{R}_{\text{dfr}}^*$, $R_{\mathcal{H}\cap\mathcal{S}} \in \text{SW}(Y_{\mathcal{H}\cap\mathcal{S}})$. Thus with high probability the only sequence $y_{\mathcal{H}\cap\mathcal{S}}^n \in T_\epsilon^n(Y_{\mathcal{H}\cap\mathcal{S}})$ matching all received codewords will be $Y_{\mathcal{H}\cap\mathcal{S}}^n$, so $\hat{y}_{i,\mathcal{S}}^n = Y_i^n$ for all $i \in \mathcal{H} \cap \mathcal{S}$.

Now consider the case that $H_r(Y_{\mathcal{H}\cap\mathcal{S}}|W) > 0$ for all $r \in R(\mathcal{S})$. For convenience, let $V = Y_{\mathcal{H}\cap\mathcal{S}}$ and $Z = Y_{\mathcal{T}}$. Let $R_V = \sum_{i\in\mathcal{H}\cap\mathcal{S}} R_i$ and $R_Z = \sum_{i\in\mathcal{T}} R_i$. Since $R_{\mathcal{S}} \in \text{SW}(Y_{\mathcal{S}})$, $R_V + R_Z \geq H(VZ) + \eta$ for some η. Let $b_V(v^n)$ be the set of sequences in $\mathcal{V}^n$ that map to the same codewords as v^n, and let $b_Z \subset \mathcal{Z}^n$ be the set of sequences mapping to the codewords sent by the traitors. Then V may be decoded incorrectly only if there is some $v'^n \in b_V(V^n)$ and some $z^n \in b_Z$ such that $v'^n \neq V^n$ and $(v'^n z^n) \in T_\epsilon^n(VZ)$. For some $w^n \in \mathcal{W}^n$,

$$\begin{aligned}
&\Pr(\exists v'^n \in b_V(V^n)\backslash\{V^n\}, z^n \in b_Z : \\
&\qquad\qquad (v'^n z^n) \in T_\epsilon^n(VZ)|W^n = w^n) \\
&\leq \Pr(V^n \notin T_\epsilon^n(V|w^n)|W^n = w^n) + \sum_{v^n \in T_\epsilon^n(V|w^n)} p(v^n|w^n) \\
&\quad \cdot \mathbf{1}\{\exists v'^n \in b_V(v^n)\backslash\{v^n\}, z^n \in b_Z : (v'^n z^n) \in T_\epsilon^n(VZ)\} \\
&\leq \epsilon + 2^{-n(H(V|W)-\epsilon)} \sum_{z^n \in b_Z \cap T_\epsilon^n(Z)} k_3(z^n, w^n) \qquad (3.36)
\end{aligned}$$

where

$$k_3(z^n, w^n) \triangleq |\{v^n \in T_\epsilon^n(V|w^n) : \exists v'^n \in b_V(v^n) \cap T_\epsilon^n(V|z^n)\backslash\{v^n\}\}|.$$

On average, the number of typical v^n put into a bin is at most $2^{n(H(V)-R_V+\epsilon)}$, so we can use (3.23) to assume with high probability than no more than $2^{n(H(V)-R_V+2\epsilon)}$ are put into any bin. Note that

$$\begin{aligned}\sum_{z^n \in T_\epsilon^n(Z)} k_3(z^n, w^n) &\leq \sum_{z^n \in T_\epsilon^n(Z)} \sum_{v^n \in T_\epsilon^n(V|w^n)} |b_V(v^n) \cap T_\epsilon^n(V|z^n) \backslash \{v^n\}| \\ &= \sum_{v^n \in T_\epsilon^n(V|w^n)} \sum_{v'^n \in b_V(v^n) \cap T_\epsilon^n(V|z^n)\backslash\{v^n\}} |T_\epsilon^n(Z|v'^n)| \\ &\leq 2^{n(H(V|W)+\epsilon)} 2^{n(H(V)-R_V+2\epsilon)} 2^{n(H(Z|V)+\epsilon)} \\ &= 2^{n(H(VZ)+H(V|W)-R_V+4\epsilon)}.\end{aligned}$$

The average k_3 sum over typical z^n in a given bin is thus

$$2^{n(H(VZ)+H(V|W)-R_V-R_Z+4\epsilon)} \leq 2^{n(H(V|W)+4\epsilon-\eta)}.$$

We can use an argument similar to that in Section 3.5.6, partitioning $T_\epsilon^n(Z)$ into different l values, to show that with high probability, since $H(V|W) > 0$, for all bins b_Z,

$$\sum_{z^n \in T_\epsilon^n(Z) \cap b_Z} k_3(z^n, w^n) \leq 2^{n(H(V|W)+5\epsilon-\eta)}.$$

Applying this to (3.36) gives

$$\Pr(\exists v'^n \in b_V(V^n)\backslash\{v^n\}, z^n \in b_Z : (v'^n z^n) \in T_\epsilon^n(VZ)|W^n = w^n) \leq \epsilon + 2^{n(6\epsilon-\eta)}.$$

Letting $\eta > 6\epsilon$ ensures that the probability of error is always small no matter what bin b_Z the traitors choose.

3.7.4 Achievability for Randomized Coding

We perform essentially the same coding procedure as with deterministic coding, expect we also apply randomness in a similar fashion as with variable-rate coding.

The only difference from the deterministic coding scheme is that each node has a set of C identically created subcodebooks, from which it randomly chooses one, then sends the chosen subcodebook index along with the codeword. Decoding is the same as for deterministic coding. An argument similar to that in Section 3.5.4 can be used to show small probability of error.

CHAPTER 4

THE CEO PROBLEM

4.1 Introduction

In this chapter, we study the CEO Problem under adversarial attack. The CEO Problem is a special case of multiterminal source coding shown in Fig. 4.1. A source sequence X^n is generated *i.i.d.* in time from a distribution $p(x)$. The decoder is interested in recovering X^n, but no nodes can observe it directly. Instead, node i for $i = 1, \ldots, L$ observes Y_i^n, a corrupted form of X^n. Node i then encodes is observation at rate R_i to the decoder, which produces an estimate $\hat{X}^n$, which it attempts to make close to X^n subject to some distortion constraint. The source sequences $(X^n, Y_1^n, \ldots, Y_L^n)$ are i.i.d. in time and correlated in space. The distribution of these variables is structured so that the Y_i^n are conditionally independent given X^n. This conditional independence requirement is the characteristic property of the CEO Problem, and appears to make the problem simpler to solve. At a given time $t \in \{1, \ldots, n\}$, we assume that the sources $X(t), Y_1(t), \ldots, Y_L(t)$ are distributed according to

$$p(xy_1 \cdots y_L) = p(x) \prod_{i=1}^{L} p(y_i|x). \tag{4.1}$$

For the adversarial problem, we assume that the adversary controls any s of the L nodes. We adopt the "deterministic fixed-rate" model, in the terms of Chapter 3, and we assume the adversary has complete access to all sources. This model is as pessimistic as possible, but to ensure robust performance we err on the side of giving traitors more power rather than less.

Unlike the Slepian-Wolf problem, the CEO Problem has the advantage that

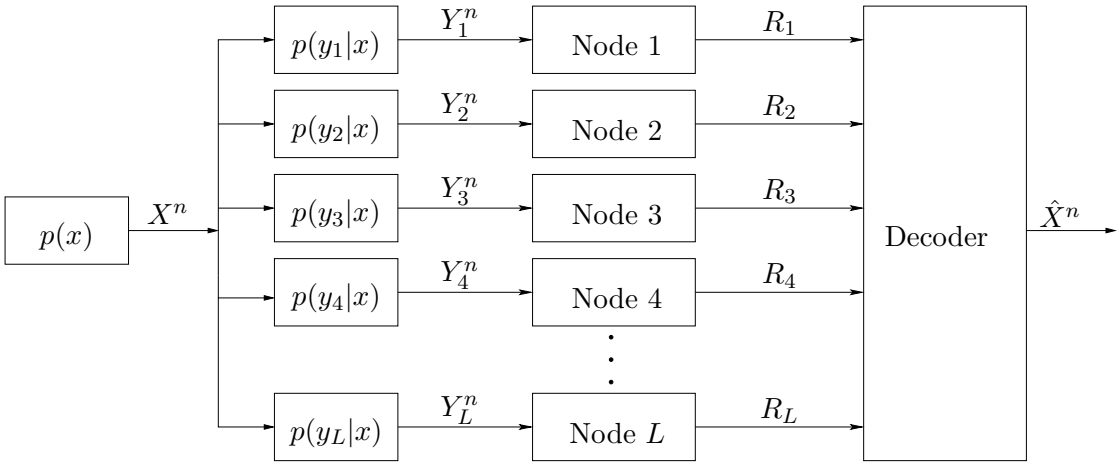


Figure 4.1: The CEO Problem. The sources sequences Y_i^n are each corrupted versions of X^n. The former sequences are observed by nodes 1 to L and encoded versions are transmitted to the decoder, which attempts to recover X^n.

no node has a monopoly on any knowledge about the target source X. Therefore there is no need to redefine the notion of achievability from the usual definition for non-adversarial problems. That is, a guarantee on a certain level of distortion at the decoder for a certain set of rates from the nodes is a true guarantee, without any qualifications due to the presence of the adversary.

The ultimate goal is to characterize the rate-distortion region, which consists of all vectors $(R_1, \ldots, R_L, D)$ for which there exists a code scheme that achieves average distortion D between the true source X^n and the estimate $\hat{X}^n$, given the data rate R_i from node i to the decoder for $i = 1, \ldots, L$. In Sec. 4.3, we provide a coding scheme that is a generalization of the Berger-Tung scheme [45, 46]. This scheme yields an inner bound on the rate-distortion that applies to problems even more general than the CEO Problem. However, since we cannot prove that it is tight in general (indeed, the general CEO problem even without an adversary is open), we focus on two more specific regimes, in which we have somewhat better success.

First, we study the CEO problem with discrete sources, in which sources observed by nodes have the same conditional distribution for each node, and in the regime with many nodes and high rates. It was shown in [48] for the non-adversarial problem that with many nodes, the distortion falls exponentially with the sum-rate, and they characterize the rate of exponential decay. In Sec. 4.4, we use the inner bound found in Sec. 4.3 to find a lower bound on the exponential decay rate with adversaries. In Sec. 4.5, we provide an upper bound on this decay rate.

The second regime in which we study the problem in more detail is the quadratic Gaussian version. Here, all sources are Gaussian and the target distortion function is quadratic. Without adversaries, the complete rate-distortion region was characterized in [55] and [56]. In Sec. 4.6, we use the inner bound from Sec. 4.3 to find an inner bound on the quadratic Gaussian problem with adversaries. In Sec. 4.7, we derive an outer bound on the rate-region of the quadratic Gaussian problem with adversaries. Furthermore, along the lines of the asymptotic results for discrete sources originally proved in [48] and extended to our results in Sec. 4.4 and 4.5, we derive some asymptotic results for the quadratic Gaussian problem. It was originally shown in [53] that for many nodes the minimum achievable distortion fell like K/R for sum-rate R. The exact value for the constant K was found in [54]. In Sec. 4.8, we use our previously derived bounds in Sec. 4.6 and Sec. 4.7 to state and prove bounds on the proportionality constant K for the adversary problem.

4.2 Problem Description

Given block length n and rates R_i for $i = 1, \ldots, L$, the encoding function for agent i is given by

$$f_i : \mathcal{Y}_i^n \to \{1, \ldots, 2^{nR_i}\}. \tag{4.2}$$

The decoding function at the decoder is given by

$$\phi : \prod_{i=1}^{L} \{1, \ldots, 2^{nR_i}\} \to \hat{\mathcal{X}}^n \tag{4.3}$$

where $\hat{\mathcal{X}}$ is the alphabet of the estimate of X, which may differ from $\mathcal{X}$. Denote by C_i the codeword from the set $\{1, \ldots, 2^{nR_i}\}$ sent by node i to the decoder. Honest node choose their transmissions by setting $C_i = f_i(Y_i^n)$. If i is a traitor, then it may select C_i in any manner it chooses, including using information about the honest coding strategy or the true values of the sources. Finally, the decoder produces its estimate of X^n by setting $\hat{X}^n = \phi(C_1, \ldots, C_L)$.

The distortion function is given by

$$d : \mathcal{X} \times \hat{\mathcal{X}} \to \mathbb{R}. \tag{4.4}$$

This function measures the quality of the estimate $\hat{X}^n$ produced at the source; our goal will be to minimize the expected value for a given set of rates. For a given set of source values $(x^n, y_1^n, \ldots, y_L^n)$, we define the maximum possible distortion over all possible actions of the traitors to be

$$D(x^n, y_1^n, \ldots, y_L^n) = \max_{\substack{T \subset \{1,\ldots,L\}: \\ |T|=s}} \max_{C_T} \frac{1}{n} \sum_{t=1}^{n} d(x(t), \hat{x}(t)). \tag{4.5}$$

In this expression T runs over all possible sets of traitors. We also maximize over C_T, the codewords sent by the traitors, ensuring that any potentially traitor actions are considered. Observe that even the choice of which agents to capture

may be a function of the source values. Note also that in (4.5) $\hat{x}^n$ is a function of C^L given by ϕ, and C_H is in turn a function of y_H^n given by the f_i.

We say the rate-distortion vector $(R_1, \ldots, R_L, D)$ is *achievable* if for sufficiently large n and any $\epsilon > 0$ there exists encoding and decoding functions $f_1, \ldots, f_L$ and ϕ as specified in (4.2) and (4.3) such that

$$\mathbb{E}\big[D(X^n, Y_1^n, \ldots, Y_L^n)\big] \leq D + \epsilon. \tag{4.6}$$

Let $D(R_1, \ldots, R_L)$ be the minimum achievable distortion for rates $R_1, \ldots, R_L$.

4.2.1 Error Exponent for Discrete Sources

We now describe the error exponent problem for discrete sources. Assume that the distribution of Y_i given X is uniform for all i. That is, the distribution $p(y_i|x)$ does not depend on i. We may therefore specify the problem in terms of a distribution $p(x, y)$. We assume a Hamming distortion given by

$$d_H(x, \hat{x}) = \begin{cases} 0 & \text{if } x = \hat{x} \\ 1 & \text{if } x \neq \hat{x} \end{cases}. \tag{4.7}$$

For a fixed number of nodes L, sum-rate R, and s traitors, let the minimum achievable distortion be

$$D(R, L, s) = \inf_{R_1, \ldots, R_L : R_1 + \cdots + R_L \leq R} D(R_1, \ldots, R_L). \tag{4.8}$$

Let the minimum achievable distortion at sum-rate R, for any number of nodes, and with the fraction of traitors no more than β, be

$$D(\beta, R) = \inf_{L, s : s \leq \beta L} D(R, L, s). \tag{4.9}$$

Observe that we assume that as the number of nodes L grows, the fraction of traitors s/L remains fixed at $\beta \in [0,1]$. Our goal is to see how the fraction β of traitors affects achievable rates. Finally, our quantity of interest is the error exponent given by

$$E(\beta) = \lim_{R\to\infty} \frac{-\log D(\beta, R)}{R}. \tag{4.10}$$

A lower bound on the error exponent is stated and proved in Sec. 4.4, and an upper bound in Sec. 4.5.

4.2.2 The Quadratic Gaussian Problem

In the quadratic Gaussian version of the problem, X is a Gaussian random variable with zero mean and variance σ_X^2. The sources observed by the nodes are given by

$$Y_i = X + N_i \text{ for } i = 1, \ldots, L \tag{4.11}$$

where N_i is a Gaussian random variable with zero mean and variance $\sigma_{N_i}^2$. The distortion function is quadratic, given by

$$d(x, \hat{x}) = (x - \hat{x})^2. \tag{4.12}$$

An inner bound on the rate-distortion region for this problem is stated and proved in Sec. 4.6, and an outer bound in Sec. 4.7.

In addition, we characterize the asymptotic behavior of the distortion as a function of the sum-rate for many nodes. In particular, the minimum achievable distortion for sum-rate R falls like $K\sigma_X^2/R$, and we are interested in K as a function of β, again the fraction of traitors s/L, which is kept fixed for large L. For formally, let $D(R, L)$ be the minimum achievable distortion for L agents where the sum-rate

is at most R. In the case that all agents have the same quality of observation (i.e. $\sigma^2_{N_i} = \sigma^2_N$ for all i), let $D(R) = \lim_{L\to\infty} D(R, L)$. Finally define

$$K(\beta) = \lim_{R\to\infty} R\frac{D(R)}{\sigma^2_X}. \tag{4.13}$$

That is, $D(R)$ goes like $K\sigma^2_X/R$ for large R. Bounds on $K(\beta)$ are stated and proved in Sec. 4.8.

4.3 Achievability Scheme for Adversarial Attacks

We give an inner bound on the rate-distortion region for a somewhat broader class of problems than the CEO Problem as described in Sec. 4.2. We keep the basic format of the problem, in that the nodes observe Y_i for $i = 1, \ldots, L$ and the decoder is interested in recovering X subject to some distortion constraint, but we relax the condition that the Y_i need by conditionally independent given X. Instead, we allow any distribution among these $L + 1$ variables given by

$$p(xy_1 \cdots y_L). \tag{4.14}$$

The following theorem gives an inner bound on the rate-distortion region for this problem.

Theorem 9 *Let U_i for $i = 1, \ldots, L$ be random variables with alphabets $\mathcal{U}_i$ respectively, jointly distributed with $X, Y_1, \ldots, Y_L$ such that the following Markov chain constraints are satisfied:*

$$U_i - Y_i - (X, Y_1, \ldots, Y_{i-1}, Y_{i+1}, \ldots, Y_L, U_1, \ldots, U_{i-1}, U_{i+1}, \ldots, U_L) \text{ for } i = 1, \ldots, L. \tag{4.15}$$

We may write the distribution of these random variables as

$$\Pr(X = x, Y_1 = y_1, \ldots, Y_L = y_L, U_1 = u_1, \ldots, U_L = u_L) \\ = p(x, y_1 \cdots y_L) \prod_{i=1}^{L} Q(u_i|y_i) \quad (4.16)$$

where $Q(u_i|y_i)$ completely specifies the variable U_i. The tuple $(R_1, \ldots, R_L, D)$ is achievable if there exist $\{U_k\}$ such that:

- *For all $S \subset \{1, \ldots, L\}$ with $|S| = L - 2s$ and all $A \subset S$,*

$$\sum_{i \in A} R_i \geq I(Y_A; U_A | U_{S \setminus A}). \quad (4.17)$$

- *For all distributions $q(u^L)$, there exists a function*

$$f_q : \prod_{i=1}^{L} \mathcal{U}_i \to \hat{\mathcal{X}} \quad (4.18)$$

  such that the following property holds for all pairs of sets $S \in \{1, \ldots, L\}$ with $|S| = L - s$ and conditional distributions $r(u_{S^c}|x, u_S)$: Let

$$r(x, u^L) = \left[\sum_{y_S} p(x, y_S) \prod_{i \in S} Q(u_i|y_i) \right] r(u_{S^c}|x, u_S). \quad (4.19)$$

  If $q(u^L) = r(u^L)$, then

$$D \leq \mathbb{E}_r \left[d(X, f_q(U_1, \ldots, U_L)) \right] \quad (4.20)$$

  where the expectation is take over the distribution $r(x, u^L)$ defined in (4.19).

We offer the following intuition for this result. Node i sends to the decoder a degraded—or quantized—version of its measurement represented by U_i. If all nodes were honest, the joint distribution of (X, U^L) would be given by

$$\sum_{y^L} p(x) \prod_{i=1}^{L} p(y_i|x) Q(u_i|y_i). \quad (4.21)$$

However, due to the presence of the traitors, the joint distribution of (X, U^L) that actually occurs, which is represented by $r(x, u^L)$, may not match the distribution that would result with no traitors. Since the decoder can observe only U^L, it can only recover $q(u^L)$, from which it must choose the estimation function f_q. From q, the decoder can identify sets of nodes S that may be the set of honest agent as the ones satisfying (4.19) for some r. However, there may be several possible sets that are indistinguishable to the decoder, and for each set many possibilities for r, each one representing a particular choice of action by the traitors. The decoder must construct its estimate by choosing a function f_q that satisfies the distortion constraint for each of these possibilities, as (4.20) stipulates.

Fig. 4.2 shows the structure of the achievability strategy. The overall configuration is the same as the standard non-adversarial Berger-Tung strategy, in that Slepian-Wolf coding is used to relay quantized versions of the sources to the destination, after which the destination estimates X from its recovered data. However, several of the blocks need to be changed from the non-adversarial strategy. In fact, for this problem the Slepian-Wolf blocks are almost exactly analogous to the strategies to defeat adversarial attack on the Slepian-Wolf problem studied in Chapter 3.

The following subsections give the proof of Theorem 9.

4.3.1 Coding Strategy

Descriptions of the codebook, and the encoding and decoding rules follow. We assume the existence of random variables U_i for $i = 1, \ldots, L$ and functions f_q for all distribution $q(u^L)$ satisfying the conditions of Theorem 9.

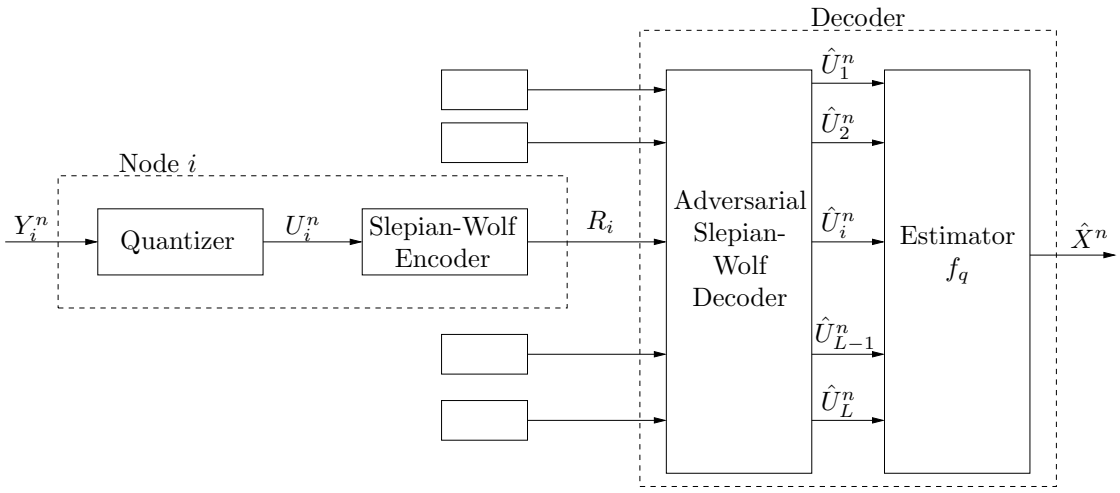


Figure 4.2: Diagram of achievable strategy for CEO-type problems. The strategy, described in detail in Sec. 4.3.1, differs from the standard Berger-Tung strategy mostly in the two blocks in the decoder. The Slepian-Wolf decoding block needs to be aware of the possibility of adversarial manipulations in recovering the U_i, and the estimation function f_q used in the final block depends on the empirical distribution of the recovered U_i.

1) Random Code Structure: Each node i forms its codebook in the following way. It generates $2^{n(I(Y_i;U_i)+\delta)}$ n-length codewords at random from the marginal distribution of U_i. Let $\mathcal{C}_i^{(n)}$ be the codeword set. These codewords are then placed into 2^{nR_i} bins uniformly at random.

2) Encoding Rule: Upon receiving Y_i^n, node i selects uniformly at random an element of

$$\mathcal{C}_i^{(n)} \cap T_\epsilon^{(n)}(U_i|Y_i^n).$$

We denote this selected sequence U_i^n. Node i then sends to the decoder the index of the bin containing U_i^n.

3) Decoding Rule: For each $S \subset \{1,\dots,L\}$ with $|S| = L-s$, the decoder looks for a group of codewords in $T_\epsilon^{(n)}(U_S)$ that matches the received bins from all agents in S. If there is exactly one such a sequence, call it $\hat{U}_i^n[S]$ for all $i \in S$. If there is no such sequence or more than one, define this to be null.

For all i, if there is exactly one non-null value of $\hat{U}_i^n[S]$ among all $S \ni i$, then call this sequence $\hat{U}_i^n$. If the values of $\hat{U}_i^n[S]$ are all null or they are inconsistent, then set $\hat{U}_i^n$ arbitrarily.

Let $t(u^L)$ be the type of $\hat{U}^{nL}$. Let $\mathfrak{V}$ be the collection of sets S for which $\hat{U}_S^n$ is jointly typical. This can be written as

$$\|t(u_S) - p(u_S)\|_\infty \leq \frac{\epsilon}{\prod_{i \in S} |\mathcal{U}_i|} \text{ for all } S \in \mathfrak{V}. \tag{4.22}$$

Let $q(u^L)$ be the distribution minimizing

$$\|t(u^L) - q(u^L)\|_\infty \tag{4.23}$$

subject to

$$q(u_S) = p(u_S) \text{ for all } S \in \mathfrak{V}. \tag{4.24}$$

The decoder chooses for its estimate $\hat{X}^n = f_q(u^L)$, using the function corresponding to this distribution $q(u^L)$.

4.3.2 Error Analysis

Consider the following error events:

1. Node i can find no conditionally typical codewords given the sequence Y_i^n. That is, the set

   $$\mathcal{C}_k^{(n)} \cap T_\epsilon^{(n)}(U_i|Y_i^n) \tag{4.25}$$

   is empty. With high probability, this does not occur by the standard proof of the point-to-point rate-distortion theorem [92].

2. The sequence U_H^n is not jointly typical, where H is the true set of honest agents. That this occurs with low probability follows from the fact that Y_H^n with be jointly typical with high probability, and the Markov Lemma [46].

3. There is a jointly typical codeword $u_H'^n$ different from U_H^n but with $u_k'^n$ in the same bin as U_i^n for all $i \in H$. It is shown in [93] that this occurs with low probability if for all $A \subset H$,

$$\sum_{i \in A} R_i \geq I(U_A; Y_A | U_{H \setminus A}). \tag{4.26}$$

This follows from (4.17) even though the size of H is $L-s$ rather than $L-2s$, by the following argument. We partition A as $A = S_1 \cup \cdots \cup S_B \cup A'$, where the sets S_b satisfy $|S_b| = L - 2s$ for $b = 1, \ldots, B$, and $|A'| \leq L - 2s$. Also let S' be a set with $S' = L - 2s$ and $A' \subset S' \subset H$. We may write

$$\sum_{i \in A} R_i = \sum_{b=1}^{B} \sum_{i \in S_b} R_i + \sum_{i \in A'} R_i \tag{4.27}$$

$$\geq \sum_{b=1}^{B} I(Y_{S_b}; U_{S_b}) + I(Y_{A'}; U_{A'} | U_{S' \setminus A'}) \tag{4.28}$$

$$= \sum_{b=1}^{B} \left[H(U_{S_b}) - H(U_{S_b} | Y_{S_b}) \right] + H(U_{A'} | U_{S' \setminus A'}) - H(U_{A'} | Y_{A'} U_{S' \setminus A'}) \tag{4.29}$$

$$\geq H(U_{S_1} \cdots U_{S_B} | U_{H \setminus A}) + H(U_{A'} | U_{H \setminus A} U_{S_1} \cdots U_{S_B}) - \sum_{b=1}^{B} H(U_{S_b} | Y_{S_b}) - H(U_{A'} | Y_{A'} U_{S' \setminus A'}) \tag{4.30}$$

$$= H(U_A | U_{H \setminus A}) - \sum_{b=1}^{B} H(U_{S_b} | Y_{S_b}) - H(U_{A'} | Y_{A'} U_{S' \setminus A'}) \tag{4.31}$$

$$= H(U_A | U_{H \setminus A}) - H(U_A | Y_A U_{H \setminus A}) \tag{4.32}$$

$$= I(Y_A; U_A | U_{H \setminus A}) \tag{4.33}$$

where (4.28) follows from several applications of (4.17), (4.30) follows because

conditioning reduces entropy, (4.31) follows from the chain rule, and (4.32) follows because $U_A - Y_A - U_{H\setminus A}$ is a Markov chain.

4. For some $S \neq H$ and $i \in H \cap S$, $\hat{U}_i^n[S] \neq U_i^n$. This can only occur if there is a jointly typical sequence that matches the bins sent by nodes in $H \cap S$ other than the true value of $U_{H\cap S}^n$. Note that $|H \cap S| \geq L - 2s$, so by (4.17) and the argument in (4.27)–(4.33), for all $A \subset H \cap S$, we have

$$\sum_{i\in A} R_i \geq I(U_A; Y_A | U_{H\cap S\setminus A}). \tag{4.34}$$

Therefore, again by the argument in [93], with high probability the only jointly typical sequence in the bins sent from nodes in $H \cap S$ will be the true value of $U_{H\cap S}^n$, so this error event does not occur.

4.3.3 Distortion Analysis

We have shown that error events (1)–(4) as described in Sec. 4.3.2 occur with small probability. Let us assume they do not occur. Hence for all $i \in H$, $\hat{U}_k^n = U_k^n$. Since U_H^n is jointly typical, $H \in \mathfrak{V}$. For all $S \in \mathfrak{V}$, we have that $\|t(u_S) - p(u_S)\|_\infty \leq \frac{\epsilon}{\prod_{i\in S} \mathcal{U}_i}$. Certainly if $\epsilon = 0$, then this implies $t(u_S) = p(u_S)$. Moreover, if $\epsilon = 0$ then the solution of the optimization problem in (4.23)–(4.24) would yield $q(u^L) = t(u^L)$. By continuity, when ϵ is nonzero, there must be some constant C for which, for sufficiently small ϵ,

$$\|q(u^L) - t(u^L)\|_\infty \leq C\epsilon. \tag{4.35}$$

Moreover, by (4.24), $q(u_H) = p(u_H)$.

Let $t(x, u^L)$ be the joint type of $(X^n, \hat{U}^{nL})$. The average distortion is given by

$$\frac{1}{n}\sum_{t=1}^{n} d(x(t), f_q(\hat{u}_1(t), \ldots, \hat{u}_L(t))) = \sum_{x,u^L} t(x, u^L) d(x, f_q(u_1, \ldots, u_L)). \tag{4.36}$$

Let

$$r(x, u^L) = q(u^L)t(x|u^L). \tag{4.37}$$

Because $q(u^L)$ and $t(u^L)$ are close as given by (4.35), we can write

$$\|r(x, u^L) - t(x, u^L)\|_\infty = \|(q(u^L) - t(u^L))t(x|u^L)\|_\infty \leq \|q(u^L) - t(u^L)\|_\infty \leq C\epsilon. \tag{4.38}$$

Therefore the average distortion is upper bounded by

$$\sum_{x,u^L} (r(x, u^L) + C\epsilon)d(x, f_q(u_1, \ldots, u_L)) \tag{4.39}$$

$$\leq \sum_{x,u^L} r(x, u^L)d(x, f_q(u_1, \ldots, u_L)) + C\epsilon \max_{x,\hat{x}} d(x, \hat{x}) \tag{4.40}$$

$$\leq D + C\epsilon \max_{x,\hat{x}} d(x, \hat{x}) \tag{4.41}$$

where (4.41) follows from (4.20), which we may apply because $r(x, u^L)$ satisfies (4.19) with $S = H$, since $r(u_S) = q(u_S) = p(u_S)$ and $r(u^L) = q(u^L)$. The theorem follows by sending $\epsilon \to 0$.

4.4 Inner Bound on Error Exponent for Discrete Sources

We use Theorem 9 to prove a lower bound on the error exponent for discrete sources. Recall that in this problem the distribution of Y_i given X is identical for all i. We therefore describe our results in terms of the distribution of X and one Y_i, given by $p(x, y)$. We introduce two auxiliary random variables U and J. The variable J takes values in $\mathcal{J}$ and is independent of (X, Y) with marginal distribution $P_J(j)$; $X \to (Y, J) \to U$ is a Markov chain. The conditional distribution of U is given by $Q(u|y, j)$, and we define for convenience

$$\tilde{Q}(u|x, j) = \sum_y p(y|x)Q(u|y, j). \tag{4.42}$$

We also introduce the vector γ_j for all $j \in \mathcal{J}$. Let

$$F(P_J, Q, \gamma) = \frac{\min_{x_1,x_2 \in \mathcal{X}} \sum_j \gamma_j D(\tilde{Q}_{\lambda,j} \| \tilde{Q}(u|x_1, j))}{I(Y; U|X, J)} \tag{4.43}$$

where

$$\tilde{Q}_{\lambda,j} = \frac{\tilde{Q}^{1-\lambda}(u|x_1, j)\tilde{Q}^{\lambda}(u|x_2, j)}{\sum_u \tilde{Q}^{1-\lambda}(u|x_1, j)\tilde{Q}^{\lambda}(u|x_2, j)} \tag{4.44}$$

and λ is chosen so that

$$\sum_j \gamma_j D(\tilde{Q}_{\lambda,j} \| \tilde{Q}(u|x_1, j)) = \sum_j \gamma_j D(\tilde{Q}_{\lambda,j} \| \tilde{Q}(u|x_2, j)). \tag{4.45}$$

It was shown in [48] that the error exponent without an adversary is given by

$$E(0) = \max_{P_J, Q} F(P_J, Q, P_J). \tag{4.46}$$

The following theorem, our lower bound, recovers this quantity as a lower bound at $\beta = 0$.

Theorem 10 *For a fraction β of traitors, the error exponent is lower bounded by*

$$E(\beta) \geq \max_{P_J, Q} \min_{\gamma} F(P_J, Q, \gamma) \tag{4.47}$$

where we impose the constraints that

$$\sum_j \gamma_j \geq 1 - 2\beta \quad \textit{and} \quad \gamma_j \leq P_J(j) \textit{ for all } j \in \mathcal{J}. \tag{4.48}$$

To prove Theorem 10, we follow the path of [48] by presenting the bound in two steps, the second a generalization of the first. In Sec. 4.4.1 we state a prove a lemma that constitutes our loose bound. Then in Sec. 4.4.2, we tighten this bound to complete the proof of Theorem 10.

4.4.1 Preliminary Bound

Lemma 9 *Let U be a random variable such that $X - Y - U$ is a Markov Chain and the distribution of U is given by $Q(u|y)$. Let*

$$\tilde{Q}(u|x) = \sum_y p(y|x)Q(u|y).$$

The error exponent is lower bounded by

$$E(\beta) \geq \max_Q \frac{\min_{x_1,x_2}(1-2\beta)D(\tilde{Q}_\lambda \| \tilde{Q}(u|x_1))}{I(Y;U|X)}$$

where

$$\tilde{Q}_\lambda(u) = \frac{\tilde{Q}(u|x_1)^{1-\lambda}\tilde{Q}(u|x_2)^\lambda}{\sum_u \tilde{Q}(u|x_1)^{1-\lambda}\tilde{Q}(u|x_2)^\lambda} \tag{4.49}$$

and λ is such that $D(\tilde{Q}_\lambda \| \tilde{Q}(u|x_1)) = D(\tilde{Q}_\lambda \| \tilde{Q}(u|x_2))$.

Proof: We prove the lemma by applying Theorem 9. To do so, we must specify the auxiliary random variables U_i as well as the function f_q as a function of $q(u^L)$. For each i, U_i has distribution conditioned on Y_i given by $Q(u|y)$. We construct f_q as follows. Given $q(u^L)$, select any set S with $|S| = L - s$ and $q(u_S) = p(u_S)$. Let

$$f_q(u^L) = \max_x p(x|u_S). \tag{4.50}$$

Set $R_i = I(Y;U|X) + \epsilon$ for all i. Note that the sum-rate is given by

$$R = L\ I(Y;U|X) + L\epsilon. \tag{4.51}$$

We now show that (4.17) is satisfied for sufficiently large L. For any S with $|S| = L - 2s$ and $A \subset S$, we may write

$$I(Y_A; U_A|U_{S\setminus A}) = H(U_A|U_{S\setminus A}) - H(U_A|Y_A U_{S\setminus A}) \tag{4.52}$$

$$\leq H(U_A|U_{S\setminus A}X) + H(X) - H(U_A|Y_A U_{S\setminus A})) \tag{4.53}$$

$$= H(U_A|X) - H(U_A|Y_A) + H(X) \tag{4.54}$$

$$= \sum_{i\in A} \left[H(U_i|X) - H(U_i|Y_i)\right] + H(X) \tag{4.55}$$

$$= \sum_{i\in A} I(Y_i; U_i|X) + H(X) \tag{4.56}$$

$$= |A| I(Y; U|X) + H(X) \tag{4.57}$$

$$= \sum_{i\in A} R_i + H(X) - |A|\epsilon \tag{4.58}$$

where (4.54) follows because $U_A - X - U_{S\setminus A}$ and $U_A - Y_A - U_{S\setminus A}$ are Markov chains, (4.55) follows because U_i does not depend on U_j or Y_j for $j \neq i$ after conditioning on Y_i or X, and (4.57) because all the U_i are distributed identically. Note that (4.58) satisfies (4.17) if $|A| \geq H(X)/\epsilon$. If $|A| < H(X)/\epsilon$, then $S \setminus A$ grows with L because $s = \beta L$ so $|S| = (1 - 2\beta)L$; thus the conditioning term causes $I(Y_A; U_A|U_{S\setminus A})$ to shrink, and (4.17) is sure to be satisfied for sufficiently large L.

We now need to evaluate the right hand side of (4.20) to find the achieved distortion. For any $r(x, u^L)$, let

$$r(x, \tilde{x}, \hat{x}, u^L) = r(x, u^L) p(\tilde{x}|u_{H\cap S}) p(\hat{x}|u_S). \tag{4.59}$$

The variables $\tilde{X}$ and $\hat{X}$ defined in this distribution are defined formally and have no counterpart in the operation of the code. However, note that we may upper

bound the achieved distortion by

$$D \leq \mathbb{E}_r\big[d_H(X, f_q(U_1, \ldots, U_L))\big] \tag{4.60}$$

$$\leq \mathbb{E}_r\big[d_H(X, \hat{X})\big] \tag{4.61}$$

$$\leq \mathbb{E}_r\big[d_H(X, \tilde{X})\big] + \mathbb{E}_r\big[d_H(\tilde{X}, \hat{X})\big] \tag{4.62}$$

where (4.61) follows because the true function f_q chooses the most likely value of X given U_S, whereas $\hat{X}$ is defined to be a randomly chosen value according to the a posterior probability, which will certainly be a worse estimate; and (4.62) follows by the triangle inequality. We proceed to evaluate the two terms in (4.62). The first term depends only on the distribution of X and $\tilde{X}$, which we may write

$$r(x, \tilde{x}) = \sum_{u^L} r(x, u^L) p(\tilde{x}|u_{H\cap S}) = \sum_{u_{H\cap S}} p(x, u_{H\cap S}) p(\tilde{x}|u_{H\cap S}) \tag{4.63}$$

because $r(x, u_H) = p(x, u_H)$. The second term in (4.62) depends only on the distribution of $\tilde{X}$ and $\hat{X}$, which we may write

$$r(\tilde{x}, \hat{x}) = \sum_{u^L} r(u^L) p(\tilde{x}|u_{H\cap S}) p(\hat{x}|u_S) \tag{4.64}$$

$$= \sum_{u_S} p(u_S) p(\tilde{x}|u_{H\cap S}) p(\hat{x}|u_S) \tag{4.65}$$

$$= \sum_{u_S} p(\hat{x}, u_S) p(\tilde{x}|u_{H\cap S}) \tag{4.66}$$

$$= \sum_{u_{H\cap S}} p(\hat{x}, u_{H\cap S}) p(\tilde{x}|u_{H\cap S}) \tag{4.67}$$

where we have used the fact that $r(u_S) = p(u_S)$. Note that the distribution of $(X, \tilde{X})$ is identical to that of $(\tilde{X}, \hat{X})$. Hence the two terms of (4.62) are the same,

and we need only bound one of them. We may therefore write

$$D/2 \leq \mathbb{E}_r\big[d_H(X,\tilde{X})\big] = \Pr_r(X \neq \tilde{X}) \tag{4.68}$$

$$= \sum_{x_1,x_2\in\mathcal{X}:x_1\neq x_2} p(x_1,u_{H\cap S})p(x_2|u_{H\cap S}) \tag{4.69}$$

$$= \sum_{x_1,x_2\in\mathcal{X}:x_1\neq x_2} \frac{p(x_1)\tilde{Q}(u_{H\cap S}|x_1)p(x_2)\tilde{Q}(u_{H\cap S}|x_2)}{p(u_{H\cap S})}. \tag{4.70}$$

Let $\gamma = |S\cap H|/L$. Certainly $\gamma \geq 1-2\beta$. Let t be the type of $u_{S\cap H}$. This is a type in space, rather than in time, and it is well defined because the alphabets for U_i is the same for each i. For $x_1 \in \mathcal{X}$,

$$\frac{p(x_1)\tilde{Q}(u_{S\cap H}|x_1)}{p(u_{S\cap H})} = \frac{p(x_1)2^{-\gamma L[D(t\|\tilde{Q}(u|x_1))+H(t)]}}{\sum_x p(x)2^{-\gamma L[D(t\|\tilde{Q}(u|x))+H(t)]}}$$

$$\leq 2^{-\gamma L[D(t\|\tilde{Q}(u|x_1))-\min_x D(t\|\tilde{Q}(u|x))-\delta]}$$

for any $\delta > 0$ and sufficiently large L. Therefore

$$\sum_{u_{S\cap H}\in\Lambda_t^{\gamma L}(U)} \frac{p(x_1)\tilde{Q}(u_{S\cap H}|x_1)p(x_2)\tilde{Q}(u_{S\cap H}|x_2)}{\Pr(u_{S\cap H})}$$

$$\leq 2^{\gamma LH(t)}2^{-\gamma L[D(t\|\tilde{Q}(u|x_1))-\min_x D(t\|\tilde{Q}(u|x))-\delta]}$$

$$\cdot\, p(x_2)2^{-\gamma L[D(t\|\tilde{Q}(u|x_2))+H(t)]}$$

$$\leq 2^{-\gamma L[D(t\|\tilde{Q}(u|x_1))+D(t\|\tilde{Q}(u|x_2))-\min_x D(t\|\tilde{Q}(u|x))-\delta]}$$

for sufficiently large L. Therefore, using the fact that the number of types t is polynomial in L,

$$D/2 \geq \min_{x_1,x_2:x_1\neq x_2} \min_t \gamma\big[D(t\|\tilde{Q}(u|x_1)) + D(t\|\tilde{Q}(u|x_2)) - \min_x D(t\|\tilde{Q}(u|x)) - \delta\big]$$

$$= \min_t \min_x 2\,\gamma D(t\|\tilde{Q}(u|x)) - \delta \tag{4.71}$$

where min2 takes the second smallest value. It can be shown that this term involving the second smallest value of x is the minimum Chernoff Information. That is,

$$\frac{-\log D}{L} \geq \min_{x_1,x_2} \gamma D(\tilde{Q}_\lambda \| \tilde{Q}(u|x_1)) - \delta - \frac{\log 2}{L}$$

where $\tilde{Q}_\lambda$ and λ are defined by (4.49). Recalling that $\gamma \geq 1 - 2\beta$ and taking the limit as $\delta \to 0$ gives

$$\lim_{L\to\infty} \frac{-\log D}{L} \geq \min_{x_1,x_2} (1-2\beta) D(\tilde{Q}_\lambda \| \tilde{Q}(u|x_1)).$$

Applying (4.51) proves Lemma 9.

□

4.4.2 Tighter Bound

Now we improve this bound by introducing the additional auxiliary random variable J. Following the essential argument of [48], we alter our application of Theorem 9 so that the nodes are split into groups, each with a different method of quantization. Partition $\{1, \cdots, L\}$ into disjoint sets R_j such that $\big||R_j| - P_J(j)L\big| \leq 1$ for all j. For all $i \in R_j$, the conditional distribution of U_i given Y_i is given by $Q(u|y, J = j)$. If $i \in R_j$, we set $R_i = I(Y; U|X, J = j)$. Checking (4.17) follows along similar lines as it did in Sec. 4.4.1. The sum-rate becomes

$$R = \sum_j |R_j| I(Y; U|X, J = j) \leq L\ I(Y; U|X, J) + \mathcal{O}(1). \tag{4.72}$$

The definition of f_q is the same as in Sec. 4.4.1, accounting for the different distribution of the underlying variables. Let $\gamma_j = |R_j \cap S \cap H|/L$. Then

$$\sum_j \gamma_j \geq 1 - 2\beta \text{ and } \gamma_j \leq P_J(j)\ \forall j \in \mathcal{J}. \tag{4.73}$$

Let t_j be the type of $u_{R_j \cap S \cap H}$. Thus

$$\tilde{Q}(u_{S\cap H}|x) = \prod_j 2^{-L\gamma_j[D(t_j\|\tilde{Q}(u|x,j))+H(t_j)]}. \tag{4.74}$$

Applying this to (4.70) yields

$$\begin{aligned}\frac{-\log D}{L} &\geq \min_{t_j} \min_x 2 \sum_{j\in\mathcal{J}} \gamma_j D(t_j\|\tilde{Q}(u|x,j)) - \delta - \frac{\log 2}{L} \\ &\geq \min_{x_1,x_2} \sum_j \gamma_j D(\tilde{Q}_{\lambda,j}\|\tilde{Q}(u|x_1,j)) - \delta - \frac{\log 2}{L} \end{aligned} \tag{4.75}$$

where $\tilde{Q}_{\lambda,j}$ is given by (4.44) and (4.45). Extending (4.75) to minimize over all γ_j satisfying (4.73), then combining the result with (4.72) completes the proof of Theorem 10.

4.5 Outer Bound on Error Exponent for Discrete Sources

Recall the definition of $F(P_J, Q, \gamma)$ in Sec. 4.4, as we use it again in the statement of our upper bound on the error exponent, which is stated as follows.

Theorem 11 *For a β fraction of traitors, the error exponent is upper bound as*

$$E(\beta) \leq \min_{\gamma} \max_{P_J,Q} F(P_J, Q, \gamma) \tag{4.76}$$

where γ and P_J are constrained so that

$$\sum_j \gamma_j \geq 1 - 2\beta \quad \textit{and} \quad \gamma_j \leq P_J(j) \textit{ for all } j \in \mathcal{J}. \tag{4.77}$$

Note that the upper bound in Theorem 11 differs from the lower bound in Theorem 10 only by a reordering of the maximum and minimum. Moreover, the two bounds meet at $\beta = 0$ and together recover the result of [48], giving the error exponent with no adversary, as stated in (4.46). The proof of Theorem 11 follows.

Proof: Recall that if node i is honest, the codeword C_i transmitted to the decoder is given by $f_i(Y_i^n)$. Define a distribution on X^n and C^L as

$$P(x^n, c^L) = \sum_{y^{nL}} p(x^n) \prod_{i=1}^{L} p(y_i^n|x^n)\mathbf{1}(c_i = f_i(y_i^n)).$$

We will refer to various marginals and conditionals of this distribution as well.

Let $\tilde{X}_t = (X(1), \ldots, X(t-1), X(t+1), \ldots, X(n))$. For any t and $\tilde{x}_t$, define $U_i(t, \tilde{x}_t)$ to be a random variable distributed with $X(t)$ and $Y_i(t)$ such that

$$\Pr(X(t) = x, Y_i(t) = y, U_i(t, \tilde{x}_t) = c) = p(x, y) \Pr(C_i = c|Y_i(t) = y, \tilde{X}_t = \tilde{x}_t).$$

Note that $X(t) - Y(t) - U_i(t, \tilde{x}_t)$ is a Markov chain.

Suppose the adversary performs the following attack. It selects a set $S \subset \{1, \ldots, L\}$ with $|S| = (1-\beta)L$ and $|H \cap S| = (1-2\beta)L$, where H is the true set of honest nodes; i.e. H^c are the traitors. The set S is the traitors' target set, that they endeavor to fool the decoder into thinking may be the true set of honest nodes. They generate a sequence X'^n from the distribution $P(x^n|c_{H\cap S})$. Finally, they construct $C_{S\setminus H}$ just as honest nodes would if X'^n were the truth. That is, from X'^n, they generate $C_{S\setminus H}$ from the distribution $P(c_{S\setminus H}|x^n)$, and transmit this $C_{S\setminus H}$ to the decoder.

Observe that X^n, X'^n, C^L will be distributed according to

$$P(x^n, c_H)P(x'^n|c_{H\cap S})P(c_{S\setminus H}|x'^n) = \frac{P(x^n, c_H)P(x'^n, c_S)}{P(c_{H\cap S})}.$$

This distribution is symmetric in x^n and x'^n. In particular, if S were the true set of honest nodes, and the traitors performed an analogous attack selecting the set H as their target set, then precisely the same distribution among X^n, X'^n, C^L would result, except that X^n and X'^n would switch roles. Hence, if the decoder achieves

a distortion of D; that is, if $\hat{X}^n$ is such that $D \geq \frac{1}{n} d_H(X^n, \hat{X}^n)$, then it must also be that $D \geq \frac{1}{n} d_H(X'^n, \hat{X}^n)$, because the decoder can only generate one estimate, but it must work in both situations. Therefore

$$\begin{aligned} D &\geq \frac{1}{2n}[d_H(X^n, \hat{X}^n) + d_H(X'^n, \hat{X}^n)] \\ &\geq \frac{1}{2n} d_H(X^n, X'^n) \qquad (4.78) \\ &= \frac{1}{2n} \sum_{t=1}^{n} \Pr(X(t) \neq X'(t)) \\ &= \frac{1}{2n} \sum_{t=1}^{n} \sum_{x(t) \neq x'(t), c^L} \frac{P(x(t), c_H) P(x'(t), c_S)}{P(c_{H \cap S})} \\ &= \frac{1}{2n} \sum_{t=1}^{n} \underbrace{\sum_{x(t) \neq x'(t), c_{H \cap S}} \frac{P(x(t), c_{H \cap S}) P(x'(t), c_{H \cap S})}{P(c_{H \cap S})}}_{D(t)} \qquad (4.79) \end{aligned}$$

where we used the triangle inequality in (4.78). The expression in (4.79) can be shown to be concave in P. We may write

$$\begin{aligned} P(x(t), c_{H \cap S}) &= \sum_{\tilde{x}_t, y^n_{H \cap S}} p(x^n) \prod_{i \in H \cap S} p(y_i^n | x^n) \mathbf{1}(c_i = f_i(y_i^n)) \\ &= \sum_{x(t^c)} p(x^n) \prod_{i \in H \cap S} \sum_{y_i(t)} p(y_i(t) | x(t)) \sum_{\tilde{y}_{i,t}} p(\tilde{y}_{i,t} | \tilde{x}_t) \mathbf{1}(c_i = f_i(y_i^n)) \\ &= \sum_{\tilde{x}_t} p(x^n) \prod_{i \in H \cap S} \sum_{y} p(y | x(t)) \Pr(C_i = c_i | \tilde{X}_t = \tilde{x}_t, Y_i(t) = y) \\ &= \mathbb{E}_{\tilde{X}_t} p(x(t)) \prod_{i \in H \cap S} \sum_{y} p(y | x(t)) \Pr(U_i(t, \tilde{X}_t) = c_i | Y_i(t) = y) \\ &= \mathbb{E}_{\tilde{X}_t} p(x(t)) \prod_{i \in H \cap S} \Pr(U_i(t, \tilde{X}_t) = c_i | X(t) = x(t)). \qquad (4.80) \end{aligned}$$

Define for convenience

$$P(x, u_{H \cap S} | t, \tilde{X}_t) = p(x) \prod_{i \in H \cap S} \Pr(U_i(t, \tilde{X}_t) = u_i | X(t) = x).$$

Substituting (4.80) and (4.5) into (4.79) and using concavity gives

$$D(t) \geq \mathbb{E}_{\tilde{X}_t} \sum_{\substack{x_1 \neq x_2 \\ u_{H\cap S}}} \frac{P(x_1, u_{H\cap S}|t, \tilde{X}_t) P(x_2, u_{H\cap S}|t, \tilde{X}_t)}{\sum_{x_3} P(x_3, u_{H\cap S}|t, \tilde{X}_t)}$$

$$\geq |X|^{-1} \mathbb{E}_{\tilde{X}_t} \max_{x_1 \neq x_2} \sum_{u_{H\cap S}} \frac{P(x_1, u_{H\cap S}|t, \tilde{X}_t) P(x_2, u_{H\cap S}|t, \tilde{X}_t)}{\max_{x_3} P(x_3, u_{H\cap S}|t, \tilde{X}_t)}$$

Let

$$\mathcal{U}_x = \left\{ u_{H\cap S} : x = \operatorname*{argmax}_{x'} p(x') \prod_{i \in H\cap S} \tilde{Q}(u_i(t, \tilde{X}_t)|x') \right\}.$$

Then

$$D(t) \geq |X|^{-1} \mathbb{E}_{\tilde{X}_t} \max_{x_1 \neq x_2} \sum_{x_3} \sum_{u_{H\cap S} \in \mathcal{U}_{x_3}} \frac{P(x_1, u_{H\cap S}|t, \tilde{X}_t) P(x_2, u_{H\cap S}|t, \tilde{X}_t)}{P(x_3, u_{H\cap S}|t, \tilde{X}_t)}$$

$$\geq |X|^{-1} \mathbb{E}_{\tilde{X}_t} \max_{x_1 \neq x_2, x_3} \sum_{u_{H\cap S} \in \mathcal{U}_{x_3}} \frac{P(x_1, u_{H\cap S}|t, \tilde{X}_t) P(x_2, u_{H\cap S}|t, \tilde{X}_t)}{P(x_3, u_{H\cap S}|t, \tilde{X}_t)}. \tag{4.81}$$

For fixed x_3, if both x_1 and x_2 are different from x_3, we can always increase the value in (4.81) by making x_1 or x_2 equal to x_3. Hence, we need only consider cases in which either $x_1 = x_3$ or $x_2 = x_3$. Thus

$$D(t) \geq |X|^{-1} \mathbb{E}_{\tilde{X}_t} \max_{x_1 \neq x_2} \sum_{u_{H\cap S} \in \mathcal{U}_{x_2}} P(x_1, u_{H\cap S}|t, \tilde{X}_t)$$

$$= |X|^{-1} \mathbb{E}_{\tilde{X}_t} \max_{x_1 \neq x_2} p(x_1) \Pr(\mathcal{U}_{x_2}|x_1, \tilde{X}_t).$$

Using ideas from [48], we have that

$$\Pr(\mathcal{U}_{x_2}|x_1, \tilde{X}_t) \geq 2^{-\sum_{i \in H\cap S} D(Q_\lambda^{(i)} \| \Pr(U_i(t, \tilde{X}_t)|x_1)) - o(L)}$$

where

$$Q_\lambda^{(i)}(u) = \frac{\Pr^{1-\lambda}(U_i(t, \tilde{X}_t) = u|x_1) \Pr^{\lambda}(U_i(t, \tilde{X}_t) = u|x_2)}{\Delta_\lambda^{(i)}} \tag{4.82}$$

with $\Delta_\lambda^{(i)}$ a normalizing constant and λ chosen such that

$$\sum_{i \in H\cap S} D(Q_\lambda^{(i)} \| \Pr(U_i(t, \tilde{X}_t)|x_1)) = \sum_{i \in H\cap S} D(Q_\lambda^{(i)} \| \Pr(U_i(t, \tilde{X}_t)|x_2)). \tag{4.83}$$

Hence

$$D(t) \geq \mathbb{E}_{\tilde{X}_t} 2^{-\min_{x_1,x_2} \sum_{i \in H \cap S} D(Q_\lambda^{(i)} \| \Pr(U_i(t,\tilde{X}_t)|x_1)) - o(L)}. \tag{4.84}$$

Putting (4.84) back into (4.79) gives

$$\begin{aligned} -\log D &\leq -\log \frac{1}{2n} \sum_{t=1}^{n} \mathbb{E}_{\tilde{X}_t} 2^{-\min_{x_1,x_2} \sum_{i \in H \cap S} D(Q_\lambda^{(i)} \| \Pr(U_i(t,\tilde{X}_t)|x_1)) - o(L)} \\ &\leq \frac{1}{n} \sum_{t=1}^{n} \mathbb{E}_{\tilde{X}_t} \min_{x_1,x_2} \sum_{i \in H \cap S} D(Q_\lambda^{(i)} \| \Pr(U_i(t,\tilde{X}_t)|x_1)) + o(L) \end{aligned} \tag{4.85}$$

where we have used Jensen's inequality in (4.85).

A chain of standard inequalities (see [48]) yields

$$R = \sum_{i=1}^{L} R_i \geq \frac{1}{n} \sum_{t=1}^{n} \mathbb{E}_{\tilde{X}_t} \sum_{i=1}^{L} I(Y_i(t); U_i(t, \tilde{X}_t | X(t)). \tag{4.86}$$

Putting (4.85) together with (4.86) and using the fact that

$$\frac{\sum_i A_i}{\sum_i B_i} \leq \max_i \frac{A_i}{B_i}$$

for any nonnegative A_i and B_i, we get

$$\begin{aligned} \frac{-\log D}{R} &\leq \max_{t,\tilde{x}_t} \frac{\min_{x_1,x_2} \sum_{i \in H \cap S} D(Q_\lambda^{(i)} \| \Pr(U_i(t,\tilde{x}_t)|x_1)) + o(L)}{\sum_{i=1}^{L} I(Y_i(t); U_i(t,\tilde{x}_t)|X(t))} \\ &\leq \max_{U_i : X \to Y_i \to U_i} \frac{\min_{x_1,x_2} \frac{1}{L} \sum_{i \in H \cap S} D(Q_\lambda^{(i)} \| \tilde{Q}(u_i|x_1))}{\frac{1}{L} \sum_{i=1}^{L} I(Y_i; U_i|X)} + \epsilon. \end{aligned} \tag{4.87}$$

Observing that the choices of H and S could have been made differently by the traitors, we introduce a vector γ_i for $i = 1, \ldots, L$ under the constraints

$$\gamma_i \in \left\{0, \frac{1}{L}\right\} \quad \text{and} \quad \sum_i \gamma_i = 1 - 2\beta. \tag{4.88}$$

This allows us to tighten (4.87) to

$$\frac{-\log D}{R} \leq \min_{\gamma_i} \max_{U_i : X \to Y_i \to U_i} \frac{\min_{x_1,x_2} \sum_{i=1}^{L} \gamma_i D(Q_\lambda^{(i)} \| \tilde{Q}(u_i|x_1))}{\frac{1}{L}\sum_{i=1}^{L} I(Y_i; U_i|X)} + \epsilon. \tag{4.89}$$

we claim that the value of (4.89) does not change if we replace (4.88) with

$$\gamma_i \leq \frac{1}{L} \quad \text{and} \quad \sum_i \gamma_i \geq 1 - 2\beta. \tag{4.90}$$

This is because we may use arbitrarily large L, so any γ_i satisfying (4.88) can be closely approximated by a γ_i satisfying (4.90). Furthermore, we introduce a variable I with values in $\{1, \ldots, L\}$ such that

$$\Pr(U = u | I = i, Y = y) = \Pr(U_i = u | Y = y)$$

and maintaining the condition $\gamma_i \leq P_I(i)$ for all $i = 1, \ldots, L$. Doing so gives

$$\begin{aligned} \frac{-\log D}{R} &\leq \min_{\gamma_i} \max_{P_I, Q} \frac{\min_{x_1,x_2} \sum_i \gamma_i D(\tilde{Q}_{\lambda,i} \| \tilde{Q}(u|x_1, i))}{I(Y; U|X, I)} \\ &= \min_{\gamma_i} \max_{P_I, Q} F(P_I, Q, \gamma). \end{aligned}$$

Replacing I with a variable J over an arbitrary alphabet proves (4.76). Note that in this process (4.82), (4.83), and (4.90) have become (4.44), (4.45), and (4.77) respectively. □

4.6 Inner Bound on Rate-Distortion Region for the Quadratic Gaussian Problem

With no adversary, the rate-distortion region for the quadratic Gaussion problem was found simultaneously in [55] and [56]. They found that with $s = 0$, the tuple $(R_1, \ldots, R_L, D)$ is achievable if and only if there exist r_i for $i = 1, \ldots, L$ such that

1. for all $A \subset \{1, \ldots, L\}$,

$$\sum_{i \in A} R_i \geq \sum_{i \in A} r_i + \frac{1}{2} \log \frac{1}{D} - \frac{1}{2} \log \left(\frac{1}{\sigma_X^2} + \sum_{i \in A^c} \frac{1 - \exp(-2r_i)}{\sigma_{N_i}^2} \right). \quad (4.91)$$

2. the distortion D is bounded by

$$\frac{1}{D} \leq \frac{1}{\sigma_X^2} + \sum_{i=1}^{L} \frac{1 - \exp(-2r_i)}{\sigma_{N_i}^2}. \quad (4.92)$$

The following theorem gives our inner bound on the rate-distortion region with an adversary.

Theorem 12 *The tuple $(R_1, \ldots, R_L, D)$ is achievable if there exist r_i for $i = 1, \ldots, L$ and for each matrix $\Sigma \in \mathbb{R}^{L \times L}$ there exist constants $c_i(\Sigma)$ such that*

1. *for all $S \subset \{1, \ldots, L\}$ with $|S| = L - 2s$ and all $A \subset S$,*

$$\sum_{i \in A} R_i \geq \sum_{i \in A} r_i + \frac{1}{2} \log \left(\frac{1}{\sigma_X^2} + \sum_{i \in S} \frac{1 - \exp(-2r_i)}{\sigma_{N_i}^2} \right) - \frac{1}{2} \log \left(\frac{1}{\sigma_X^2} + \sum_{i \in S \setminus A} \frac{1 - \exp(-2r_i)}{\sigma_{N_i}^2} \right) \quad (4.93)$$

2. *for every $S \subset \{1, \ldots, L\}$ with $|S| = L - s$ and every vector $\lambda \in \mathbb{R}^L$ for which*

$$\Sigma_{i,j} = \sigma_X^2 + \frac{\sigma_{N_i}^2}{1 - \exp(-2r_i)} \delta_{i,j} \text{ for all } i, j \in S \quad (4.94)$$

and $\lambda_i = \sigma_X^2$ for $i \in H$,

$$D \geq \mathbb{E}_{\Sigma, \lambda} \left(X - \sum_{i=1}^{L} c_i(\Sigma) U_i \right)^2 \quad (4.95)$$

where by $\mathbb{E}_{\Sigma,\lambda}$ we mean an expectation taken over a distribution on the variables $(X, U_1, \ldots, U_L)$ with covariance matrix

$$\begin{pmatrix} \sigma_X^2 & \lambda^T \\ \lambda & \Sigma \end{pmatrix}. \quad (4.96)$$

Proof: Again we apply Theorem 9. We define U_i as

$$U_i = Y_i + W_i \tag{4.97}$$

where W_i is a Gaussian random variable with zero mean and variance $\sigma^2_{W_i}$. The estimation function f_q is determined by the sample covariance matrix of $q(u^L)$, which we denote Σ. Then let

$$f_q(u^L) = \sum_{i=1}^{L} c_i(\Sigma) u_i. \tag{4.98}$$

Consider first the rate condition in the statement of Theorem 12. Define (just for the section)

$$r_i = I(Y_i; U_i|X) = \frac{1}{2} \log \frac{\sigma^2_{N_i} + \sigma^2_{W_i}}{\sigma^2_{W_i}}.$$

There is a one-to-one correspondence between r_i and $\sigma^2_{W_i}$, so we can write everything in terms of r_i instead of $\sigma^2_{W_i}$. It is not hard to show that

$$I(Y_A; U_A|U_{S\setminus A}) = \sum_{k\in A} r_k + \frac{1}{2} \log \left(\frac{1}{\sigma^2_X} + \sum_{i\in S} \frac{1-\exp(-2r_i)}{\sigma^2_{N_i}} \right) - \frac{1}{2} \log \left(\frac{1}{\sigma^2_X} + \sum_{i\in S\setminus A} \frac{1-\exp(-2r_i)}{\sigma^2_{N_i}} \right).$$

Hence (4.93) follows from (4.17).

Now consider the distortion condition in the statement of Theorem 12. Any distribution $r(x, u^L)$ has a covariance matrix which we parameterize as in (4.96). The condition (4.94) is precisely the same as (4.19) in that the marginal distribution of U_S is exactly the honest distribution. Therefore (4.95) follows from (4.20). □

4.7 Outer Bound on Rate-Distortion Region for the Quadratic Gaussian Problem

The following theorem gives our outer bound on the rate-distortion region for the quadratic Gaussian CEO Problem with an adversary.

Theorem 13 *If the tuple* $(R_1, \ldots, R_L, D)$ *is achievable, then there exist* r_i *for* $r_i = 1, \ldots, L$ *such that for all* $S \subset \{1, \ldots, L\}$ *with* $|S| = L - 2s$ *and all* $A \subset S$,

$$\sum_{i \in A} R_i \geq \sum_{i \in A} r_i + \frac{1}{2} \log \frac{1}{D} - \frac{1}{2} \log \left(\frac{1}{\sigma_X^2} + \sum_{i \in S \setminus A} \frac{1 - \exp(-2r_i)}{\sigma_{N_i}^2} \right), \tag{4.99}$$

$$\frac{1}{D} \leq \frac{1}{\sigma_X^2} + \sum_{i \in S} \frac{1 - \exp(-2r_i)}{\sigma_{N_i}^2}. \tag{4.100}$$

The region specified in our outer bound in Theorem 13 is identical to the rate region for the non-Byzantine problem given in [55, 56], and stated in (4.91)–(4.92), except that the two conditions on $\{1, \ldots, L\}$ have been replaced with conditions on S for all sets of size $L - 2s$. Together our inner and outer bounds match at $s = 0$ and recover the non-adversary result of [55, 56].

Proof: Assume $(R_1, \ldots, R_L, D)$ is achievable, and consider a code that achieves it with codewords $(C_1, \ldots, C_L)$. We may assume without loss of generality that the code achieves distortion D with probability at least $1 - \epsilon$, because we can always repeat the code multiple times and apply the law of large numbers. Fix

$S \subset \{1, \ldots, L\}$ with $|S| = L - 2s$. We may write

$$\begin{aligned}
\sum_{i \in A} R_i &\geq \sum_{i \in A} \frac{1}{n} H(C_i) \\
&\geq \frac{1}{n} H(C_A) \\
&\geq \frac{1}{n} H(C_A | C_{S \setminus A}) \\
&\geq \frac{1}{n} I(Y_A^n; C_A | C_{S \setminus A}) \\
&= \frac{1}{n} I(Y_A^n, X^n; C_A | C_{S \setminus A}) \\
&= \frac{1}{n} I(X^n; C_A | C_{S \setminus A}) + \frac{1}{n} I(Y_A^n; C_A | X^n, C_{S \setminus A}) \\
&= \frac{1}{n} I(X^n; C_S) - \frac{1}{n} I(X^n; C_{S \setminus A}) + \sum_{i \in A} \frac{1}{n} I(Y_i^n; C_i | X^n). \qquad (4.101)
\end{aligned}$$

We define (for this section)

$$r_i = \frac{1}{n} I(Y_i^n; C_i | X^n). \qquad (4.102)$$

Lemma 3.1 in [56] states that for any $B \subset \{1, \ldots, L\}$,

$$\frac{1}{\sigma_X^2} \exp\left(\frac{2}{n} I(X^n; C_B)\right) \leq \frac{1}{\sigma_X^2} + \sum_{i \in B} \frac{1 - \exp(-2r_i)}{\sigma_{N_i}^2} \qquad (4.103)$$

which allows us to bound the second term in (4.101). Only the first term remains, which we may rewrite as

$$\frac{1}{n} I(X^n; C_S) = \frac{1}{n} h(X^n) - \frac{1}{n} h(X^n | C_S) = \frac{1}{2} \log 2\pi e \sigma_X^2 - \frac{1}{n} h(X^n | C_S). \qquad (4.104)$$

We will proceed to show that

$$\frac{1}{n} h(X^n | C_S) \leq \frac{1}{2} \log 2\pi e D \qquad (4.105)$$

which, combined with (4.102), (4.103), and (4.104), allows us to extend (4.101) to (4.99). Taking $A = \emptyset$ gives (4.100).

We now prove (4.105). Let H_1, H_2 be sets of size $L - s$ such that $S = H_1 \cap H_2$. If H_1 is the true set of honest nodes, then they use the deterministic encoding

functions f_i to get C_{H_1} from $Y_{H_1}^n$. Meanwhile, the traitors, ${H_1}^c$, choose $C_{H_1^c}$. The decoder's estimate $\hat{X}^n$ is a deterministic function of C^L, but when H_1 are the honest nodes, we can think of it as a deterministic function of $Y_{H_1}^n$ and $C_{{H_1}^c}$. Thus we can define the set

$$S_D(X, Y_{H_1}) = \left\{ (x^n, y_{H_1}^n) : \forall c_{H_1^c}, \frac{1}{n} d(x^n, \hat{X}^n(y_{H_1}^n, c_{H_1^c})) \leq D \right\}.$$

This is the set of all $(x^n, y_{H_1}^n)$ pairs for which $\hat{X}^n$ achieves the distortion constraint no matter what the traitors do. Because we assume that distortion D is achieved with probability nearly one, the probability of the set $S_D(X, Y_H)$ is also nearly one. We define the set $S_D(X, Y_{H_2})$ in an analogous fashion, in the case that H_2 is the true set of honest nodes. Since a code achieving distortion D must perform no matter which nodes are the traitors, $S_D(X, Y_{H_2})$ is also a set with probability nearly one.

Now define

$$Q_D(X, Y_S) = \{(x^n, y_S^n) : \exists y_{H_1 \setminus H_2}^n, y_{H_2 \setminus H_1}^n : \\ (x^n, y_{H_1}^n) \in S_D(X, Y_{H_1}), (x^n, y_{H_2}^n) \in S_D(X, Y_{H_2})\}. \quad (4.106)$$

That is, $Q_D(X, Y_S)$ is the set of pairs (x^n, y_S^n) such that $\hat{X}^n$ achieves the distortion constraint for some $y_{H_1 \setminus H_2}^n$ when H_1 are the honest nodes and some $y_{H_2 \setminus H_1}^n$ when H_2 are the honest nodes. Because the S_D sets have probability nearly one, so does Q_D.

For a fixed $y_{H \cap S}^n$, define the conditional version of Q_D as

$$Q_D(X | y_S^n) = \{x^n : (x^n, y_S) \in Q_D(X, Y_S)\}.$$

Note that

$$\begin{aligned}1-\epsilon &\leq \Pr(Q_D(X,Y_S))\\ &= \int_{Q_D(X,Y_S)} dx^n dy_S^n p(x^n, y_S^n)\\ &= \int dy_S^n p(y_S^n) \int_{Q_D(X|y_S^n)} dx^n p(x^n|y_S^n)\\ &= \int dy_S^n p(y_S^n) \Pr(Q_D(X|y_S^n)|y_S^n).\end{aligned}$$

Since this is a convex combination nearly equal 1, each individual value must nearly equal 1, so in particular the probability of $Q_D(X|y_S^n)$ is nearly 1 given y_S^n.

Fix a codeword c_S. Define

$$Q_D(X|c_S) = \bigcup_{y_S^n : f_S(y_S^n)=c_S} Q_D(X|y_S^n).$$

From the high probability property of $Q_D(X|y_S^n)$, it follows that $Q_D(X|c_S)$ has high probability conditioned on c_S being sent. Hence

$$\frac{1}{n} h(X^n|C_S) \leq \frac{1}{n} \max_{c_{H\cap S}} \log \mathrm{Vol}(Q_D(X|c_{H\cap S})). \tag{4.107}$$

Consider two elements x^n, x'^n of $Q_D(X|c_S)$. By definition, there must be some sequences $y_S^n, y_S'^n$ such that $(x^n, y_S^n), (x'^n, y_S'^n) \in Q_D(X, Y_{H\cap S})$. From the definition of Q_D, there must be sequences $y_{H_1\setminus H_2}^n$ and $y_{H_2\setminus H_1}'^n$ extending y_S^n and $y_S'^n$ respectively such that $(x^n, y_{H_1}^n) \in S_D(X, Y_{H_1})$ and $(x'^n, y_{H_2}'^n) \in S_D(X, Y_{H_2})$. Consider the case that c_S, $c_{H_1\setminus H_2} = f_{H_1\setminus H_2}(y_{H_1\setminus H_2}^n)$, and $c_{H_2\setminus H_1} = f_{H_2\setminus H_1}(y_{H_2\setminus H_1}'^n)$ are sent. First observe that this set of messages could have been produced if $X^n = x^n$, $Y_{H_1}^n = y_{H_1}^n$, and H_1 were the set of honest nodes. Then the nodes in $H_2 \setminus H_1$, which are all traitors, could send $c_{H_2\setminus H_1}$. Since $(x^n, y_{H_1}^n) \in S_D(X, Y_{H_1})$, by definition the estimate $\hat{x}^n$ produced at the decoder must satisfy $\frac{1}{n} d(x^n, \hat{x}^n) \leq D$. However, the same set of messages could have been produced if $X^n = x'^n$, $Y_{H_2}^n = y_{H_2}^n$, and H_2 were the set of honest nodes, where $H_1 \setminus H_2$ decide to send $c_{H_1\setminus H_2}$. Since the

decoder produces just one estimate for any input messages, the very same estimate $\hat{x}^n$, by the same reasoning, must satisfy $\frac{1}{n}d(x'^n, \hat{x}^n) \leq D$. Hence, we have

$$\frac{1}{n}\sum_{t=1}^{n}(x(t) - \hat{x}(t))^2 \leq D,$$
$$\frac{1}{n}\sum_{t=1}^{n}(x'(t) - \hat{x}(t))^2 \leq D.$$

We may rewrite this as

$$\|x - \hat{x}\|_2 \leq \sqrt{nD},$$
$$\|x' - \hat{x}\|_2 \leq \sqrt{nD}.$$

Therefore by the triangle inequality, for any $x^n, x'^n \in Q_D(X|c_S)$,

$$\|x - x'\|_2 \leq 2\sqrt{nD}.$$

That is, $Q_D(X|c_S)$ has diameter at most $2\sqrt{nD}$. The following lemma bounds the volume of subsets of $\mathbb{R}^n$ as a function of their diameter. It is proved is Sec. 4.7.1.

Lemma 10 *The volume of any subset of $\mathbb{R}^n$ is no more than that of the n-ball with the same diameter.*

Using Lemma 10, we have that the volume of $Q_D(X|c_S)$ is no more than the volume of an n-ball with radius $\sqrt{nD}$. It can be easily shown that such a ball has volume no more than $(2\pi eD)^{n/2}$. Applying this to (4.107) gives (4.105), completing the proof. □

4.7.1 Proof of Lemma 10

Fix a set $A \subset \mathbb{R}^n$ with diameter $2r$. That is, for any $x, y \in A$, $\|x - y\|_2 \leq 2r$. Consider the set sum

$$A - A = \{x - y : x, y \in A\}.$$

Certainly for any point $z \in A - A$, $\|z\|_2 \leq 2r$. Therefore, $A - A$ is contained in the n-ball of radius $2r$. Let C_n be the volume of a unit n-ball, so an n-ball of radius r has volume $C_n r^n$. Hence

$$\text{Vol}(A - A) \leq C_n (2r)^n = 2^n C_n r^n. \tag{4.108}$$

The Brunn-Minkowski inequality [92] states that for any $A, B \subset \mathbb{R}^n$,

$$\text{Vol}(A + B)^{1/n} \geq \text{Vol}(A)^{1/n} + \text{Vol}(B)^{1/n}.$$

Therefore

$$\text{Vol}(A - A) \geq [\text{Vol}(A)^{1/n} + \text{Vol}(-A)^{1/n}]^n = 2^n \text{Vol}(A). \tag{4.109}$$

Combining (4.108) with (4.109) gives

$$\text{Vol}(A) \leq C_n r^n.$$

That is, the volume of A is no more than that of an n-ball with the same diameter.

4.8 Asymptotic Results for the Quadratic Gaussian Problem

The following theorem bounds the asymptotic proportionality constant $K(\beta)$.

Theorem 14 *For a fraction β of traitors*

$$\frac{\sigma_N^2}{2\sigma_X^2} \frac{1}{1 - 2\beta} \leq K(\beta) \leq \frac{\sigma_N^2}{2\sigma_X^2} \frac{\sqrt{1-\beta} + \sqrt{\beta}}{(1-\beta)(\sqrt{1-\beta} - \sqrt{\beta})}. \tag{4.110}$$

At $\beta = 0$, the two bounds meet at $\sigma_N^2/(2\sigma_X^2)$, matching the result of [54]. They also both diverge at $\beta = 1/2$. The ratio between them is monotonically increasing

in β and is never more than 4. The proof is stated in the next two subsections, and both sides make use of the bounds already found on the rate-distortion region in Sec. 4.6 and Sec. 4.7.

4.8.1 Proof of the Upper Bound on the Asymptotic Proportionality Constant

We apply Theorem 12. For a given sum-rate R, let $R_i = R/L$ for all i. Let r be the largest possible value satisfying (4.93) where $r_i = r/L$. It is not hard to show that for large L and R, r is nearly equal to R.

We need to specify the function $c_i(\Sigma)$. First define for all $A \subset \{1, \ldots, L\}$

$$\hat{X}_A = \mathbb{E}(X|U_A) = \frac{\sum_{i \in A} \frac{U_i}{\sigma_N^2}}{\frac{1}{\sigma_X^2} + |A| \frac{1-\exp(-2r/L)}{\sigma_N^2}}.$$

When X and U_A are related according to the nominal distribution,

$$\mathbb{E}(X - \hat{X}_A)^2 = \frac{1}{\frac{1}{\sigma_X^2} + |A| \frac{1-\exp(-2r/L)}{\sigma_N^2}}.$$

If we fix $|A|/L$, for large L and R,

$$\mathbb{E}(X - \hat{X}_A)^2 \approx \frac{\sigma_N^2}{2R} \frac{L}{|A|}.$$

Also observe that if $B \subset A$,

$$\mathbb{E}(\hat{X}_A - \hat{X}_B)^2 = \mathbb{E}(X - \hat{X}_B)^2 - \mathbb{E}(X - \hat{X}_A)^2.$$

We choose the c_i in the following way. Given Σ, we look for a set $\hat{H} \subset \{1, \ldots, L\}$ of size $(1 - \beta)L$ that has the expected distribution if H were the set of honest agents. That is, for all $i, j \in \hat{H}$,

$$\Sigma_{i,j} = \sigma_X^2 + \frac{\sigma_N^2}{1 - \exp(-2r/L)} \delta_{i,j}.$$

If there is more than one such $\hat{H}$, choose between them arbitrarily. Then define c_i such that

$$\sum_{i=1}^{L} c_i U_i = \hat{X}_{\hat{H}}.$$

Now we show that this choice achieves the upper bound given in Theorem 14. In the worst case, the true set of honest agents H shares just $(1-2\beta)L$ agents with $\hat{H}$. Because $U_{\hat{H}}$ is distributed according to the nominal distribution,

$$\begin{aligned}
\mathbb{E}(\hat{X}_{\hat{H}} - \hat{X}_{\hat{H}\cap H})^2 &= \mathbb{E}(X - \hat{X}_{\hat{H}\cap H})^2 - \mathbb{E}(X - \hat{X}_{\hat{H}})^2 \\
&\approx \frac{\sigma_N^2}{2R}\left(\frac{L}{|\hat{H}\cap H|} - \frac{L}{|\hat{H}|}\right) \\
&\leq \frac{\sigma_N^2}{2R}\left(\frac{1}{1-2\beta} - \frac{1}{1-\beta}\right)).
\end{aligned}$$

Furthermore, since $\hat{H}\cap H$ contains only honest agents,

$$\mathbb{E}(\hat{X}_{\hat{H}\cap H} - X)^2 \approx \frac{\sigma_N^2}{2R}\frac{L}{|\hat{H}\cap H|} \leq \frac{\sigma_N^2}{2R}\frac{1}{1-2\beta}.$$

Therefore by the Cauchy-Shwartz inequality

$$\begin{aligned}
\mathbb{E}(\hat{X}_{\hat{H}} - X)^2 &\leq \left(\sqrt{\mathbb{E}(\hat{X}_{\hat{H}} - \hat{X}_{\hat{H}\cap H})^2} + \sqrt{\mathbb{E}(\hat{X}_{\hat{H}\cap H} - X)^2}\right)^2 \\
&\leq \frac{\sigma_N^2}{2R}\left(\sqrt{\frac{1}{1-2\beta} - \frac{1}{1-\beta}} + \sqrt{\frac{1}{1-2\beta}}\right)^2 \\
&= \frac{\sigma_N^2}{2R}\frac{\sqrt{1-\beta}+\sqrt{\beta}}{(1-\beta)(\sqrt{1-\beta}-\sqrt{\beta})}.
\end{aligned}$$

Therefore in the for large L and R,

$$R\frac{\mathbb{E}(\hat{X}_{\hat{H}} - X)^2}{\sigma_X^2} \leq \frac{\sigma_N^2}{2\sigma_X^2}\frac{\sqrt{(}1-\beta)+\sqrt{\beta}}{(1-\beta)(\sqrt{1-\beta}-\sqrt{\beta})}.$$

4.8.2 Proof of the Lower Bound on the Asymptotic Proportionality Constant

We apply Theorem 13. Let $r = \sum_{i=1}^{L} r_i$. Certainly

$$R = \sum_{i=1}^{L} R_i \geq \sum_{i=1}^{L} r_i = r.$$

We have that

$$\frac{1}{D} \leq \frac{1}{\sigma_X^2} + \min_{S:|S|=(1-2\beta)L} \sum_{i \in S} \frac{1 - \exp(-2r_i)}{\sigma_N^2}.$$

By concavity of the function $1 - \exp(-2r_i)$ in r_i, this is maximized when all the r_i are equal. Hence

$$\frac{1}{D} \leq \frac{1}{\sigma_X^2} + (1-2\beta)L \frac{1 - \exp(-2r/L)}{\sigma_N^2}.$$

Observe that

$$L(1 - \exp(-2r/L)) = L\left(\frac{2r}{L} + \mathcal{O}(L^{-2})\right) = 2r + \mathcal{O}(L^{-1}).$$

Taking the limit as $L \to \infty$ gives

$$\frac{1}{D} \leq \frac{1}{\sigma_X^2} + \frac{(1-2\beta)2r}{\sigma_N^2}.$$

Therefore

$$K(\beta) = \lim_{R\to\infty} \frac{RD}{\sigma_X^2} \geq \lim_{r\to\infty} \frac{rD}{\sigma_X^2} \geq \lim_{r\to\infty} \frac{r}{\sigma_X^2\left(\frac{1}{\sigma_X^2} + \frac{(1-2\beta)2r}{\sigma_N^2}\right)} = \frac{\sigma_N^2}{2\sigma_X^2}\ \frac{1}{1-2\beta}.$$

CHAPTER 5

MALICIOUS DATA ATTACKS ON POWER SYSTEM STATE ESTIMATION

5.1 Introduction

Since the beginning of the development of power system state estimation [69], it has been necessary to deal with bad data. Traditionally, bad data were assumed to be caused by random errors resulting from a fault in a power meter and/or its attendant communication system. These errors are modeled by a change of variance in Gaussian noise, which leads to an energy (L_2) detector. In this chapter, we study the problem that several meters are seized by an adversary that is able to corrupt the measurements from those meters that are received by the control center. This differs from previous investigations of the problem in that the malicious data at various meters can be simultaneously crafted by the adversary to defeat the state estimator, as opposed to independent errors caused by random faults.

This problem was first studied in [78], in which it was observed that there exist cooperative and malicious attacks on meters that all known bad data techniques will fail to detect. The authors of [78] gave a method to adjust measurements at just a few meters in the grid in such a way that bad data detector will fail to perceive the corruption of the data. In the sequel, we describe the attacks an *unobservable* attacks, as they are closely related to the classical notion of unobservability of an estimation problem. We regard the existence of unobservable attacks as a fundamental limit on the ability to detect malicious data attacks. We therefore study the problem in two regimes: when the adversary can executed an unobservable attack, and when it cannot or does not. In Sec. 5.3, we study the

former case, by characterizing the conditions under which an unobservable attack exists, and giving an efficient algorithm for finding small unobservable attacks. This can provide some insight into how vulnerable a given power network is to such an attack.

In the regime that an unobservable attack cannot be performed, it is possible for the control center to detect malicious data attacks. Moreover, it is less clear what the worst attacks are for the adversary. Therefore we study we study two aspects of the problem: (i) attack detection and localization strategies at the control center; (ii) attack strategies by the adversary.

We present in Sec. 5.4 a decision theoretic formulation of detecting malicious data injection by an adversary. Because the adversary can choose where to attack the network and design the injected data, the problem of detecting malicious data cannot be formulated as a simple hypothesis test, and the uniformly most power test does not exist in general. We propose a detector based on the generalized likelihood ratio test (GLRT). GLRT is not optimal in general, but it is known to perform well in practice and it has well established asymptotic optimality [87, 88, 89]. In other words, if the detector has many data samples, the detection performance of GLRT is close to optimal.

We note that the proposed detector has a different structure from those used in conventional bad data detectors which usually employ a test on the state estimator residues errors [69, 70, 94]. The proposed the GLRT detector does not compute explicitly the residue error. We show, however, that when there is at most one attacked meter (a single attacked data), the GLRT is identical to the classical largest normalized residue (LNR) test using the residue error from the minimum mean square error (MMSE) state estimator. The asymptotic optimality of GLRT

lends a stronger theoretic basis for the LNR test for the single bad data test.

Next we investigate malicious data attack from the perspective of an adversary who must make a tradeoff between inflicting the maximum damage on state estimation and being detected by the EMS at the control center. We define in Sec. 5.5 the notion of *Attacker Operating Characteristic* (AOC) that characterizes the tradeoff between the probability of being detected vs. resulting (extra) mean-square error at the state estimator. We therefore formulate the problem of optimal attack as minimizing the probability of being detected subject to causing the mean square error (MSE) to increase beyond a predetermined level. Finding the attack with the optimal AOC is intractable, unfortunately. We present a heuristic that allows us to obtain attacks that with minimum attack power leakage to the detector while increasing the mean square error at the state estimator beyond a predetermined objective. This heuristic reduces to an eigenvalue problem that can be solved off line.

Finally, in Sec. 5.6 we conduct numerical simulations on a small scale example using the IEEE 14 bus network. For the control center, we present simulation results that compare different detection schemes based on the *Receiver operating Characteristics* (ROC) that characterize the tradeoff between the probability of attack detection vs. the probability of false alarm. We show that there is a substantial difference between the problem of detecting randomly appearing bad data from detecting malicious data injected by an adversary. Next we compare the GLRT detector with two classical detection schemes: the $J(\hat{\mathbf{x}})$ detector and the (Bayesian) largest normalized residue (LNR) detector [69, 70]. Our test shows improvement over the two well established detection schemes. From the adversary perspective, we compare the *Attacker Operating Characteristics* (AOC). Our result

shows again that the GLRT detector gives higher probability of detection than that those of conventional detectors for the same amount MSE increase at the state estimator.

5.2 Problem Formulation

A power system is composed of a collection of busses, transmission lines, and power flow meters. We adopt a graph-theoretic model for such a system. Therefore the power system is modeled as an undirected graph (V, E), where V represents the set of busses, and E is the set of transmission lines. Each line connects two meters, so each element $e \in E$ is an unordered pair of busses in V. Fig 5.1 shows the graph structure of the IEEE 14-bus test system, which we use in our simulations. The control center receives measurements from various meters deployed throughout the system, from which it performs state estimation. Meters come in two varieties: transmission line flow meters, which measure the power flow through a single transmission line, and bus injection meters, which measure the total outgoing flow on all transmission lines connected to a single bus. Therefore each meter is associated with either a bus in V or a line in E. We allow for the possibility of multiple meters on the same bus or line. Indeed, in our simulations, we assume that a meter is placed in every bus, and two meters on every line, one in each direction.

The graph-theoretic model for the power system yields the following DC power

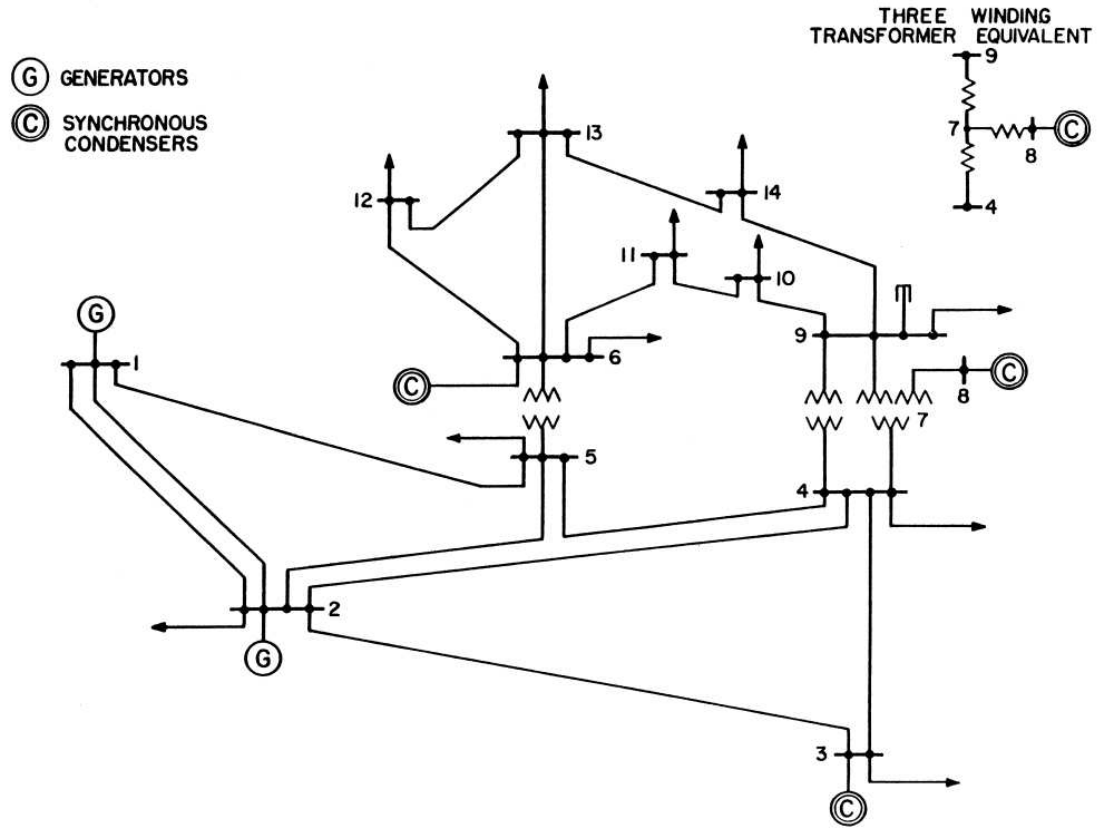


Figure 5.1: IEEE 14 bus test system.

flow model, a linearized version of the AC power flow model [95]:

$$\begin{aligned} \mathbf{z} &= \mathbf{H}\mathbf{x} + \mathbf{a} + \mathbf{e} \\ &\mathbf{e} \sim \mathcal{N}(\mathbf{0}, \mathbf{\Sigma}_e), \\ &\mathbf{a} \in \mathcal{A}_s = \{\mathbf{a} \in \mathbb{R}^m : \|\mathbf{a}\|_0 \leq s\} \end{aligned} \tag{5.1}$$

where $\mathbf{z} \in \mathbb{R}^m$ is the vector of power flow measurements, $\mathbf{x} \in \mathbb{R}^n$ is the system state, $\mathbf{e}$ is the Gaussian measurement noise with zero mean and covariance matrix $\mathbf{\Sigma}_e$, and vector $\mathbf{a}$ is malicious data injected by an adversary. Here we assume that the adversary can at most control s meters. That is, $\mathbf{a}$ is a vector with at most s non-zero entries ($\|\mathbf{a}\|_0 \leq s$). A vector $\mathbf{a}$ is said to have sparsity s if $\|\mathbf{a}\|_0 = s$. The $\mathbf{H}$ matrix in (5.1) arises from the graph theoretic model as follows. For each transmission line $(b_1, b_2) \in E$, the DC power flow through this line from bus b_1 to

bus b_2 is given by

$$\begin{bmatrix} 0 \cdots 0 & \underbrace{Y_{(b_1,b_2)}}_{b_1\text{th element}} & 0 \cdots 0 & \underbrace{-Y_{(b_1,b_2)}}_{b_2\text{th element}} & 0 \cdots 0 \end{bmatrix} \mathbf{x} \tag{5.2}$$

where $A_{(b_1,b_2)}$ is the susceptance of the transmission line (b_1, b_2). Let $h_{(b_1,b_2)}$ be the row vector in (5.2). If a meter measures the flow through the transmission line connecting busses b_1 and b_2, then the associated row of $\mathbf{H}$ is given by $h_{(b_1,b_2)}$. If a meter measures the power injection for bus b_1, then the associated row of $\mathbf{H}$ is given by

$$\sum_{b_2:(b_1,b_2)\in E} h_{(b_1,b_2)}. \tag{5.3}$$

5.2.1 A Bayesian Framework and MMSE Estimation

We consider in this paper a Bayesian framework where the state variables are random vectors with Gaussian distribution $\mathcal{N}(\boldsymbol{\mu}_x, \boldsymbol{\Sigma}_x)$. We assume that, in practice, the mean $\boldsymbol{\mu}_x$ and covariance $\boldsymbol{\Sigma}_x$ can be estimated from historical data. By subtracting the mean from the data, we can assume without loss of generality that $\boldsymbol{\mu}_x = \mathbf{0}$.

In the absence of an attack, i.e. $\mathbf{a} = \mathbf{0}$ in (5.1), $(\mathbf{z}, \mathbf{x})$ are jointly Gaussian. The minimum mean square error (MMSE) estimator of the state vector $\mathbf{x}$ is a linear estimator given by

$$\hat{\mathbf{x}}(\mathbf{z}) = \underset{\hat{\mathbf{x}}}{\operatorname{argmin}}\, \mathbb{E}(\|\mathbf{x} - \hat{\mathbf{x}}(\mathbf{z})\|^2) = \mathbf{K}\mathbf{z} \tag{5.4}$$

where

$$\mathbf{K} = \boldsymbol{\Sigma}_x \mathbf{H}^T (\mathbf{H}\boldsymbol{\Sigma}_x \mathbf{H}^T + \boldsymbol{\Sigma}_e)^{-1}. \tag{5.5}$$

The minimum mean square error, in the absence of attack, is given by

$$\mathcal{E}_0 = \min_{\hat{\mathbf{x}}} \mathbb{E}(||\mathbf{x} - \hat{\mathbf{x}}(\mathbf{z})||^2) = \operatorname{Tr}\left(\boldsymbol{\Sigma}_x - \mathbf{K}_x \mathbf{H}\boldsymbol{\Sigma}_x\right).$$

If an adversary injects malicious data $\mathbf{a} \in \mathcal{A}_s$ but the control center is unaware of it, then the state estimator defined in (5.4) is no longer the true MMSE estimator (in the presence of attack); the estimator $\hat{\mathbf{x}} = \mathbf{Kz}$ is a "naive" MMSE estimator that ignores the possibility of attack, and it will incur a higher mean square error (MSE). In particular, it is not hard to see that the MSE in the presence of $\mathbf{a}$ is given by

$$\mathcal{E}_0 + \|\mathbf{Ka}\|_2^2. \tag{5.6}$$

The impact on the estimator from a particular attack $\mathbf{a}$ is given by the second term in (5.6). To increase the MSE at the state estimator, the adversary necessarily has to increase the "energy" of attack, which increases the probability of being detected at the control center.

5.3 Unobservable Attacks

Liu, Ning and Reiter observe in [78] that if there exists a nonzero s-sparse $\mathbf{a}$ for which $\mathbf{a} = \mathbf{Hc}$ for some $\mathbf{c}$, then

$$\mathbf{z} = \mathbf{Hx} + \mathbf{a} + \mathbf{e} = \mathbf{H}(\mathbf{x} + \mathbf{c}) + \mathbf{e}.$$

Therefore $\mathbf{x}$ cannot be distinguished from $\mathbf{x} + \mathbf{c}$ at the control center. If both $\mathbf{x}$ and $\mathbf{x} + \mathbf{c}$ are valid network states, the adversary's injection of data $\mathbf{a}$ when the true state is $\mathbf{x}$ will lead the control center to believe that the true network state is $\mathbf{x} + \mathbf{c}$, and vector $\mathbf{c}$ can be scaled arbitrarily. Since no detector can distinguish $\mathbf{x}$ from $\mathbf{x} + \mathbf{c}$, we call hereafter an attack vector $\mathbf{a}$ *unobservable* if it has the form $\mathbf{a} = \mathbf{Hc}$.

Note that it is unlikely that random bad data $\mathbf{a}$ will satisfy $\mathbf{a} = \mathbf{Hc}$. But an adversary can synthesize its attack vector to satisfy the unobservable condition.

5.3.1 Characterization of Unobservable Attacks

The following theorem demonstrates that this type of attack is closely related to the classical notion of network observability [75].

Theorem 15 *An s-sparse attack vector* $\mathbf{a}$ *comprises an unobservable attack if and only if the network becomes unobservable when the s meters associated with the nonzero entries of* $\mathbf{a}$ *are removed from the network; that is, the* $(m-s) \times n$ *submatrix of* $\mathbf{H}$ *taken from the rows of* $\mathbf{H}$ *corresponding to the zero entries of* $\mathbf{a}$ *does not have full column rank.*

Proof: Without loss of generality, let $\mathbf{H}$ be partitioned into $\mathbf{H}^T = [\mathbf{H}_1^T \mid \mathbf{H}_2^T]$, and submatrix $\mathbf{H}_1$ does not have full column rank, i.e. there exists a vector $\mathbf{c} \neq \mathbf{0}$ such that $\mathbf{H}_1\mathbf{c}=\mathbf{0}$. We now have $\mathbf{a} = \mathbf{H}\mathbf{c} \in \mathcal{A}_s$, which is unobservable by definition. Conversely, consider an unobservable $\mathbf{a} = \mathbf{H}\mathbf{c} \in \mathcal{A}_s$. Without loss of generality, we can assume that the first $m-s$ entries of $\mathbf{a}$ are zero. We therefore have $\mathbf{H}_1\mathbf{c} = \mathbf{0}$ where $\mathbf{H}_1$ is the submatrix made of the first $m-s$ rows of $\mathbf{H}$. □

The implication from the above theorem is that the attack discovered in [78] is equivalent to removing s meters from the network thus making the network not observable.

Note that even though an unobservable attack is equivalent to the network being made unobservable, the adversarial attack is still much more destructive. When the network is unobservable because there are insufficient meters, the control center can easily determine this; it knows exactly what aspects about the system state it can gather information about, and which it cannot. However, in the case of an unobservable adversarial attack, the control center does not know it is under

attack, nor which of several possible attacks is being executed. Therefore the situation is much more precarious, because the control center does not even know what it does not know.

5.3.2 Graph-Theoretic Approach to Minimum Size Unobservable Attacks

To know how susceptible a power system is to this highly damaging unobservable attack, it is important to know how few meters must be controlled by the adversary before the attack can be performed. From Theorem 15, we know that there is an unobservable s-sparse attack vector a if and only if it is possible to remove s rows from $\mathbf{H}$ and cause $\mathbf{H}$ not to have full column rank. Finding the minimum such s for an arbitrary $\mathbf{H}$ is a hard problem. However, it becomes easier given the extra structure on $\mathbf{H}$ imposed by the network topology.

We now give a simple method to find sets of meters whose removal make the system unobservable. Moreover, we show that it is possible to efficiently minimize the size of the set of meters produced by this method; thereby one may efficiently compute small sets of meters from which an adversary may execute an unobservable attack.

For a set of lines $\mathcal{A} \subseteq E$, let $g(\mathcal{A})$ be the set of meters either on lines in $\mathcal{A}$ or on busses adjacent to lines in $\mathcal{A}$. Let $h(\mathcal{A})$ be the number of connected components in the graph $(V, E \setminus \mathcal{A})$; i.e. the original graph after all lines in $\mathcal{A}$ have been removed. The following theorem gives a simple method for determining a number of meters in $g(\mathcal{A})$ to remove from the network to make it unobservable. The proof relies on [77], which gave an efficient method to determine the observability of a network

based only on its topology.

Theorem 16 (Sufficient condition for unobservable attacks) *For all $\mathcal{A} \subseteq E$, removing an arbitrary subset of $g(\mathcal{A})$ of size $|g(\mathcal{A})| - h(\mathcal{A}) + 2$ makes the system unobservable.*

Proof: Let $\bar{V}$ and $\bar{E}$ be the sets of busses and lines respectively with a meter placed on them. Theorem 5 in [77] states that the power system given by $(V, E, \bar{V}, \bar{E})$ is observable if and only if there exists a $\mathcal{F} \subseteq E$ comprising a spanning tree of V and an assignment function

$$\phi : \mathcal{F} \to \bar{V} \cup \bar{E} \tag{5.7}$$

satisfying:

1. If $l \in \bar{E}$, then $\phi(l) = l$.
2. If $\phi(l) \in \bar{V}$, then line l is incident to the bus $\phi(l)$.
3. If $l_1, l_2 \in \mathcal{F}$ are distinct, then $\phi(l_1) \neq \phi(l_2)$.

The principle behind this theorem is that a bus injection meter may "impersonate" a single line meter on a line incident to the bus. If a bus $b = \phi(l)$ for some line l, this represents the meter at b impersonating a meter on line l. The system is observable if and only if a spanning tree $\mathcal{F}$ exists made up of transmission lines with either real meters or impersonated meters by bus meters.

Not including the lines in $\mathcal{A}$, the network splits into $h(\mathcal{A})$ separate pieces. Therefore, any spanning tree $\mathcal{F}$ must include at least $h(\mathcal{A}) - 1$ lines in $\mathcal{A}$. Any assignment ϕ satisfying the conditions above must therefore employ at least $h(\mathcal{A}) -$

1 meters in $g(\mathcal{A})$. Hence, if any $|g(\mathcal{A})| - h(\mathcal{A}) + 2$ of these meters are removed from the network, only $h(\mathcal{A}) - 2$ remain, which is not enough to create a full spanning tree, so the network becomes unobservable. □

Example: Consider the IEEE 14-bus test system, shown in Fig. 5.1. Take $\mathcal{A} = \{(7, 8)\}$. Since bus 8 is only connected to the system through bus 7, removing this line from the network cuts it into two pieces. Therefore $h(\mathcal{A}) = 2$. The set of meters $g(\mathcal{A})$ consists of meters on the line $(7, 8)$, and bus injection meters at bus 7 and 8. Theorem 16 states that if we remove $|g(\mathcal{A})|$ meters from this set—that is, all the meters in $g(\mathcal{A})$—the system becomes unobservable. In our simulation examples, we assume there are two meters on each line, therefore it takes 4 meters to execute an unobservable attack. Furthermore, it is not hard to employ Theorem 16 to find similar 4-sparse unobservable attacks on the 30-bus, 118-bus, and 300-bus test systems.

Theorem 16 provides a method to find unobservable attacks, but we would like to find attacks using as few meters as possible. We use the theory of submodular functions to show that the quantity $|g(\mathcal{A})| - h(\mathcal{A}) + 2$ can be efficiently minimized over all sets of edges $\mathcal{A}$. This significantly increases the usefulness of Theorem 16, because it means we can efficiently find small unobservable attacks for arbitrary power systems.

A submodular function is a real-valued function f defined on the collection of subsets of a set W such that for any $A, B \subseteq W$,

$$f(A \cup B) + f(A \cap B) \leq f(A) + f(B). \tag{5.8}$$

Moreover, a function f is supermodular if $-f$ is submodular. There are several known techniques to find the set $A \subseteq W$ minimizing $f(A)$ in time polynomial in the size of W [84, 85, 86]. It is not hard to see that $|g(\mathcal{A})|$ is submodular in $\mathcal{A}$,

and $h(\mathcal{A})$ is supermodular. Therefore, their difference is submodular, so it can be efficiently minimized.

5.4 Detection of Malicious Data Attack

In this section, we study the problem in the regime that the adversary cannot or does not perform an unobservable attack as described in Sec. 5.3. In this regime, it is possible to detect the adversary's presence. We first formulate the detection problem, then introduce the generalized likelihood ratio test (GLRT), as well as some classical detectors.

5.4.1 Statistical Model and Attack Hypotheses

We now present a formulation of the detection problem at the control center. We assume a Bayesian model where the state variables are random with a multivariate Gaussian distribution $\mathbf{x} \sim \mathcal{N}(\mathbf{0}, \mathbf{\Sigma}_x)$. Our detection model, on the other hand, is not Bayesian in the sense that we do not assume any prior probability of the attack nor do we assume any statistical model for the attack vector $\mathbf{a}$.

Under the observation model (5.1), we consider the following composite binary hypothesis:

$$\mathcal{H}_0 : \mathbf{a} = \mathbf{0} \quad \text{versus} \quad \mathcal{H}_1 : \mathbf{a} \in \mathcal{A}_s \setminus \{\mathbf{0}\}. \tag{5.9}$$

Given observation $\mathbf{z} \in \mathbb{R}^m$, we wish to design a detector $\delta : \mathbb{R}^m \to \{0, 1\}$ with $\delta(\mathbf{z}) = 1$ indicating a detection of attack ($\mathcal{H}_1$) and $\delta(\mathbf{z}) = 0$ the null hypothesis.

An alternative formulation, one we will not pursue here, is based on the extra

MSE $\|\mathbf{Ka}\|_2^2$ at the state estimator. See (5.6). In particular, we may want to distinguish, for $\|\mathbf{a}\|_0 \leq s$,

$$\mathcal{H}_0' : \|\mathbf{Ka}\|_2^2 \leq C, \quad \text{versus} \quad \mathcal{H}_1' : \|\mathbf{Ka}\|_2^2 > C. \tag{5.10}$$

Here both null and alternative hypotheses are composite and the problem is more complicated. The operational interpretation, however, is significant because one may not care in practice about small attacks that only marginally increase the MSE of the state estimator.

5.4.2 Generalized Likelihood Ratio Detector with L_1 Norm Regularization

For the hypotheses test given in (5.9), the uniformly most powerful test does not exist. We propose a detector based on the generalized likelihood ratio test (GLRT). We note in particular that, if we have multiple measurements under the same $\mathbf{a}$, the GLRT proposed here is asymptotically optimal in the sense that it offers the fastest decay rate of miss detection probability [96].

The distribution of the measurement $\mathbf{z}$ under the two hypotheses differ only in their means

$$\begin{aligned} \mathcal{H}_0 &: \mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{\Sigma}_z) \\ \mathcal{H}_1 &: \mathbf{z} \sim \mathcal{N}(\mathbf{a}, \mathbf{\Sigma}_z),\ \mathbf{a} \in \mathcal{A}_s \setminus \{\mathbf{0}\} \end{aligned}$$

where $\mathbf{\Sigma}_z \triangleq \mathbf{H}\mathbf{\Sigma}_x\mathbf{H}^T + \mathbf{\Sigma}_e$. The GLRT is given by

$$L(\mathbf{z}) \triangleq \frac{\max\limits_{\mathbf{a}\in\mathcal{A}_s} f(\mathbf{z}|\mathbf{a})}{f(\mathbf{z}|\mathbf{a}=\mathbf{0})} \underset{\mathcal{H}_0}{\overset{\mathcal{H}_1}{\gtrless}} \tau, \tag{5.11}$$

where $f(\mathbf{z}|\mathbf{a})$ be the Gaussian density function with mean $\mathbf{a}$ and covariance $\mathbf{\Sigma}_z$, and the threshold τ is chosen from under null hypothesis for a certain false alarm rate. This is equivalent to

$$\min_{\mathbf{a}\in\mathcal{A}_s} \mathbf{a}^T\mathbf{\Sigma}_z^{-1}\mathbf{a} - 2\mathbf{z}^T\mathbf{\Sigma}_z^{-1}\mathbf{a} \underset{\mathcal{H}_1}{\overset{\mathcal{H}_0}{\gtrless}} \tau. \tag{5.12}$$

Thus the GLRT reduces to solving

$$\begin{aligned} &\text{minimize} \quad \mathbf{a}^T\mathbf{\Sigma}_z^{-1}\mathbf{a} - 2\mathbf{z}^T\mathbf{\Sigma}_z^{-1}\mathbf{a} \\ &\text{subject to} \quad \|\mathbf{a}\|_0 \leq s. \end{aligned} \tag{5.13}$$

For a fixed sparsity pattern, i.e. if we know the support but not necessarily the actual values of $\mathbf{a}$, the above optimization is easy to solve. In other words, if we know a small set of suspect meters from which malicious may be injected, the above test is easily computable. The sparsity condition on $\mathbf{a}$ makes the above optimization problem non-convex, but for small s it can be solved exactly simply by exhaustively searching through all sparsity patterns. For larger s, this is not feasible. It is a well known technique that (5.13) can be approximated by a convex optimization:

$$\begin{aligned} &\text{minimize} \quad \mathbf{a}^T\mathbf{\Sigma}_z^{-1}\mathbf{a} - 2\mathbf{z}^T\mathbf{\Sigma}_z^{-1}\mathbf{a} \\ &\text{subject to} \quad \|\mathbf{a}\|_1 \leq \nu \end{aligned} \tag{5.14}$$

where the L_1 norm constraint is a heuristic for the sparsity of $\mathbf{a}$. The constant ν needs to be adjusted until the solution involves an $\mathbf{a}$ with sparsity s. This requires solving (5.14) several times. A similar approach was taken in [79].

5.4.3 Classical Detectors with MMSE State Estimation

We will compare the performance of the GLRT detector with two classical bad data detectors [69, 70], both based on the residual error $\mathbf{r} = \mathbf{z} - \mathbf{H}\hat{\mathbf{x}}$ resulted from

the MMSE state estimator.

The first is the $J(\hat{\mathbf{x}})$ detector, given by

$$\mathbf{r}^T \mathbf{\Sigma}_e^{-1} \mathbf{r} \underset{\mathcal{H}_0}{\overset{\mathcal{H}_1}{\gtrless}} \tau. \tag{5.15}$$

The second is the largest normalized residue (LNR) test given by

$$\max_i \frac{|r_i|}{\sigma_{r_i}} \underset{\mathcal{H}_0}{\overset{\mathcal{H}_1}{\gtrless}} \tau, \tag{5.16}$$

where σ_{r_i} is the standard deviation of the ith residual error r_i. We may regard this is a test on the l_∞-norm of the measurement residual, normalized so that each element has unit variance.

The asymptotic optimality of the GLRT detector implies a better performance of GLRT over the above two detectors when the sample size is large. For the finite sample case, numerical simulations shown in Sec 5.6 confirm that the GLRT detector improves the performance of the $J(\hat{\mathbf{x}})$ and LNR detectors. The interesting exception is the case when only one meter is under attack, i.e. $\|\mathbf{a}\|_0 = 1$ and $\mathbf{\Sigma}_e = \sigma_e^2 \mathbf{I}$. In this case, the GLRT turns out to be identical to the LNR detector. Therefore, the GLRT can be viewed as a generalization of the LNR detector, in that it can be tuned to any sparsity level. Moreover, this provides some theoretical justification for the LNR detector. The equivalence of the two detectors is stated and proved in the following Proposition.

Proposition 1 *When $s = 1$, the GLRT detector given in* (5.12) *is equivalent to the LNR detector given in* (5.16)*.*

Proof: If $s = 1$, the left hand side of (5.12) becomes

$$\min_i \min_{a_i} (\mathbf{\Sigma}_z^{-1})_{ii} a_i^2 - 2\mathbf{z}^T (\mathbf{\Sigma}_z^{-1})_i a_i \tag{5.17}$$

where $(\mathbf{\Sigma}_z^{-1})_{ii}$ is the ith diagonal element of $\mathbf{\Sigma}_z^{-1}$, and $(\Sigma_z^{-1})_i$ is the ith row of $\mathbf{\Sigma}_z^{-1}$. The second minimization can be solved in closed form, so (5.17) becomes

$$-\max_i \frac{[\mathbf{z}^T(\Sigma_z^{-1})_i]^2}{(\Sigma_z^{-1})_{ii}}. \tag{5.18}$$

We may therefore write the GLRT as

$$\max_i \frac{|\mathbf{z}^T(\mathbf{\Sigma}_z^{-1})_i|}{\sqrt{(\mathbf{\Sigma}_z^{-1})_{ii}}} \underset{\mathcal{H}_0}{\overset{\mathcal{H}_1}{\gtrless}} \tau. \tag{5.19}$$

The vector of numerators in (5.19) is given by $\mathbf{r}' = \mathbf{\Sigma}_z^{-1}\mathbf{z}$. Note that the covariance matrix of $\mathbf{r}'$ is simply Σ_z^{-1}. Therefore we may regard (5.19) as a test on the maximum element of the $\mathbf{r}'$ after each element is normalized to unit variance.

We now show that $\mathbf{r}'$ is just a constant multiple of $\mathbf{r}$, meaning that (5.19) is identical to (5.16), saving a constant factor. Recall that $\mathbf{r} = (\mathbf{I} - \mathbf{HK})\mathbf{z}$, where

$$\begin{aligned}
\mathbf{I} - \mathbf{HK} &= I - \mathbf{H\Sigma}_x\mathbf{H}^T(\mathbf{H\Sigma}_x\mathbf{H}^T + \mathbf{\Sigma}_e)^{-1} \\
&= (\mathbf{H\Sigma}_x\mathbf{H}^T + \mathbf{\Sigma}_e - \mathbf{H\Sigma}_x\mathbf{H}^T)(\mathbf{H\Sigma}_x\mathbf{H}^T + \mathbf{\Sigma}_e)^{-1} \\
&= \mathbf{\Sigma}_e\mathbf{\Sigma}_z^{-1} = \sigma_e^2\mathbf{\Sigma}_z^{-1}.
\end{aligned}$$

Thus $\mathbf{r} = \sigma_e^2\mathbf{r}'$; the two detectors are identical. □

5.5 Attack Operating Characteristics and Optimal Attacks

We now study the impact of malicious data attack from the perspective of an attacker. We assume that the attacker knows the (MMSE) state estimator and the (GLRT) detector used by the control center. We also assume that the attacker can choose s meters arbitrarily in which to inject malicious data. In practice, however, the attacker may be much more limitted. Thus our results here are perhaps more pessimistic than in reality.

5.5.1 AOC and Optimal Attack Formulations

The attacker faces two conflicting objectives: maximizing the MSE by choosing the best data injection $\mathbf{a}$ vs. avoiding being detected by the control center. The tradeoff between increasing MSE of the state estimator and lower the probability of detection is characterized by *attacker operating characteristics* (AOC), analogous to the receiver operating characteristics (ROC) at the control center. Specifically, AOC is the probability of detection of the detector $\Pr(\delta(\mathbf{z}) = 1 \mid \mathbf{a})$ as a function of the extra MSE $\mathcal{E}(\mathbf{a}) = \mathcal{E}_0 + \|\mathbf{K}\mathbf{a}\|_2^2$ (5.6) at the state estimator, where $\mathcal{E}_0$ is the MMSE in the absence of attack.

The optimal attack in the sense of maximizing the MSE while limiting the probability of detection can be formulated as the following constrained optimization

$$\max_{\mathbf{a}\in\mathcal{A}_s} \|\mathbf{K}\mathbf{a}\|_2^2 \quad \text{subject to} \quad \Pr(\delta(\mathbf{z}) = 1|\mathbf{a}) \leq \beta, \tag{5.20}$$

or equivalently,

$$\min_{\mathbf{a}\in\mathcal{A}_s} \Pr(\delta(\mathbf{z}) = 1|\mathbf{a}) \quad \text{subject to} \quad \|\mathbf{K}\mathbf{a}\|_2^2 \leq C. \tag{5.21}$$

In order to evaluate the true worst-case performance for any detector, (5.20) or (5.21) would need to be solved. This is very difficult, due to the lack of analytical expressions for the detection error probability $\Pr(\delta(\mathbf{z}) = 1|\mathbf{a})$. We propose a heuristic for $\Pr(\delta(\mathbf{z}) = 1|\mathbf{a})$, which will allow us to approximate the above optimization with one that is easier to solve.

5.5.2 Minimum Residue Energy Attack

Given the naive MMSE state estimator $\hat{\mathbf{x}} = \mathbf{K}\mathbf{z}$ (5.4-5.5), the estimation residue error is given by

$$\mathbf{r} = \mathbf{G}\mathbf{z}, \quad \mathbf{G} \triangleq \mathbf{I} - \mathbf{H}\mathbf{K} \tag{5.22}$$

Substituting the measurement model, we have

$$\mathbf{r} = \mathbf{G}\mathbf{H}\mathbf{x} + \mathbf{G}\mathbf{a} + \mathbf{G}\mathbf{e}.$$

where $\mathbf{G}\mathbf{a}$ is the only term from the attack. Therefore, an attack vector $\mathbf{a}$ will be more difficult to detect at the control center if $\mathbf{G}\mathbf{a}$ is small. Recall from (5.6), the damage in MSE done by injecting $\mathbf{a}$ is $\|\mathbf{K}\mathbf{a}\|_2^2$. We therefore consider the following equivalent problems:

$$\max_{\mathbf{a}\in\mathcal{A}_s} \|\mathbf{K}\mathbf{a}\|_2^2 \quad \text{subject to} \quad \|\mathbf{G}\mathbf{a}\|_2^2 \leq \eta, \tag{5.23}$$

or equivalently,

$$\min_{\mathbf{a}\in\mathcal{A}_s} \|\mathbf{G}\mathbf{a}\|_2^2 \quad \text{subject to} \quad \|\mathbf{K}\mathbf{a}\|_2^2 \geq C. \tag{5.24}$$

The above optimizations remain difficult due to the constraint $\mathbf{a} \in \mathcal{A}_s$. However, given a specific sparsity pattern $\mathcal{S} \subset \{1, \cdots, n\}$ for which $a_i = 0$ for all $i \notin \mathcal{S}$, solving the optimal attack vector $\mathbf{a}$ for the above two formulations is a standard generalized eignevalue problem.

In particular, for fixed sparsity pattern $\mathcal{S}$, let $\mathbf{a}_\mathcal{S}$ be the nonzero subvector of $\mathbf{a}$, $\mathbf{K}_\mathcal{S}$ the corresponding submatrix of $\mathbf{K}$, and $\mathbf{G}_\mathcal{S}$ similarly defined. The problem (5.24) becomes

$$\min_{\mathbf{u}\in\mathbb{R}^{n-s}} \|\mathbf{G}_\mathcal{S}\mathbf{u}\|_2^2 \quad \text{subject to} \quad \|\mathbf{K}_\mathcal{S}\mathbf{u}\|_2^2 \geq C. \tag{5.25}$$

Let $\mathbf{Q}_G \triangleq \mathbf{G}_\mathcal{S}^T\mathbf{G}_\mathcal{S}$, $\mathbf{Q}_K \triangleq \mathbf{K}_\mathcal{S}^T\mathbf{K}_\mathcal{S}$. It can be shown that the optimal attack pattern has the form

$$\mathbf{a}_\mathcal{S}^* = \sqrt{\frac{C}{\|\mathbf{K}_\mathcal{S}\mathbf{v}\|_2^2}}\mathbf{v} \tag{5.26}$$

where $\mathbf{v}$ is the generalized eigenvector corresponding to the smallest generalized eigenvalue $\lambda_{\min}$ of the following matrix pencil

$$\mathbf{Q}_G\mathbf{v} - \lambda_{\min}\mathbf{Q}_K\mathbf{v} = \mathbf{0}.$$

The s dimensional symmetrical generalized eigenvalue problem can be solved the QZ algorithm [97].

5.6 Numerical Simulations

We present some simulation results on the IEEE 14 bus system shown in Fig. 5.1 to compare the performance of the GLRT with the $J(\hat{x})$ test and the LNR test [69, 70]. For various sparsity levels, we find the minimum energy residue attack as discussed in Sec. 5.5.2. The adversary may then scale this attack vector depending on how much it wishes to influence the mean square error. We plot both the ROC and AOC curves for various sparsity levels and all three detectors. For the AOC curve, we fix a probability of false alarm and vary the length of the attack vector along the direction minimizing the energy residue, plotting the MSE vs. the probability of detection. For the ROC curve, we fix the length of the attack vector, but very the detector's threshold and plot the probability of false alarm vs. probability of detector. In our simulations, we characterize the mean square error increase at the control center using the ratio between the resulting MSE from the attack and the MSE under no attack (i.e. $\mathbf{a} = 0$) in dB.

Fig. 5.2 shows the ROC and AOC curves for the worst-case 2-sparse attack. We implement the GLRT using exhaustive search over all possible sparsity patterns. This is feasible because of the low sparsity level, so we need not resort to the L_1 minimization as in (5.14). Observe that the GLRT performs consistently better

than the other two conventional detectors.

Fig. 5.3 shows the ROC and AOC curves for the worst-case 3-sparse attack, again using exhaustive search for the GLRT. Interestingly, the LNR test outperforms the GLRT at this sparsity level. We believe the reason for this is that the GLRT has little recourse when there is significant uncertainty in the sparsity pattern of the attack. In particular, the meters being controlled by the adversary here are the bus injection meter at bus 1, and the two meters on the transmission line between bus 1 and 2. These constitute three of the seven meters that hold any information about the state at bus 1. Thus, it may be difficult for the detector to determine which of the several meters around bus 1 are the true adversarial meters. The GLRT does not react to this uncertainty: it can only choose the most likely sparsity pattern, which is often wrong. Indeed, in our simulations the GLRT identified the correct sparsity pattern only 4.2% of the time.

Continuing our analysis of 3-sparsity attacks, we conduct simulations when the adversaries are placed randomly in the network, instead of at the worst-case meters. Once their random meters are chosen, we find the worst-case attack vector using the energy residual heuristic. This simulates the situation that the adversaries cannot choose their locations, but are intelligent and cooperative in their attack. The resulting performance of the three detectors is shown in Fig. 5.4. Observe that we have recovered the outperformance of the GLRT as compared to the conventional detectors, if only slightly. When the placement of the adversaries is random, they are not as capable of cooperating with one another, therefore their attack is easier to detect.

We increase the sparsity level to 6, at which it is impossible to perform exhaustive search for the GLRT. At this sparsity level, it becomes possible to perform an

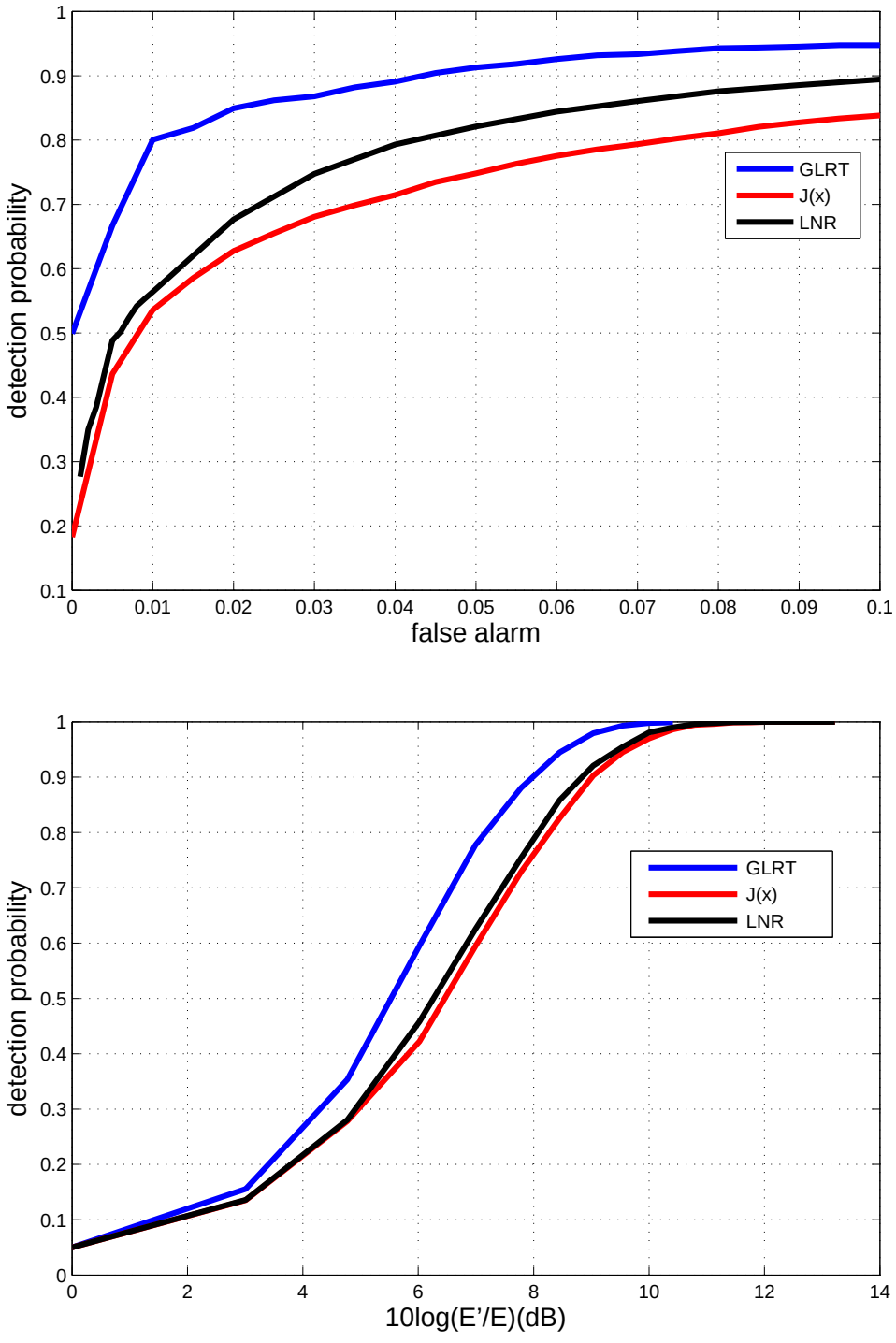
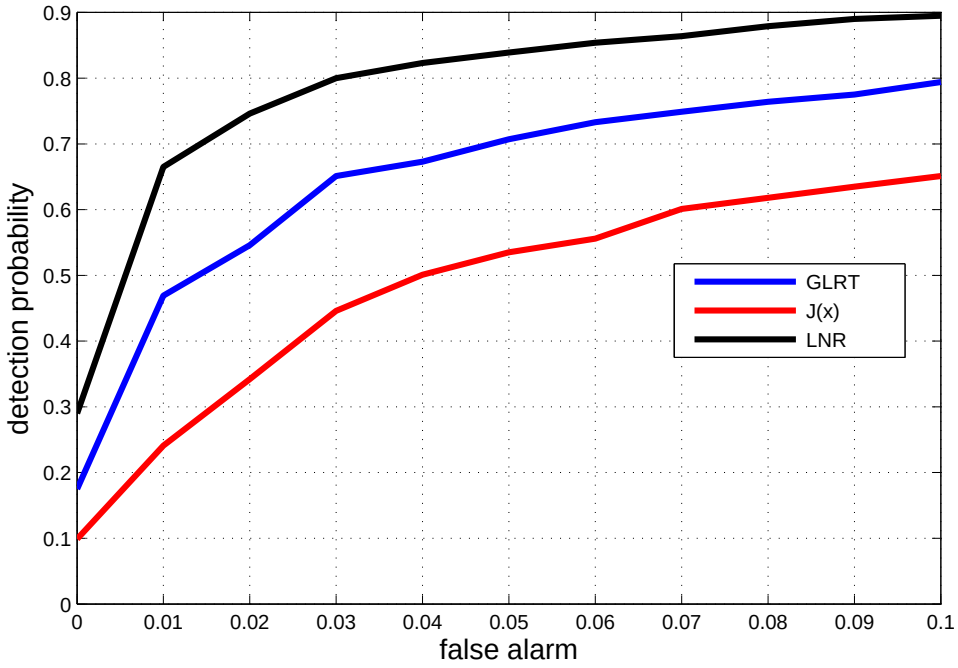


Figure 5.2: Above: ROC Performance of GLRT for the 2 sparsity case. MSE with attack is 8db. SNR=10db. Below: AOC Performance of GLRT for the 2 sparsity case. False alarm rate is 0.05. SNR=10dB.



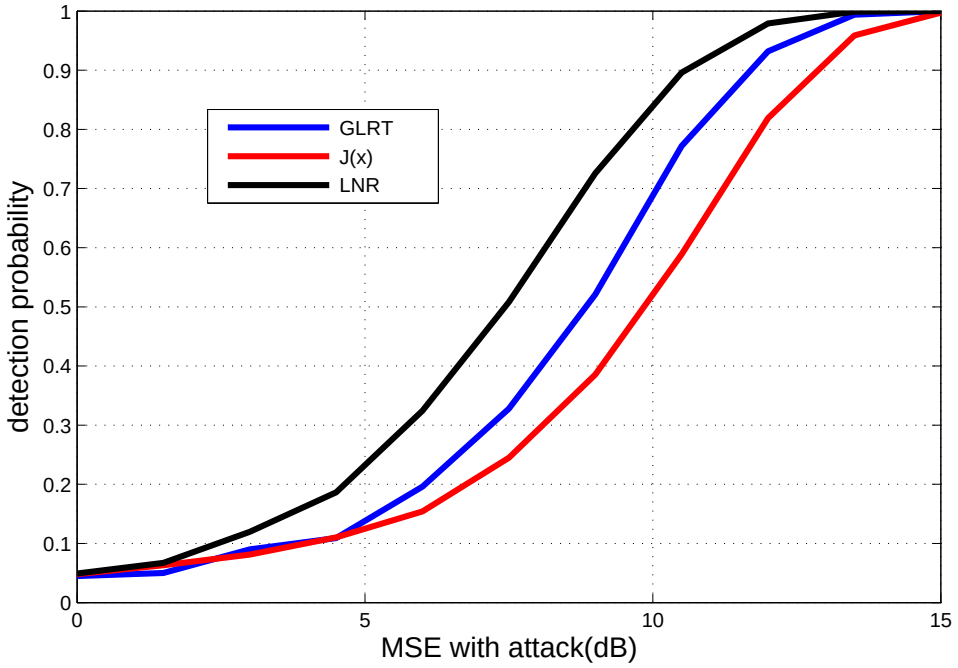


Figure 5.3: Above: ROC Performance of GLRT for the 3 sparsity case. MSE with attack is 10db. SNR=10db. Below: AOC Performance of GLRT for 3 sparsity case. False alarm rate is 0.05. SNR=10dB

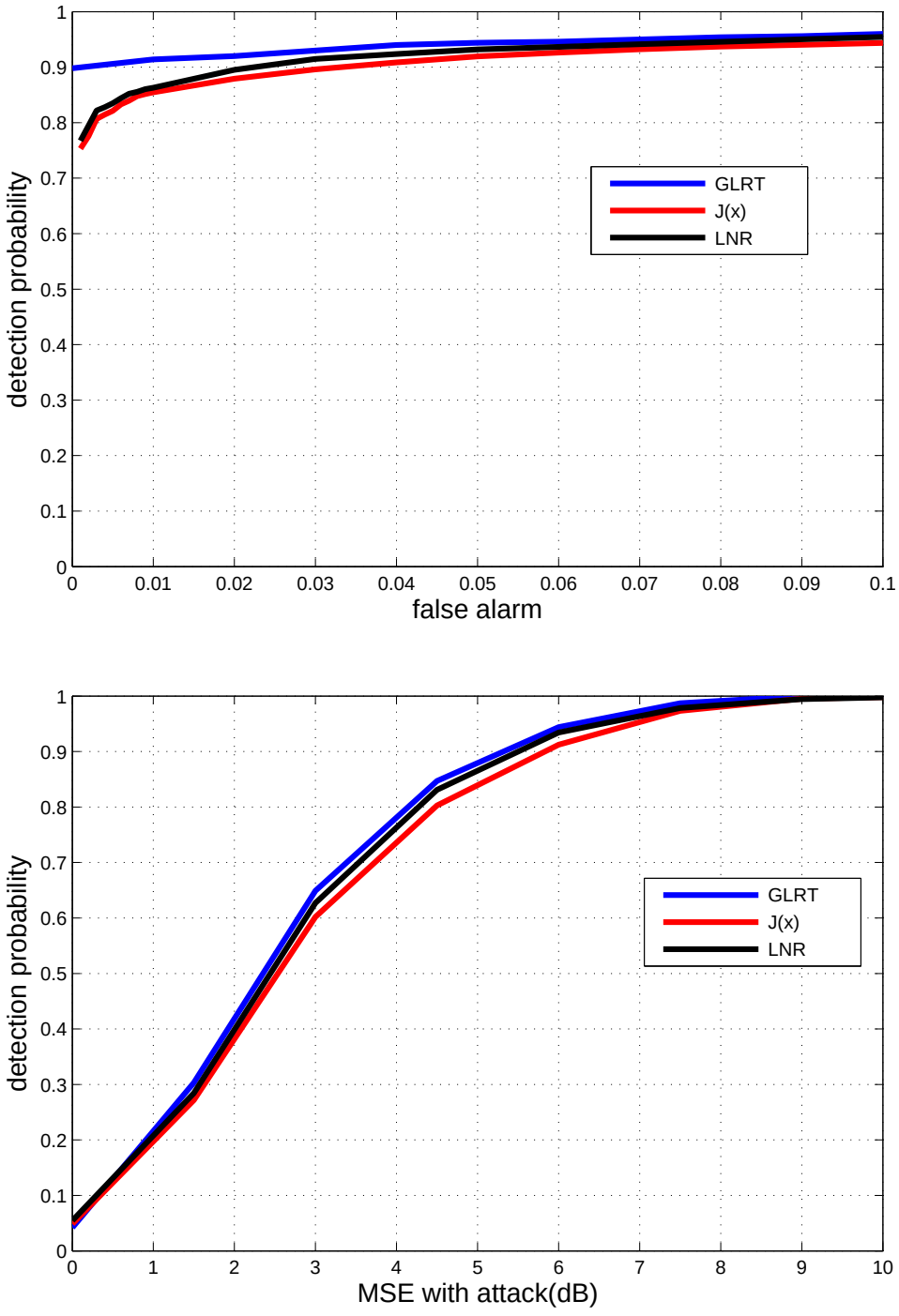


Figure 5.4: Above: ROC Performance of GLRT under random attack for 3 sparsity case. MSE with attack is 6db. SNR=10db. Below: AOC Performance of GLRT under random attack for 3 sparsity case. False alarm rate is 0.05. SNR=10dB

unobservable attack, so it is not as illuminating to choose the worst-case sparsity pattern, as that would be very difficult to detect. Instead, we again choose the sparsity pattern randomly but optimize the attack within it. Fig. 5.5 compares the performance of the GLRT implemented via L_1 minimization as in (5.14) to the two conventional detectors. Note again that the GLRT outperforms the others.

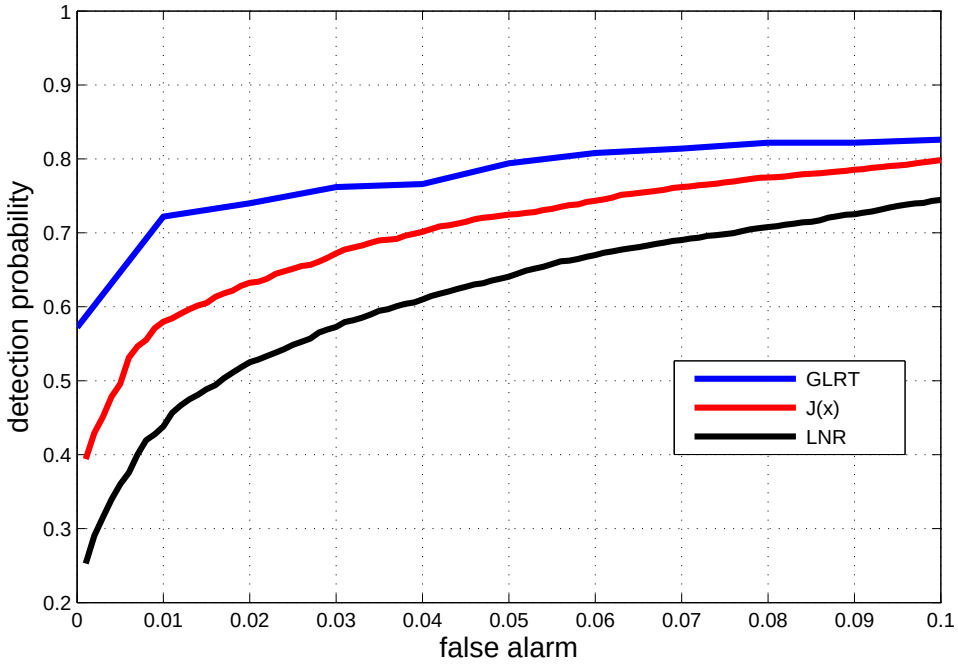


Figure 5.5: ROC Performance of GLRT under random attack for 6 sparsity case. MSE with attack is 6db. SNR=10db.

Finally, we present some numerical evidence that the residue energy described in Sec. 5.5.2 works well as a heuristic in that it is roughly increasing with the probability of detection $\Pr(\delta(\mathbf{z}) = 1|\mathbf{a})$ no matter what detector is used. For the $J(\hat{\mathbf{x}})$ and LNR detectors, we consider the detection probability for all 1-sparse vectors $\mathbf{a}$ satisfying $\|\mathbf{K}\mathbf{a}\|_2^2 = C$. on the 14-bus test system. We plot in Fig. 5.6 the value of the residue energy vs. the true probability of detector of $\mathbf{a}$ for both detectors. Observe that the scatter plots are roughly increasing.

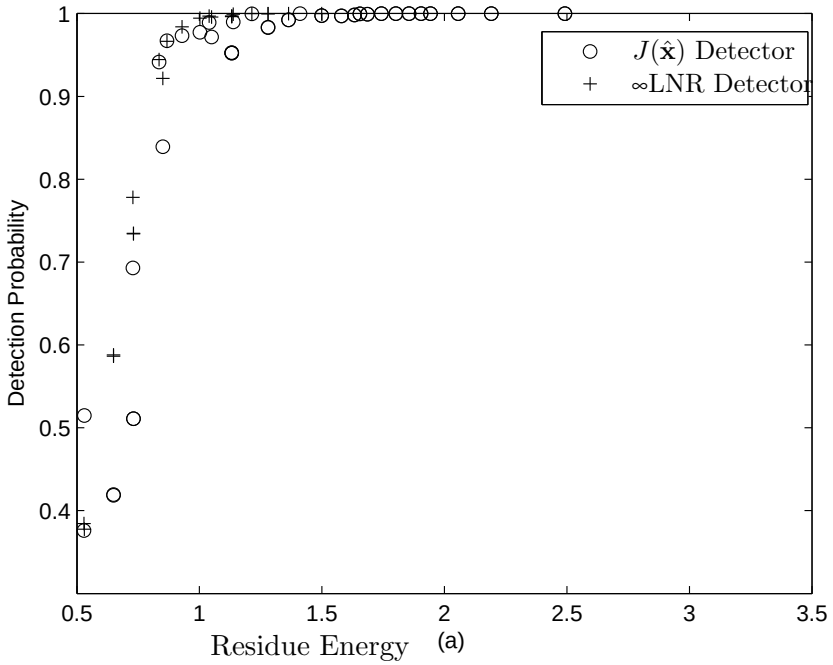


Figure 5.6: Comparison of the residue energy heuristic with the true detection probability for 1-sparse attack vectors for both $J(\hat{\mathbf{x}})$ and LNR detectors.

We evaluate the performance of the residue energy heuristic on 2-sparse vectors in the following way. For each pair of entries i, j of $\mathbf{a}$, we optimize (5.24) where $\mathbf{a}$ is constraint to have sparsity pattern $\{i, j\}$. We then evaluate the true probability of detection for the two detectors, with the same parameter values as above. The results are shown in Fig. 5.7 for the $J(\hat{\mathbf{x}})$ and LNR detectors. Again, the heuristic appears to track the true probabilities reasonably well. This provides some justification for our use earlier in the ROC and AOC curves of approximating the worst-case performance of these detectors by assume the maximum residue energy attack.

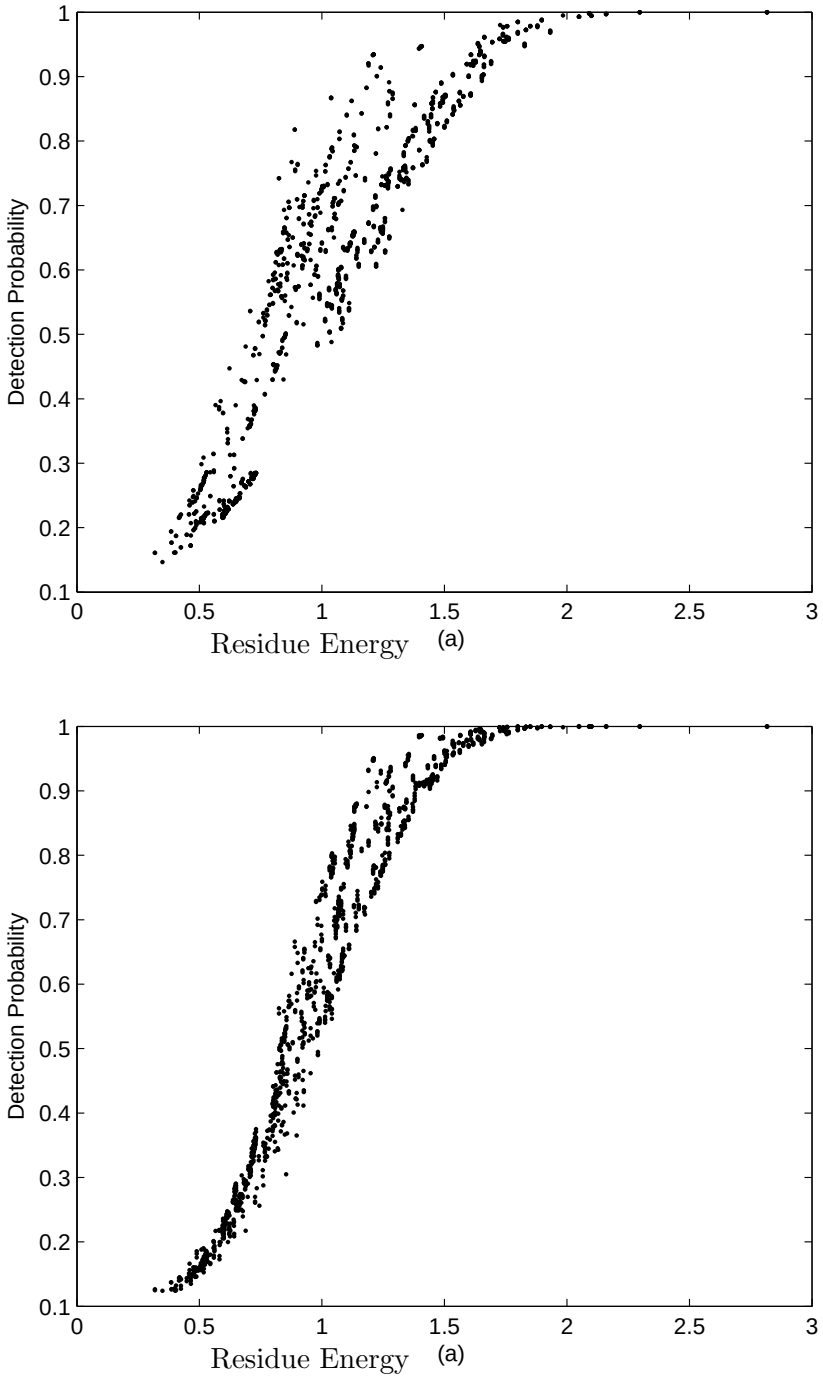


Figure 5.7: Comparison of the residue energy heuristic with the true detection probability for 2-sparse attack vectors. Above: Scatter plot for the $J(\hat{\mathbf{x}})$ detector. Below: Scatter plot for the LNR detector.

CHAPTER 6
CONCLUSIONS

This thesis studied the problem of an adversary entering a network and taking control of several nodes in it. We looked at several specific problems, and found strategies to defeat the adversary for each. We believe that our most significant contribution, at least for the information theory problems studied in Chapters 2–4, is the idea that adversaries can be detecting by observing joint empirical statistics. If the statistics do not match what was expected, then a traitor must be present. This simple idea forms the basis of Polytope Codes against adversaries in network coding, discussed in Chapter 2, as well as the achievable strategies against adversaries in the Slepian-Wolf problem in Chapter 3, and the Berger-Tung-like achievable strategy against adversaries in various multiterminal source coding problems in Chapter 4. We believe that this basic idea can be applied to more general network information theory problems. We now make some more specific comments on possible future directions in each of the areas.

6.1 Network Coding

There are numerous networks for which the results of Chapter 2 do not solve the network coding problem under node-based adversarial attack. The main result in Chapter 2 is Theorem 4, which states that the cut-set upper bound is achievable for a certain class of planar graph. Certainly it may be possible to generalize Theorem 4, and find larger classes of networks for which the cut-set bound is achievable. We believe that this should be possible with Polytope Codes. It would be interesting to analyze the planarity condition in more depth: perhaps it could lead to a more general theory of achievable rates given topological properties of

the network.

However, as we have shown, the cut-set bound is not always achievable, so to solve the general problem work would need to be done upper bounds as well. From the complicated nature of the tighter upper bound given in Sec. 2.11, we suspect that the solution to the general problem may be very difficult, and may require significant tools that have yet to be developed.

Perhaps the most interesting question regarding this problem is whether Polytope Codes can achieve capacity for general networks, or at least for all one-source one-destination problems (or perhaps even multicast). As far as we know, they are the best known strategy for defeating adversarial attacks on network coding, as they do at least as well as linear codes, which are used to solve most problems.

6.2 Multiterminal Source Coding

The results of Chapter 3 find tight bounds on the set of achievable rates for various forms of the Slepian-Wolf problem. Therefore we do not believe there is much additional work that could be done in that area. However, the more general multiterminal source coding problems studied in Chapter 4 are wide open. Much more work could be done on these problems in the presence of an adversary. One must tread carefully, however, because many multiterminal source coding problems are open even without adversaries, so there seems to be little hope to find tight results with adversaries. This was the reason that we chose problems to study in Chapter 4 that had been completely solved in the no-adversary case, in the hope that they could also be solved with adversaries. We provided bounds for these problems in Chapter 4, but did not quite solve them. We conjecture that the in-

ner bounds resulting from our Berger-Tung-like achievable scheme in Theorem 9 are tight for both the error exponent of the discrete CEO Problem, and the rate-distortion region for the quadratic Gaussian CEO Problem, but we were unable to prove either.

6.3 Power System Sensing and Estimation

Study of malicious data attacks on power systems is still in its infancy. Chapter 5 exclusively studied the effect of these attacks on state estimation. The data taken by meters in the power system is used for other things, and it may be more interesting to study the effect of malicious data attacks on these. What is primarily missing from Chapter 5 is a sense of what the *result* of these attacks are. For example, can they cause a black-out? The answer is unclear, because all we know is that they may increase the mean square error of the state estimate. How this affects the operation of the power grid depends on how the state estimate is employed to make decisions at the control center. Indeed, it is often the case that control decisions are made directly from measurements, without being processed by the state estimator; this could induce further dangers if corrupted measurements are not even corroborated against other measurements.

Another application of power measurements relates to the pricing of power in the network. If measurements strongly influence the compensation of generators, there may be a strong economic incentive to manipulate them to one's own advantage.

Finally, phasor measurement units (PMUs) are increasingly being installed at busses in the power grid [98]. These allow much more high quality measurements

of voltage levels than has been previously available, including, for the first time, phase differences between busses. How this new wealth of data may affect the problem of malicious data attacks is as yet unclear.

BIBLIOGRAPHY

[1] A. Koestler, *Spanish Testament*, 1937.

[2] M. Pease, S. R, and L. Lamport, "Reaching Agreement in the Presence of Faults," *Journal of the Association for Computing Machinery*, vol. 27, no. 2, April 1980.

[3] L. Lamport, "The Implementation of Reliable Distributed Multiprocess Systems," *Computer Networks*, vol. 2, pp. 95–114, 1978.

[4] J. H. Wensley, L. Lamport, J. Goldborg, M. W. Greet, K. N. Levitt, P. M. Melliar-Smith, R. E. Shostak, and C. B. Weinstock, "SIFT: Design and Analysis of a Fault-Tolerant Computer for Aircraft Control," *Proceedings of the IEEE*, vol. 66, no. 10, pp. 1240–1255, Oct. 1978.

[5] L. Lamport, R. Shostak, and M. Pease, "The Byzantine generals problem," *ACM Transactions on Programming Languages and Systems*, vol. 4, no. 1, pp. 14–30, 1982.

[6] R. W. Yeung and N. Cai, "Network error correction, part I: Basic concepts and upper bounds," *Comm. in Inf. and Syst.*, vol. 6, no. 1, pp. 19–36, 2006.

[7] N. Cai and R. W. Yeung, "Network error correction, part II: Lower bounds," *Comm. in Inf. and Syst.*, vol. 6, no. 1, pp. 37–54, 2006.

[8] L. R. Ford and D. R. Fulkerson, "Maximal flow through a network," *Canadian Journal of Mathematics*, vol. 8, pp. 399–404, 1956.

[9] R. Ahlswede, N. Cai, S. R. Li, and R. W. Yeung, "Network information flow," *IEEE Trans. Inf. Theory*, vol. 46, pp. 1204–1216, 2000.

[10] S. Li, R. Yeung, and N. Cai, "Linear network coding," *IEEE Trans. Inf. Theory*, vol. 49, no. 2, pp. 371–381, 2003.

[11] R. Koetter and M. Médard, "An algebraic approach to network coding," *IEEE/ACM Trans. Netw.*, vol. 11, no. 5, pp. 782–795, 2003.

[12] T. Ho, R. Koetter, M. Médard, D. R. Karger, and M. Effros, "The benefits of coding over routing in a randomized setting," in *Proc. Int. Symp. Information Theory*, 2003.

[13] T. Ho, M. Medard, R. Koetter, D. Karger, M. Effros, J. Shi, and B. Leong, "A Random Linear Network Coding Approach to Multicast," *IEEE Trans. Inf. Theory*, vol. 52, no. 10, pp. 4413–4430, 2006.

[14] S. Jaggi, P. Sanders, P. Chou, M. Effros, S. Egner, K. Jain, and L. Tolhuizen, "Polynomial time algorithms for multicast network code construction," *IEEE Trans. Inf. Theory*, vol. 51, no. 6, pp. 1973–1982, 2005.

[15] P. A. Chou, Y. Wu, and K. Jain, "Practical network coding," in *Proc. Allerton Conference on Communication, Control, and Computing*, 2003.

[16] Y. Zhu, B. Li, and J. Guo, "Multicast with network coding in application-layer overlay networks," *IEEE Journal on Selected Areas in Comm.*, vol. 22, no. 1, pp. 107–120, 2004.

[17] T. Noguchi, T. Matsuda, and M. Yamamoto, "Performance evaluation of new multicast architecture with network coding," *IEICE Trans. Commun*, vol. 86, pp. 1788–1795, 2002.

[18] S. Katti, H. Rahul, W. Hu, D. Katabi, M. Médard, and J. Crowcroft, "XORs in the Air: Practical Wireless Network Coding," *IEEE/ACM Transactions on Networking*, vol. 16, no. 3, pp. 497–510, 2008.

[19] A. R. Lehman and E. Lehman, "Complexity classification of network information flow problems," in *Proc. Symp. Discrete Algorithms*, New Orleans, LA, 2004, pp. 142–150.

[20] M. Médard, M. Effros, T. Ho, and D. R. Karger, "On coding for nonmulticast networks," in *Proc. Allerton Conference on Communication, Control, and Computing*, 2003.

[21] R. Dougherty, C. Freiling, and K. Zeger, "Insufficiency of linear coding in network information flow," *IEEE Trans. Inf. Theory*, vol. 51, no. 8, pp. 2745–2759, 2005.

[22] R. W. Yeung, "A framework for linear information inequalities," *IEEE Trans. Inf. Theory*, vol. 43, no. 6, pp. 1924–1934, 1997.

[23] Z. Zhang and R. W. Yeung, "On characterization of entropy function via information inequalities," *IEEE Trans. Inf. Theory*, vol. 44, no. 4, pp. 1440–1452, 1998.

[24] R. Dougherty, C. Freiling, and K. Zeger, "Networks, Matroids, and Non-Shannon Information Inequalities," *IEEE Trans. Inf. Theory*, vol. 53, no. 6, pp. 1949–1969, 2007.

[25] T. H. Chan and R. W. Yeung, "On a relation between information inequalities and group theory," *IEEE Trans. Inf. Theory*, vol. 48, no. 7, pp. 1992–1995, 2002.

[26] N. Cai and R. W. Yeung, "Secure network coding," in *Proc. Int. Symp. Information Theory*, 2002.

[27] C. E. Shannon, "Communication theory of secrecy systems," *Bell System Technical Journal*, vol. 28, pp. 656–715, 1948.

[28] A. Wyner, "The wiretap channel," *Bell System Technical Journal*, vol. 54, pp. 1355–1387, 1975.

[29] T. Cui, T. Ho, and J. Kliewer, "Achievable Strategies for General Secure Network Coding," in *Information Theory and Applications Workshop*, 2010.

[30] T. Ho, B. Leong, R. Koetter, M. Médard, M. Effros, , and D. R. Korger, "Byzantine modification detection in multicast networks using randomized network coding," in *Proc. Int. Symp. Information Theory*, 2004.

[31] S. Jaggi, M. Langberg, S. Katti, T. Ho, D. Katabi, and M. Médard, "Resilient network coding in the presence of Byzantine adversaries," in *Proc. INFOCOM*, 2007, pp. 616–624.

[32] R. Koetter and F. R. Kschischang, "Coding for errors and erasures in random network coding," *IEEE Trans. Inf. Theory*, vol. 54, no. 8, pp. 3579–3591, 2008.

[33] M. Kim, M. Médard, J. Barros, and R. Koetter, "An algebraic watchdog for wireless network coding," in *Proc. Int. Symp. Information Theory*, July 2009.

[34] G. Liang and N. H. Vaidya, "When watchdog meets coding," in *Proc. INFOCOM*, San Diego, CA, 2010.

[35] O. Kosut, L. Tong, and D. Tse, "Nonlinear network coding is necessary to combat general Byzantine attacks," in *Proc. Allerton Conference on Communication, Control, and Computing*, Sep. 2009.

[36] ——, "Polytope codes against adversaries in networks," in *Proc. Int. Symp. Information Theory*, 2010.

[37] S. Kim, T. Ho, M. Effros, and S. Avestimehr, "Network error correction with unequal link capacities," in *Proc. Allerton Conference on Communication, Control, and Computing*, Sep. 2009.

[38] ——, "New results on network error correction: capacities and upper bounds," in *Information Theory and Applications Workshop*, 2010.

[39] R. Singleton, "Maximum distance q -nary codes," *Information Theory, IEEE Transactions on*, vol. 10, no. 2, pp. 116 – 118, apr 1964.

[40] D. Slepian and J. Wolf, "Noiseless coding of correlated information sources," *IEEE Trans. Inf. Theory*, vol. 19, no. 4, pp. 471–480, 1973.

[41] T. M. Cover, "A proof of the data compression theorem of Slepian and Wolf for ergodic sources," *IEEE Trans. Inf. Theory*, vol. 21, no. 2, pp. 226–228, 1975.

[42] A. D. Wyner, "On Source Coding with Side Information at the Decoder," *IEEE Trans. Inf. Theory*, vol. 21, no. 3, pp. 294–300, 1975.

[43] R. Ahlswede and J. Körner, "Source Coding with Side Information and a Converse for Degraded Broadcast Channels," *IEEE Trans. Inf. Theory*, vol. 21, no. 6, pp. 629–637, 1975.

[44] A. Wyner and J. Ziv, "The rate-distortion function for source coding with side information at the decoder," *IEEE Trans. Inf. Theory*, vol. 22, no. 1, pp. 1–10, 1976.

[45] T. Berger, *The Information Theory Approach to Communications*, G. Longo, Ed. Springer-Verlag, 1978.

[46] S. Y. Tung, "Multiterminal source coding," Ph.D. dissertation, Cornell University, Ithaca, NY, 1978.

[47] J. Korner and K. Marton, "How to encode the modulo-two sum of binary sources (corresp.)," *Information Theory, IEEE Transactions on*, vol. 25, no. 2, pp. 219 – 221, mar 1979.

[48] T. Berger, Z. Zhang, and H. Viswanathan, "The CEO problem [multiterminal source coding]," *IEEE Trans. Inf. Theory*, vol. 42, no. 3, pp. 887–902, 1996.

[49] C. E. Shannon, "A Mathematical Theory of Communication," *Bell System Technical Journal*, vol. 27, pp. 379–423, 623–656, 1948.

[50] A. Stam, "Some inequalities satisfied by the quantities of information of Fisher and Shannon," *Inf. Control*, vol. 2, pp. 101–112, 1959.

[51] N. Blachman, "The convolution inequality for entropy powers," *IEEE Trans. Inf. Theory*, vol. 11, pp. 267–271, 1965.

[52] A. Wagner, S. Tavildar, and P. Viswanath, "Rate region of the quadratic gaussian two-encoder source-coding problem," *Information Theory, IEEE Transactions on*, vol. 54, no. 5, pp. 1938 –1961, may 2008.

[53] H. Viswanathan and T. Berger, "The quadratic Gaussian CEO problem," *IEEE Trans. Inf. Theory*, vol. 43, pp. 1549–1559, 1997.

[54] Y. Oohama, "The rate-distortion function for the quadratic Gaussian CEO problem," *IEEE Trans. Inf. Theory*, vol. 44, pp. 55–67, 1998.

[55] ——, "Rate-distortion theory for Gaussian multiterminal source coding systems with several side informations at the decoder," *IEEE Trans. Inf. Theory*, vol. 51, pp. 2577–2593, 2005.

[56] V. Prabhakaran, D. Tse, and K. Ramchandran, "Rate region of the quadratic Gaussian CEO problem," in *Proc. Int. Symp. Information Theory*, 2004.

[57] L. Ozarow, "On a Source Coding Problem with Two Channels and Three Receivers," *Bell System Technical Journal*, vol. 59, no. 1909–1921, 1980.

[58] H. Witsenhausen, "On source networks with minimal breakdown degradation," *Bell Syst. Tech. J.*, vol. 59, no. 6, pp. 1083–1087, 1980.

[59] J. Wolf, A. Wyner, and J. Ziv, "Source coding for multiple descriptions," *Bell Syst. Tech. J.*, vol. 59, no. 8, pp. 1417–1426, 1980.

[60] H. S. Witsenhausen and A. D. Wyner, "Source coding for multiple descriptions II: A binary source," *Bell Syst. Tech. J.*, vol. 60, pp. 2281–2292, 1981.

[61] A. A. E. Gamal and T. M. Cover, "Achievable rates for multiple descriptions," *IEEE Trans. Inf. Theory*, vol. 28, no. 6, pp. 851–857, 1982.

[62] R. Ahlswede, "The rate-distortion region for multiple descriptions without excess rate," *IEEE Trans. Inf. Theory*, vol. 31, no. 6, pp. 721–726, 1985.

[63] J. Chen and T. Berger, "Robust Distributed Source Coding," *IEEE Trans. Inf. Theory*, vol. 54, pp. 3385–3398, 2008.

[64] O. Kosut and L. Tong, "Variable-rate distributed source coding in the presence of byzantine sensors," in *Information Theory, 2007. ISIT 2007. IEEE International Symposium on*, 24-29 2007, pp. 2121 –2125.

[65] ——, "Distributed source coding in the presence of Byzantine sensors," *IEEE Trans. Inf. Theory*, vol. 54, no. 6, pp. 2550–2565, 2008.

[66] ——, "The byzantine ceo problem," in *Information Theory, 2008. ISIT 2008. IEEE International Symposium on*, 6-11 2008, pp. 46 –50.

[67] ——, "A characterization of the error exponent for the Byzantine CEO problem," in *Proc. Allerton Conference on Communication, Control, and Computing*, 2008.

[68] ——, "The quadratic gaussian ceo problem with byzantine agents," in *Information Theory, 2009. ISIT 2009. IEEE International Symposium on*, june 2009, pp. 1145 –1149.

[69] F. C. Schweppe, J. Wildes, and D. P. Rom, "Power system static state estimation, Parts I, II, III," *IEEE Tran. on Power Appar. & Syst.*, vol. 89, pp. 120–135, 1970.

[70] E. Handschin, F. C. Schweppe, J. Kohlas, and A. Fiechter, "Bad data analysis for power system state estimation," *IEEE Trans. Power Apparatus and Systems*, vol. 94, pp. 329–337, 1975.

[71] A. Garcia, A. Monticelli, and P. Abreu, "Fast decoupled state estimation and bad data processing," *IEEE Transactions on Power Apparatus and Systems*, vol. 98, no. 5, pp. 1645–1652, September 1979.

[72] T. V. Cutsem, M. Ribbens-Pavella, and L. Mili, "Bad Data Identification Methods In Power System State Estimation—A Comparative Study," *IEEE*

Trans. on Power Apparatus and Systems, vol. 104, no. 11, pp. 3037–3049, November 1985.

[73] H.-J. Koglin, T. Neisius, G. Beissler, and K. D. Schmitt, "Bad data detection and identification," *Int. J. Elect. Power*, vol. 12, no. 2, pp. 94–103, April 1990.

[74] J. Chen and A. Abur, "Improved bad data processing via strategic placement of PMUs," *IEEE Power Engineering Society General Meeting*, vol. 1, pp. 509–513, June 2005.

[75] A. Monticelli and F. Wu, "Network observability: Theory," *IEEE Trans. Power Apparatus and Systems*, vol. 104, pp. 1042–1048, 1985.

[76] F. F. Wu and W. E. Liu, "Detection of topology errors by state estimation," *IEEE Trans. Power Systems*, vol. 4, pp. 176–183, 1989.

[77] G. R. Krumpholz, K. A. Clements, and P. W. Davis, "Power system observability: a practical algorithm using network topology," *IEEE Trans. Power Apparatus and Systems*, vol. 99, no. 4, pp. 1534–1542, July 1980.

[78] Y. Liu, P. Ning, and M. K. Reiter, "False data injection attacks against state estimation in electric power grids," in *ACM Conference on Computer and Communications Security*, 2009, pp. 21–32.

[79] D. Gorinevsky, S. Boyd, and S. Poll, "Estimation of faults in DC electrical power systems," in *Proc. 2009 American Control Conf.*, St. Louis, MO, June 2009, pp. 4334–4339.

[80] D. Donoho, "Compressed sensing," *IEEE Trans. Inf. Theory*, vol. 52, no. 4, pp. 1289–1306, 2006.

[81] O. Kosut, L. Jia, R. J. Thomas, and L. Tong, "Limiting false data attacks on power system state estimation," in *Proc. 2010 Conference on Information Sciences and Systems*, March 2010.

[82] ——, "On false data attacks on power system state estimation," in *Proc. 45th International Universities' Power Engineering Conference*, 2010.

[83] ——, "Malicious Data Attacks on Smart Grid State Estimation: Attack Strategies and Countermeasures," in *IEEE SmartGridComm [submitted]*, 2010.

[84] A. S. M. Grötschel, L. Lovász, "The ellipsoid method and its consequences in combinatorial optimization," *Combinatorica*, vol. 1, no. 2, pp. 169–197, June 1981.

[85] W. H. Cunningham, "On submodular function minimization," *Combinatorica*, vol. 5, no. 3, pp. 185–192, September 1985.

[86] A. Schrijver, "A combinatorial algorithm minimizing submodular functions in strongly polynomial time," *Journal of Combinatorial Theory Series B*, vol. 80, no. 2, pp. 346–355, November 2000.

[87] O. Zeitouni, J. Zi, and N. Merhav, "When is the generalized likelihood ratio test optimal," *IEEE Trans. Inf. Theory*, vol. 38, pp. 1597–1602, 1991.

[88] B. C. Levy, *Principles of Signal Detection and Parameter Estimation.* Springer, 2008.

[89] A. Dembo and O. Zeitouni, *Large Deviations Techniques and Applications, 2nd ed.* Springer, 1998.

[90] F. Harary, *Graph Theory.* Addison-Wesley Publishing Company, 1972.

[91] A. Wyner, "The common information of two dependent random variables," *IEEE Trans. Inf. Theory*, vol. 21, no. 2, pp. 163–179, 1975.

[92] T. M. Cover and J. Thomas, *Elements of Information Theory.* John Wiley, 1991.

[93] P. Viswanath, "Sum rate of a class of Gaussian multiterminal source coding problems," *Advances in Network Information Theory*, vol. 66, pp. 43–60, 2004.

[94] A. Monticelli, "Electric power system state estimation," *Proceedings of the IEEE*, vol. 88, no. 2, pp. 262–282, 2000.

[95] J. D. Glover, M. S. Sarma, and T. J. Overbye, *Power system analysis and design.* Cengage Learning, 2008.

[96] S. Kourouklis, "A large deviation result for the likelihood ratio statistic in exponential families," *The Annals of Statistics*, vol. 12, no. 4, pp. 1510–1521, 1984.

[97] G. Golub and C. V. Loan, *Matrix Computations*. The Johns Hopkins University Press, 1990.

[98] A. Phadke, J. S. Thorp, R. F. Nuqui, and M. Zhou, "Recent developments in state estimation with phasor measurements," in *Proc. IEEE Power Sys. Conf. Exposition*, March 2009, pp. 1–7.