3-25-2014

Foundations of Quantitative Information Flow: Channels, Cascades, and the Information Order

Barbara Espinoza Becerra
Florida International University, bespi009@fiu.edu

DOI: 10.25148/etd.FI14040824

FLORIDA INTERNATIONAL UNIVERSITY

Miami, Florida

FOUNDATIONS OF QUANTITATIVE INFORMATION FLOW: CHANNELS, CASCADES, AND THE INFORMATION ORDER

A dissertation submitted in partial fulfillment of the

requirements for the degree of

DOCTOR OF PHILOSOPHY

in

COMPUTER SCIENCE

by

Barbara Espinoza Becerra

2014

To: Dean Amir Mirmiran
College of Engineering and Computing

This dissertation, written by Barbara Espinoza Becerra, and entitled Foundations of Quantitative Information Flow:
Channels, Cascades, and the Information Order, having been approved in respect to style and intellectual content, is referred to you for judgment.

We have read this dissertation and recommend that it be approved.

Bogdan Carbunar

Peter J. Clarke

Dev K. Roy

Jinpeng Wei

Geoffrey Smith, Major Professor

Date of Defense: March, 25 2014

The dissertation of Barbara Espinoza Becerra is approved.

Dean Amir Mirmiran
College of Engineering and Computing

Dean Lakshmi N. Reddi
University Graduate School

Florida International University, 2014

DEDICATION

To my family.

ACKNOWLEDGMENTS

First of all, I thank my advisor Geoffrey Smith for guidance, patience, encouragement, and countless fascinating discussions during my years as a PhD student. I am particularly grateful to him for his continuous support that allowed me to bring this thesis to completion while keeping a balanced personal life.

I thank Prof. Giri Narasimhan and Prof. Tao Li for their guidance in the solution of the channel matrix factorization problem.

I should also thank the members of the INRIA associated team PRINCESS ("Privacy in Data Access"), for their feedback and collaboration.

The members of my dissertation committee, Bogdan Carbunar, Peter J. Clarke, Dev K. Roy, and Jinpeng Wei, provided insightful comments and suggestions that improved the presentation of this thesis.

I also wish to thank the IT and office support staff of the department, especially Olga Carbonell, for assisting me with all the required paperwork.

I am especially grateful to my husband, Luis Useche, for all the encouragement, understanding, technical feedback, and for being my best advocate through all these years. I also thank my family, particularly my mother, for always believing in me.

I finally thank my professors at Simón Bolívar University, especially Prof. Jesús Ravelo, for providing me with a world-class education and cultivating in me an appreciation for elegant solutions to mathematical and algorithmic problems.

This work was partially supported by the National Science Foundation under grants CNS-0831114 and CNS-1116318.

ABSTRACT OF THE DISSERTATION

FOUNDATIONS OF QUANTITATIVE INFORMATION FLOW: CHANNELS, CASCADES, AND THE INFORMATION ORDER

by

Barbara Espinoza Becerra

Florida International University, 2014

Miami, Florida

Professor Geoffrey Smith, Major Professor

Secrecy is fundamental to computer security, but real systems often cannot avoid leaking some secret information. For this reason, the past decade has seen growing interest in quantitative theories of information flow that allow us to quantify the information being leaked. Within these theories, the system is modeled as an information-theoretic channel that specifies the probability of each output, given each input. Given a prior distribution on those inputs, entropy-like measures quantify the amount of information leakage caused by the channel.

This thesis presents new results in the theory of min-entropy leakage. First, we study the perspective of secrecy as a resource that is gradually consumed by a system. We explore this intuition through various models of min-entropy consumption. Next, we consider several composition operators that allow smaller systems to be combined into larger systems, and explore the extent to which the leakage of a combined system is constrained by the leakage of its constituents. Most significantly, we prove upper bounds on the leakage of a cascade of two channels, where the output of the first channel is used as input to the second. In addition, we show how to decompose a channel into a cascade of channels.

We also establish fundamental new results about the recently-proposed g-leakage family of measures. These results further highlight the significance of channel cascading. We prove that whenever channel A is composition refined by channel B, that is, whenever A is the cascade of B and R for some channel R, the leakage of A never exceeds that of B, regardless of the prior distribution or leakage measure (Shannon leakage, guessing entropy leakage, min-entropy leakage, or g-leakage). Moreover, we show that composition refinement is a partial order if we quotient away channel structure that is redundant with respect to leakage alone. These results are strengthened by the proof that composition refinement is the only way for one channel to never leak more than another with respect to g-leakage. Therefore, composition refinement robustly answers the question of when a channel is always at least as secure as another from a leakage point of view.

TABLE OF CONTENTS

CHAPTER PAGE

1. INTRODUCTION . . . 1
1.1 Quantitative Information Flow . . . 1
1.2 Research Goals . . . 3
1.3 Motivation, Contribution, and Outline . . . 4
1.4 Publications . . . 8

2. PRELIMINARIES . . . 9
2.1 Channels . . . 9
2.2 Measures of Information Flow . . . 13
2.2.1 Shannon Leakage . . . 13
2.2.2 Guessing Entropy Leakage . . . 15
2.2.3 Min-Entropy Leakage . . . 16
2.2.4 g-Leakage . . . 21
2.3 Order Relations . . . 22
2.4 Partition Refinement . . . 23
2.5 Linear Algebra . . . 25

3. MIN-ENTROPY AS A RESOURCE . . . 28
3.1 Creation of Min-Entropy . . . 30
3.2 Dynamic Min-Entropy Leakage . . . 31
3.3 Worst-Case Min-Entropy Leakage . . . 33
3.4 Average-Case Min-Entropy Leakage . . . 35
3.5 A Case Study: the Crowds Protocol . . . 37
3.6 Related Work . . . 40
3.7 Summary . . . 42
3.8 Credits . . . 43

4. MIN-ENTROPY LEAKAGE OF COMBINED CHANNELS . . . 44
4.1 Leakage of a Cascade of Channels . . . 45
4.1.1 Foundations of Cascades of Channels . . . 46
4.1.2 Min-Entropy Leakage of a Cascade of Channels . . . 53
4.1.3 Min-Capacity of a Cascade of Channels . . . 60
4.2 Leakage of a Repeated Independent Runs Channel . . . 62
4.3 Leakage of an Adaptive Composition of Channels . . . 68
4.4 Related Work . . . 72
4.5 Summary . . . 74
4.6 Credits . . . 75

5. ABSTRACT CHANNELS AND THEIR LEAKAGE ORDERING 76
5.1 Abstract Channels 79
5.2 Structural Ordering 84
5.3 Equivalence of the Structural and Leakage Orderings 96
5.4 Relationship to Other Leakage Orderings 101
5.4.1 Guessing Entropy Leakage 102
5.4.2 Shannon leakage 106
5.4.3 Min-Entropy Leakage 107
5.5 Expressing Other Leakage Measures as g-Leakages 108
5.5.1 Guessing Entropy Leakage as a g-Leakage 109
5.5.2 Shannon Leakage as a g-Leakage 110
5.6 Compositionality of Abstract Channels 112
5.7 Related Work 115
5.8 Summary 116
5.9 Credits 117

6. CHANNEL MATRIX FACTORIZATION 119
6.1 Channel Matrix Factorization 121
6.2 Non-Negative Matrix Factorization Characterization 123
6.3 Algorithms for Solving NMF with the Generalized KL Divergence 129
6.4 Applicability 131
6.5 Related Work 133
6.6 Summary 134
6.7 Credits 134

7. CONCLUSION 136

BIBLIOGRAPHY 141

VITA 149

LIST OF FIGURES

FIGURE PAGE

2.1 The partition induced by C_{state} refines that of $C_{country}$ 25

2.2 Lattice of Information for $\mathcal{X} = \{x_1, x_2, x_3\}$ 25

3.1 Crowds protocol 29

4.1 Cascade of channels A and B 46

4.2 Repeated independent runs of channel C 63

4.3 Adaptive channel composition A + B 69

6.1 D(A||BR) for each factorization rank of a channel matrix A(20 × 20). . . . 132

CHAPTER 1

INTRODUCTION

In this chapter we introduce the research area of quantitative information flow, present our research goals, describe the motivation and significance of this thesis, delineate our most important contributions, provide an outline for the dissertation, and enumerate the scientific publications where we have previously presented this research.

1.1 Quantitative Information Flow

Protecting confidential information from improper disclosure is a fundamental security goal, made more challenging due to the practical difficulty of preventing all leakage of secret information. As a basic example, a login program that rejects an incorrect password unavoidably reveals that the secret password differs from the one that was entered. Similarly, revealing the tally of votes in an election leaks some information about the secret ballots—if the election happens to be unanimous, for example, then we learn what *all* of the ballots were. More subtly, an adversary may be able to observe *side information* about a system's implementation that may leak secret information. For example, Kocher's celebrated timing attack on RSA [Koc96] demonstrates that the timings of a set of RSA decryptions can allow an adversary to deduce the private key.

One promising way to address information leakage is to consider it *quantitatively*, based on the intuition that a login program is acceptable in practice because it leaks only a "small" amount of information about the secret password. This viewpoint has led to the area of *quantitative information flow*, which has seen growing interest in the past decade. (See, for example, [CHM01, Bor06, Mal07, KB07, CPP08a, CPP08b, Smi09, KS10].)

The notion of *channel* from information theory [Sha48] provides a general framework for quantifying the transmission of information in systems. Channels capture the relationship between the inputs and the outputs of a system through a channel matrix which specifies, for each input, the conditional probability of observing each output of the system. Given a channel, it is then natural to measure the leakage of confidential information based on how much the adversary's *uncertainty* with respect to the secret input is reduced after observing the channel's public output:

$$\text{leakage} = \text{initial uncertainty} - \text{remaining uncertainty}.$$

The information flow community is currently studying a variety of theories of information flow, each of them characterized by the choice of uncertainty measure. The first uncertainty measures to be considered were *Shannon entropy* and *guessing entropy*. Shannon entropy [Sha48] is a classic measure from information theory that quantifies the average information content of a random variable. The Shannon entropy of a random variable X can be understood as the expected number of bits required to transmit X using an optimal encoding scheme. Guessing entropy [Mas94], on the other hand, quantifies the expected number of guesses that an adversary would have to make to correctly guess the value of the secret. Both measures have been used to quantify information flow in a variety of scenarios, including programs that handle sensitive information (e.g. [CHM01]), side-channel attacks (e.g. [KB07]), and even anonymity protocols (e.g. [CPP08a]); all of which can be modeled as channels.

Another theory of quantitative information flow that has received considerable attention recently [Smi09, BCP09, APvRS10, HSP10, AAP10b] is based on measuring uncertainty using Rényi's *min-entropy* [Rén61], rather than guessing entropy or Shannon entropy. An advantage of min-entropy leakage is that it is based directly on the secret's *vulnerability* to being guessed in one try by an adversary, resulting in stronger operational security guarantees than are obtained with Shannon entropy [Smi09]. In contrast, Shannon entropy and guessing entropy provide guarantees in terms of the expected number of guesses that the adversary has to make, which can be arbitrarily high even if there is a high probability that the adversary will make the correct guess at the first attempt.

More recently, *g-leakage* [ACPS12], a generalization of min-entropy leakage, has been proposed to allow for a wider variety of operational scenarios to be modeled. With g-leakage, the benefit that an adversary obtains from making a particular guess is specified with a *gain function*. This allows to model adversaries that benefit not just from guessing the complete secret, but also from guessing values close to the secret, that are part of the secret, that are properties of the secret, or even guessing the secret within a number of tries. (We review the most commonly discussed measures of leakage in Section 2.2.)

1.2 Research Goals

This thesis advances the current understanding of quantitative theories of information flow by filling knowledge gaps, re-examining the rationale behind current models, and providing new perspectives for the understanding of quantitative information flow analysis. The main research problems we address in this thesis are:

1. Exploring the perspective that secrecy can be viewed as a resource that is gradually consumed by a system.
2. Analyzing the information flow of combined channels.
3. Determining the conditions under which a channel always leaks more information than another.

4. Studying techniques for factoring a channel matrix into the product of channel matrices.

1.3 Motivation, Contribution, and Outline

This thesis begins with a review of important background concepts on Chapter 2. We describe information-theoretic channels, review a variety of measures of information flow, basic notions of order theory, and define an order relation for deterministic channels (those channels in which each input maps to exactly one output) known as partition refinement.

We start our study in Chapter 3 by noticing that the initial uncertainty or secrecy of an input can be viewed as a resource that is gradually consumed by the system execution. After choosing min-entropy as our measure of secrecy, we identify three different models for its consumption: a new dynamic model of min-entropy leakage that quantifies the information flow in a single run of the system, a new worst-case run model, and Smith's average-case run model [Smi09]. Our results show that the min-entropy leakage of a single run of the system can be negative, so min-entropy does not behave as a reasonable resource in this case. In contrast, both worst-case and average-case min-entropy leakage are always non-negative, so both of these models are well suited for the viewpoint of min-entropy as a resource. However, caution should be observed with the worst-case model, as it is overly sensitive to unlikely "bad" outputs of the system. We conclude the chapter with a quantitative information flow analysis of the Crowds anonymity protocol with respect to both worst-case and average-case min-entropy. This case study due to Smith [ES13] demonstrates the importance of choosing leakage models based on the characteristics of the scenario being studied.

The viewpoint of secrecy as a resource naturally leads us to consider its consumption when multiple channels are combined. Accordingly, in Chapter 4 we turn our attention to the min-entropy leakage associated to combinations of channels. We first look at the leakage of a cascade of channels. A cascade [Des53, Abr63] is a classic construction on two channels, where the output of the first channel is used as input to the second. Our main contribution with respect to this subject is proving that the information flow of a cascade of channels cannot exceed the flow of the first channel, that is, the first channel in the cascade behaves as a bottleneck. This property is a min-entropy analogue to the classic *data-processing inequality* [CT06, p. 34] for Shannon entropy leakage. Curiously, we found that such upper bound does not hold with respect to the second channel of the cascade. However, we found that when we turn to the min-entropy capacity of a cascade, that is, the maximum min-entropy leakage among all possible distributions of the secret input, both channels of the cascade behave as bottlenecks for the information flow. In addition to studying cascading, we discuss and present refinements to other channel composition operators that have previously appeared in the literature. Specifically, we study the min-entropy leakage when repeated independent runs of a channel are allowed, and the leakage in an adaptive composition of two channels where the output of both channels is public and the second channel receives as input both the input and the output of the first channel.

Establishing bounds on the leakage of combined channels based on the leakage of their constituents is a good starting point towards developing compositional analysis and design techniques for secure programs. But another useful research direction towards this goal is the study of leakage ordering relations of channels. In particular, knowing that a channel is always more secure than another is essential if we aim to develop secure software through stepwise refinement where, at each refinement step, the implementation must be guaranteed to be at least as secure as the specification.

Note that, in general, the leakage ordering of two channels A and B (both taking X as input) varies with the choice of leakage measure and the distribution of the secret input *prior* to the system execution. For example, if an adversary that tries to guess the secret input knows that the input can only take one particular value, then the channel cannot possibly leak any additional information. Therefore, it is interesting and useful to determine the conditions under which channels satisfy a robust leakage ordering relation, that is, a leakage ordering that is independent of the prior distribution and the leakage measure.

We study leakage ordering relations of channels in Chapter 5. Our results show that whenever a channel A is equivalent to a channel B followed by post-processing with some channel R, that is, whenever A is the cascade of channels B and R, then the leakage of channel A cannot exceed the leakage of channel B for any prior distribution or any of the leakage measures that we have mentioned: Shannon leakage, guessing-entropy leakage, min-entropy leakage, or g-leakage. Following [ACPS12] we call this relation composition refinement, and say that channel A is composition refined by channel B. Our main result with respect to this topic is the proof that composition refinement is in fact a partial order relation on channels provided that we quotient away redundant information contained in the channel structure, such as duplication, scaling or permutation of columns. Moreover, we explain that composition refinement is the only way for a channel to always leak more information than another with respect to g-leakage, a result that was first proved by McIver et al. [MMM12].

These results combined indicate that composition refinement is an order relation on channels with both structural and leakage characterizations. A similar relation on deterministic channels called *partition refinement* was previously studied by Landauer and Redmond [LR93]. As we explain in Chapter 2, partition refinement, is an order relation on deterministic channels such that, whenever channel A is partition refined by channel B, A is guaranteed to never leak more information than B on any prior distribution, and under of Shannon leakage, guessing-entropy leakage, min-entropy leakage, or g-leakage. Similar to composition refinement, partition refinement induces a partial order on channels (in fact it induces a lattice called the Lattice of Information) when we abstract away the redundant information contained in channels. Moreover, partition refinement and composition refinement coincide on deterministic channels [ACPS12]. Hence, our results imply that composition refinement can be viewed as a generalization of partition refinement from deterministic to probabilistic channels.

The importance of cascading and composition refinement within the area of quantitative information flow leads us to explore, in Chapter 6, techniques for decomposing a channel into a cascade of channels. More specifically, our main contribution in this chapter is a procedure for *approximately* factoring a channel matrix into the product of two channel matrices. This procedure is derived from a previous result from Ho and Van Dooren [HvD08], and relies on already existing algorithms for factoring non-negative matrices into the product of non-negative matrices. Channel matrix factorization can be applied in a variety of scenarios including finding channels that composition refine a particular channel, or even statistical disclosure control of sensitive data sets.

This thesis concludes with Chapter 7, where we present a discussion of our results and suggest future research directions.

1.4 Publications

Most of the material presented in this thesis has appeared previously in the following peer-reviewed publications and presentations:

- Barbara Espinoza and Geoffrey Smith. *Min-entropy Leakage of Channels in Cascade.* In Gilles Barthe, Anupam Datta, and Sandro Etalle, editors, *Formal Aspects of Security and Trust*, volume 7140 of *Lecture Notes in Computer Science*, pages 70-84. Springer Berlin Heidelberg, 2012. [ES12]
- Barbara Espinoza and Geoffrey Smith. *Min-entropy as a Resource.* In *Special Issue: Information Security as a Resource*, volume 226 of *Information and Computation*, pages 57-75. May 2013. [ES13]
- Barbara Espinoza and Geoffrey Smith. *Channels and the Information Order (Poster).* Presented at the *34th IEEE Symposium on Security and Privacy (Oakland 2013).* San Francisco, California, May 20, 2013.
- Barbara Espinoza, Annabelle McIver, Larissa Meinicke, Carroll Morgan and Geoffrey Smith. *Abstract Channels, Gain Functions and the Information Order (Abstract).* Presented at the *Workshop on Foundations of Computer Security.* Tulane University, New Orleans, Louisiana, June 29, 2013.
- Annabelle McIver, Carroll Morgan, Geoffrey Smith, Barbara Espinoza, and Larissa Meinicke. *Abstract Channels and their Robust Information-Leakage Ordering.* To appear in proceedings of the *3rd Conference on Principles of Security and Trust (POST 2014).* Grenoble, France, April 2014. [MMS+14]

Each core chapter of this thesis includes a credits section describing the contributions of my co-authors for that particular chapter.

CHAPTER 2

PRELIMINARIES

In this chapter we review the concept of channel, a variety of measures of information flow discussed in the literature, basic notions of order theory, the partition refinement relation, and useful properties of linear algebra.

2.1 Channels

Throughout this thesis, we model probabilistic systems as information-theoretic channels [Sha48] that receive a secret input and produce an observable output with some probability. In particular, whenever we use the term channel we will be referring to discrete memoryless channels [Mac03].

Formally, a *channel* is a triple $(\mathcal{X}, \mathcal{Y}, C)$, where $\mathcal{X}$ is a finite set of secret input values, $\mathcal{Y}$ is a finite set of observable output values, and C is a $|\mathcal{X}| \times |\mathcal{Y}|$ matrix, called the *channel matrix*. The intent is that $C[x,y]$ is the conditional probability[1] $p(y|x)$ of obtaining output y when the input is x. Note that each entry of C is between 0 and 1, and each row sums to 1:

$$\text{for every } x \in \mathcal{X},\ \sum_{y} C[x,y] = 1. \tag{2.1}$$

An important special case is a *deterministic channel*, where each input yields a unique output. In terms of C, this means that each entry is either 0 or 1, and each row contains exactly one 1.

[1] Recall that in traditional probability theory, conditional probabilities are defined in terms of joint distributions. So, in the absence of a joint distribution, how can we speak of C as giving the conditional probabilities $p(y|x)$? We believe that it is actually best to view these conditional probabilities as a *primitive notion*—they simply say that *if* the input is x, *then* output y will occur with probability $C[x,y]$. Indeed, Rényi argued that "the basic notion of probability theory should be the notion of the conditional probability of A under the condition B" [Rén70, p. 35].

Given a *prior distribution* π on $\mathcal{X}$, we can define a *joint distribution* p on $\mathcal{X} \times \mathcal{Y}$ by

$$p(x,y) = \pi[x]C[x,y]. \tag{2.2}$$

This gives jointly distributed random variables X and Y with marginal probabilities $p(x) = \sum_y p(x,y)$ and $p(y) = \sum_x p(x,y)$ respectively, and conditional probabilities

$$p(y|x) = \frac{p(x,y)}{p(x)},$$

provided that $p(x)$ is nonzero.

Notice that after observing output y of the channel, an adversary $\mathcal{A}$ usually doesn't know for certain what the secret input was. However, if we make the worst-case assumption that $\mathcal{A}$ knows the prior distribution π and the channel matrix C, then $\mathcal{A}$ can infer the posterior probabilities $p(x|y)$ of input x given the observation of output y that we can infer using *Bayes's theorem* as follows:

$$p(x|y) = \frac{p(y|x)p(x)}{p(y)} = \frac{p(x,y)}{p(y)}.$$

For every $y \in \mathcal{Y}$ (with $p(y)$ nonzero), these probabilities can be organized into a *posterior distribution* $p_{X|y}$. The posterior distribution $p_{X|y}$ gives $\mathcal{A}$'s updated knowledge about X if it sees output y. Note that we will write subscripts on p when necessary to avoid ambiguity, e.g. p_{XY} or p_Y.

Example 2.1.1. *Consider channel* $(\{x_1, x_2, x_3, x_4\}, \{y_1, y_2, y_3, y_4\}, C)$*, where* C *is:*[2]

[2] Note that for clarity we represent channel matrices in tabular form, instead of their more standard notation within box brackets or parentheses. This way we can make explicit the input and output sets of the channels, which also serve as indexing sets for the matrices.

C	y_1	y_2	y_3	y_4
x_1	0	0	$1/3$	$2/3$
x_2	0	$4/5$	0	$1/5$
x_3	$4/7$	0	$2/7$	$1/7$
x_4	$4/9$	0	$4/9$	$1/9$

Then, if the prior is $\pi = (3/16, 5/16, 7/32, 9/32)$*, we get the following joint distribution matrix:*

p_{XY}	y_1	y_2	y_3	y_4
x_1	0	0	$1/16$	$1/8$
x_2	0	$1/4$	0	$1/16$
x_3	$1/8$	0	$1/16$	$1/32$
x_4	$1/8$	0	$1/8$	$1/32$

By summing the columns of the joint matrix we obtain the output distribution $p_Y = (1/4, 1/4, 1/4, 1/4)$ *which happens to be uniform in this case. And by normalizing the columns of the joint matrix we get four posterior distributions:*

$$p_{X|y_1} = (0, 0, 1/2, 1/2)$$

$$p_{X|y_2} = (0, 1, 0, 0)$$

$$p_{X|y_3} = (1/4, 0, 1/4, 1/2)$$

$$p_{X|y_4} = (1/2, 1/4, 1/8, 1/8)$$

Hence, after observing output y_2 *we know that the secret input must have been* x_2*. But, if we observe output* y_1*, all we know is that the input could have been either* x_3 *or* x_4 *each with equal probability.* □

Note that, as shown in [ES12], p is the *unique* joint distribution that recovers π by marginalization and the conditional probabilities in C whenever they are defined:

Theorem 2.1.2. *p_{XY} is the* unique *joint distribution that recovers π and C, in that $p(x) = \pi[x]$ and $p(y|x) = C[x,y]$ (if $p(x)$ is nonzero).*

Proof. First, we observe that p_{XY} recovers π by marginalization, since for any x,

$$p(x) = \sum_y p(x,y) = \sum_y \pi[x]C[x,y] = \pi[x]\sum_y C[x,y] = \pi[x].$$

From this, we also see that p_{XY} is a valid distribution, since

$$\sum_{x,y} p(x,y) = \sum_x \sum_y p(x,y) = \sum_x \pi[x] = 1.$$

Finally, p_{XY} recovers the conditional probabilities C, whenever they are defined. For if $\pi[x] \neq 0$, then for any $y \in \mathcal{Y}$,

$$p(y|x) = \frac{p(x,y)}{\pi[x]} = \frac{\pi[x]C[x,y]}{\pi[x]} = C[x,y].$$

Now, to see that there is at most one such joint distribution, note first that if $\pi[x] = 0$, then we must have $0 = p(x) = \sum_y p(x,y)$, which implies that $p(x,y) = 0$, for every $y \in \mathcal{Y}$. Second, if $\pi[x] \neq 0$, then we must have for every $y \in \mathcal{Y}$,

$$C[x,y] = p(y|x) = \frac{p(x,y)}{p(x)} = \frac{p(x,y)}{\pi[x]}$$

which implies that $p(x,y) = \pi[x]C[x,y]$. Finally, observe that these two cases can be merged into our definition: $p(x,y) = \pi[x]C[x,y]$.

□

Note that we can equivalently define p_{XY} as the product of a diagonal matrix with π on its diagonal, and C:

$$p_{XY} = \operatorname{diag}(\pi)C. \tag{2.3}$$

In the rest of this thesis we will write sometimes C as a shorthand for a channel $(\mathcal{X}, \mathcal{Y}, C)$.

2.2 Measures of Information Flow

Given a channel $(\mathcal{X}, \mathcal{Y}, C)$, we consider an adversary $\mathcal{A}$ that wishes to guess the value of X. We assume that $\mathcal{A}$ knows both the prior distribution π and the channel. It is then natural to measure the amount of information that flows from X to Y by considering the reduction in $\mathcal{A}$'s uncertainty about X after observing the value of Y, giving the following intuitive equation:

$$\text{leakage} = \text{initial uncertainty} - \text{remaining uncertainty}. \tag{2.4}$$

The different measures of leakage we now present are characterized by the choice of uncertainty measure.

2.2.1 Shannon Leakage

Shannon entropy [Sha48] is a measure of uncertainty that quantifies the expected amount of information contained in an outcome of a random variable. For Shannon entropy, the information content of an individual outcome is measured with its *self-information.* The self information of outcome x of a random variable X distributed according to π is given by:

$$I(x) = -\log \pi[x].$$

Note that events with a higher associated probability have lower self information. Intuitively, events that are certain to happen convey no information. In contrast, highly improbable events are very unexpected, so their associated information content is very high.

Hence, given prior distribution π and channel C, the prior Shannon entropy, denoted $H(\pi)$, is the expected self information for all possible values of X,

$$H(\pi) = -\sum_x \pi[x] \log \pi[x]$$

and the posterior Shannon entropy is the expected Shannon entropy of the posterior distributions,

$$H(\pi, C) = \sum_y p(y) H(p_{X|y}).$$

The Shannon leakage, better known in the information theory community as mutual information, is then the reduction in the Shannon entropy given the knowledge of C:

$$I(\pi, C) = H(\pi) - H(\pi, C).$$

Mutual information is traditionally denoted as $I(X;Y)$, using the random variables X and Y as parameters—instead of the prior distribution and the channel—to indicate that the information flows from X to Y. Similarly, the prior Shannon entropy is usually denoted as $H(X)$ and the posterior Shannon entropy as $H(X|Y)$.

An important property of mutual information is that it is symmetric, so that $I(X;Y) = I(Y;X)$, and in the case of the deterministic channels we have:

$$I(X;Y) = I(Y;X) = H(Y) - H(Y|X) = H(Y) - 0 = H(Y),$$

where the second from last step follows because in a deterministic channel the input X fully determines the output Y.

Both Shannon entropy and Shannon leakage are non-negative quantities. This implies that the knowledge of C cannot increase the uncertainty with respect to the secret: $H(\pi) \geq H(\pi, C)$. Moreover, Shannon entropy is maximized by the uniform prior distribution:

$$H(p(x_1), p(x_2), \dots, p(x_n)) \leq H\left(\frac{1}{n}, \dots, \frac{1}{n}\right) = \log n.$$

An important notion in information theory is the *channel capacity*, which is the maximum leakage over all possible prior distributions. For Shannon leakage the capacity is given by the expression $\sup_\pi I(\pi, C)$. Finding the Shannon capacity of a channel is a convex optimization problem [Mac03]. Note, however, that in the special case where C is deterministic, the channel capacity is simply the logarithm of the number of outputs of the channel[3]:

$$\sup_\pi I(\pi, C) = \sup_\pi H(Y) = \log |\mathcal{Y}|$$

2.2.2 Guessing Entropy Leakage

Guessing entropy [Mas94] quantifies uncertainty in terms of the expected number of guesses, using an optimal guessing strategy, to correctly guess the value of X. With the elements of $\mathcal{X}$ indexed in non-increasing order with respect to their probability $\pi[x]$, the prior guessing entropy is

$$G(\pi) = \sum_{i=1}^{n} i\pi[x_i].$$

The posterior guessing entropy is the expected guessing entropy of the posterior distributions:

$$G(\pi, C) = \sum_y p(y) G(p_{X|y}).$$

Then, the guessing entropy leakage is then the difference of these quantities:

$$I_G(\pi, C) = G(\pi) - G(\pi, C).$$

[3] Here we assume that all the outputs of the channel are feasible, so there must exist a prior distribution that causes the outputs to be equally likely.

2.2.3 Min-Entropy Leakage

The operational significance of both Shannon entropy and guessing entropy can be stated in terms of the expected number of guesses that the adversary would need to determine the secret [Mas94]. But the expected number of guesses can be very high even if the adversary has a high probability of guessing the secret successfully in just one try.

Example 2.2.1. *Let*

$$\pi = (1/2, 2^{-1000}, 2^{-1000}, 2^{-1000}, \ldots, 2^{-1000}).$$

Then $H(\pi) = 1/2 \log 2 + 2^{999} \cdot 2^{-1000} \cdot \log 2^{1000} = 500.5$ *bits, even though* $\mathcal{A}$ *has probability* $1/2$ *of guessing the value of* X *correctly in one try.* □

For this reason, Smith [Smi09] proposed measuring information flow with *min-entropy leakage*, which is based on the *vulnerability* of the secret to being guessed by the adversary in one try.

We distinguish between the vulnerability before and after observing the output of the channel. The former is called the prior vulnerability and defined as

$$V(\pi) = \max_x \pi[x].$$

The latter is the posterior vulnerability and is defined as the expected vulnerability after observing the output of C.

$$\begin{aligned} V(\pi, C) &= \sum_y \max_x \pi[x]C[x,y] \\ &= \sum_y \max_x p(x,y) \\ &= \sum_y p(y) \max_x p(x|y) \\ &= \sum_y p(y) V(p_{X|y}) \end{aligned}$$

We can convert from probability measures to bit measures by taking the negative logarithm. Using this method, we obtain our measures of uncertainty.

- initial uncertainty: $H_\infty(\pi) = -\log V(\pi)$.
- remaining uncertainty: $H_\infty(\pi, C) = -\log V(\pi, C)$.

In information theory, the quantity H_∞ is known as Rényi *min-entropy.* The notation $H_\infty(\pi, C)$ should then be read as the posterior min-entropy of π given the knowledge of channel C. We remark that there is no consensus in the literature with respect to what the definition of posterior min-entropy should be, so here we adopting Smith's definition [Smi09].

Substituting our uncertainty measures in equation (2.4) we can define the *min-entropy leakage*, denoted by $\mathcal{L}(\pi, C)$, [4] to be

$$\begin{aligned}\mathcal{L}(\pi, C) &= H_\infty(\pi) - H_\infty(\pi, C)\\ &= -\log V(\pi) - (-\log V(\pi, C))\\ &= \log \frac{V(\pi, C)}{V(\pi)}.\end{aligned}$$

Vulnerability is always positive. Moreover, vulnerability can only increase, in that $V(\pi, C) \geq V(\pi)$ for any prior π. Thus, min-entropy leakage is the logarithm of the factor by which knowledge of C increases the vulnerability of the secret.

Note that with min-entropy, the secrecy of the distribution in Example 2.2.1 is now $H_\infty(\pi) = -\log V(\pi) = -\log {}^1\!/_2 = 1$, a quantity that accurately reflects the large threat to the secret.

Of course min-entropy is a rather crude measure, in that it depends only on the maximum probability in π. So, for example, the min-entropy is also 1 on the distri-

[4] Note that we deviate from the notation $V(X)$, $V(X|Y)$, $H_\infty(X|Y)$, and $\mathcal{L}_{XY}$ used in [Smi09, Smi11]. Instead we follow [ACPS12] in adopting a notation that makes explicit the dependence on X's prior distribution.

bution $(1/2, 1/2)$, which is clearly less secure than $(1/2, 2^{-1000}, 2^{-1000}, 2^{-1000}, \dots, 2^{-1000})$. Still, it is reasonable to say that both are situations where there is little initial secrecy.

In the case of min-entropy leakage, we will refer to the capacity of the channel as the *min-capacity* and use the notation $\mathcal{ML}(C)$:

$$\mathcal{ML}(C) = \sup_{\pi} \mathcal{L}(\pi, C).$$

Min-capacity is always realized by a uniform distribution on X (and possibly by other distributions as well) [BCP09, KS10], and can be easily calculated as the logarithm of the sum of the column maximums in C:

Theorem 2.2.2. *$\mathcal{ML}(C) = \log \sum_y \max_x C[x, y]$, and it is realized on a uniform prior π.*

Proof. For any prior π, we have

$$\begin{aligned}
\mathcal{L}(\pi, C) &= \log \frac{V(\pi, C)}{V(\pi)} \\
&= \log \frac{\sum_y \max_x \pi[x] C[x, y]}{\max_x \pi[x]} \\
&\leq \log \frac{\sum_y \max_x (\max_x \pi[x]) C[x, y]}{\max_x \pi[x]} \\
&= \log \sum_y \max_x C[x, y]
\end{aligned}$$

The upper bound is realized when π is uniform. (It can also be realized on nonuniform π, provided that some proper subset of the rows of C includes at least one maximum from each column.) □

Theorem 2.2.2 gives two useful corollaries:

Corollary 2.2.3. *If C is deterministic, then $\mathcal{ML}(C)$ is the logarithm of the number of feasible outputs.*

Corollary 2.2.4. $\mathcal{ML}(C) = 0$ *iff the rows of* C *are identical.*

If we turn from min-capacity to min-entropy leakage, we find that leakage of 0 is more subtle. In fact, we have $\mathcal{L}(\pi, C) = 0$ whenever the adversary's best guess is unaffected by the output y. This can sometimes be surprising, as in the following example from [Smi11], which illustrates the so-called *base-rate fallacy.*

Example 2.2.5. *Suppose that* C *is the channel matrix of a good, but imperfect, test for cancer:*

C	*positive*	*negative*
cancer	0.90	0.10
no cancer	0.07	0.93

Moreover, suppose that for the population under consideration (say, age 40–50, no symptoms, no family history) the prior π *is*

$$\pi[\textit{cancer}] = 0.008 \quad \pi[\textit{no cancer}] = 0.992$$

Then, although the channel might appear to be quite reliable, we find that the min-entropy leakage is 0. For we find that the joint matrix p_{XY} *is*

p_{XY}	*positive*	*negative*
cancer	0.00720	0.00080
no cancer	0.06944	0.92256

Hence the sum of the column maximums coincides with $\pi[\textit{no cancer}]$*, since both column maximums occurs in the no cancer row. This implies that* $V(\pi) = V(\pi, C)$*, giving* $\mathcal{L}(\pi, C) = 0$*. Operationally, this reflects the fact that the adversary should* *guess no cancer, regardless of whether the test was positive or negative. (In particular, $p(cancer|positive) \approx 0.094$, which is much greater than $p(cancer) = 0.008$, but still much less than 0.500.)* □

Min-capacity is a useful measure in situations where the prior π is unknown. Moreover, because min-capacity is relatively simple to calculate, particularly in the case of deterministic channels, it may facilitate the design of practical quantitative information flow analyses. For whenever we can show that min-capacity is small, we know that min-entropy leakage must also be small, whatever the prior π may be. Furthermore, it has been shown [ACPS12], that min-capacity is also an upper bound for Shannon capacity (i.e. the maximum Shannon leakage over all priors π), a result that further highlights the significance of min-capacity as an upper bound on a channel's leakage. On the other hand, for a particular prior π we may find that $\mathcal{L}(\pi, C)$ is far less than $\mathcal{ML}(C)$. So in cases where the prior π is known, it is more precise to use the min-entropy leakage with respect to π.

We should finally emphasize that all the information flow measures that we have considered are information theoretic, rather than computational. To see this, consider the following example from [Smi11]. Let C be a channel that takes as input a uniformly-distributed 100-digit prime p, and that outputs pq, where q is a uniformly-distributed 101-digit prime. Then the posterior vulnerability $V(\pi, C) = 1$, since each column of the channel matrix has a unique nonzero entry. Hence C's min-entropy leakage exceeds 322 bits, since (by the prime number theorem) $V(\pi) < 10^{-97}$. Nevertheless, *finding* the input given the output requires factoring a very large number into the product of two roughly equally-sized primes, a problem strongly believed to be computationally hard.

2.2.4 g-Leakage

The theory of min-entropy assumes that the adversary can benefit only from exactly guessing the entire secret. With g-leakage [ACPS12], the benefit obtained by the adversary is instead modeled with a *gain* function g that returns a value between 0 and 1 to indicate the adversary's gain, given a guess and the secret's actual value. With gain functions it is possible to model a wide variety of operational scenarios, for example, we can model adversaries that benefit from guessing values close to the secret, that are part of the secret, that are properties of the secret, or even guessing the secret within a number of tries.

Formally, a gain function is a function $g : \mathcal{W} \times \mathcal{X} \to [0,1]$, such that $\mathcal{W}$ is a finite non-empty set of allowable guesses, and $\mathcal{X}$ is the set of possible secrets of the channel.

We can then consider the gain function when calculating the secret's vulnerability. This results in the prior g-vulnerability, which is the maximum expected gain over all possible guesses:

$$V_g(\pi) = \max_w \sum_x \pi[x] g(w,x).$$

Similarly, the posterior g-vulnerability, is the expected g-vulnerability of the posterior distributions:

$$\begin{aligned} V_g(\pi) &= \sum_y \max_w \sum_x \pi[x] C[x,y] g(w,x) \\ &= \sum_y \max_w \sum_x p(x,y) g(w,x) \\ &= \sum_y p(y) V_g(p_{X|y}). \end{aligned}$$

Note that g-vulnerability coincides with vulnerability when we choose the identity gain function:

$$g_{id}(w,x) = \begin{cases} 1, & \text{if } w = x, \\ 0, & \text{if } w \neq x. \end{cases}$$

Similar to min-entropy, we can now define g-entropy, g-leakage and g-capacity:

$$\begin{aligned} H_g(\pi) &= -\log V_g(\pi) \\ H_g(\pi, C) &= -\log V_g(\pi, C) \\ \mathcal{L}_g(\pi, c) &= H_g(\pi) - H_g(\pi, C) = \log \frac{V_g(\pi, C)}{V_g(\pi)} \\ \mathcal{ML}_g(C) &= \sup_{\pi} \mathcal{L}_g(\pi, C) \end{aligned}$$

Unlike min-capacity, g-capacity is not realized on the uniform prior in general. Furthermore, calculating a g-vulnerability is not as straightforward as calculating a vulnerability, since we now have to determine the guessing strategy that maximizes the adversary's gain. However, an important property of g-leakage is the "miracle" theorem which tells us that for any channel C and gain function g, min-capacity is an upper bound for g-capacity: $\mathcal{ML}_g(C) \leq \mathcal{ML}(C)$. Hence, if the min-capacity of C is small, then the g-leakage with respect to any gain function g and prior π must also be small.

2.3 Order Relations

In this section we briefly review the definitions of some order-theoretic concepts that we use throughout this thesis [Dav02].

Definition 2.3.1. *Let S be a set. A partial order on S is a binary relation $\leq$ on S such that, for all $x, y, z \in S$,*

- $x \leq x$ *(reflexivity),*
- $x \leq y \wedge y \leq x \Rightarrow x = y$ *(antisymmetry),*
- $x \leq y \wedge y \leq z \Rightarrow x \leq z$ *(transitivity).*

Definition 2.3.2. *Let S be a set. A pre-order on S is a binary relation $\leq$ on S such that, for all $x, y, z \in S$,*

- $x \leq x$ *(reflexivity),*
- $x \leq y \wedge y \leq z \Rightarrow x \leq z$ *(transitivity).*

Definition 2.3.3. *The least upper bound $x \vee y$ is an element of S such that:*

- $x \leq (x \vee y)$ *and* $y \leq (x \vee y)$*.*
- $(x \vee y) \leq w$ *for all $w \in S$ such that $x \leq w$ and $y \leq w$.*

Definition 2.3.4. *The greatest lower bound $x \wedge y$ is an element of S such that:*

- $(x \wedge y) \leq x$ *and* $(x \wedge y) \leq y$*.*
- $w \leq (x \wedge y)$ *for all $w \in S$ such that $w \leq x$ and $w \leq y$.*

Definition 2.3.5. *A lattice is a set S partially ordered by a relation $\leq$ such that the least upper bound $x \vee y$ and greatest lower bound $x \wedge y$ exist for all $x, y \in S$.*

2.4 Partition Refinement

Let us now turn our attention to deterministic channels. Note that any deterministic channel $(\mathcal{X}, \mathcal{Y}, C)$ is essentially a function $C : \mathcal{X} \to \mathcal{Y}$, and as such, induces an equivalence relation $\sim_C$ on its domain $\mathcal{X}$, where two inputs are equivalent if and only if they map to the same output [LR93, ACPS12]. Using the standard function notation for C instead of the matrix notation, the relation $\sim_C$ is given by:

$$x_1 \sim_C x_2 \quad \text{iff} \quad C(x_1) = C(x_2)$$

The partitions induced by deterministic channels with set of secret inputs $\mathcal{X}$ can be ordered by the *partition refinement* relation. It is said that partition $\sim_{C_1}$ is refined by partition $\sim_{C_2}$, denoted by $C_1 \sqsubseteq C_2$, if each equivalence class of $\sim_{C_2}$ is contained within some equivalence class of $\sim_{C_1}$.

Example 2.4.1. *To illustrate partition refinement, suppose that $C_{country}$ and C_{state} are deterministic channels that receive as input information about an individual. As depicted in figure 2.1, let $C_{country}$ output only the individual's country of birth and C_{state} output also the state of birth for those individuals born in the United States. In this case we can say that $C_{country} \sqsubseteq C_{state}$, since the information provided by C_{state} is finer grained that the information provided by $C_{country}$.*

The relevance of partition refinement in the context of information flow security is that finer partitions are associated to greater amounts of information leakage [Mal11, ACPS12]. For if $A \sqsubseteq B$ then, for any prior distribution π, A never leaks more information than B with respect to Shannon leakage, guessing-entropy leakage, min-entropy leakage, or g-leakage. Note that this is an intuitive result, since finer partitions convey all the information of coarser ones, plus some additional details.

The partition refinement relation together with the set of all partitions induced by deterministic channels from $\mathcal{X}$ constitutes a lattice [LR93]. Figure 2.2 illustrates the resulting lattice for $\mathcal{X} = \{x_1, x_2, x_3\}$. From the point of view of quantitative information flow, this lattice is known as the *Lattice of Information.*

As a final remark, note that the Lattice of Information is an order relation over the partitions induced by channels, rather than an order relation over channels

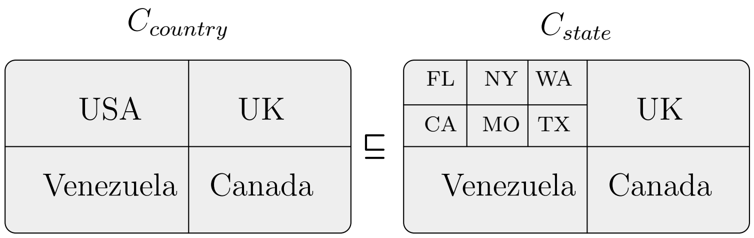

Figure 2.1: The partition induced by C_{state} refines that of $C_{country}$

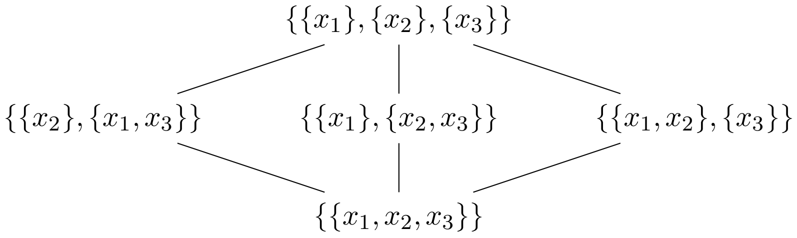
$\{\{x_1\}, \{x_2\}, \{x_3\}\}$
$\{\{x_2\}, \{x_1, x_3\}\}$
$\{\{x_1\}, \{x_2, x_3\}\}$
$\{\{x_1, x_2\}, \{x_3\}\}$
$\{\{x_1, x_2, x_3\}\}$

Figure 2.2: Lattice of Information for $\mathcal{X} = \{x_1, x_2, x_3\}$

themselves. The rationale behind this is that each partition of $\mathcal{X}$ is associated to more than one channel—indeed renaming the output labels of a channel while preserving the same mapping from inputs to outputs results in a different channel that induces the same partition of $\mathcal{X}$. Hence, there exist pairs of *distinct* channels that induce the same partition and hence trivially partition refine each other. As a consequence, partition refinement is not antisymmetric and, therefore, not a partial order on deterministic channels directly. It is only a pre-order.

2.5 Linear Algebra

In this section we enumerate some definitions and properties from linear algebra that we use in this thesis.

We start with some notions of convex algebra [Rom08]:

1. A subset $X \subseteq \mathbb{R}^n$ is *convex* if for any two points x and y in X, the line segment between x and y falls in X.

2. A *convex combination* of a set of vectors $x_1, x_2, ..., x_k \in \mathbb{R}^n$ is a linear combination $a_1x_1 + a_2x_2 + ... + a_kx_k$ such that $0 \le a_i \le 1$ and $\sum_{i=1}^k a_i = 1$.

3. The *convex hull* of a set $X \subseteq \mathbb{R}^n$ is the smallest convex set in $\mathbb{R}^n$ that contains X.

4. The *convex hull* of a set $X \subseteq \mathbb{R}^n$ is the set of all convex combinations of vectors in X.

5. A *polytope* is a generalization of a polyhedron in higher dimensional space. It is a finite region in n-dimensional space enclosed by a finite number of hyperplanes.

6. A *convex polytope* in n-dimensional space, is a polytope in n-dimensional space that is also a convex set.

7. The vertex representation of a convex polytope is an enumeration of the vertices of the polytope.

The following properties of matrix multiplication are also relevant.

Theorem 2.5.1. *Let A, B, C be real matrices such that $A = BC$. Then the i-th row of A is a linear combination of the rows of C with coefficients given by the i-th of B.*

Proof. Let A, B, and C be of sizes $m \times n$, $m \times r$, and $r \times n$ respectively. Let $A^{(1)}, A^{(2)}, ..., A^{(m)}$ denote the rows of A and $C^{(1)}, C^{(2)}, ..., C^{(r)}$ denote the rows of C. Then we can rewrite the equation $A = BC$ using matrix notation as

$$\begin{pmatrix} A^{(1)} \\ \vdots \\ A^{(m)} \end{pmatrix} = \begin{pmatrix} b_{11} & \cdots & b_{1r} \\ \vdots & \ddots & \\ b_{m1} & & b_{mr} \end{pmatrix} \cdot \begin{pmatrix} C^{(1)} \\ \vdots \\ C^{(r)} \end{pmatrix}$$

Using the standard matrix multiplication procedure we get that

$$A^{(i)} = b_{i1}C^{(1)} + b_{i2}C^{(2)} + ... + b_{ir}C^{(r)}.$$

Therefore, the i-th row of A is a linear combination of the rows of C with coefficients given by the i-th of B. □

Corollary 2.5.2. *Let A, B, and C be channel matrices such that $A = BC$. Then the rows of A are a convex combination of the rows of C with coefficients given by the rows of B.*

Proof. Since B is a channel matrix, the coefficients given by the rows of B are non-negative and add up to one. □

Note that if A, B, and C are channel matrices such that $A = BC$, the rows of A (viewed as points in $\mathbb{R}^n$) are within the convex hull of the rows of C.

Theorem 2.5.3. *Let A, B and C be real matrices such that $A = BC$. Then the i-th column of A is a linear combination of the columns of B with coefficients given by the i-th column of C.*

Proof. The proof follows the same reasoning as the one for Theorem 2.5.1. □

CHAPTER 3

MIN-ENTROPY AS A RESOURCE

In this chapter, we explore particularly the idea that secrecy can be viewed as a *resource* that is “created” through a random process, and then gradually “consumed” by the execution of a system. As a first intuitive example, suppose that a system takes as input a 32-bit integer X, where we assume that all 2^{32} possible values are equally likely. If the system performs the bitwise “and” operation

```
Y = X & 0x007f;
```

then an adversary $\mathcal{A}$ seeing the value of Y learns exactly the last 7 bits of X, and remains entirely ignorant of the first 25 bits. Thus it seems clear here that the system starts with 32 bits of secrecy and consumes 7 bits of it, leaving 25 bits of remaining secrecy.

But other systems are considerably more subtle. As an example, consider the Crowds anonymity protocol of Reiter and Rubin [RR98]. In this protocol, a *crowd* of m users cooperates to communicate anonymously with a server. A crowd member wanting to send a message to the server initially sends it to a randomly-chosen *forwarder* (possibly itself). Then, with probability p_f, each forwarder sends the message to another randomly-chosen forwarder (again, possibly itself) or, with probability $1 - p_f$, sends it to the server. Figure 3.1 illustrates a possible message path. Note that when the server receives a message from user i, i is always a randomly-chosen forwarder, and hence no information about the initiator is revealed. However, we further assume that c of the crowd members are actually *collaborators* trying to compromise anonymity. (The collaborators are shaded in Figure 3.1.) If user i forwards a message to a collaborator, then the collaborator reveals i to the server—this weakens anonymity, since i is now more likely than the other crowd members to have

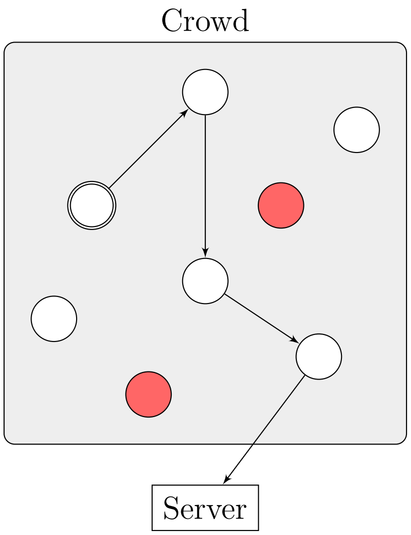

Figure 3.1: Crowds protocol

been the initiator. But quantifying *how much* secrecy is consumed here is not at all obvious. (We will give answers to this question in Section 3.5.)

The main goal of this chapter is to develop the viewpoint of secrecy as a resource in the general setting of probabilistic systems (like the Crowds protocol), which we model as information-theoretic channels. Note that, in light of the significance of min-entropy leakage that we argued in Section 2.2, in this chapter we adopt min-entropy as our measure of secrecy. Viewing min-entropy as a resource naturally leads us to introduce in Section 3.2 a new dynamic measure of min-entropy leakage resulting from a particular system execution, but we will argue that this measure does not work very well, both because it makes policy enforcement difficult and because it can result in a system "consuming" a *negative* amount of secrecy. We will then argue that it is more useful to adopt a *static* measure of the consumption caused by the system as a whole. Accordingly, in Section 3.3 we introduce a new static measure of the min-entropy leakage resulting from the worst-case system execution, and in Section 3.4 we contrast the newly introduced measures with the average-case min-entropy leakage of Smith [Smi09] that we reviewed in Section

2.2.3. Finally, in Section 3.5 we present an information flow analysis of the Crowds anonymity protocol that illustrates how both the worst-case and the average-case static measures of min-entropy leakage can be useful.

3.1 Creation of Min-Entropy

To begin with, we assume that the secret is *created* by a random process, so that the secret is a random variable X that ranges over some finite set $\mathcal{X}$ according to some distribution π. We then model the system as a channel $(\mathcal{X}, \mathcal{Y}, C)$ and make the worst-case assumption that the adversary $\mathcal{A}$, that tries to guess the value of X, knows both the channel and π.

Having established our threat scenario, we measure secrecy with min-entropy. Recall from Section 2.2.3, that the prior min-entropy of the secret is the negative logarithm of the vulnerability of the secret, that is, the worst-case probability that the adversary will guess the value of the secret in one try:

$$H_\infty(\pi) = -\log V(\pi) = -\log \max_x \pi[x]$$

Note that the prior min-entropy quantifies the uncertainty of $\mathcal{A}$ with respect to the value of X before the system execution. In order to quantify the consumption of secrecy that results from the execution of the system we also need a measure of the remaining uncertainty. Then, the consumption of secrecy corresponds to the information being leaked, so it is given by equation 2.4:

$$\text{leakage} = \text{initial uncertainty} - \text{remaining uncertainty}.$$

In the following sections we study three different ways of measuring the remaining min-entropy of the secret.

3.2 Dynamic Min-Entropy Leakage

One view of the consumption of secrecy is *dynamic*, considering the change of secrecy when the adversary observes a *particular* channel output during an individual run of the channel.

Definition 3.2.1. *Given channel $(\mathcal{X}, \mathcal{Y}, C)$ and prior π, the* dynamic min-entropy leakage *associated with output y is the decrease in min-entropy caused by y:*

$$\mathcal{L}^{dynamic}(\pi, C, y) = H_\infty(\pi) - H_\infty(p_{X|y}).$$

Equivalently, it is the logarithm of the factor by which y increases the vulnerability:

$$\mathcal{L}^{dynamic}(\pi, C, y) = \log \frac{V(p_{X|y})}{V(\pi)}.$$

□

For instance, in Example 2.1.1, observing output y_2 reveals that the secret must be x_2, since $p_{X|y_2} = (0, 1, 0, 0)$. In this case, we can say that the secrecy of X decreases from $H_\infty(\pi) = -\log {}^5\!/_{16} \approx 1.678$ down to $H_\infty(p_{X|y_2}) = -\log 1 = 0$. Hence we have $\mathcal{L}^{dynamic}(\pi, C, y_2) \approx 1.678$.

Moreover, if the adversary can run the channel multiple times, using the same value of X each time, then it can repeatedly refine the posterior distribution on $\mathcal{X}$, by using the posterior distribution from one output as the prior distribution for the next run.

Example 3.2.2. *If we run repeatedly the channel from Example 2.1.1 and observe the output sequence y_3, y_1 and y_3, the distribution on $\mathcal{X}$ is refined as follows:*

$$\begin{aligned} p_X &= (3/16, 5/16, 7/32, 9/32) \\ p_{X|y_3} &= (1/4, 0, 1/4, 1/2) \\ p_{X|y_3,y_1} &= (0, 0, 9/23, 14/23) \\ p_{X|y_3,y_1,y_3} &= (0, 0, 81/277, 196/277) \end{aligned}$$

Respectively, the min-entropy for these distributions is:

$$\begin{aligned} H_\infty(p_X) &= -\log(5/16) \approx 1.678 \\ H_\infty(p_{X|y_3}) &= -\log(1/2) = 1 \\ H_\infty(p_{X|y_3,y_1}) &= -\log(14/23) \approx 0.716 \\ H_\infty(p_{X|y_3,y_1,y_3}) &= -\log(196/277) \approx 0.708 \end{aligned}$$

These decreasing entropy values reflect the gradual consumption of the secrecy of X through the repeated observations. □

While this dynamic view of leakage is natural, it suffers from some significant drawbacks. First, dynamic (run-time) policy enforcement may actually reveal information about the secret. For instance, an execution monitor could track the amount of remaining min-entropy, verifying that it stays above some threshold. But if a run produces an output that leaks too much, what can the monitor do? It might try to respond by aborting the execution, but the very act of aborting might in itself leak a lot of information to the adversary. For example, in the case of a password checker, aborting the execution when the adversary enters the correct password reveals the entire password to the adversary (and also makes the password checker useless).

Moreover, under the dynamic view it turns out that min-entropy need not decrease monotonically—it can actually *increase* as a result of an observation.

Example 3.2.3. *Let channel* $(\mathcal{X}, \mathcal{Y}, C)$ *be as follows:*

C	y_1	y_2
x_1	1	0
x_2	0	1
x_3	0	1
x_4	0	1
x_5	0	1

and suppose that $\pi = [{}^9\!/\!{}_{10}, {}^1\!/\!{}_{40}, {}^1\!/\!{}_{40}, {}^1\!/\!{}_{40}, {}^1\!/\!{}_{40}]$. *Then* $V(\pi) = {}^9\!/\!{}_{10}$ *and* $H_\infty(\pi) \approx 0.152$. *However, after observing output* y_2 *the vulnerability* decreases *to* $V(p_{X|y_2}) = {}^1\!/\!{}_4$ *and the secrecy* increases *to* $H_\infty(p_{X|y_2}) = 2$. *Hence we get* negative leakage*:*

$$\mathcal{L}^{dynamic}(\pi, C, y_2) = \log \frac{{}^1\!/\!{}_4}{{}^9\!/\!{}_{10}} = \log {}^5\!/\!{}_{18} \approx -1.848.$$

□

A real-world scenario corresponding to this example is the case of a doctor trying to diagnose an unknown disease. Based on the symptoms, there might be only one likely diagnosis, making the prior "secrecy" small. But if a medical test refutes that diagnosis, then the doctor is left with no idea of the disease, making the posterior "secrecy" large.

We conclude that, under the dynamic perspective, min-entropy does not behave as a reasonable resource. For this reason, we henceforth restrict our attention to *static* viewpoints.

3.3 Worst-Case Min-Entropy Leakage

Perhaps the most straightforward way to achieve a static measure of the secrecy consumption of a channel is to consider the *worst-case* loss of min-entropy, over all channel outputs. (Such worst-case measures were considered previously by Köpf and Basin [KB07], although using guessing entropy rather than min-entropy, and by Mardziel et al. [MMHS11]; we will discuss the latter paper further in Section 3.6.)

We first define worst-case posterior vulnerability:

Definition 3.3.1. *Given prior* π *and channel* C*, the* worst-case posterior vulnerability *is given by*

$$V^{worst}(\pi, C) = \max_{y \in \mathcal{Y}} V(p_{X|y}).$$

□

Notice that the worst-case posterior vulnerability is simply the maximum posterior probability over all the inputs and outputs:

$$V^{worst}(\pi, C) = \max_{x \in \mathcal{X}, y \in \mathcal{Y}} p(x|y).$$

We define the *worst-case posterior min-entropy* by taking the negative logarithm, as before:

Definition 3.3.2. $H_\infty^{worst}(\pi, C) = -\log V^{worst}(\pi, C)$. □

Finally, we define *worst-case min-entropy leakage* as the difference between the prior and posterior min-entropy; equivalently, it is the logarithm of the ratio of the posterior and prior vulnerability:

Definition 3.3.3.

$$\mathcal{L}^{worst}(\pi, C) = H_\infty(\pi) - H_\infty^{worst}(\pi, C) = \log \frac{V^{worst}(\pi, C)}{V(\pi)}.$$

□

Worst-case min-entropy leakage is sometimes a useful measure, but it has the serious drawback that it is highly sensitive to a channel's worst output, even if that output is very unlikely. For instance a password checker

```
  if (X == g) Y = 1; else Y = 0;
```

has the *same* worst-case leakage as a program that always leaks the entire secret:

```
  Y = X;
```

To see this, notice that under the password checker $p_{X|1}(g) = p(X = g|Y = 1) = 1$, which means that $V(p_{X|1}) = 1$.

3.4 Average-Case Min-Entropy Leakage

To minimize the sensitivity to unlikely "bad" outputs, it seems generally more useful to define leakage by considering the *average* posterior vulnerability over all outputs.

Definition 3.4.1. *Given prior* π *and channel* C*, the* average posterior vulnerability *is given by*

$$V^{average}(\pi, C) = \sum_{y \in \mathcal{Y}} p(y) V(p_{X|y}).$$

□

Now we define entropy and leakage as before:

Definition 3.4.2. *The* average posterior min-entropy *is given by*

$$H_\infty^{average}(\pi, C) = -\log V^{average}(\pi, C),$$

and the average min-entropy leakage *is given by*

$$\mathcal{L}^{average}(\pi, C) = H_\infty(\pi) - H_\infty^{average}(\pi, C) = \log \frac{V^{average}(\pi, C)}{V(\pi)}.$$

□

Note that $V(\pi)$, $V^{average}(\pi,C)$, $H_\infty^{average}(\pi,C)$, and $\mathcal{L}^{average}(\pi,C)$ are the leakage measures that are advocated by Smith [Smi09, Smi11] and (with a slight variation) by Braun, Chatzikokolakis, and Palamidessi [BCP09]. We presented min-entropy leakage in more detail in Section 2.2.3.

Example 3.4.3. *Returning to Example 2.1.1, we have*

$$V^{average}(\pi,C) = \sum_{y\in\mathcal{Y}} \max_{x\in\mathcal{X}} p(x,y) = {}^1\!/\!_8 + {}^1\!/\!_4 + {}^1\!/\!_8 + {}^1\!/\!_8 = {}^5\!/\!_8.$$

Hence we get

$$\mathcal{L}^{average}(\pi,C) = \log \frac{{}^5\!/\!_8}{{}^5\!/\!_{16}} = \log 2 = 1,$$

reflecting the fact that the average vulnerability is doubled by C.

Note that the adversary's best prior guess for X is x_2, since $\pi[x_2] = {}^5\!/\!_{16}$ is maximal. Interestingly, the maximums in the calculation of $V^{average}$ reflect the adversary's best guess about X, given each output. On output y_1, the best guess is x_3 (or x_4); on output y_2, the best guess is x_2; on output y_3, the best guess is x_4; and on output y_4, the best guess is x_1. □

In contrast to what we observed about dynamic posterior vulnerability, the average posterior vulnerability cannot be less than the prior vulnerability:

Theorem 3.4.4. *For any π and C, $V(\pi) \le V^{average}(\pi,C)$. Hence $\mathcal{L}^{average}(\pi,C) \ge 0$.*

Proof. We have

$$V(\pi) = \max_{x\in\mathcal{X}} \sum_{y\in\mathcal{Y}} p(x,y) \le \sum_{y\in\mathcal{Y}} \max_{x\in\mathcal{X}} p(x,y) = V^{average}(\pi,C).$$

□

Also, as expected, the average posterior vulnerability is upper bounded by the worst-case posterior vulnerability:

Theorem 3.4.5. *For any π and C, $V^{average}(\pi, C) \leq V^{worst}(\pi, C)$.*

Proof.

$$V^{average}(\pi, C) = \sum_{y \in \mathcal{Y}} p(y) V(p_{X|y}) \leq \sum_{y \in \mathcal{Y}} p(y) V^{worst}(\pi, C) = V^{worst}(\pi, C).$$

□

As an immediate consequence, worst-case min-entropy leakage must be non-negative and must be an upper bound on average min-entropy leakage.

3.5 A Case Study: the Crowds Protocol

We conclude this chapter with a case study in which we calculate the average and worst-case min-entropy leakage associated with the Crowds protocol of Reiter and Rubin [RR98], which was briefly described at the beginning of this chapter. This analysis of the Crowds protocol is due to Smith [ES13].

We assume that there are n honest users and c collaborators in the Crowd, whose total size is m, where $m = n + c$. An initiator wishing to send a message to the server first sends it to a randomly-chosen forwarder, possibly itself, each with probability $\frac{1}{m}$. With probability p_f, each forwarder will forward the message again to another randomly-chosen forwarder, possibly itself; with probability $1 - p_f$, it sends the message to the server. If some user forwards a message to a collaborator, then that user is said to be *detected.* Once a user is detected or the message reaches the server, the protocol stops. We assume that the forwarding probability p_f satisfies $0 \leq p_f < 1$, since if $p_f = 1$ then the message can never reach the server.

As a channel, the set of secret inputs $\mathcal{X} = \{u_1, \ldots, u_n\}$, where u_i means that user i is the initiator. The set of observable outputs $\mathcal{Y} = \{d_1, \ldots, d_n, s\}$, where d_i means that user i was detected, and s means that the message reached the server without ever going to a collaborator.

Now we calculate the channel matrix C, using techniques like those in [RR98]. To compute $p(s|u_i)$, we observe that output s occurs iff the message is forwarded one or more times among honest users, and then is sent to the server:

$$\begin{aligned} p(s|u_i) &= \sum_{j=0}^{\infty} \frac{n}{m}\left(p_f \frac{n}{m}\right)^j (1 - p_f) \\ &= \frac{n}{m}(1 - p_f)\left(\frac{1}{1 - \frac{p_f n}{m}}\right) \\ &= \frac{n - p_f n}{m - p_f n} \end{aligned}$$

Let D be the event that *some* user is detected. We have

$$p(D|u_i) = 1 - p(s|u_i) = 1 - \frac{n - p_f n}{m - p_f n} = \frac{c}{m - p_f n}$$

Now let D_2 be the event that some user is detected after two or more steps. We have

$$p(D_2|u_i) = p(D|u_i) - \frac{c}{m} = \frac{c(m - (m - p_f n))}{m(m - p_f n)} = \frac{c p_f n}{m(m - p_f n)}$$

Note that *every* user is equally likely to be detected after two or more steps, since forwarders are all randomly chosen. Hence we can compute $p(d_i|u_i)$ as follows:

$$p(d_i|u_i) = \frac{c}{m} + \frac{1}{n}\frac{c p_f n}{m(m - p_f n)} = \frac{c(m - p_f(n-1))}{m(m - p_f n)}$$

Finally, we can calculate $p(d_j|u_i)$, for $j \neq i$, by noting that user j cannot be detected in one step when i is the initiator:

$$p(d_j|u_i) = \frac{c p_f}{m(m - p_f n)}$$

Assume now that the prior distribution π is uniform, so that each user has probability $\frac{1}{n}$ of being the initiator. This means that the prior vulnerability is

$$V(\pi) = \frac{1}{n}.$$

We now determine the posterior vulnerability for each output. First, since $p(s|u_i)$ is the same, for every i, it is immediate that $p_{X|s}$ is a uniform distribution. Hence

$$V(p_{X|s}) = \frac{1}{n}.$$

Next consider $p_{X|d_i}$, for any i. We have seen that column d_i of the channel matrix contains its largest entry at $p(d_i|u_i)$, and $n-1$ smaller entries $p(d_i|u_j)$, for $j \neq i$. Moreover the sum of the column can be calculated easily, using the symmetries of the matrix:

$$\sum_{j=1}^{n} p(d_i|u_j) = \sum_{j=1}^{n} p(d_j|u_i) = p(D|u_i)$$

Hence we have

$$V(p_{X|d_i}) = \frac{p(d_i|u_i)}{p(D|u_i)} = \frac{c(m-p_f(n-1))}{m(m-p_fn)}\frac{m-p_fn}{c} = \frac{m-p_f(n-1)}{m}.$$

Note that $V(p_{X|d_i})$ does not depend on i.

Now we calculate the average posterior vulnerability:

$$\begin{aligned}
V^{average}(\pi, C) &= p(s)V(p_{X|s}) + \sum_{i=1}^{n} p(d_i)V(p_{X|d_i}) \\
&= \frac{n-p_fn}{m-p_fn}\frac{1}{n} + \left(1 - \frac{n-p_fn}{m-p_fn}\right)\frac{m-p_f(n-1)}{m} \\
&= \frac{m-p_fn-p_fc+cm-cp_fn+cp_f}{m(m-p_fn)} \\
&= \frac{m(c+1)-p_fn(c+1)}{m(m-p_fn)} \\
&= \frac{c+1}{c+n}.
\end{aligned}$$

And from this, we can calculate the average min-entropy leakage:

$$\mathcal{L}^{average}(\pi, C) = \log\frac{(c+1)n}{c+n}$$

Remarkably, we see that the average posterior vulnerability and the average min-entropy leakage *do not depend* on the forwarding probability p_f.

But this is not to say that p_f is irrelevant to the effectiveness of the Crowds protocol. For example, if $p_f = 0$, then on output d_i the adversary *knows* that user i was the initiator. For this reason it is interesting to consider as well the worst-case posterior vulnerability:

$$V^{worst}(\pi, C) = \frac{m - p_f(n-1)}{m} = \frac{c + (1 - p_f)n + p_f}{c + n}$$

and the worst-case min-entropy leakage:

$$\mathcal{L}^{worst}(\pi, C) = \log \frac{(c + (1 - p_f)n + p_f)n}{c + n}.$$

Note that the worst-case vulnerability is maximized by $p_f = 0$, which gives a vulnerability of 1. It is minimized by choosing p_f close to 1, but of course this has the drawback of making message transmission slower and also increasing the probability that the message will go at some point to a collaborator.

The worst-case vulnerability analysis of Crowds gives useful insight, showing the importance of carefully choosing information flow measures based the characteristics of the scenario being studied. However, considering that worst-case vulnerability is highly sensitive to a channel's worst output (no matter how unlikely it may be), it usually seems more informative to focus on the average posterior vulnerability and average min-entropy leakage.

3.6 Related Work

In this section, we briefly discuss some additional related work.

Worst-case posterior vulnerability, discussed in Section 3.3, is used by Mardziel et al. [MMHS11]. They consider a scenario where the confidentiality of a user's private information may be gradually consumed by a sequence of queries. After each query output y, the adversary's knowledge is updated dynamically, going from distribution p_X to posterior distribution $p_{X|y}$. The system wishes to ensure that the vulnerability never exceeds some threshold t. But (as we discussed in Section 3.2) aborting a query in the case when $V(p_{X|y}) > t$ would itself reveal information about X. For this reason, they decide whether or not to answer a query based on the worst-case posterior vulnerability, requiring in their Definition 3 that query C be answered only when $V^{worst}(p_X, C) \leq t$. In their implementation, they use abstract interpretation to compute a safe upper bound on $V^{worst}(p_X, C)$.

Besson et al. [BBJ13] adopt self-information, an alternative dynamic measure of information flow, to quantify the amount of information that a fingerprinting script learns by observing the web browser configuration of a user.[1] The self-information $I(x)$ of outcome x of random variable X is defined as $-\log p(x)$. Note that this measure is closely related to Shannon entropy, in that the Shannon entropy of a random variable X is the expected self information over all possible outcomes: $H(X) = E[I(x)] = -\sum_x p(x) \log p(x)$. Because they restrict their study to deterministic scripts, and assuming that the probability of each browser configuration is known, they measure the leakage about a user's browser configuration x as the self-information of the equivalence class of all the browser configurations that map to the same output as x. Using this measure, they implement a hybrid information flow monitor for fingerprinting scripts that uses a combination of static analysis and dynamic enforcement to overapproximate the information leakage.

The channels we have considered here are known in information theory as *discrete memoryless channels*; they are appropriate for modeling non-interactive scenarios. But it is also interesting to consider interactive scenarios in which secret inputs and observable outputs *alternate*. Min-entropy leakage in interactive scenarios is briefly

[1] Browser fingerprinting is a technique that allows websites to profile and track their users' browsers without storing information about them on the client side.

explored by Andrés et al. [APvRS10]. A fuller treatment, though considering only Shannon leakage, is given by Alvim et al. [AAP10a], making use of channels with *memory* and *feedback*.

In this thesis we have assume that the prior π is known to the adversary. It is also interesting to consider the case where the adversary has possibly incorrect *beliefs* about the prior, a scenario explored by Clarkson et al. [CMS05] and by Hamadou et al. [HSP10].

Algorithmic techniques for calculating or bounding min-entropy leakage or min-capacity have seen considerable interest, including work on probabilistic automata by Andrés at al. [APvRS10] and work on deterministic imperative programs by Backes et al. [BKR09], Newsome et al. [NMS09], Köpf and Rybalchenko [KR10], Heusser and Malacaria [HM10], and Meng and Smith [MS11]. Also, negative complexity results have been given by Yasuoka and Terauchi [YT10].

The relationship between min-entropy leakage and *differential privacy* (see Dwork [Dwo11]) has been studied by Alvim et al. [AAC+11] and Barthe and Köpf [BK11].

Finally, recall that min-entropy leakage assumes implicitly that the adversary gains only by guessing the secret *exactly*, in one try. For this reason Alvim et al. [ACPS12] introduced *g-leakage*, a generalization of min-entropy leakage that we presented in Chapter 2.

3.7 Summary

In this chapter, we have explored the viewpoint of secrecy as a *resource* that may be gradually *consumed* by a system. Having adopted min-entropy as our measure of secrecy, we considered three measures of min-entropy consumption: a new dynamic model of min-entropy leakage that quantifies the information flow in a single run of the system, a new worst-case run model, and the average-case model of Smith [Smi09]. We concluded that the average-case model is particularly useful in that the leakage measurements are not overly sensitive to unlikely "bad" outputs of the system, and are always non-negative. We showed, however, that both the worst-case and average-case measures can be useful depending on the characteristics of the scenario being studied.

In the following chapter we study the consumption of secrecy when multiple channels are combined through a variety of channel composition operators.

3.8 Credits

The results that we presented in this chapter are joint work with my advisor Geoffrey Smith and previously appeared in our journal paper titled *Min-entropy as a Resource*, published in 2013 in a special issue of the Information and Computation Journal called Information Security as a Resource.

The original idea of exploring the perspective of min-entropy as a resource came from an invitation to my advisor to give a talk at the Information Security as a Resource Workshop which took place at the University of Oxford in 2011. The goal of his talk was to discuss whether secrecy could be modeled as a resource.

The analysis of the Crowds protocol that we presented in this chapter is due to Geoffrey Smith.

CHAPTER 4

MIN-ENTROPY LEAKAGE OF COMBINED CHANNELS

In this chapter we turn our attention to the behavior of min-entropy leakage when we combine multiple channels. Ideally, we would be able to compute the leakage of a system in terms of the leakage of its constituents. However, such closed formulas are typically not possible. Still, as the following sections describe, we can derive a number of useful bounds depending on what mechanism we use to combine the channels.

In Section 4.1 we start by studying *cascading* [Des53, Abr63], a classic construction on two channels where the output of the first channel is used as the input to the second. A natural question concerns the amount of information flow in a cascade of channels, as compared with in each of the two channels. In the theory of Shannon leakage, the classic *data-processing inequality* [CT06, p. 34] says that the Shannon leakage on a cascade of channels cannot exceed that of either channel; this straightforwardly implies similar bounds for Shannon capacity. In this section, our main goal is to investigate whether similar properties hold for min-entropy leakage. In particular, we show that under any prior distribution, the min-entropy leakage of a cascade of channels cannot exceed the leakage of the first link, and show that, contrary to our intuition, it *can* exceed the leakage of the second link. Given the cascade C of channels $(\mathcal{X}, \mathcal{Y}, A)$ and $(\mathcal{Y}, \mathcal{Z}, B)$ conditional vulnerabilities $V(\pi, C)$, $V(\pi, A)$, and $V(\pi, B)$. We show that $V(\pi, C) \leq V(\pi, A)$, but that no relationship need hold between $V(\pi, C)$ and $V(p_y, B)$. In the case when A is deterministic, however, we show that $V(\pi, C) \leq V(p_y, B)$. Turning to min-capacity, we generalize the results of Köpf and Smith [KS10], showing that the min-capacity of a cascade of channels is upper bounded not just by the logarithm of the number of intermediate results, but also by the min-capacity of each of the links. These results give us a general technique for bounding the min-entropy leakage of any channel that can be factored into a cascade of channels.

In Sections 4.2 and 4.3 we consider the information flow associated to other channel composition operators that have previously appeared in the literature, while providing some refinements to the models and illustrative examples throughout. Concretely, in Section 4.2 we study the information flow when repeated independent runs of a channel are allowed. We show that the min-capacity in repeated independent runs of a channel grows logarithmically with respect to the number of runs n, a result that was first proved by Köpf and Smith [KS10] within the context of timing attacks against a cryptosystem. Later, in Section 4.3 we look at the min-entropy leakage in an adaptive composition of channels A and B, where the output of A is observable and B receives as input not only the output from A but also the input to A. In this case, as shown by Barthe and Köpf [BK11], the min-capacity of the combined channel is upper bounded by the sum of the min-capacities of A and B. We also analyze the case of non-adaptive composition where B ignores the output from A, extend the upper bound from Barthe and Köpf to the general case of n adaptive and non-adaptive compositions, and show that these more general upper bounds are actually tight.

4.1 Leakage of a Cascade of Channels

In this section we establish upper bounds on the min-entropy leakage and min-capacity of a cascade of channels under a given prior distribution. In subsection 4.1.1 we first carefully study the semantics of cascading of channels, exposing some technical subtleties with non-uniqueness of joint distributions and also taking care to deal with undefined conditional probabilities. Once we have settled the foundations

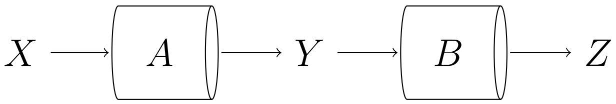

Figure 4.1: Cascade of channels A and B

of cascades of channels, in subsection 4.1 we study their min-entropy leakage and, in subsection 4.1.3, we establish upper bounds on their min-capacity.

4.1.1 Foundations of Cascades of Channels

Given channels $(\mathcal{X}, \mathcal{Y}, A)$ and $(\mathcal{Y}, \mathcal{Z}, B)$, where the set of outputs of the first is the same as the set of inputs of the second, it makes sense to form a *cascade of channels* that composes the channels sequentially [Abr63]. Intuitively, given a prior distribution π, the cascade of channels will proceed in two steps. First, the information in X flows through the first channel and determines a distribution p_y and a random variable Y. Then, the information in Y flows through the second channel to produce the final output Z distributed according to p_z. This construction is depicted in Figure 4.1.

When we consider the formal semantics of a cascade of channels, we might expect that there is a unique joint distribution p_{XYZ} that recovers π and the conditional probabilities A and B, whenever they are defined. Curiously, this turns out not to be true.

Example 4.1.1. *Let* $\mathcal{X} = \mathcal{Y} = \mathcal{Z} = \{0, 1\}$*, and let* A*,* B*, and* π *be as follows:*

A	0	1
0	$1/4$	$3/4$
1	$1/2$	$1/2$

B	0	1
0	$1/2$	$1/2$
1	$1/4$	$3/4$

$$\pi = (2/3, 1/3).$$

With this setup, we can pinpoint at least two scenarios for the joint distribution p_{XYZ}. Recall that any joint distribution must satisfy the product rule

$$p(x,y,z) = p(x)p(y|x)p(z|x,y)$$

whenever the conditional probabilities are defined. Since we demand $p(x) = \pi[x]$ and $p(y|x) = A[x,y]$, it is clear that our only freedom is in choosing $p(z|x,y)$.

For our first scenario, we make Z the exclusive or *of X and Y:*

$$q(z|x,y) = \begin{cases} 1, & \textit{if } z = x \oplus y \\ 0, & \textit{otherwise} \end{cases}$$

Using the product rule, we obtain the following joint distribution:

X	Y	Z	$q(x,y,z)$
0	0	0	$1/6$
0	0	1	0
0	1	0	0
0	1	1	$1/2$
1	0	0	0
1	0	1	$1/6$
1	1	0	$1/6$
1	1	1	0

This joint distribution q recovers π as well as the conditional probabilities A and B. For example, we can verify that $q_{Z|Y}(0|1) = 1/4 = B[1,0]$:

$$q_{Z|Y}(0|1) = \frac{q_{YZ}(1,0)}{q_Y(1)} = \frac{\sum_x q_{XYZ}(x,1,0)}{\sum_{x,z} q_{XYZ}(x,1,z)} = \frac{0 + 1/6}{0 + 1/2 + 1/6 + 0} = 1/4.$$

Note, however, that the definition of q is contrary to our intended "cascading" behavior, since it makes the conditional probability of Z depend on both *X and Y.*[1]

[1] A strange consequence is that $q_{Z|Y}$ depends on prior π. For instance, if we change π to $(1/2, 1/2)$, we find that $q_{Z|Y}$ no longer coincides with B.

For our second scenario, we instead make the conditional probability of Z depend only on Y, choosing $p(z|x,y) = p(z|y)$. This gives a second joint distribution that recovers π and the conditional probabilities A and B:

X	Y	Z	$p(x,y,z)$
0	0	0	$1/12$
0	0	1	$1/12$
0	1	0	$1/8$
0	1	1	$3/8$
1	0	0	$1/12$
1	0	1	$1/12$
1	1	0	$1/24$
1	1	1	$1/8$

□

Using the intuitions developed in Example 4.1.1, we formally define the semantics of a cascade of channels:

Definition 4.1.2. *The* cascade *of channels* $(\mathcal{X}, \mathcal{Y}, A)$ *and* $(\mathcal{Y}, \mathcal{Z}, B)$ *under prior distribution* π *has joint distribution* p_{XYZ}*, where*

$$p_{XYZ}(x,y,z) = \pi[x]A[x,y]B[y,z].$$

We now establish the properties of p_{XYZ} in a series of theorems.

Theorem 4.1.3. p_{XYZ} *recovers the prior* π *by marginalization, in that* $p(x) = \pi[x]$.

Proof. For any $x \in \mathcal{X}$, we have

$$\begin{aligned} p(x) &= \sum_{y,z} p(x,y,z) \\ &= \sum_{y,z} \pi[x]A[x,y]B[y,z] \\ &= \pi[x]\sum_{y} A[x,y]\sum_{z} B[y,z] \\ &= \pi[x]\sum_{y} A[x,y] \\ &= \pi[x] \end{aligned}$$

□

Theorem 4.1.4. *p_{XYZ} is a valid joint distribution.*

Proof. Each $p(x,y,z)$ is non-negative because it is the product of non-negative factors $\pi[x]$,$A[x,y]$ and $B[y,z]$. Moreover, we have

$$\begin{aligned} \sum_{x,y,z} p(x,y,z) &= \sum_{x,y,z} \pi[x]A[x,y]B[y,z] \\ &= \sum_{x} \pi[x]\sum_{y} A[x,y]\sum_{z} B[y,z] \\ &= \sum_{x} \pi[x] \\ &= 1. \end{aligned}$$

□

Theorem 4.1.5. *p_{XYZ} recovers the conditional probabilities in A and B, in that $p(y|x) = A[x,y]$ whenever $p(x) \neq 0$, and $p(z|y) = B[y,z]$ whenever $p(y) \neq 0$.*

Proof. Assuming that $p(x) \neq 0$, we have

$$\begin{aligned}
p(y|x) &= \frac{p(x,y)}{p(x)} \\
&= \frac{\sum_z p(x,y,z)}{\pi[x]} \\
&= \frac{\sum_z \pi[x]A[x,y]B[y,z]}{\pi[x]} \\
&= \sum_z A[x,y]B[y,z] \\
&= A[x,y]\sum_z B[y,z] \\
&= A[x,y]
\end{aligned}$$

Similarly, assuming that $p(y) \neq 0$, we can verify that $p(z|y) = B[y,z]$. □

Theorem 4.1.6. *Whenever* $p(x,y) \neq 0$, *we have* $p(z|x,y) = p(z|y)$.

Proof. Assuming that $p(x,y) \neq 0$, we have

$$\begin{aligned}
p(z|x,y) &= \frac{p(x,y,z)}{p(x,y)} \\
&= \frac{\pi[x]A[x,y]B[y,z]}{\sum_{z\in\mathcal{Z}} \pi[x]A[x,y]B[y,z]} \\
&= \frac{\pi[x]A[x,y]B[y,z]}{\pi[x]A[x,y]\sum_z B[y,z]} \\
&= B[y,z] \\
&= p(z|y)
\end{aligned}$$

□

This last property gives the intended cascading behavior: the conditional probability of output z depends only on the intermediate result y and not directly on the secret input x. Moreover, p is the *unique* joint distribution that satisfies these four theorems:

Theorem 4.1.7. *If q_{XYZ} is any joint distribution that recovers π, gives the correct conditional probabilities when they are defined, and satisfies $q(z|x,y) = q(z|y)$ when they are defined, then q_{XYZ} is equal to p_{XYZ}.*

Proof. If all the conditional probabilities are defined and $q(z|x,y) = q(z|y)$, we know that

$$\frac{q(x,y,z)}{q(x,y)} = \frac{q(x,y,z)}{\sum_z q(x,y,z)} = q(z|y).$$

Moreover, since q_{XYZ} recovers B we have

$$q(x,y,z) = \left(\sum_z q(x,y,z)\right) B[y,z].$$

Since it also recovers A we know that $q(y|x) = \frac{\sum_z q(x,y,z)}{q(x)} = A[x,y]$. Then, substituting $\sum_z q(x,y,z)$ into the previous equation we get

$$q(x,y,z) = q(x)A[x,y]B[y,z].$$

Finally, since $q(x,y,z)$ recovers the prior distribution π we conclude that

$$q(x,y,z) = \pi[x]A[x,y]B[y,z].$$

Now we consider the cases where conditional probabilities are undefined. If $q(x) = 0$, we must have $0 = \sum_y \sum_z q(x,y,z)$, which means that $p(x,y,z) = 0$ for every $y \in \mathcal{Y}$ and $z \in \mathcal{Z}$. Similarly, $q(x,y) = 0$ implies that $q(x,y,z) = 0$ for every $z \in \mathcal{Z}$. Also, if $q(y) = 0$, then $q(x,y,z)$ should be zero for every $x \in \mathcal{X}$ and $z \in \mathcal{Z}$. Since we can merge all the cases into $q(x,y,z) = p(x,y,z)$, we have concluded the proof. □

We next turn our attention to the conditional probabilities $p(z|x)$, showing that these can be obtained by matrix multiplication:

Theorem 4.1.8. *Whenever $\pi[x] \neq 0$, we have $p(z|x) = AB[x,z]$.*

Proof. If $\pi[x] \neq 0$, then

$$\begin{aligned} p(z|x) &= \frac{p(x,z)}{p(x)} \\ &= \frac{\sum_y p(x,y,z)}{p(x)} \\ &= \frac{\sum_y \pi[x]A[x,y]B[y,z]}{\pi[x]} \\ &= \sum_y A[x,y]B[y,z] \\ &= AB[x,z] \end{aligned}$$

□

This last property motivates the following definition, which specifies the channel matrix of the cascade in terms of the channel matrices of the links, and independently of a prior distribution:

Definition 4.1.9. *The* cascade *of channels $(\mathcal{X},\mathcal{Y},A)$ and $(\mathcal{Y},\mathcal{Z},B)$ is the channel $(\mathcal{X},\mathcal{Z},AB)$.*

In the rest of this document, we will sometimes write $C = AB$ to indicate that channel C is the cascade of channels A and B. But we must be careful with this notation, since definition 4.1.9 does not suffice to determine p_{XYZ} as the following example illustrates.

Example 4.1.10. *Recalling Example 4.1.1, we can calculate the conditional probabilities $p(z|x)$ by multiplying the matrices A and B. For convenience, here we organize these probabilities in matrix form and denote them with $p_{Z|X}$.*

$$\begin{array}{|c|cc|} \hline p_{Z|X} & 0 & 1 \\ \hline 0 & 5/16 & 11/16 \\ 1 & 3/8 & 5/8 \\ \hline \end{array} = \begin{array}{|c|cc|} \hline A & 0 & 1 \\ \hline 0 & 1/4 & 3/4 \\ 1 & 1/2 & 1/2 \\ \hline \end{array} \cdot \begin{array}{|c|cc|} \hline B & 0 & 1 \\ \hline 0 & 1/2 & 1/2 \\ 1 & 1/4 & 3/4 \\ \hline \end{array}$$

In contrast, the conditional probabilities $q(z|x)$ do not coincide with the matrix product:

$q_{Z\|X}$	0	1
0	$1/4$	$3/4$
1	$1/2$	$1/2$

This might make us wonder whether the property that $p_{Z|X}$ is given by matrix multiplication might suffice to determine p_{XYZ}. But this turns out not to be true. Consider the channels

A	y_1	y_2
x_1	$1/2$	$1/2$
x_2	$1/2$	$1/2$

B	z_1	z_2
y_1	$2/3$	$1/3$
y_2	$1/3$	$2/3$

$\pi = (2/3, 1/3)$.

If we define q_{XYZ} as in Example 4.1.1, then we get $q_{XYZ} \neq p_{XYZ}$, but nevertheless

$q_{Z\|X}$	0	1
0	$1/2$	$1/2$
1	$1/2$	$1/2$

$= AB.$

□

4.1.2 Min-Entropy Leakage of a Cascade of Channels

If we imagine channels as pipes, and information as water that flows through these pipes, then we might anticipate that the leakage in a cascade of channels cannot exceed the leakage of the first link. We prove this property in Theorem 4.1.11.

Theorem 4.1.11. *Let $(\mathcal{X}, \mathcal{Z}, C)$ be the cascade of $(\mathcal{X}, \mathcal{Y}, A)$ and $(\mathcal{Y}, \mathcal{Z}, B)$. Then for any prior distribution π, we have $\mathcal{L}(\pi, C) \leq \mathcal{L}(\pi, A)$.*

Proof. Unfolding the formula of min-entropy leakage, we observe that the desired inequality is equivalent to an inequality on the posterior vulnerabilities:

$$\mathcal{L}(\pi, C) \leq \mathcal{L}(\pi, A) \iff \log \frac{V(\pi, C)}{V(\pi)} \leq \log \frac{V(\pi, A)}{V(\pi)} \iff V(\pi, C) \leq V(\pi, A).$$

Those posterior vulnerabilities are the sum of the column maximums in the corresponding joint matrices:

$$V(\pi, C) = \sum_z \max_x p_{XZ}(x, z) \qquad\qquad V(\pi, A) = \sum_y \max_x p_{XY}(x, y).$$

Recall from equation (2.3) that we can express the joint matrices as a matrix product:

$$p_{XZ} = \mathrm{diag}(\pi) C \qquad\qquad p_{XY} = \mathrm{diag}(\pi) A.$$

Considering that $(\mathcal{X}, \mathcal{Z}, C)$ is a cascade of channels we get

$$p_{XZ} = \mathrm{diag}(\pi) C = \mathrm{diag}(\pi)(AB) = (\mathrm{diag}(\pi) A) B = p_{XY} B.$$

Hence, it is our goal to prove that the sum of the column maximums in p_{XY} must be at least as large as the sum of the column maximums in $p_{XY} B$.[2]

Let α_y for $y \in \mathcal{Y}$ denote the maximum of column y of p_{XY}:

$$\alpha_y = \max_x p_{XY}(x, y).$$

Also, let μ_z denote the maximum of column z of p_{XZ}:

$$\mu_z = \max_x p_{XZ}(x, z).$$

Then, for every $z \in \mathcal{Z}$, the elements in column z of P_{XZ} satisfy

$$p_{XZ}(x, z) = \sum_y p_{XY}(x, y) B[y, z] \leq \sum_y \alpha_y B[y, z].$$

[2] Notice that the number of columns in p_{XY} and p_{XZ} need not match, so the task cannot be reduced to comparing the matrices column by column.

In particular, this property is satisfied by the column maximum:

$$\mu_z \leq \sum_y \alpha_y B[y,z].$$

Then, using these properties we proceed with the proof:

$$\begin{aligned}
V(\pi, C) &= \sum_z \max_x p_{XZ}(x,z) \\
&= \sum_z \mu_z \\
&\leq \sum_z \sum_y \alpha_y B[y,z] \\
&= \sum_y \sum_z \alpha_y B[y,z] \\
&= \sum_y \alpha_y \sum_z B[y,z] \\
&= \sum_y \alpha_y \\
&= \sum_y \max_x p_{XY}(x,y) \\
&= V(\pi, A).
\end{aligned}$$

□

Alternative proof. A more compact proof of this theorem can be obtained as follows. For any prior π,

$$\begin{aligned}
V(\pi, C) &= \sum_z \max_x p(x,z) \\
&= \sum_z \max_x \sum_y \pi[x]A[x,y]B[y,z] \\
&\leq \sum_z \sum_y \max_x \pi[x]A[x,y]B[y,z] \\
&= \sum_y \sum_z B[y,z] \max_x \pi[x]A[x,y] \\
&= \sum_y \left(\sum_z B[y,z]\right)\left(\max_x \pi[x]A[x,y]\right)
\end{aligned}$$

$$
\begin{aligned}
&= \sum_y \max_x \pi[x]A[x,y] \\
&= \sum_y \max_x p(x,y) \\
&= V(\pi, A).
\end{aligned}
$$

Therefore,

$$\mathcal{L}(\pi, C) = \log \frac{V(\pi, C)}{V(\pi)} \leq \log \frac{V(\pi, C_1)}{V(\pi)} = \mathcal{L}(\pi, C_1).$$

□

It is interesting to note that Theorem 4.1.11 can be viewed as the min-entropy analogue to the classic *data-processing inequality* for *mutual information* (Shannon leakage). Recall from Section 2.2.1, that mutual information is the measure of leakage obtained by measuring uncertainty using Shannon entropy. The data-processing inequality tells us that post-processing can only destroy information.

The standard formulation of the data-processing inequality [CT06, p. 34] starts with the hypothesis that X, Y, Z form a *Markov chain*, denoted $X \rightarrow Y \rightarrow Z$, which means that the joint distribution satisfies the equality

$$p(x,y,z) = p(x)p(y|x)p(z|y). \tag{4.1}$$

It says then that the flow from X to Z cannot exceed the flow from X to Y, as measured by mutual information:

$$I(X;Z) \leq I(X;Y).$$

A drawback of this formulation is that the prior π is "hard coded" into the Markov chain, rather than being a separate parameter as in the formulation of Theorem 4.1.11. Moreover, equation (4.1) runs into undefined conditional probabilities if some values of $\mathcal{X}$ or $\mathcal{Y}$ have probability 0. We can provide an alternative and arguably more expressive formulation in terms of cascades.

Theorem 4.1.12. *Let $(\mathcal{X}, \mathcal{Z}, C)$ be the cascade of $(\mathcal{X}, \mathcal{Y}, A)$ and $(\mathcal{Y}, \mathcal{Z}, B)$, that is, $C = AB$. Then for any prior distribution π, we have*

$$I(\pi, C) \le I(\pi, A).$$

Proof. From Theorem 4.1.1, we know that for any values $x \in \mathcal{X}$, $y \in \mathcal{Y}$, and $z \in \mathcal{Z}$ such that $p(x, y) > 0$, we have $p(z|x, y) = p(z|y)$.

The proof immediately follows, since the data-processing inequality [Gal68, p. 26] tells us that whenever the previous condition holds, we have $I(X; Z) \le I(X; Y)$ which is an alternative notation for $I(\pi, C) \le I(\pi, A)$. □

Returning now to min-entropy leakage, when we consider the leakage in the *second* link of a cascade of channels, we find that it does not behave in the same way as the leakage in the first link. In fact, as the following example shows, the leakage of a cascade of channels may exceed the leakage of the second link.

Example 4.1.13.

A	y_1	y_2
x_1	1	0
x_2	1	0
x_3	0	1

B	z_1	z_2
y_1	1	0
y_2	0	1

C	z_1	z_2
x_1	1	0
x_2	1	0
x_3	0	1

$= AB$

If $\pi = ({}^1\!/_3, {}^1\!/_3, {}^1\!/_3)$, then $p_Y = p_Z = ({}^2\!/_3, {}^1\!/_3)$, and:

$$\mathcal{L}(\pi, C) = \log \frac{V(\pi, C)}{V(\pi)} = \log \frac{{}^2\!/_3}{{}^1\!/_3} = \log 2$$

$$\mathcal{L}(p_Y, B) = \log \frac{V(p_Y, B)}{V(p_Y)} = \log \frac{1}{{}^2\!/_3} = \log {}^3\!/_2$$

Hence, the cascade C leaks more than the second link B. Moreover, the knowledge of the output of B could not have possibly doubled the prior vulnerability as in the *case of C, since doubling a vulnerability of* $^2/_3$ *would mean a posterior vulnerability of* $^4/_3$*, a value that falls outside the range of valid probabilities.* □

To understand why the second link behaves differently, notice that when we compare the leakages $\mathcal{L}(\pi, C)$ and $\mathcal{L}(\pi, A)$ in Theorem 4.1.11, we are comparing $\log \frac{V(\pi,C)}{V(\pi)}$ and $\log \frac{V(\pi,A)}{V(\pi)}$, which reduces to comparing the numerators $V(\pi, C)$ and $V(\pi, A)$. In contrast, when we try to compare $\mathcal{L}(\pi, C)$ and $\mathcal{L}(p_Y, B)$, we are actually comparing $\log \frac{V(\pi,C)}{V(\pi)}$ and $\log \frac{V(p_Y,B)}{V(p_Y)}$, which differ in both the numerators and denominators.

Exploring further, we found that it is not even possible to establish that $V(\pi, C) \leq V(p_y, B)$ in general. As an example, if there is only one value of X and multiple possible values of Y, then $V(\pi, C)$ is certainly equal to 1, while $V(p_y, B)$ could be less than 1.

However, given the additional assumption that A is deterministic, we would expect that $V(\pi, C) \leq V(p_y, B)$. Intuitively, if we correctly guess X, then we can use A to deduce Y as well. We prove this in the following theorem:

Theorem 4.1.14. *If $(\mathcal{X}, \mathcal{Z}, C)$ is the cascade of $(\mathcal{X}, \mathcal{Y}, A)$ and $(\mathcal{Y}, \mathcal{Z}, B)$, where A is deterministic, then for any prior π we have $V(\pi, C) \leq V(p_y, B)$.*

Proof. Let $f : \mathcal{X} \to \mathcal{Y}$ denote the function described by the deterministic channel A, that is, $f(x) = y \iff A[x, y] = 1$. Also, let $[x]_f$ be the set of elements in $\mathcal{X}$ that map to $f(x)$, that is, $[x]_f = \{x' \in \mathcal{X} \mid f(x') = f(x)\}$. Since A is deterministic, for each $x \in \mathcal{X}$ the probability $\pi[x]$ is at most the probability of its image $p_Y[f(x)]$:

$$\begin{aligned}
p_Y[f(x)] &= \sum_{x' \in \mathcal{X}} \pi[x'] A[x', f(x)] \\
&= \sum_{x' \in [x]_f} \pi[x'] A[x', f(x)] + \sum_{x' \in \mathcal{X} \setminus [x]_f} \pi[x'] A[x', f(x)] \\
&= \sum_{x' \in [x]_f} \pi[x'] A[x', f(x)]
\end{aligned}$$

$$= \sum_{x' \in [x]_f} \pi[x']$$

$$\geq \pi[x]$$

Furthermore, we can see that $C[x,z] = B[f(x),z]$:

$$\begin{aligned}
C[x,z] &= \sum_y A[x,y]B[y,z] \\
&= A[s,f(x)]B[f(x),z] + \sum_{y \setminus \{f(x)\}} A[x,y]B[y,z] \\
&= A[s,f(x)]B[f(x),z] \\
&= B[f(x),z].
\end{aligned}$$

Then, using the previous two properties we can proceed with the proof:

$$\begin{aligned}
V(\pi,C) &= \sum_z \max_{x \in \mathcal{X}} (\pi[x]C[x,z]) \\
&= \sum_z \max_{x \in \mathcal{X}} (\pi[x]B[f(x),z]) \\
&\leq \sum_z \max_{x \in \mathcal{X}} (p_Y[f(x)]B[f(x),z]) \\
&= \sum_z \max_y (p_Y[y]B[y,z]) \\
&= V(p_y, B).
\end{aligned}$$

□

Unlike our results for min-entropy leakage, with Shannon mutual information leakage we get bounds on *both* links of the cascade [Abr63]. We can easily prove the bound on the second link if we consider that a Markov chain $X \to Y \to Z$ implies another Markov chain $Z \to Y \to X$. So, by the data-processing inequality, we have $I(Z;X) \leq I(Z;Y)$. But now we can use the *symmetry* of mutual information (i.e. the fact that $I(X;Y) = I(Y;X)$) to deduce that $I(X;Z) \leq I(Y;Z)$. This gives the following corollary.

Corollary 4.1.15. *Let $(\mathcal{X}, \mathcal{Z}, C)$ be the cascade of $(\mathcal{X}, \mathcal{Y}, A)$ and $(\mathcal{Y}, \mathcal{Z}, B)$. Then for any prior distribution π, we have*

$$I(\pi, C) \le I(p_Y, B).$$

Remark 4.1.16. *The symmetry of mutual information is key in proving the data-processing inequality for the second link of a cascade. But it is arguably a strange property; it seems counterintuitive that the mutual information leakage from X to Z should be the same as the mutual information leakage from Z to X. Min-entropy leakage, in contrast, is not symmetric in general. As an example, consider the following $n \times (n+1)$ channel matrix:*

$$\begin{array}{|c|cccccc|}
\hline
C & z_1 & z_2 & z_3 & z_4 & \dots & z_{n+1} \\
\hline
x_1 & 1/2 & 1/2 & 0 & 0 & \dots & 0 \\
x_2 & 1/2 & 0 & 1/2 & 0 & \dots & 0 \\
x_3 & 1/2 & 0 & 0 & 1/2 & \dots & 0 \\
\vdots & \vdots & \vdots & \vdots & \vdots & & \vdots \\
x_n & 1/2 & 0 & 0 & 0 & \dots & 1/2 \\
\hline
\end{array}.$$

Under a uniform prior distribution, $V(\pi) = \frac{1}{n}$ and $V(\pi, C) = \frac{n+1}{2n}$, which implies that $\mathcal{L}(\pi, C) = \log \frac{n+1}{2}$. But when we view p_{XZ} as a channel from Z to X, we find that $V(p_Z) = \frac{1}{2}$ but also the posterior vulnerability of Z given the output of the channel is $\sum_x \max_z p_{XZ}(x, z) = \frac{1}{2}$, which implies that the leakage from Z to X is 0. □

4.1.3 Min-Capacity of a Cascade of Channels

A bound on the min-capacity of a cascade of channels was shown earlier by Köpf and Smith [KS10]. They showed that if a channel C can be factored into the cascade of channels A and B, then the min-capacity of C is at most the logarithm of the number of the number of feasible outputs of A. They used this result to establish security guarantees of blinded cryptography under timing attacks, modeling such an attack as a channel whose input is a secret decryption key and whose output is a sequence of timings of decryption operations using that key. They showed that this channel can be factored into the cascade of two channels such that the set of intermediate results is small, which implies that its min-capacity is small.

Here we go beyond the upper bound established in [KS10], by extending Theorem 4.1.11 to the capacity of a cascade of channels. Moreover, we show that unlike our result for min-entropy leakage under prior π, we can prove that the min-capacity of a cascade of channels cannot exceed the min-capacity of the second link.

Theorem 4.1.17. *If* $(\mathcal{X},\mathcal{Z},C)$ *is the cascade of* $(\mathcal{X},\mathcal{Y},A)$ *and* $(\mathcal{Y},\mathcal{Z},B)$*, then*

$$\mathcal{ML}(C) \le \min(\mathcal{ML}(A), \mathcal{ML}(B)).$$

Proof. By Theorem 4.1.11 and the definition of min-capacity we know that for any prior π,

$$\mathcal{L}(\pi, C) \le \mathcal{L}(\pi, A) \le \mathcal{ML}(A).$$

Hence

$$\mathcal{ML}(C) = \sup_{\pi} \mathcal{L}(\pi, C) \le \mathcal{ML}(A).$$

To obtain the upper bound with respect to B, we observe that

$$\begin{aligned}
\mathcal{ML}(C) &= \log \sum_z \max_x C[x,z] \\
&= \log \sum_z \max_x (AB)[x,z] \\
&= \log \sum_z \max_x \left(\sum_y A[x,y] B[y,z] \right) \\
&\le \log \sum_z \max_x \left(\sum_y A[x,y] \max_{y' \in \mathcal{Y}} B[y',z] \right)
\end{aligned}$$

$$\begin{aligned}
&= \log \sum_{z \in \mathcal{Z}} \max_x \left(\max_{y' \in \mathcal{Y}} B[y', z] \right) \\
&= \log \sum_z \max_{y' \in \mathcal{Y}} B[y', z] \\
&= \mathcal{ML}(B).
\end{aligned}$$

□

We can also provide an alternative proof for the upper bound on the capacity of a cascade of channels from [KS10]. That is, the capacity of a cascade of channels cannot exceed the logarithm of number of intermediate results:

Corollary 4.1.18. *If* $(\mathcal{X}, \mathcal{Z}, C)$ *is the cascade of* $(\mathcal{X}, \mathcal{Y}, A)$ *and* $(\mathcal{Y}, \mathcal{Z}, B)$*, then* $\mathcal{ML}(C) \leq \log |\mathcal{Y}|$.

Proof. We have $\mathcal{ML}(C) \leq \mathcal{ML}(A)$. But $\mathcal{ML}(A)$ is the logarithm of the sum of the column maximums of A. Since A has $|\mathcal{Y}|$ columns, and each maximum is at most 1, we have $\mathcal{ML}(C) \leq \log |\mathcal{Y}|$. □

4.2 Leakage of a Repeated Independent Runs Channel

We now turn our attention to the behavior of min-entropy leakage when repeated independent runs of a channel are allowed. We can model this scenario with a channel that receives an input in $\mathcal{X}$, feeds it to multiple copies of a channel $(\mathcal{X}, \mathcal{Y}, C)$, and collects the outputs from each run into a tuple in $\mathcal{Y}^n$. Figure 4.2 illustrates this combination of channels. The effects of repeated independent runs of a channel on min-entropy were first studied by Köpf and Smith [KS10] within the concrete scenario of a timing attack against a cryptosystem that implements input blinding and bucketing.

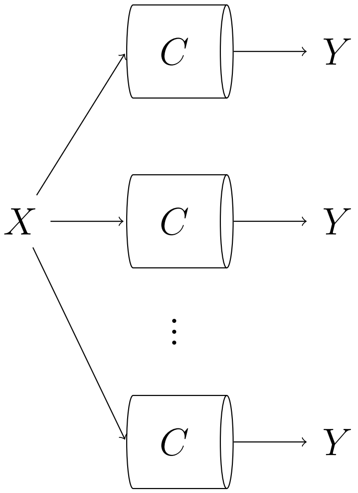

Figure 4.2: Repeated independent runs of channel C

The resulting combined channel $C^{(n)}$ is a channel from $\mathcal{X}$ to $\mathcal{Y}^n$ and the entries $[x,(y_1,\ldots,y_n)]$ of its matrix give the probability of producing the sequence of outputs $(y_1,\ldots,y_n)$ when the input to the channel is x. Since the probabilities of producing a particular output in each run are conditionally independent given a secret input, we calculate entries of the combined channel matrix by multiplying the individual conditional probabilities.

Definition 4.2.1. *The matrix of the channel $(\mathcal{X},\mathcal{Y}^n,C^{(n)})$ that results from n independent runs of channel $(\mathcal{X},\mathcal{Y},C)$ is given by*

$$C^{(n)}[x,(y_1,\ldots,y_n)] = \prod_{i=1}^{n} C[x,y_i].$$

Note that the study of repeated independent runs of a channel makes sense only in the case of a probabilistic channel C; otherwise given a secret input every run (of channel C) would return the same output.

Curiously, even when $\mathcal{L}(\pi,C) = 0$, the leakage under two repeated independent runs $\mathcal{L}(\pi,C^{(2)})$ can be greater than zero.

Example 4.2.2. *Let the matrix of channel* $(\mathcal{X}, \mathcal{Y}, C)$ *be*

C	y_1	y_2
x_1	0.9	0.1
x_2	0.1	0.9

Then, if the prior is given by $\pi = (0.05, 0.95)$ *we get joint distribution*

C	y_1	y_2
x_1	0.045	0.005
x_2	0.095	0.855

and the min-entropy leakage is

$$\mathcal{L}(\pi, C) = \log \frac{0.095 + 0.855}{0.95} = \log \frac{0.95}{0.95} = 0.$$

However, if we consider two independent runs of C*, the resulting combined matrix* $C^{(2)}$ *satisfies* $C^{(2)}(x, (y_1, y_2)) = C[x, y_1]C[x, y_2]$*, giving*

$C^{(2)}$	(y_1, y_1)	(y_1, y_2)	(y_2, y_1)	(y_2, y_2)
x_1	0.81	0.09	0.09	0.01
x_2	0.01	0.09	0.09	0.81

and associated joint matrix

p_{XY}	(y_1, y_1)	(y_1, y_2)	(y_2, y_1)	(y_2, y_2)
x_1	0.0405	0.0045	0.0045	0.0005
x_2	0.0095	0.0855	0.0855	0.7695

But now, the sum of the column maximums of the joint matrix (posterior vulnerability) exceeds the prior vulnerability and we get a non-zero min-entropy leakage

$$\mathcal{L}(\pi, C^{(2)}) = \log \frac{0.0405 + 0.0855 + 0.0855 + 0.7695}{0.95} = \log \frac{0.981}{0.95} > 0.$$

$\square$

It can be shown that the vulnerability of the secret cannot decrease after each additional independent run of channel C. Similarly, neither the leakage nor the capacity of $C^{(n)}$ can decrease as n grows. It is then of interest to understand how fast the capacity $\mathcal{ML}(C^{(n)})$ increases as more repetitions are allowed.

Boreale et al. [BPP11] proved that $\mathcal{ML}(C^{(n)})$ converges exponentially quickly to the logarithm of the number of distinct rows in C. (Intuitively, distinct rows of the channel matrix can be distinguished by repeatedly sampling the output.) More precisely, their Theorem 1 restricted to the case of a uniform prior (which realizes min-capacity) can be restated in the following form:

Theorem 4.2.3. *Let K denote the number of distinct rows in C. Then there is an $\epsilon > 0$ such that*

$$\log K \geq \mathcal{ML}(C^{(n)}) \geq \log K + \log r(n),$$

where $r(n) = 1 - (n+1)^{|\mathcal{Y}|} 2^{-n\epsilon}$.

However, the min-capacity $\mathcal{ML}(C^{(n)})$ grows only logarithmically with respect to the number of runs n. This result was first proved by Köpf and Smith [KS10] within the context of timing attacks against a cryptosystem, but the proof holds in general for any n independent runs channel $C^{(n)}$.

Theorem 4.2.4. *For any channel $(\mathcal{X}, \mathcal{Y}, C)$ and number of repetitions n,*

$$\mathcal{ML}(C^{(n)}) \leq |\mathcal{Y}| \log(n+1).$$

Proof. The proof relies on the information-theoretic method of *types*; see for example [CT06]. The key idea is that, in view of Definition 4.2.1, the conditional probability of an output sequence $\bar{y}$ does not depend on the *ordering* of the outputs, but only on the *number of occurrences* of each element of $\mathcal{Y}$ within $\bar{y}$; this information is called the *type* of $\bar{y}$. For example, if $\mathcal{Y} = \{y_1, y_2, y_3, y_4\}$ and $n = 10$, then the output sequence $(y_3, y_2, y_2, y_4, y_2, y_3, y_2, y_2, y_2, y_2)$ has type $(0, 7, 2, 1)$. In general, a type is a length-$|\mathcal{Y}|$ sequence of numbers, each between 0 and n, whose sum is n. We write $t_{\bar{y}}$ to denote the type of $\bar{y}$, $|t_{\bar{y}}|$ to denote the number of sequences with type $t_{\bar{y}}$, and $\mathcal{T}$ to denote the set of all types.

Because two output sequences with the same type have the same conditional probability given the secret, it follows that we can factor $C^{(n)}$ into the cascade of a channel A from $\mathcal{X}$ to $\mathcal{T}$ and a channel B from $\mathcal{T}$ to $\mathcal{Y}^n$. More precisely, we define

$$A[x, t_{\bar{y}}] = |t_{\bar{y}}| C^{(n)}[x, \bar{y}]$$

and

$$B[t_{\bar{y}}, \bar{y}'] = \begin{cases} \frac{1}{|t_{\bar{y}}|}, & \text{if } \bar{y}' \text{ has type } t_{\bar{y}} \\ 0, & \text{otherwise.} \end{cases}$$

It is easy to see that A and B are well-defined channel matrices and $C^{(n)} = AB$.

Hence we can apply Corollary 4.1.18 to deduce that $\mathcal{ML}(C^{(n)}) \leq \log |\mathcal{T}|$. Since each type is a length-$|\mathcal{Y}|$ sequence of numbers between 0 and n, we see that $|\mathcal{T}| \leq (n+1)^{|\mathcal{Y}|}$, and the theorem follows. □

The bound on the size of $\mathcal{T}$ in the above proof is quite crude, since it ignores the fact that the numbers in a type must sum to n. Köpf and Smith [KS10] also showed a tighter bound by calculating $|\mathcal{T}|$ precisely:

Theorem 4.2.5. *For any channel $(\mathcal{X}, \mathcal{Y}, C)$ and number of repetitions n,*

$$\mathcal{ML}(C^{(n)}) \leq \log \binom{n + |\mathcal{Y}| - 1}{n}.$$

Proof. We show that

$$|\mathcal{T}| = \binom{n + |\mathcal{Y}| - 1}{n}.$$

Counting $|\mathcal{T}|$ can be viewed as an "Occupancy Problem" as discussed in Section II.5 of Feller [Fel68]. We want to know in how many ways we can place n indistinguishable "balls" (the outputs) into $|\mathcal{Y}|$ "bins" (the possibilities for each output). In general, the number of ways of putting n indistinguishable balls into b bins turns out to be the binomial coefficient

$$\binom{n+b-1}{n}.$$

To see this, note that each such placement can be represented as a string of n stars (representing the balls) with $b-1$ bars inserted (representing the boundaries between the bins). For example, with $n = 5$ and $b = 4$, the string

$$* * \,|\, * \,||\, * *$$

represents the case when we put 2 balls in the first bin, 1 ball in the second bin, 0 balls in the third bin, and 2 balls in the fourth bin. If the symbols were all distinguishable, then the number of such strings would be $(n+b-1)!$. But since the n stars and $b-1$ bars are indistinguishable, then the total number of strings is

$$\frac{(n+b-1)!}{n!(b-1)!},$$

which is equal to the above binomial coefficient. □

The following is an example where these bounds are useful to prove formal security guarantees of a system.

Example 4.2.6 (Bounding the leakage of timing attacks on cryptosystems)**.** *Input blinding and bucketing are countermeasures against timing attacks to public-key cryptosystems. Input blinding consists of randomizing the cyphertext before decryption and de-randomizing it after decryption. Hence, with blinding, the time required to decrypt a cyphertext is a randomized function of the decryption key and is independent of the cyphertext. Bucketing, on the other hand, consists of limiting the decryption operation to take one of only a small number of possible times; this requires sometimes delaying the response of the decryption algorithm.*

Köpf and Smith [KS10] observed that a timing attack against a cryptosystem that implements blinding can be modeled as a repeated independent runs channel $(\mathcal{X}, \mathcal{Y}^n, C^{(n)})$ *that receives the secret decryption key and outputs a sequence of* n *timing observations with that key.*

Hence, Theorem 4.2.5 can be directly applied to establish an upper bound for the min-capacity of the timing attack channel:

$$\mathcal{ML}(C^{(n)}) \leq \log \binom{n + |\mathcal{Y}| - 1}{n}.$$

Moreover, bucketing allows the assumption that $|\mathcal{Y}|$ *is small.*

Concretely, consider a cryptosystem that implements input blinding and bucketing with a total of 5 buckets ($|\mathcal{Y}| = 5$*). Then, the channel of a timing attack with* 2^{40} *timing observations is given by* $(\mathcal{X}, \mathcal{Y}^{2^{40}}, C^{(2^{40})})$ *and its capacity is at most*

$$\mathcal{ML}(C^{(2^{40})}) \leq \log \binom{2^{40} + 4}{2^{40}} \approx 155.4 \text{ bits.}$$

$\square$

4.3 Leakage of an Adaptive Composition of Channels

Barthe and Köpf [BK11] studied a form of channel composition that is more powerful than cascading. Their model considers two channels, A and B, where the second channel receives as input not only the output from A but also the input to A. Furthermore, the outputs from both channels are revealed. We illustrate this construction in Figure 4.3.

An example scenario for this kind of composition consists of two privacy preserving randomized queries to the same dataset, where the second query adapts its results depending on the results of the first one. Each of the queries can then be modeled by a probabilistic channel with the special restriction that the channel

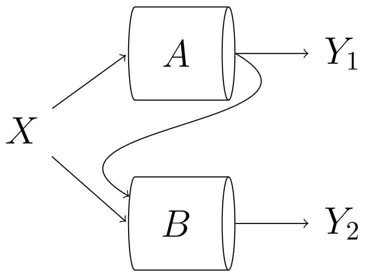

Figure 4.3: Adaptive channel composition $A + B$

of the second query must be ready to handle two inputs: the secret dataset and the result from the previous query. The following is a formal definition of adaptive composition:

Definition 4.3.1. *The adaptive composition of channels* $(\mathcal{X}, \mathcal{Y}_1, A)$ *and* $(\mathcal{Y}_1 \times \mathcal{X}, \mathcal{Y}_2, B)$ *is the channel* $(\mathcal{X}, \mathcal{Y}_1 \times \mathcal{Y}_2, A + B)$ *where*

$$(A + B)[x, (y_1, y_2)] = A[x, y_1]B[(y_1, x), y_2].$$

□

Note that this definition makes sense because, when the conditional probabilities are defined, the entries of the matrix $(A + B)$ are consistent with the chain rule for probabilities:

$$\begin{aligned} p(y_1, y_2|x) &= (A + B)[x, (y_1, y_2)] \\ &= A[x, y_1]B[(y_1, x), y_2] \\ &= p(y_1|x)p(y_2|y_1, x) \end{aligned}$$

Barthe and Köpf established an upper bound on the min-capacity of $A + B$:

Theorem 4.3.2. *The min-capacity of the adaptive composition of channels* $(\mathcal{X}, \mathcal{Y}_1, A)$ *and* $(\mathcal{Y}_1 \times \mathcal{X}, \mathcal{Y}_2, B)$ *is at most the sum of the min-capacities of the channels:*

$$\mathcal{ML}(A + B) \leq \mathcal{ML}(A) + \mathcal{ML}(B).$$

Contrast this with the case of cascading, where Theorem 4.1.17 tells us that the capacity of a cascade of two channels is upper bounded by the *minimum* of the capacities of the channels. Both bounds provide nice guarantees in terms of the behavior of channels when combined under specific conditions. However, we observe that the more restrictive conditions of cascading result in stronger capacity bounds.

We can also think of composing channels A and B *non-adaptively* by having B "ignore" the output from A. We denote this kind of composition with $A +_{na} B$. Note that letting B ignore the output from A amounts to assuming that Y_1 and Y_2 are conditionally independent given the knowledge of X:

$$p(y_1, y_2|x) = p(y_1|x)p(y_2|x).$$

Hence the formula for the combined channel is unchanged since this conditional independence also implies that

$$p(y_2|x) = p(y_2|y_1, x) = B[(y_1, x), y_2].$$

Therefore, we can still establish the same upper bound for the min-capacity of non-adaptive composition $A +_{na} B$.

Corollary 4.3.3. *The min-capacity of the non-adaptive composition of channels* $(\mathcal{X}, \mathcal{Y}_1, A)$ *and* $(\mathcal{Y}_1 \times \mathcal{X}, \mathcal{Y}_2, B)$ *is at most the sum of the min-capacities of the channels:*

$$\mathcal{ML}(A +_{na} B) \leq \mathcal{ML}(A) + \mathcal{ML}(B)$$

Of course, the upper bound from both the adaptive and non-adaptive cases can be extended to the scenario where n compositions are performed:

Corollary 4.3.4. $\mathcal{ML}(A + B + \cdots + C_n) \leq \sum_{i=1}^{n} \mathcal{ML}(C_i)$.

Corollary 4.3.5. $\mathcal{ML}(A +_{na} B +_{na} \cdots +_{na} C_n) \leq \sum_{i=1}^{n} \mathcal{ML}(C_i)$.

But we can wonder whether the upper bounds from corollaries 4.3.4 and 4.3.5 can be strengthened, particularly in the non-adaptive case. The following example shows that these bounds are actually tight.

Example 4.3.6. *Let us consider some deterministic channels with two feasible outputs; the min-capacity of any such channel is 1 bit. First, consider a family of channels E_i that (like a password checker) output* 1 *exactly when a guess c_i matches the secret input:*

E_i: `if (X ==` c_i`) Y = 1; else Y = 0;`

Composing either adaptively or non-adaptively n different instances of the family E_i yields a channel with $n+1$ distinct outputs, since at most one of the E_i's may produce the output 1. *Hence, the min-capacity of the composition is logarithmic in n:*

$$\mathcal{ML}(E_1 + E_2 + \cdots + E_n) = \log(n+1).$$

Consider now a variant family G_i, where the equality test is replaced with a greater-than-or-equal test:

G_i: `if (X >=` c_i`) Y = 1; else Y = 0;`

Now it makes a difference whether we compose adaptively or non-adaptively. With a non-adaptive composition, we can get at most $n+1$ outputs from the combined channel. To understand why, assume that the c_is are chosen in increasing order. Then, the sequence of outputs must consist of k 1's (meaning that the value of X is greater than k of the c_i's) followed by $(n-k)$ 0's, for some k between 0 and n. Hence the combined channel has logarithmic *capacity:*

$$\mathcal{ML}(G_1 +_{na} G_2 +_{na} \cdots +_{na} G_n) = \log(n+1).$$

In contrast, with an adaptive composition, the resulting combined channel may yield a binary search where each G_i checks a different bit of the secret. For instance, if X is a 10-bit secret, we can build a combined channel that behaves as follows:

```
G1:  if (X >= 512) Y1 = 1; else Y1 = 0;

G2:  if (X >= 512*Y1 + 256) Y2 = 1; else Y2 = 0;

G3:  if (X >= 512*Y1 + 256*Y2 + 128) Y3 = 1; else Y3 = 0;

...
```

This results in 2^n feasible outputs, giving linear *capacity:*

$$\mathcal{ML}(G_1 + G_2 + \cdots + G_n) = n.$$

Since each G_i has capacity 1, and n compositions of G_i yield a capacity of at most n, we conclude that the upper bound from Corollary 4.3.4 cannot be strengthened. But what if we restrict ourselves to non-adaptive composition? Curiously, that bound turns out to be tight as well.

Consider finally a family of programs A_i that use a bitwise "and" operation to test the ith bit of the secret:

```
Ai:  if (X & 2^(i−1)) Y = 1; else Y = 0;
```

Clearly, composing non-adaptively n instances of A_i gives 2^n feasible outputs, again giving linear capacity:

$$\mathcal{ML}(A_1 +_{na} A_2 +_{na} \cdots +_{na} A_n) = n.$$

□

4.4 Related Work

In this section, we briefly discuss some additional related work.

The problem of transmitting information through channels in cascade has been studied from the dawn of information theory, as in telecommunications it is very common to split a channel into multiple links. For the case of discrete memoryless channels with a common alphabet for the inputs and outputs, Desoer [Des53] proves that the Shannon capacity of a cascade of channels cannot exceed the Shannon capacity of each link in the cascade. Focusing on the same type of channels, Kiely and Coffey [KC93] study the effect of the *ordering* of the links on the Shannon capacity of a cascade.

The work of El-Sayed [ES78] provides a proof of the data processing inequality for Rényi entropies of order α (for $0 \leq \alpha \leq 1$), while we consider min-entropy, which is Rényi entropy of order ∞. Moreover, El-Sayed's definition of conditional Rényi entropy is different from the one that we use.

Alvim et al. [AAC+11] study the relationship between min-entropy leakage and *differential privacy* [Dwo11], a popular approach to protecting privacy in databases that allow statistical queries. They model a differentially-private query on a secret database X as a cascade of a deterministic channel A that returns the query's real answer Y (which might reveal too much about X), followed by a second channel B that returns a randomized answer Z. The goal is to *minimize* the leakage through A, $\mathcal{L}(\pi, A)$, while simultaneously *maximizing* the *utility* of Z with respect to Y, which is formalized as $V(p_Y, B)$. We can see that our results are somehow consistent with their goals: Theorem 4.1.11 says that $\mathcal{L}(\pi, C) \leq \mathcal{L}(\pi, A)$, which means that the randomization mechanism might help but cannot hurt; and Theorem 4.1.14 says that $V(p_Y, B) \geq V(\pi, C)$, which means that Z's utility with respect to Y may exceed but cannot be less than its utility with respect to X (which in turn corrolates closely with the leakage from X to Z).

4.5 Summary

In this chapter, we have shown that min-entropy leakage satisfies a number of compositionality results that allow the leakage of a complex system to be bounded by the leakage of its constituents.

First, we presented a careful account of channel cascading, and showed that cascading satisfies some nice properties with respect to min-entropy leakage, most importantly, that the min-entropy leakage of a cascade of channels is upper bounded by the min-entropy leakage of the first link. This property is a min-entropy analogue to the data-processing inequality for Shannon leakage. Curiously, we found that such upper bound does not hold with respect to the second link of the cascade. Although, when we turned our attention to min-capacity, we found that both links of the cascade behave as bottlenecks to the information flow of the combined channel.

We also studied the information flow when repeated independent runs of a channel are allowed, showing the min-capacity of the combined channel grows logarithmically with respect to the number of runs, a result that was first proved by Köpf and Smith [KS10] within the context of timing attacks against a cryptosystem. Finally, we reviewed the results from Barthe and Köpf [BK11] regarding the min-entropy leakage in an adaptive composition of channels A and B, who showed that the min-capacity of the combined channel is upper bounded by the sum of the min-capacities of A and B. Going further, we analyzed the case of non-adaptive composition where B ignores the output from A, extended the upper bound from Barthe and Köpf to the general case of n adaptive and non-adaptive compositions, and showed that these more general bounds cannot be strengthened.

In the following channel we will expand our study of channel cascading, showing its important role in determining a leakage ordering relation of channels regardless or prior distribution or leakage measure.

4.6 Credits

The results that we presented in this chapter are joint work with my advisor Geoffrey Smith and previously appeared in two of my earlier publications:

- Barbara Espinoza and Geoffrey Smith. *Min-entropy leakage of channels in cascade.* In Gilles Barthe, Anupam Datta, and Sandro Etalle, editors, *Formal Aspects of Security and Trust*, volume 7140 of *Lecture Notes in Computer Science*, pages 70-84. Springer Berlin Heidelberg, 2012.
- Barbara Espinoza and Geoffrey Smith. *Min-entropy as a Resource.* In *Special Issue: Information Security as a Resource*, volume 226 of *Information and Computation*, pages 57-75. May 2013.

The proof that the min-capacity of a repeated independent runs channel grows only logarithmically with respect to the number of runs of a channel is due to Geoffrey Smith.

The proofs of Theorems 4.1.11, and 4.1.14 are my own work, as well as the proof that the min-capacity of a cascade of two channels is upper bounded by the min-capacity of the second link.

CHAPTER 5

ABSTRACT CHANNELS AND THEIR LEAKAGE ORDERING

"I cannot believe that something so ugly as multiplication of matrices is an essential part of the scheme of nature."

Sir Arthur Eddington

Consider a channel C from $\mathcal{X}$ to $\mathcal{Y}$. Recall from Chapter 2 that if an adversary knows the prior distribution π and C, then its initial uncertainty about X is a function of π. But each separate output value y allows it to update its knowledge about X from π to a posterior $p_{X|y}$ via Bayesian reasoning. Hence, its expected remaining uncertainty bout X, after seeing the output of C, will depend on the set of possible posterior distributions on X and their probabilities. The leakage is then the difference between the initial and remaining uncertainties.

This general quantitative framework is clear enough; but there is of course more than one way to measure the uncertainty associated with a probability distribution as we discussed in Section 2.2. Each leakage measure has its own operational significance, which might or might not suit a given operational scenario. Moreover, the leakage caused by some C will also depend on its prior π. As a result, if we consider the *leakage ordering* of two channels A and B (both taking X as input), it is difficult to give an answer that is *robust*, i.e that does not depend on the particular prior and leakage measure. But such a robust ordering is indispensable if we aim to develop software through stepwise refinement, based on general laws that hold in *all contexts.*

There is such a robust order for deterministic channels, provided by the partition refinement relation ($\sqsubseteq$) [LR93]. As we discussed in Section 2.4, any deterministic channel from $\mathcal{X}$ to $\mathcal{Y}$ induces a partition on $\mathcal{X}$, where x_1 and x_2 belong to the same block iff they map to the same output. That is, each block of the partition is the pre-image of some output y.

Recall channels C_{state} and $C_{country}$ from Example 2.4.1, where the partition induced by C_{state} refines that of $C_{country}$ by mapping all states back to USA. It is intuitively clear that an adversary will always prefer the finer grained partition induced by C_{state} to that of $C_{country}$, whatever the input prior π. This is supported by a theorem due to Yasuoka and Terauchi [YT10], Malacaria, [Mal11], and Alvim et al. [ACPS12], which says that channel B partition refines channel A iff A never leaks more information than channel B on any prior π and under Shannon leakage, min-entropy leakage, guessing entropy leakage or g-leakage.

This is an outstanding property, not only it says that channels in the partition refinement relation satisfy a robust leakage ordering, but it also states that the only way for a deterministic channel A to never leak more than a deterministic channel B is for A's partition to be refined by B's.

In this chapter we generalize these nice properties from *deterministic* to *probabilistic* channels. A first issue, however, is that the story for deterministic channels is not quite as nice as it appears, in that partition refinement is not in fact a *partial order* on deterministic channels, but only a *pre-order.* Because distinct deterministic channels can induce the same partition on $\mathcal{X}$ (since the particular names of the outputs and their order do not matter), partition refinement is not antisymmetric. While this problem is rather obvious in the case of deterministic channels, we will see that it is more subtle for probabilistic channels, and this will lead us to introduce *abstract channels* formed by quotienting away the redundant structure of channel matrices. In Section 5.1 we present abstract channels and explore their fundamental properties, including their canonical representation by *reduced channels* and by *hyper-distributions.*

Turning to the robust leakage ordering of abstract channels, in Section 5.2 we consider a generalization of partition refinement called *composition refinement* ($\sqsubseteq_\circ$) [ACPS12], where $A \sqsubseteq_\circ B$ holds if A can be expressed as B followed by "post-processing". We show that composition refinement is antisymmetric and, therefore, a partial order on abstract channels.

In Section 5.3 we show that composition refinement and the strong g-leakage ordering ($\leq_\mathcal{G}$) *coincide*, where $A \leq_\mathcal{G} B$ holds if A never leaks more than B with respect to g-leakage on any prior distribution or gain function. Hence, composition refinement is partial order on abstract channels that has both structural and leakage characterizations, and is therefore a compelling generalization of partition refinement to probabilistic channels.

In Section 5.4 we explore the relationship between composition refinement and other leakage orderings showing that, like partition refinement, composition refinement entails a robust leakage ordering. A byproduct of this result is a family of data-processing inequalities that hold for any choice of concave uncertainty function, in particular, a new data-processing inequality for guessing entropy. However, we explain that with respect to min-entropy alone, a leakage ordering for all priors does not guarantee composition refinement, and conjecture that this is also the case for Shannon entropy.

Given that the strong g-leakage ordering implies composition refinement, and that composition refinement implies a robust leakage ordering, in Section 5.5 we investigate whether, like min-entropy leakage, Shannon leakage and guessing entropy leakage can be expressed as g-leakages for some choice of gain function. Specifically, we show that casting these measures as g-leakages is possible if we generalize gain functions beyond what was considered by Alvim et al. [ACPS12].

Finally, in Section 5.6 we discuss the compositionality properties of abstract channels. We show that, surprisingly, abstract channels cannot be combined to form more complex structures (e.g. cascades) because they lack critical information about the real channel outputs.

5.1 Abstract Channels

Given a channel $(\mathcal{X}, \mathcal{Y}, C)$ and a prior distribution π, it can easily happen that distinct output values y and y' in $\mathcal{Y}$ give rise to the same posterior distribution on $\mathcal{X}$. In that case there is actually no benefit to the adversary from distinguishing outputs y and y', since each gives the same knowledge about X. Furthermore, the output values themselves make no difference: all that matters to an adversary that observes output y is its associated posterior distribution $p_{X|y}$. In fact, assuming that the adversary knows C and π, the posterior distributions $p_{X|y}$ and their probabilities are what C reveals to the adversary about X.

In light of these observations, from the perspective of information flow security, a channel simply maps prior distributions on $\mathcal{X}$ to distributions of posterior distributions on $\mathcal{X}$, which following [MMM10] we call *hyper-distributions*.

Example 5.1.1. *Consider channel* $(\mathcal{X}, \mathcal{Y}, C)$ *with secret inputs* $\mathcal{X} = \{x_1, x_2, x_3\}$, *public outputs* $\mathcal{Y} = \{y_1, y_2, y_3, y_4\}$, *and channel matrix*

C	y_1	y_2	y_3	y_4
x_1	1	0	0	0
x_2	0	$1/2$	$1/4$	$1/4$
x_3	$1/2$	$1/3$	$1/6$	0

Then, given the uniform prior $\pi = (1/3, 1/3, 1/3)$*, we get joint distribution*

p_{YX}	y_1	y_2	y_3	y_4
x_1	$1/3$	0	0	0
x_2	0	$1/6$	$1/12$	$1/12$
x_3	$1/6$	$1/9$	$1/18$	0

Now, the (marginal) distribution on the outputs is $p_Y = (1/2, 5/18, 5/36, 1/12)$*, and normalizing the columns we get the posterior distributions:*

$$p_{X|y_1} = (2/3, 0, 1/3)$$

$$p_{X|y_2} = (0, 3/5, 2/5)$$

$$p_{X|y_3} = (0, 3/5, 2/5)$$

$$p_{X|y_4} = (0, 1, 0)$$

For clarity, we represent these posterior distributions in table form, together with the output probabilities at the bottom margin:

$p_{X\|Y}$	y_1	y_2	y_3	y_4
x_1	$2/3$	0	0	0
x_2	0	$3/5$	$3/5$	1
x_3	$1/3$	$2/5$	$2/5$	0
	$1/2$	$5/18$	$5/36$	$1/12$

Notice then that outputs y_2 *and* y_3 *produce the same posterior distribution, i.e.* $p_{X|y_2} = p_{X|y_3}$*. Hence, the hyper-distribution produced by* C *on* π*, which we denote with* $\mathcal{C}(\pi)$*, has only three columns rather than four:*

$$\mathcal{C}(\pi) = \begin{array}{c|ccc} x_1 & 2/3 & 0 & 0 \\ x_2 & 0 & 3/5 & 1 \\ x_3 & 1/3 & 2/5 & 0 \\ \hline & 1/2 & 15/36 & 1/12 \end{array}$$

In this representation, the columns of the hyper-distribution are the distinct posterior distributions of X, with their corresponding probability on the bottom margin. Note that the probability $^{15}/_{36}$ *of the middle posterior distribution is obtained by adding* $p(y_2)+p(y_3)$*, that is,* $^{5}/_{18}+^{5}/_{36}$*. We also dropped the column labels, since there is no longer a one-to-one correspondence between the posterior distributions and the set* $\mathcal{Y}$*.* □

We have seen in the previous example how a channel maps a prior distribution to a hyper distribution. We capture this relation in the following definition.

Definition 5.1.2 (Abstract channel)**.** *The leakage semantics* $\mathcal{C}$ *of channel* $(\mathcal{X},\mathcal{Y},C)$ *is the mapping that* C *gives from prior distributions on* $\mathcal{X}$ *to hyper-distributions on* $\mathcal{X}$*, that is,*

$$\mathcal{C} : \mathbb{D}\mathcal{X} \to \mathbb{D}\mathbb{D}\mathcal{X}$$

We call this mapping an abstract channel*.*

Any reasonable leakage measure should be well defined on abstract channels. In fact, as the following theorem confirms, this is the case for the usual leakage measures.

Theorem 5.1.3. *Shannon leakage, guessing entropy leakage, min-entropy leakage, and* g*-leakage are well defined on abstract channels.*

Proof. As we saw in Section 2.2.3, min-entropy leakage is the logarithm of the ratio between the prior and posterior vulnerabilities: $\mathcal{L}(\pi,C) = \log\frac{V(\pi,C)}{V(\pi)}$, where $V(\pi) = \max_x \pi[x]$, and $V(\pi,C) = \sum_y p(y)V(p_{X|y})$. Hence, the column labels make no difference in the calculations. Moreover, if $p_{X|y} = p_{X|y'}$ then the posterior vulnerability is unaffected by merging outputs y and y', since then

$$p(y)V(p_{X|y}) + p(y')V(p_{X|y'}) = (p(y)+p(y'))\,V(p_{X|y}).$$

A similar argument can be applied to the other three leakage measures. □

This abstracted semantic perspective of channels makes us realize that their conventional channel-matrix representation can contain *redundant structure* as far as leakage is concerned, namely (1) *labels* on columns, (2) columns that are *all zero*, representing outputs that can never occur, and (3) *similar* columns, which are columns that are scalar multiples of each other and therefore yield the same posterior distributions. By eliminating this redundancy, we obtain a well defined *reduced channel*:

Definition 5.1.4 (Reduced channel). *The reduced channel C^r of channel $(\mathcal{X}, \mathcal{Y}, C)$ is formed by deleting the output labels, dropping all-zero columns from C, then adding similar columns together, and finally ordering the resulting columns lexicographically.*

Theorem 5.1.5. *Any channel C has the same leakage semantics as its reduction C^r.*

Proof. Given a prior distribution π, output labels, all-zero columns, and column ordering all have no effect on the resulting hyper-distribution. Moreover, similar columns contribute weight to the same posterior distribution; hence merging them leaves the hyper-distribution unchanged. □

In fact, a reduced channel serves as a canonical representation of an abstract channel.

Corollary 5.1.6. *Channels A and B represent the same abstract channel if and only if $A^r = B^r$.*

Example 5.1.7. *Given $X = \{x_1, x_2, x_3\}$ consider the following two channels A and B:*

$$\begin{array}{c|ccc} A & y_1 & y_2 & y_3 \\ \hline x_1 & 1 & 0 & 0 \\ x_2 & 1/4 & 1/2 & 1/4 \\ x_3 & 1/2 & 1/3 & 1/6 \end{array} \qquad \begin{array}{c|ccc} B & z_1 & z_2 & z_3 \\ \hline x_1 & 2/5 & 0 & 3/5 \\ x_2 & 1/10 & 3/4 & 3/20 \\ x_3 & 1/5 & 1/2 & 3/10 \end{array}$$

While these channels may look very different, they are semantically the same channel as far as leakage of X is concerned; that is, as abstract channels they are the same. Indeed both map prior distribution $\pi = (p_1, p_2, p_3)$ to the same hyper-distribution:

$$\mathcal{A}(\pi) = \mathcal{B}(\pi) = \begin{array}{c|cc} x_1 & \frac{4p_1}{4p_1+p_2+2p_3} & 0 \\ x_2 & \frac{p_2}{4p_1+p_2+2p_3} & \frac{3p_2}{3p_2+2p_3} \\ x_3 & \frac{2p_3}{4p_1+p_2+2p_3} & \frac{2p_3}{3p_2+2p_3} \\ \hline & \frac{(4p_1+p_2+2p_3)}{4} & \frac{(3p_2+2p_3)}{4} \end{array}$$

To understand this, note that the second and third columns of A are similar, in that column 2 is two times column 3. In the same way, columns 1 and 3 of B are similar, in that column 1 is two-thirds times column 3. Merging these similar columns we find that A and B have the same reduced channel:

$$A^r = B^r = \begin{array}{c|cc} x_1 & 1 & 0 \\ x_2 & 1/4 & 3/4 \\ x_3 & 1/2 & 1/2 \end{array}$$

□

While we have said that an abstract channel is a mapping from priors to hyper distributions, in fact the mappings that come from channel matrices are highly constrained. Let $\lceil \pi \rceil$ be the *support* of distribution π, that is, those elements of $\mathcal{X}$ for which $\pi[x] \neq 0$. Then we have that:

Theorem 5.1.8. *An abstract channel $\mathcal{C}$ with input $\mathcal{X}$ is completely determined by its behavior on any full-support prior π, that is, one with $\lceil \pi \rceil = \mathcal{X}$.*

Proof. Let π be a full support prior distribution on $\mathcal{X}$. If π yields a certain hyper-distribution then by scaling each posterior distribution with its probability we recover the joint matrix of C^r under π. Then, normalizing the rows of the joint matrix gives C^r. □

It follows that we can also canonically represent an abstract channel by the hyper-distribution that it produces on (for instance) the uniform prior. We showed such a hyper-distribution in Example 5.1.1.

5.2 Structural Ordering

Recall from section 2.4 that *deterministic* channels are essentially functions and, as such, induce a partition on their set of inputs. Moreover, these partitions are *partially ordered* by the *partition refinement* relation ($\sqsubseteq$). An important property of partition refinement is that channels that are in the partition refinement order also satisfy a robust leakage ordering, that is, a leakage ordering that is independent of the leakage measure and prior distribution. Even more remarkable, the converse is also true, so the only way for a deterministic channel A to never leak more than a deterministic channel B is for A's partition to be refined by B's. Here we formalize this property due to Yasuoka and Terauchi [YT10], Malacaria, [Mal11], and Alvim et al. [ACPS12]:

Theorem 5.2.1. *If $(\mathcal{X}, \mathcal{Y}, A)$ and $(\mathcal{X}, \mathcal{Z}, B)$ are deterministic channels, then $A \sqsubseteq B$ iff A never leaks more than B on any prior π (and gain function g) and under Shannon leakage, min-entropy leakage, guessing-entropy leakage, or g-leakage.*

We remark that, in general, the leakage ordering of a pair of channels may vary according to the choice of prior distribution and leakage measure, so Theorem 5.2.1 is a very nice property from the perspective of information flow security. For it allows a robust comparison of the leakage of channels that operate over a common set of secret inputs.

A relevant question is then whether we can generalize Theorem 5.2.1 to probabilistic channels. Note that, unlike deterministic channels, probabilistic channels do not partition the set of secret inputs, so in order to generalize partition refinement we need to find an alternative (more general) structural order relation on probabilistic channels and show that it is associated to a robust leakage ordering.

Fortunately, as first noted in [ACPS12], partition refinement is strongly connected to cascading, as the following example illustrates.

Example 5.2.2. *Coming back to example 2.4.1, consider now a deterministic channel C_{merge} that maps all the American states in C_{state} back to USA. Then, channel $C_{country}$ is equivalent to C_{state} followed by postprocessing with C_{merge}, that is, $C_{country} = C_{state}C_{merge}$.*

In general, it has been shown [LR93, ACPS12] that given deterministic channels A and B, $A \sqsubseteq B$ iff there exists a deterministic channel R such that $A = BR$. This observation motivates the following definition due to Alvim et al. [ACPS12].

Definition 5.2.3. *For channels A and B, we say that A is* composition refined *by B, written $A \sqsubseteq_{\circ} B$ whenever there exists a channel R such that $A = BR$.*

It follows then that, for deterministic channels, $A \sqsubseteq B$ iff $A \sqsubseteq_{\circ} B$, making composition refinement a promising candidate for generalizing partition refinement to probabilistic channels.

A first step towards proving that composition refinement is a generalization of partition refinement is to verify that it is also a partial order. Here we show that composition refinement is easily seen to be reflexive and transitive on probabilistic channels, and thus a pre-order.

Theorem 5.2.4. *Composition refinement is reflexive and transitive on the domain of probabilistic channels, and thus, a preorder.*

Proof. To prove reflexivity we just need to observe that for any channel matrix C, $C = CI$ where I is the identity matrix. Hence, for all channels, $C \sqsubseteq_\circ C$.

Transitivity follows from the associativity of matrix multiplication since, for any channel matrices A, B, C, R_1, R_2, whenever $A = BR_1$ and $B = CR_2$, we have that $A = (CR_2)R_1 = C(R_2R_1)$. Hence, $A \sqsubseteq_\circ B$ and $B \sqsubseteq_\circ C$ implies that $A \sqsubseteq_\circ C$. □

However, composition refinement is not antisymmetric as can be seen from matrices A and B in Example 5.1.7, which composition refine each other despite being distinct:

$$\begin{array}{|c|ccc|}\hline A & y_1 & y_2 & y_3 \\ \hline x_1 & 1 & 0 & 0 \\ x_2 & 1/4 & 1/2 & 1/4 \\ x_3 & 1/2 & 1/3 & 1/6 \\ \hline \end{array} = \begin{array}{|c|ccc|}\hline B & z_1 & z_2 & z_3 \\ \hline x_1 & 2/5 & 0 & 3/5 \\ x_2 & 1/10 & 3/4 & 3/20 \\ x_3 & 1/5 & 1/2 & 3/10 \\ \hline \end{array} \cdot \begin{array}{|c|ccc|}\hline R_1 & y_1 & y_2 & y_3 \\ \hline z_1 & 1 & 0 & 0 \\ z_2 & 0 & 2/3 & 1/3 \\ z_3 & 1 & 0 & 0 \\ \hline \end{array}$$

$$\begin{array}{|c|ccc|}\hline B & z_1 & z_2 & z_3 \\ \hline x_1 & 2/5 & 0 & 3/5 \\ x_2 & 1/10 & 3/4 & 3/20 \\ x_3 & 1/5 & 1/2 & 3/10 \\ \hline \end{array} = \begin{array}{|c|ccc|}\hline A & y_1 & y_2 & y_3 \\ \hline x_1 & 1 & 0 & 0 \\ x_2 & 1/4 & 1/2 & 1/4 \\ x_3 & 1/2 & 1/3 & 1/6 \\ \hline \end{array} \cdot \begin{array}{|c|ccc|}\hline R_2 & z_1 & z_2 & z_3 \\ \hline y_1 & 2/5 & 0 & 3/5 \\ y_2 & 0 & 1 & 0 \\ y_3 & 0 & 1 & 0 \\ \hline \end{array}$$

But if we restrict ourselves to the domain of abstract channels, we find that composition refinement is better behaved: it becomes a true partial order. This is analogous to the way partition refinement, as described in Section 2.4, is only a pre-order on deterministic channels directly, but becomes a partial order when we restrict ourselves to the domain of the partitions induced by deterministic channels. Note that, like abstract channels, the partition induced by a deterministic channel abstract away from information that is not relevant to leakage alone, such as the names of the channel outputs and their order.

We next prove that composition refinement is antisymmetric on abstract channels, and therefore a partial order. The first part of this proof is Lemma 5.2.8, which can be seen as a generalized data-processing inequality for abstract channels and *concave* $(\frown)$ uncertainty functions. A function is said to be concave $(\frown)$ if the line segment between any two points on the graph of the function lies below the graph. We write $(\frown)$ after concave to give the reader a quick reminder of the shape of a concave function.

Definition 5.2.5 (Concave function)**.** *A function f is concave $(\frown)$ if the domain of f is a convex set and for all $x, y \in \operatorname{dom} f$, and $0 \le \lambda \le 1$,*

$$\lambda f(x) + (1 - \lambda) f(y) \le f(\lambda x + (1 - \lambda) y)$$

Definition 5.2.6 (Strictly concave function)**.** *A function f is strictly concave $(\frown)$ if the domain of f is a convex set and for all $x, y \in \operatorname{dom} f$, and $0 < \lambda < 1$,*

$$\lambda f(x) + (1 - \lambda) f(y) < f(\lambda x + (1 - \lambda) y)$$

Definition 5.2.7 (Convex function)**.** *A function f is convex if $-f$ is a concave function.*

Lemma 5.2.8 (Generalized data-processing inequality for abstract channels)**.** *Let $\mathcal{A}$ and $\mathcal{B}$ be abstract channels, with $(A, \mathcal{X}, \mathcal{Y})$ and $(B, \mathcal{X}, \mathcal{Z})$ their representation as* *reduced channels,* [1] *and let F be a concave ($\frown$) function from distributions on $\mathcal{X}$ to the reals, and for any channel $(C, \mathcal{X}, \mathcal{Y})$, let*

$$F(\pi, C) = \sum_y p(y) F(p_{X|y}).$$

If $A = BR$ for some channel $(R, \mathcal{Z}, \mathcal{Y})$ then, for any full-support prior π, we have $F(\pi, A) \geq F(\pi, B)$.

Furthermore, if $\mathcal{A} \neq \mathcal{B}$ and F is strictly concave, then the inequality is strict.

Proof. Our proof relies on Jensen's inequality [CT06], that is, if $\lambda_1, \lambda_2, \ldots \lambda_n$ are coefficients in $[0, 1]$ that sum to one, and F is concave, then

$$\sum_n \lambda_n F(x_n) \leq F(\sum_n \lambda_n x_n).$$

We use the following matrix notation. Given matrix M with row labels $\mathcal{X}$ and column labels $\mathcal{Y}$ we write $M_{x,y}$ (instead of $M[x, y]$) to denote the (x, y) entry of M and $M_{-,y}$ to denote column y of M. A fundamental property of matrix multiplication is that $(MN)_{-,y} = M(N_{-,y})$, i.e. that column y of MN is a linear combination of the columns of M with column y of N as the coefficients, and thus that in fact the parentheses above are not necessary.

We write D_π to denote the diagonal matrix with π on its diagonal; hence $D_\pi A$ is the joint matrix giving p_{XY}. Note that because A is reduced and π is full support, the columns of $D_\pi A$ are all non-zero and non-similar; hence normalizing these columns is well defined and gives the posterior distributions $p_{X|y}$, which are all distinct. Since the (necessarily non-zero) sum of column y of $D_\pi A$ is $p(y)$, we have $p_{X|y} = \frac{1}{p(y)} D_\pi A_{-,y}$. For B, similarly, the posterior distributions $p_{X|z}$ are all distinct and we also have that $p_{X|z} = \frac{1}{p(z)} D_\pi B_{-,z}$.

[1] For clearer notation, we specify labels for the columns of reduced matrices A and B.

We now show that $F(\pi, A) \geq F(\pi, B)$ under these conditions. First we have

$$F(\pi, A)$$

$$= \qquad \ll \text{definition of } F(\pi, A) \;\gg$$

$$\sum_y p(y) F(p_{X|y})$$

$$= \qquad \ll p_{X|y} = \frac{1}{p(y)} D_\pi A_{\_,y} \gg$$

$$\sum_y p(y) F\left(\frac{1}{p(y)} D_\pi A_{\_,y}\right)$$

$$= \qquad \ll A = BR \gg$$

$$\sum_y p(y) F\left(\frac{1}{p(y)} D_\pi (BR)_{\_,y}\right)$$

$$= \qquad \ll (BR)_{\_,y} = \sum_z B_{\_,z} R_{z,y} \gg$$

$$\sum_y p(y) F\left(\frac{1}{p(y)} D_\pi \sum_z B_{\_,z} R_{z,y}\right)$$

$$= \qquad \ll \text{ multiplying and dividing by } p(z) \text{ and reorganizing} \gg$$

$$\sum_y p(y) F\left(\sum_z \frac{R_{z,y} p(z)}{p(y)} \left(\frac{1}{p(z)} D_\pi B_{\_,z}\right)\right)$$

$$= \qquad \ll \; p_{X|z} = \frac{1}{p(z)} D_\pi B_{\_,z} \gg$$

$$\sum_y p(y) F\left(\sum_z \frac{R_{z,y} p(z)}{p(y)} p_{X|z}\right)$$

which contains F applied to a linear combination of the posterior distributions, whose coefficients $\frac{R_{z,y} p(z)}{p(y)}$ we now show are convex and thus suitable for applying Jensen's inequality. They sum to one because

$$\sum_z R_{z,y} p(z)$$

$$= \qquad \ll p(z) = \sum_x (D_\pi B)_{x,z} \;\gg$$

$$\sum_z R_{z,y} \sum_x (D_\pi B)_{x,z}$$

$$= \qquad \ll \text{distributive law} \gg$$

$$\sum_{x,z} (D_\pi B)_{x,z} R_{z,y}$$

$=$ $\ll$ definition of matrix multiplication $\gg$

$$\sum_x (D_\pi BR)_{x,y}$$

$=$ $\ll A = BR \gg$

$$\sum_x (D_\pi A)_{x,y}$$

$=$ $\ll$ definition of $p(y)$ $\gg$

$$p(y).$$

Hence we can continue our reasoning

$$\sum_y p(y) F\left(\sum_z \frac{R_{z,y} p(z)}{p(y)} p_{X|z}\right)$$

$\geq$ $\ll$ (*) Jensen's inequality $\gg$

$$\sum_y p(y) \sum_z \frac{R_{z,y} p(z)}{p(y)} F(p_{X|z})$$

$=$ $\ll$ simplifying and reorganizing $\gg$

$$\sum_z p(z) F(p_{X|z}) \sum_y R_{z,y}$$

$=$ $\ll \sum_y R_{z,y} = 1 \gg$

$$\sum_z p(z) F(p_{X|z})$$

$=$ $\ll$ definition of $F(\pi, B)$ $\gg$

$$F(\pi, B)$$

so that $F(\pi, A) \geq F(\pi, B)$ as claimed.

Now suppose that $\mathcal{A} \neq \mathcal{B}$ and F is strictly concave.

A strict form of Jensen's inequality is that if $\lambda_1, \lambda_2, \ldots \lambda_n$ are coefficients in $[0,1]$ that sum to one, with at least one $\lambda_i \neq 1$, and F is strictly concave, and the x_n's are all distinct, then

$$\sum_n \lambda_n F(x_n) < F(\sum_n \lambda_n x_n).$$

We now show that this gives strict inequality at $(*)$ above. Because B is reduced, the distributions $p_{X|z}$ (the normalized columns of $D_\pi B$) are distinct; otherwise B would have similar columns. Those are the distinct x_n's for the strict Jensen's inequality.

We now consider the λ_n's, showing that at least one of them is not one. No two columns of R can have a single non-zero entry in the same row, since those two columns would generate similar columns in A, contradicting the fact that A is reduced. Thus, if all columns of R have exactly one non-zero value, since those values are alone in their rows and R is a channel matrix, in fact R must be a permutation of the identity. But that makes A a column permutation of B, impossible if A and B are reduced and distinct.

Therefore, channel matrix R must have some column $R_{-,y'}$ in which at least two entries are non-zero. But from $\sum_z R_{z,y'}p(z) = p(y')$, proved just above, plus the fact that $p(z)$ is nowhere zero, we have at least one z' (in fact, two) with $\frac{R_{z',y'}p(z')}{p(y')} \neq 1$. This z' gives the $\lambda_n \neq 1$ (for that y') that the strict Jensen's inequality requires.

These facts taken all together allow us to make step (*) above strict, since for all y's (the non-strict) Jensen's inequality applies, and for y' it applies strictly.

□

A simple consequence of Lemma 5.2.8 is the following theorem, which is itself of interest—it is a strict version of the classic *data-processing inequality* for Shannon leakage.

Theorem 5.2.9 (Strict data-processing inequality)**.** *Let $\mathcal{A}$ and $\mathcal{B}$ be distinct abstract channels with $(A, \mathcal{X}, \mathcal{Y})$ and $(B, \mathcal{X}, \mathcal{Z})$ their representation as reduced channels. If $\mathcal{A} \sqsubseteq_\circ \mathcal{B}$, then for any full-support prior π, the Shannon leakage of $\mathcal{A}$ is strictly less than than that of $\mathcal{B}$: that is $I(\pi, \mathcal{A}) < I(\pi, \mathcal{B})$.*

Proof. We appeal to the strict concavity $(\frown)$ of Shannon entropy [Gal68, p. 85] and use H for F in Lemma 5.2.8, to conclude that $H(\pi, A) > H(\pi, B)$. Hence $I(\pi, A) < I(\pi, B)$. □

Given Theorem 5.2.9, our desired result is now easy.

Theorem 5.2.10. $(\sqsubseteq_\circ)$ *is a partial order on abstract channels.*

Proof. Since $(\sqsubseteq_\circ)$ is reflexive and transitive, we just need to show that it is antisymmetric on abstract channels. Let $\mathcal{A}$ and $\mathcal{B}$ be distinct abstract channels with $(A, \mathcal{X}, \mathcal{Y})$ and $(B, \mathcal{X}, \mathcal{Z})$ their representation as reduced channels. If $A \sqsubseteq_\circ B \sqsubseteq_\circ A$ then by Theorem 5.2.9 we have, for any full-support prior π, that $I(\pi, \mathcal{A}) < I(\pi, \mathcal{B}) < I(\pi, \mathcal{A})$, which is impossible. □

We now illustrate Theorem 5.2.9 through a concrete example.

Example 5.2.11. *Let A, B, and R be as shown, satisfying $A = BR$:*

$$\begin{array}{|c|ccc|} \hline A & y_1 & y_2 & y_3 \\ \hline x_1 & \frac{7}{12} & \frac{1}{3} & \frac{1}{12} \\ x_2 & \frac{3}{8} & \frac{3}{8} & \frac{1}{4} \\ \hline \end{array} = \begin{array}{|c|ccc|} \hline B & z_1 & z_2 & z_3 \\ \hline x_1 & \frac{1}{2} & \frac{1}{4} & \frac{1}{4} \\ x_2 & \frac{1}{4} & \frac{3}{4} & 0 \\ \hline \end{array} \cdot \begin{array}{|c|ccc|} \hline R & y_1 & y_2 & y_3 \\ \hline z_1 & \frac{1}{2} & \frac{1}{2} & 0 \\ z_2 & \frac{1}{3} & \frac{1}{3} & \frac{1}{3} \\ z_3 & 1 & 0 & 0 \\ \hline \end{array}$$

Note that A and B are distinct reduced matrices where $A \sqsubseteq_\circ B$.

Given prior distribution $\pi = (\frac{3}{4}, \frac{1}{4})$, we get joint matrices $D_\pi A$ and $D_\pi B$, and the factorization is preserved (since $A = BR$ implies $D_\pi A = D_\pi BR$):

$$\begin{array}{|c|ccc|} \hline D_\pi A & y_1 & y_2 & y_3 \\ \hline x_1 & \frac{7}{16} & \frac{1}{4} & \frac{1}{16} \\ x_2 & \frac{3}{32} & \frac{3}{32} & \frac{1}{16} \\ \hline \end{array} = \begin{array}{|c|ccc|} \hline D_\pi B & z_1 & z_2 & z_3 \\ \hline x_1 & \frac{3}{8} & \frac{3}{16} & \frac{3}{16} \\ x_2 & \frac{1}{16} & \frac{3}{16} & 0 \\ \hline \end{array} \cdot \begin{array}{|c|ccc|} \hline R & y_1 & y_2 & y_3 \\ \hline z_1 & \frac{1}{2} & \frac{1}{2} & 0 \\ z_2 & \frac{1}{3} & \frac{1}{3} & \frac{1}{3} \\ z_3 & 1 & 0 & 0 \\ \hline \end{array}$$

Notice that column y_1 of $D_\pi A$ is a linear combination of the columns of $D_\pi B$, with column y_1 of R as coefficients:

$$D_\pi A_{\_,y_1} = \frac{1}{2} D_\pi B_{\_,z_1} + \frac{1}{3} D_\pi B_{\_,z_2} + 1 D_\pi B_{\_,z_3}$$

Also, we can see from row z_1 of R that column z_1 of $D_\pi B$ gets split into two equal pieces, which are used in forming columns y_1 and y_2 of $D_\pi A$.

We can also rewrite each column of the joint matrices as the scalar product of the column sum (which is the probability of the corresponding output) and the normalized column (which is the corresponding posterior distribution):

$$D_\pi A_{\_,y_1} = \frac{17}{32}\begin{bmatrix} \frac{14}{17} \\ \frac{3}{17} \end{bmatrix} \quad D_\pi A_{\_,y_2} = \frac{11}{32}\begin{bmatrix} \frac{8}{11} \\ \frac{3}{11} \end{bmatrix} \quad D_\pi A_{\_,y_3} = \frac{1}{8}\begin{bmatrix} \frac{1}{2} \\ \frac{1}{2} \end{bmatrix}$$

$$D_\pi B_{\_,z_1} = \frac{7}{16}\begin{bmatrix} \frac{6}{7} \\ \frac{1}{7} \end{bmatrix} \quad D_\pi B_{\_,z_2} = \frac{3}{8}\begin{bmatrix} \frac{1}{2} \\ \frac{1}{2} \end{bmatrix} \quad D_\pi B_{\_,z_3} = \frac{3}{16}\begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

Hence the equation for $D_\pi A_{\_,y_1}$ can be rewritten to

$$\begin{aligned} \frac{17}{32}\begin{bmatrix} \frac{14}{17} \\ \frac{3}{17} \end{bmatrix} &= \frac{1}{2}\frac{7}{16}\begin{bmatrix} \frac{6}{7} \\ \frac{1}{7} \end{bmatrix} + \frac{1}{3}\frac{3}{8}\begin{bmatrix} \frac{1}{2} \\ \frac{1}{2} \end{bmatrix} + 1\frac{3}{16}\begin{bmatrix} 1 \\ 0 \end{bmatrix} \\ &= \frac{7}{32}\begin{bmatrix} \frac{6}{7} \\ \frac{1}{7} \end{bmatrix} + \frac{1}{8}\begin{bmatrix} \frac{1}{2} \\ \frac{1}{2} \end{bmatrix} + \frac{3}{16}\begin{bmatrix} 1 \\ 0 \end{bmatrix} \\ &= \frac{17}{32}\left(\frac{7}{17}\begin{bmatrix} \frac{6}{7} \\ \frac{1}{7} \end{bmatrix} + \frac{4}{17}\begin{bmatrix} \frac{1}{2} \\ \frac{1}{2} \end{bmatrix} + \frac{6}{17}\begin{bmatrix} 1 \\ 0 \end{bmatrix}\right) \end{aligned}$$

This shows that the posterior distribution $p_{X|y_1}$ can be written as a convex combination of the posterior distributions $p_{X|z_1}$, $p_{X|z_2}$, and $p_{X|z_3}$.

Finally, we can show the steps that establish that the posterior Shannon entropy of channel A is strictly greater than the posterior Shannon entropy of channel B: $H(\pi, A) > H(\pi, B)$.

$$
\begin{aligned}
& H(\pi, A) \\
= \;& \sum_y p(y) H(p_{X|y}) \\
= \;& \frac{17}{32} H \begin{bmatrix} \frac{14}{17} \\ \frac{3}{17} \end{bmatrix} + \frac{11}{32} H \begin{bmatrix} \frac{8}{11} \\ \frac{3}{11} \end{bmatrix} + \frac{1}{8} H \begin{bmatrix} \frac{1}{2} \\ \frac{1}{2} \end{bmatrix} \\
= \;& \frac{17}{32} H \left(\frac{7}{17} \begin{bmatrix} \frac{6}{7} \\ \frac{1}{7} \end{bmatrix} + \frac{4}{17} \begin{bmatrix} \frac{1}{2} \\ \frac{1}{2} \end{bmatrix} + \frac{6}{17} \begin{bmatrix} 1 \\ 0 \end{bmatrix} \right) + \frac{11}{32} H \left(\frac{7}{11} \begin{bmatrix} \frac{6}{7} \\ \frac{1}{7} \end{bmatrix} + \frac{4}{11} \begin{bmatrix} \frac{1}{2} \\ \frac{1}{2} \end{bmatrix} \right) \\
& + \frac{1}{8} H \begin{bmatrix} \frac{1}{2} \\ \frac{1}{2} \end{bmatrix} \\
> \;& \frac{17}{32} \left(\frac{7}{17} H \begin{bmatrix} \frac{6}{7} \\ \frac{1}{7} \end{bmatrix} + \frac{4}{17} H \begin{bmatrix} \frac{1}{2} \\ \frac{1}{2} \end{bmatrix} + \frac{6}{17} H \begin{bmatrix} 1 \\ 0 \end{bmatrix} \right) + \frac{11}{32} \left(\frac{7}{11} H \begin{bmatrix} \frac{6}{7} \\ \frac{1}{7} \end{bmatrix} + \frac{4}{11} H \begin{bmatrix} \frac{1}{2} \\ \frac{1}{2} \end{bmatrix} \right) \\
& + \frac{1}{8} H \begin{bmatrix} \frac{1}{2} \\ \frac{1}{2} \end{bmatrix} \\
= \;& \left(\frac{7}{32} H \begin{bmatrix} \frac{6}{7} \\ \frac{1}{7} \end{bmatrix} + \frac{1}{8} H \begin{bmatrix} \frac{1}{2} \\ \frac{1}{2} \end{bmatrix} + \frac{3}{16} H \begin{bmatrix} 1 \\ 0 \end{bmatrix} \right) + \left(\frac{7}{32} H \begin{bmatrix} \frac{6}{7} \\ \frac{1}{7} \end{bmatrix} + \frac{1}{8} H \begin{bmatrix} \frac{1}{2} \\ \frac{1}{2} \end{bmatrix} \right) \\
& + \frac{1}{8} H \begin{bmatrix} \frac{1}{2} \\ \frac{1}{2} \end{bmatrix} \\
= \;& \frac{7}{32} H \begin{bmatrix} \frac{6}{7} \\ \frac{1}{7} \end{bmatrix} + \frac{7}{32} H \begin{bmatrix} \frac{6}{7} \\ \frac{1}{7} \end{bmatrix} + \frac{1}{8} H \begin{bmatrix} \frac{1}{2} \\ \frac{1}{2} \end{bmatrix} + \frac{1}{8} H \begin{bmatrix} \frac{1}{2} \\ \frac{1}{2} \end{bmatrix} + \frac{1}{8} H \begin{bmatrix} \frac{1}{2} \\ \frac{1}{2} \end{bmatrix} + \frac{3}{16} H \begin{bmatrix} 1 \\ 0 \end{bmatrix}
\end{aligned}
$$

$$
\begin{aligned}
&= \frac{7}{16}H\begin{bmatrix} \frac{6}{7} \\ \frac{1}{7} \end{bmatrix} + \frac{3}{8}H\begin{bmatrix} \frac{1}{2} \\ \frac{1}{2} \end{bmatrix} + \frac{3}{16}H\begin{bmatrix} 1 \\ 0 \end{bmatrix} \\
&= \sum_z p(z)H(p_{X|z}) \\
&= H(\pi, B)
\end{aligned}
$$

Recall now that by the strict concavity of Shannon entropy and the strict Jensen's inequality, the Shannon entropy of a convex combination of distinct distributions (with at least one coefficient $\lambda_i \neq 1$) is strictly greater than the convex combination of the entropies of the distributions; note that the first two terms in the crucial "$>$" step are of this form.

Finally, we have $I(\pi, A) = H(\pi) - H(\pi, A) < H(\pi) - H(\pi, B) = I(\pi, B)$. □

We now look at the link between composition refinement and reduced channels. For channels A and B we write $A \equiv_\circ B$ to mean that A and B are equivalent with respect to composition refinement, that is, $A \sqsubseteq_\circ B$ and $B \sqsubseteq_\circ A$.

Theorem 5.2.12. *For any channel C we have that $C \equiv_\circ C^r$.*

Proof. The reduced form C^r of channel C is defined via a series of operations: deleting all-zero columns, summing similar columns, and permuting columns. Each of those can be achieved by cascading channel C with a channel matrix that performs the post-processing with a matrix R_1 so that $C^r = CR_1$. Hence, $C^r \sqsubseteq_\circ C$.

For the reverse direction the operations are adding an all-zero column, splitting a column into several similar columns, and reordering columns. Again, all of these can be achieved by cascading with a matrix R_2 so that $C = C^r R_2$. Hence, $C \sqsubseteq_\circ C^r$, and have proved that $C \equiv_\circ C^r$.

□

We remark that, in the proof above, by multiplying C on the right with R_1 we cannot delete non-zero columns, so it is essential for the proof of this theorem that we need only delete all-zero columns from C to obtain C^r. Also, by multiplying C^r on the right with R_2, we cannot decompose a column into a sum of dissimilar columns, so it is essential that we need only sum similar columns of C to obtain C^r.

Theorem 5.2.13. *$A \equiv_\circ B$ iff $A^r = B^r$.*

Proof. If $A \equiv_\circ B$ then by Theorem 5.2.12 we have that $A^r \equiv_\circ A \equiv_\circ B \equiv_\circ B^r$, so by transitivity $A^r \equiv_\circ B^r$ and by the antisymmetry of $\sqsubseteq_\circ$ (Theorem 5.2.10) we conclude that $A^r = B^r$. On the other hand, if $A^r = B^r$, then by reflexivity of $\sqsubseteq_\circ$ we get that $A^r \equiv_\circ B^r$, so by Theorem 5.2.12 and transitivity we find that $A \equiv_\circ B$. $\square$

We conclude this section reporting that preliminary investigations by McIver, Morgan and Meinicke [MEM+13] suggest that composition refinement is not a lattice, as is the case with partition refinement. Future research should confirm this conjecture and analyze its implications for the quantitative information flow analysis of programs.

5.3 Equivalence of the Structural and Leakage Orderings

We have so far seen that composition refinement is a partial order on abstract channels (Theorem 5.2.10) so, in order to determine whether it generalizes partition refinement to probabilistic channels, we now consider its relationship to leakage ordering relations.

First, let us consider the situations in which $A \sqsubseteq_\circ B$ implies that A leaks no more than B. We can argue informally that this should be the case for any reasonable leakage measure: if $A = BR$ for some channel R, then an adversary should never prefer channel A to channel B, because given channel B the adversary can always *simulate* channel A by simply post-processing the output from channel B according to channel R. And indeed this property does hold for Shannon leakage (Theorem 4.1.12), min-entropy leakage (Theorem 4.1.11), g-leakage [ACPS12], and, as we will explain in Section 5.4, guessing entropy leakage. It is a generalized *data-processing inequality*, proved here for the case of g-leakage.

Theorem 5.3.1. *For any channels* $(A, \mathcal{X}, \mathcal{Y})$ *and* $(B, \mathcal{X}, \mathcal{Z})$*, If* $A \sqsubseteq_\circ B$ *then the* g*-leakage of* A *never exceeds that of* B*, for any prior* π *and any gain function* g *(We denote this by* $A \leq_{\mathcal{G}} B$*.)*

Proof. Note first that given channel $(\mathcal{X}, \mathcal{Y}, C)$, because $\mathcal{L}_g(\pi, C) = \log \frac{V_g(\pi, C)}{V_g(\pi)}$ and $V_g(\pi, C)$ and $V_g(\pi)$ are positive, we have $\mathcal{L}_g(\pi, A) \leq \mathcal{L}(\pi, B)$ iff $V_g(\pi, A) \leq V_g(\pi, B)$. Also,

$$V_g(\pi, C) = \sum_y \max_w \sum_x \pi[x]C[x,y]g(w,x),$$

and we represent the adversary's strategy for choosing w, given y, as a probabilistic channel S from $\mathcal{Y}$ to $\mathcal{W}$. Hence we have,

$$\begin{aligned} V_g(\pi, C) &= \max_S \sum_{x,y,w} \pi[x]C[x,y]S[y,w]g(w,x) \\ &= \max_S \sum_{x,w} \pi[x](CS)[x,w]g(w,x). \end{aligned} \tag{5.1}$$

Now notice that in the case where $A = BR$, any optimal strategy S for A is *equivalent* to a strategy for B, namely RS; but of course RS might not be optimal for B—there might be a better strategy S'. This allows us to calculate

$$V_g(\pi, A)$$

$$= \qquad \ll \text{equation } 5.1 \gg$$

$$\max_S \sum_{x,w} \pi[x](AS)[x,w]g(w,x)$$

$$= \qquad \ll A = BR \gg$$

$$\max_{S} \sum_{x,w} \pi[x](BRS)[x,w]g(w,x)$$

$$\leq \qquad \ll S' \text{ can be } RS \gg$$

$$\max_{S'} \sum_{x,w} \pi[x](BS')[x,w]g(w,x)$$

$$= \qquad \ll \text{equation 5.1} \gg$$

$$V_g(\pi, B)$$

which gives the inequality $V_g(\pi, A) \leq V_g(\pi, B)$ that we seek. □

Now, if $A \not\sqsubseteq_\circ B$, does it mean that there exist a prior π and gain function g that causes A to leak strictly more than B? The following theorem, due to McIver et al. [MMM12], establishes exactly that: the strong g-leakage order implies composition refinement. This implication was first studied in [ACPS12], but not proved in full generality—it was shown only in the case when the columns of B are linearly independent—and the general result was as denominated the *Coriaceous Conjecture*.

Theorem 5.3.2. *For any channels* $(A, \mathcal{X}, \mathcal{Y})$ *and* $(B, \mathcal{X}, \mathcal{Z})$*, if* $A \leq_{\mathcal{G}} B$ *then* $A \sqsubseteq_\circ B$*.*

Proof. We argue the contrapositive, showing that if $A \not\sqsubseteq_\circ B$, then we can construct a gain function g and a prior π such that $V_g(\pi, A) > V_g(\pi, B)$; note that this implies that $\mathcal{L}_g(\pi, A) > \mathcal{L}_g(\pi, B)$ and hence $A \not\leq_{\mathcal{G}} B$.

If $A \not\sqsubseteq_\circ B$, then there exists no channel $(R, \mathcal{X}, \mathcal{Y})$ such that $A = BR$. If we write $B^\uparrow$ for the channel matrices $\{BR \mid R \text{ is any channel from } \mathcal{Z} \text{ to } \mathcal{Y}\}$, then our assumption becomes $A \notin B^\uparrow$.

Because matrix A and the matrices in $B^\uparrow$ go from $\mathcal{X}$ to $\mathcal{Y}$, they can be embedded into Euclidean space of dimension $N = |\mathcal{X}| \times |\mathcal{Y}|$ by gluing their columns together in order. Then $B^\uparrow$ becomes a set of points in N-space which we observe by linearity of matrix multiplication is both convex and closed. Furthermore, A is a point in N-space that does not belong to $B^{\uparrow}$.

By the *Separating Hyperplane Lemma* [Tru71] there is therefore a hyperplane in N-space such that point A strictly on one side, an all of the set $B^{\uparrow}$ strictly on the other side. If G is the normal of the hyperplane, also an N-vector, this gives us that $A \cdot G > B' \cdot G$ for all $B' \in B^{\uparrow}$, were $(\cdot)$ denotes the dot product of the vectors. Note that we can assume a $(>)$-separation without loss of generality, because we can negate G if necessary. Moreover we can assume without loss of generality that the elements of G are in $[0,1]$. First, we can eliminate negative elements of G by adding a constant k to each entry; this has the effect of increasing both sides of the inequalities above by exactly $k|\mathcal{X}|$, since A and each B' are channel matrices, so as vectors they all sum to $|\mathcal{X}|$. Second, we can eliminate elements of G that are greater than 1 by scaling G, which simply scales both sides of the inequalities.

Now by "ungluing" we can view G, a vector in N-space, as a matrix (though not necessarily a channel matrix) from $\mathcal{X}$ to $\mathcal{Y}$. Hence we can view G as a *gain function* $g : \mathcal{Y} \times \mathcal{X} \to [0,1]$ using $\mathcal{Y}$ as the set of guesses and defined by $g(y,x) = G[x,y]$.

It turns out that this g is precisely the gain function that causes A to leak more than B under the uniform prior π_u. For by the definition of g-vulnerability we have

$$V_g(\pi_u, A) = \max_{S_A} \sum_{x,y} \pi_u[x](AS_A)[x,y]g(y,x)$$

and

$$V_g(\pi_u, B) = \max_{S_B} \sum_{x,y} \pi_u[x](AS_B)[x,y]g(y,x)$$

where strategies S_A for A are channel matrices from $\mathcal{Y}$ to $\mathcal{Y}$ and strategies S_B for B are channel matrices from $\mathcal{Z}$ to $\mathcal{Y}$. Note then that the identity matrix I is a strategy for A, and that each BS_B is in $B^{\uparrow}$. Hence, letting S_B^o denote any optimal strategy for B, we have

$$
\begin{array}{ll}
 & V_g(\pi_u, B) \\
= & \qquad \ll S_B^o \text{ is optimal} \gg \\
 & \sum_{x,y} \pi_u[x](BS_B^o)[x,y]g(y,x) \\
= & \qquad \ll \pi_u \text{ is uniform over } \mathcal{X} \gg \\
 & \frac{1}{|\mathcal{X}|}\sum_{x,y}(BS_B^o)[x,y]G[x,y] \\
= & \qquad \ll \text{taking dot-product in vector form} \gg \\
 & \frac{1}{|\mathcal{X}|}\sum_{x,y}(BS_B^o) \cdot G \\
< & \qquad \ll \text{Separating Hyperplane Lemma and } BS_B^o \in B^{\uparrow} \gg \\
 & \frac{1}{|\mathcal{X}|}\sum_{x,y} A \cdot G \\
= & \qquad \ll \text{multiplying by the identity } I \gg \\
 & \sum_{x,z} \pi_u[x](AI)[x,y]g(y,x) \\
= & \qquad \ll S_A \text{ can be } I \gg \\
 & \sum_{x,z} \pi_u[x](AS_A)[x,y]g(y,x) \\
\leq & \qquad \ll \text{definition of } V_g \gg \\
 & V_g(\pi_u, A).
\end{array}
$$

□

Hence, the structural order given by composition refinement is equivalent to the strong g-leakage order, so composition refinement is a compelling generalization of partition refinement to probabilistic channels.

Note, however, that the strong leakage ordering with respect to other leakage measures may not be sufficient to guarantee their composition refinement ordering. In particular, as we will show in the following section, the strong min-entropy order is strictly weaker than composition refinement. But we do not know whether the strong leakage orders with respect to Shannon entropy or guessing entropy are also strictly weaker.

5.4 Relationship to Other Leakage Orderings

We have so far proved that composition refinement is equivalent to the strong g-leakage ordering ($\leq_{\mathcal{G}}$), and thus, can be understood as a generalization of partition refinement from deterministic to probabilistic channels. However, recall that partition refinement is associated to a robust leakage ordering, that is, a leakage ordering not just with respect to g-leakage, but with respect to any of the leakage measures (Theorem 5.2.1). In Section 5.3 we argued informally that any reasonable leakage measure should consider that if $A = BR$ for some channel R, then an adversary should never prefer channel A to channel B. In this section we prove that indeed composition refinement implies strong leakage orderings also for guessing entropy leakage, Shannon leakage, and min-entropy leakage. Therefore, these are all weaker leakage ordering relations than ($\leq_{\mathcal{G}}$). We define these strong leakage orderings as follows:

Definition 5.4.1 (Strong guessing entropy leakage order). *Channels A and B are in the strong guessing entropy leakage order, written $A \leq_{guessing} B$, if $I_G(\pi, A) \leq I_G(\pi, B)$ for any prior distribution π.*

Definition 5.4.2 (Strong Shannon leakage order). *Channels A and B are in the strong Shannon leakage order, written $A \leq_{Shannon} B$, if $I(\pi, A) \leq I(\pi, B)$ for any prior distribution π.*

Definition 5.4.3 (Strong min-entropy entropy leakage order). *Channels A and B are in the strong min-entropy leakage order, written $A \leq_{min-entropy} B$, if $\mathcal{L}(\pi, A) \leq \mathcal{L}(\pi, B)$ for any prior distribution π.*

However, we explain that ($\leq_{min-entropy}$) does not imply composition refinement, and thus ($\leq_{min-entropy}$) is *strictly* weaker than ($\leq_{\mathcal{G}}$). In addition, we conjecture that ($\leq_{Shannon}$) is also strictly weaker than ($\leq_{\mathcal{G}}$). It is not yet clear, however, whether ($\leq_{guessing}$) implies ($\leq_{\mathcal{G}}$).

5.4.1 Guessing Entropy Leakage

Recall from Section 2.2.2 that guessing entropy is the expected number of guesses, using an optimal guessing strategy, to correctly guess the value of X. Note that the adversary's optimal guessing strategy consists of guessing the values of X in non-increasing order of probability. Hence, if the elements of $\mathcal{X}$ are indexed in non-increasing order with respect to their probability $\pi[x_i]$, the guessing entropy is

$$G(\pi) = \sum_{i=1}^{n} i\pi[x_i].$$

In this section we show that composition refinement implies the strong guessing entropy leakage ordering ($\leq_{guessing}$) by using Lemma 5.2.8 to derive a version of data-processing inequality for guessing entropy. In order to apply Lemma 5.2.8, we first prove that guessing entropy is a concave function.

Theorem 5.4.4. *Guessing entropy is a concave function.*

Proof. For our argument, we use an auxiliary sorting function $\texttt{sort}(\pi)$ that takes a distribution π and sorts it in non-increasing order of probability. With this function, we no longer need to assume a special indexing of the distribution π in the definition of $G(\pi)$ which is now given by

$$G(\pi) = \sum_{i=1}^{n} i(\texttt{sort}(\pi))[x_i].$$

The first step in our proof is to observe that an arbitrary convex combination of two probability distributions yields a higher guessing entropy than it would if we first sort the probability distributions before combining them. We express this formally in the following inequality, where p and q are probability distributions and λ is a scalar between 0 and 1.

$$G(\lambda p + (1 - \lambda)q) \geq G(\lambda \texttt{sort}(p) + (1 - \lambda)\texttt{sort}(q)) \quad (*)$$

Intuitively, sorting the distributions ahead of the combination step will cause the larger probability values in both distributions to be combined together, leaving less probability mass available for the smaller elements of the resulting probability distribution. But such smaller elements have a greater weight—given by the adversary's guessing attempt number—in the formula of guessing entropy, so the greater the probability mass we make available for later guessing attempts the greater the guessing entropy of the probability distribution.

We now continue the proof, showing that guessing entropy is concave since it satisfies Definition 5.2.5.

$$G(\lambda p + (1 - \lambda)q)$$

$$\geq \qquad \ll \text{by } (*) \gg$$

$$G(\lambda \texttt{sort}(p) + (1 - \lambda)\texttt{sort}(q))$$

$$= \qquad \ll \text{definition of } G(\pi) \gg$$

$$\sum_{i=1}^{n} i\left(\texttt{sort}(\lambda \texttt{sort}(p) + (1 - \lambda)\texttt{sort}(q))\right)[x_i]$$

$$= \qquad \ll \lambda \texttt{sort}(p) + (1 - \lambda)\texttt{sort}(q) \text{ is already sorted} \gg$$

$$\sum_{i=1}^{n} i(\lambda\mathtt{sort}(p) + (1-\lambda)\mathtt{sort}(q))[x_i]$$

$$= \qquad \ll \text{distributive law} \gg$$

$$\sum_{i=1}^{n} i(\lambda\mathtt{sort}(p))[x_i] + \sum_{i=1}^{n} i((1-\lambda)\mathtt{sort}(q))[x_i]$$

$$= \qquad \ll \text{pushing } \lambda \text{ and } 1-\lambda \text{ outside of the sums} \gg$$

$$\lambda\sum_{i=1}^{n} i(\mathtt{sort}(p))[x_i] + (1-\lambda)\sum_{i=1}^{n} i(\mathtt{sort}(q))[x_i]$$

$$= \qquad \ll \text{definition of } G(\pi) \gg$$

$$\lambda G(p) + (1-\lambda)G(q).$$

□

Now that we have proved that guessing entropy is concave, we can use Lemma 5.2.8 to show that guessing entropy satisfies the data processing inequality.

Theorem 5.4.5 (Data-processing inequality for guessing entropy)**.** *Let A and B be channels such that $A = BR$ for some channel R. Then $I_G(\pi, A) \leq I_G(\pi, B)$ for any prior distribution π.*

Proof. First, we show that if A is the cascade of B and R, that is, $A = BR$, then $A^r = B^r T$ for some channel matrix T.

$$A = BR$$

$$\Rightarrow \qquad \ll A^r \sqsubseteq_\circ A, \text{ so } A^r = AR_1 \text{ for some } R_1 \gg$$

$$A^r = (BR)R_1$$

$$\Rightarrow \qquad \ll B \sqsubseteq_\circ B^r, \text{ so } B = B^r R_2 \text{ for some } R_2 \gg$$

$$A^r = ((B^r R_2)R)R_1$$

$$\Rightarrow \qquad \ll \text{associativity of matrix multiplication} \gg$$

$$A^r = B^r(R_2 R R_1)$$

$$\Rightarrow \qquad \ll \text{Let } T = R_2 R R_1 \gg$$

$$A^r = B^r T$$

Since G is a concave function (Theorem 5.4.4) we can use G for F in Lemma 5.2.8 to conclude that $G(\pi, A^r) \geq G(\pi, B^r)$ for any full-support prior π. Moreover, given that $G(\pi, A^r) = G(\pi, A)$ (Theorem 5.1.3) we have showed that $G(\pi, A) \geq G(\pi, B)$ for any full-support prior π.

We are now left with proving that $G(\pi, A) \geq G(\pi, B)$ for any non full-support prior. If π is not full-support then the zero probability inputs do not contribute toward the calculation of the guessing entropy. Therefore, if we remove from A, B and π the rows and probabilities corresponding to those inputs we obtain truncated versions of the matrices and the prior distribution, which we denote with A', B' and π' respectively, with the property that that $G(\pi, A) = G(\pi', A')$ and $G(\pi, B) = G(\pi', B')$. Moreover, if $A = BR$ then row i of A is a combination of the rows of R with coefficients in row i of B, so after truncating rows from both A and B we still get that $A' = B'R$. Hence, we can apply Lemma 5.2.8 as before, but this time with the reduced versions of the truncated matrices A' and B' to conclude that $G(\pi, A') \geq G(\pi, B')$ for any full-support prior π. But given that π' is full-support, and that the guessing entropy is not affected by truncating the matrices and the prior, we have shown that if $A = BR$ then $G(\pi, A) \geq G(\pi, B)$ for *any* prior π.

At this point it follows, from the definition guessing entropy leakage, that whenever $A = BR$, we have that for all π:

$$\begin{aligned} G(\pi, A) \geq G(\pi, B) \quad & \text{iff} \quad -G(\pi, A) \leq -G(\pi, B) \\ & \text{iff} \quad G(\pi) - G(\pi, A) \leq G(\pi) - G(\pi, B) \\ & \text{iff} \quad I_G(\pi, A) \leq I_G(\pi, B) \end{aligned}$$

□

Therefore, $A \sqsubseteq_\circ B$ implies $A \leq_{guessing} B$, so from Theorem 5.3.2 we have that

$$A \leq_{\mathcal{G}} B \Rightarrow A \sqsubseteq_\circ B \Rightarrow A \leq_{guessing} B.$$

Hence, the strong guessing entropy leakage ordering ($\leq_{guessing}$) is a weaker leakage ordering than the strong g-leakage ordering ($\leq_{\mathcal{G}}$). Of course, the two relations could possibly be equal.

5.4.2 Shannon leakage

From the data-processing inequality (Theorem 4.1.12), it follows that composition refinement implies the strong Shannon leakage order ($\leq_{Shannon}$), that is, $A \sqsubseteq_\circ B$ implies $A \leq_{Shannon} B$. So we also have that $\leq_{Shannon}$ is a weaker leakage ordering relation than $\leq_{\mathcal{G}}$:

$$A \leq_{\mathcal{G}} B \Rightarrow A \sqsubseteq_\circ B \Rightarrow A \leq_{Shannon} B.$$

We remark that we can provide an alternative proof of the data-processing inequality by deriving it from Lemma 5.2.8 and appealing to the concavity of Shannon entropy [Gal68, p. 85]. We can simply follow the same argument we used to show the data-processing inequality for guessing entropy.

Furthermore, we conjecture that $\leq_{Shannon}$ is *strictly* weaker than $\leq_{\mathcal{G}}$ based on the following pair of channels from [ACPS12, Section VI].

A	y_1	y_2
x_1	$1/4$	$3/4$
x_2	$1/4$	$3/4$
x_3	$3/5$	$2/5$

B	z_1	z_2	z_3
x_1	$1/2$	0	$1/2$
x_2	0	$1/2$	$1/2$
x_3	$1/2$	$1/2$	0

It can be shown that $A \not\sqsubseteq_\circ B$, and yet, based on experimental evidence (first attempted by Alvim et al.[ACPS12]), it is likely that $A \leq_{Shannon} B$. In particular, after a brute force search approach we were unable to find a prior π that causes $I(\pi, A)$ to exceed $I(\pi, B)$. In addition, we observed that the surface plot of $I(\pi, B) - I(\pi, A)$ with respect to $\pi = (p_1, p_2, 1 - (p_1 + p_2))$ appears to lie above the zero plane.

5.4.3 Min-Entropy Leakage

We have three different ways to show that $A \leq_{\mathcal{G}} B$ implies $A \leq_{min-entropy} B$. The most straightforward is to note that min-entropy leakage is realized from g-leakage on the identity gain function, so it immediately follows that $(\leq_{min-entropy})$ is a weaker ordering relation than $(\leq_{\mathcal{G}})$. We can also appeal to Theorem 4.1.11, which is a data-processing inequality for min-entropy leakage, in combination with theorem 5.3.2 to conclude that

$$A \leq_{\mathcal{G}} B \Rightarrow A \sqsubseteq_\circ B \Rightarrow A \leq_{min-entropy} B.$$

We can also use Lemma 5.2.8, although in a slightly less direct way than we did for guessing entropy or Shannon entropy. Vulnerability is actually a convex function [BCP09], so instead of $V(\pi)$ we choose $-V(\pi)$ for the concave function F of Lemma 5.2.8. Note that with this choice of F,

$$F(\pi, C) = \sum_y p(y) F(p_{X|y}) = -\sum_y p(y) V(p_{X|y}) = -V(\pi, C).$$

We can then conclude that if $A = BR$ for some channel matrix R then $-V(\pi, A) \geq -V(\pi, B)$ for all π, or equivalently, $V(\pi, A) \leq V(\pi, B)$. Therefore, $\mathcal{L}(\pi, A) \leq \mathcal{L}(\pi, B)$ for all π so we have an alternative proof for Theorem 4.1.11.

However, $A \leq_{min-entropy} B$ does not imply $A \sqsubseteq_\circ B$. In fact, there exist channels A and B such that for all π, $\mathcal{L}(\pi, A) \leq \mathcal{L}(\pi, B)$ and yet $A \not\sqsubseteq_\circ B$. We illustrate this in the following example:

Example 5.4.6. *Let channels A and B be as follows:*

A	y_1	y_2
x_1	$2/3$	$1/3$
x_2	$2/3$	$1/3$
x_3	$1/4$	$3/4$

B	z_1	z_2	z_3
x_1	$1/2$	$1/2$	0
x_2	$1/2$	0	$1/2$
x_3	0	$1/2$	$1/2$

It can be verified using the linear programming algorithm from section VI.F of [ACPS12] that for all π, $\mathcal{L}(\pi, A) \leq \mathcal{L}(\pi, B)$. However, $A \not\sqsubseteq_{\circ} B$, since B is invertible and assuming that $A = BR$ for some R we find that $R = B^{-1}A$, but the unique matrix R that results from this calculation contains negative entries and, thus, is not a channel matrix.

$\square$

Therefore, the strong min-entropy leakage ordering is strictly weaker than the strong g-leakage ordering.

5.5 Expressing Other Leakage Measures as g-Leakages

In the previous section we showed that the strong leakage orderings with respect to guessing entropy, Shannon entropy, and min-entropy are all implied by the strong g-leakage ordering. In addition, we know that min-entropy leakage is the g-leakage resulting when we choose the g_{id} gain function. In this section we show that by generalizing gain functions beyond what was considered in [ACPS12], we are also able to express both guessing entropy leakage and Shannon leakage as *additive g-leakages*. Additive leakage measures leakage as the *difference* between the posterior and prior vulnerabilities, rather than the (logarithm of) their *ratio*; for min-entropy leakage, this idea was introduced in [BCP09].

In [ACPS12], gain functions are limited to functions $g : \mathcal{W} \times \mathcal{X} \to [0, 1]$, where $\mathcal{W}$ is a finite set of guesses. Here we make two major extensions: we allow the set $\mathcal{W}$ to be uncountably infinite, and we allow gain functions to return values in the range $[-\infty, \infty)$.

5.5.1 Guessing Entropy Leakage as a g-Leakage

We can express $G(\pi)$ as the negation of a g-vulnerability if we let $\mathcal{W}$ be the set of all *permutations* σ of $\mathcal{X}$. We can then define the gain function

$$g(\sigma, x) = -i$$

where i is the unique *index* of x within permutation σ, assumed to range from 1 to n. Note that this gain function has range in $[-n, -1]$.

It is easy to see that the permutation $\sigma = (x_1, x_2, \ldots, x_n)$ (based on the indexing above) realizes the "max" in the vulnerability. Hence we have

$$\begin{aligned}
V_g(\pi) &= \max_{w \in \mathcal{W}} \sum_{x \in \mathcal{X}} \pi[x] g(w, x) \\
&= \sum_{i=1}^{n} \pi[x_i] g((x_1, x_2, ..., x_n), x_i) \\
&= \sum_{i=1}^{n} \pi[x_i](-i) \\
&= -\sum_{i=1}^{n} i\pi[x_i] \\
&= -G(\pi).
\end{aligned}$$

Note that the g-vulnerability is always negative in this case. Also, as expected, a higher g-vulnerability corresponds to a lower guessing entropy.

The posterior guessing entropy is the expected g-vulnerability of the posterior distributions:

$$G(\pi, C) = \sum_{y \in \mathcal{Y}} p(y) G(p_{X|y}).$$

Hence with the $\mathcal{W}$ and g that we have defined, the posterior g-vulnerability becomes

$$\begin{aligned} V_g(\pi, C) &= \sum_{y \in \mathcal{Y}} p(y) V_g(p_{X|y}) \\ &= \sum_{y \in \mathcal{Y}} p(y) \left(-G(p_{X|y}) \right) \\ &= -G(\pi, C). \end{aligned}$$

Now that we know both $V_g(\pi)$ and $V_g(\pi, C)$, we can calculate the additive g-leakage:

$$\begin{aligned} \mathcal{L}_g^+(\pi, C) &= V_g(\pi, C) - V_g(\pi) \\ &= -G(\pi, C) - (-G(\pi)) \\ &= G(\pi) - G(\pi, C). \end{aligned}$$

This last expression is exactly the guessing entropy leakage $I_G(\pi, C)$. Thus, $I_G(\pi, C)$ can be expressed as an additive g-leakage if we allow gains outside of the range $[0, 1]$.

Note that we could additionally ensure that we have only positive g-vulnerabilities by adding n to all gain values, thus shifting the range to $[0, n-1]$. However, negative gains will still be necessary to express Shannon entropy below.

5.5.2 Shannon Leakage as a g-Leakage

By further relaxing gain functions, we can also express Shannon entropy as a (negative) g-vulnerability.

To do this, we let $\mathcal{W}$ be the (uncountably infinite) set of all probability distributions q on $\mathcal{X}$, and we define g by

$$g(q, x) = \log q[x].$$

Since $q[x]$ is in $[0, 1]$, this gain function has range $[-\infty, 0]$.

The g-vulnerability associated to this gain function is

$$V_g(p) = \max_{q \in \mathcal{W}} \sum_{x \in \mathcal{X}} p[x] \log q[x].$$

It turns out that V_g is realized when $q = p$, which implies that $V_g(p) = -H(p)$.

To prove this, we start with the equivalent formulation

$$V_g(p) = -\min_{q \in \mathcal{W}} \sum_{x \in \mathcal{X}} p[x] \log \frac{1}{q[x]}.$$

Then, we just need show that $\sum_{x \in \mathcal{X}} p[x] \log \frac{1}{q[x]}$ is minimized by choosing $q = p$, and that the minimum is $H(p)$. This follows immediately from the following lemma:

Lemma 5.5.1. *For all p and q, $H(p) \leq \sum_x p[x] \log \frac{1}{q[x]}$.*

Proof. The proof can be achieved using Gibbs' inequality [Mac03] which says that the relative entropy or Kullback-Leibler divergence [Mac03] of distribution q from distribution p, written $D_{KL}(p \| q)$ is non-negative. The Kullback-Leibler divergence is a non-symmetric measure of the similarity between two probability distributions and is defined as:

$$D_{KL}(p \| q) = \sum_x p[x] \log \frac{p[x]}{q[x]}$$

Our proof then proceeds as follows:

$$\begin{aligned}
\sum_x p[x] \log \frac{1}{q[x]} - H(p) &= \sum_x p[x] \log \frac{1}{q[x]} + \sum_x p[x] \log p[x] \\
&= \sum_x p[x] \log \frac{p[x]}{q[x]} \\
&= D_{KL}(p \| q) \\
&\geq 0.
\end{aligned}$$

□

The posterior Shannon entropy is then given by

$$H(\pi, C) = \sum_{y \in \mathcal{Y}} p(y) H(p_{X|y}).$$

Therefore, in the same way as the posterior guessing entropy, the posterior g-vulnerability becomes

$$\begin{aligned} V_g(\pi, C) &= \sum_{y \in \mathcal{Y}} p(y) V_g(p_{X|y}) \\ &= \sum_{y \in \mathcal{Y}} p(y) \left(-H(p_{X|y})\right) \\ &= -H(\pi, C). \end{aligned}$$

With our choice of gain function, the additive g-leakage coincides with the Shannon leakage:

$$\begin{aligned} \mathcal{L}_g^+(\pi, C) &= V_g(\pi, C) - V_g(\pi) \\ &= -H(\pi, C) - (-H(\pi)) \\ &= H(\pi) - H(\pi, C) \\ &= I(\pi, C). \end{aligned}$$

Note, however, that with this gain function the gain values are unbounded, so we cannot get rid of the negative g-vulnerabilities by adding a constant to all gain values like we suggested in the case of guessing entropy.

5.6 Compositionality of Abstract Channels

This section is a note of caution about abstract channels. The reader might be surprised to learn that abstract channels are not compositional, meaning that the abstract channel of a composition of channels cannot be determined based on the abstract channels of its constituents. For example, given reduced channels A^r and B^r, we do not have enough information to calculate the reduced channel $(AB)^r$ of the cascade of A and B.

To understand this, recall that an abstract channel is just a mapping from prior distributions to hyper-distributions. Indeed, when we calculate the reduced channel we discard all the information about the original column labels of the channel and their order. As a consequence, if we try to cascade reduced channels A^r and B^r we will surely find ourselves in trouble. First of all, the sizes of the matrices may no longer match to allow for matrix multiplication. But most importantly, the columns of A^r do not represent real channel outputs and therefore it would be semantically incorrect to match the outputs of A^r with the inputs to B^r in order to deduce the matrix of the cascade.

But this issue is not exclusive to probabilistic channels. The same thing occurs for deterministic channels and the partitions they induce on the set of secret inputs $\mathcal{X}$. Given the partition of $\mathcal{X}$ induced by deterministic channel C we know that all the inputs within a particular block of the partition map to the same output of C, but we do not know to which one specifically. Therefore, given just the partitions induced by deterministic channels A and B, we cannot calculate the partition associated to AB.

However, since reducing a channel has no effect on the set of inputs or their order, it is still possible to cascade a reduced channel matrix B^r after a concrete channel A. Curiously, by further reducing the resulting combined channel AB^r we can actually find the reduced channel $(AB)^r$ of the cascade without knowing matrix B:

$$(AB^r)^r = (AB)^r.$$

We can also cascade a reduced channel after any number of cascaded channels and obtain the reduced channel of the cascade:

$$(ABC^r)^r = (A(BC^r))^r = (A(BC^r)^r)^r = (A(BC)^r)^r = (ABC)^r.$$

In general, it is possible to obtain the reduced channel of the cascade in just one step. We can formally express this property if we follow Landauer and Redmond's work for deterministic systems [LR93] and define the # function of any channel A for any reduced channel B^r as follows:

$$A\#(B^r) = (AB)^r.$$

Then, we obtain the following theorem:

Theorem 5.6.1. *The composition of the # functions for channels A and B is equivalent to the # function of the cascade AB:*

$$A\# \circ B\# = (AB)\#.$$

Proof. Given channels $(\mathcal{X},\mathcal{Y},A)$ and $(\mathcal{Y},\mathcal{Z},B)$, let C^r be any reduced channel from $\mathcal{Z}$. Then,

$$\begin{aligned} A\# \circ B\#(C^r) &= A\#(B\#(C^r)) \\ &= A\#((BC)^r) \\ &= (A(BC))^r \\ &= ((AB)C)^r \\ &= (AB)\#(C^r) \end{aligned}$$

□

However, we remark that only the last link in a sequence of cascaded channels can be a reduced channel.

We end this section with a simple observation about the relationship between composition refinement and reduced channels. Note that if $A \sqsubseteq_\circ B$, then for some reduced channel C^r, we have that $A^r = (BC^r)^r$. This is just an immediate consequence of the fact that $(AB^r)^r = (AB)^r$ and the definition of composition refinement.

5.7 Related Work

There has long been interest in the robustness of information flow measures and on the leakage orderings on channels that they give. Such studies can both establish and refute relationships among measures. For instance, Massey [Mas94] compares Shannon entropy H and guessing entropy G, showing that $G(\pi) > 2^{H(\pi)-2}$, but that there is no interesting upper bound on $G(\pi)$ in terms of $H(\pi)$. Another negative result is given by Pliam [Pli00], who shows the incomparability of Shannon entropy and *marginal guesswork*, which is the minimum number of brute-force guesses required to guess a secret with some specified probability of success. With respect to vulnerability and min-entropy, Santhi and Vardy [SV06] prove a bound between posterior Shannon entropy and *Bayes risk*, which is the complement of posterior vulnerability; in our notation their bound can equivalently be written as $H(\pi, C) \geq -\log V(\pi, C) = H_\infty(\pi, C)$. Further study of similar bounds is done by Chatzikokolakis, Palamidessi, and Panangaden [CPP08b].

Turning to comparisons between channels, we have the results of Yasuoka and Terauchi [YT10] and Malacaria [Mal11] described in Section 1 that establish the robustness of partition refinement in comparing *deterministic* channels. For *probabilistic* channels, Braun, Chatzikokolakis, and Palamidessi [BCP09] compare the leakage ordering resulting form *multiplicative* and *additive* versions of min-entropy leakage. They show that when comparing two channels on a *given* prior, it makes no difference whether multiplicative or additive leakage is used. But when channels are compared with respect to their *capacity* then multipicative and additive leakage can produce inconsistent results.

Finally, Sabelfeld and Sands [SS01] describe a partial equivalence relation or "PER" model of security specifications, based on partitions of the hidden-value space; and there are some similarities between their treatment of partitions and ours: in particular, refining a PER that specifies a program's input could be understood as allowing the program to be less secure; and refining an output PER would require the program to be more secure. Their extension to probability, however, does not seem to lead to the same relation between channels as ours does.

5.8 Summary

In this chapter we looked into the mathematical foundations of quantitative information flow. We argued that, from the information-theoretic perspective, *abstract channels* are the fundamental objects of study. For when we consider the information leakage caused by channel C, the essential fact is precisely the mapping that C gives from prior distributions to hyper-distributions—and any of the multitude of possible leakage measures can be seen as simply *summarizing* this mapping. Concretely, then, we have seen that classical channel matrices contain structural redundancies, and that quotienting away these redundancies leads to *reduced channels*. The usefulness of the abstract-channel framework is further clarified by our study of *composition refinement*, which is only a pre-order on channel matrices, but which we have proved is a *partial order* on abstract channels. And, given that composition refinement coincides with the *strong g-leakage ordering* ($\leq_{\mathcal{G}}$), it is a partial order with both structural and information leakage characterizations—and is therefore a compelling generalization (from deterministic to probabilistic channels) of *partition refinement*.

Having showed the equivalence between composition refinement and the strong g-leakage ordering, we looked at the relationship with other leakage orderings, showing that composition refinement also implies strong leakage orderings with respect to guessing entropy, Shannon entropy and min-entropy. Note that these results, combined with the similar property for g-leakage [ACPS12], constitute a generalized data-processing inequality. We also remarked that the strong min-entropy leakage ordering is strictly weaker than $(\leq_{\mathcal{G}})$, and conjectured that this is also the case for the strong Shannon entropy ordering.

Since $(\leq_{\mathcal{G}})$ implies all the strong leakage orderings with respect to guessing entropy, Shannon entropy, and min-entropy, we wondered if g-leakage encompasses all of these measures. Indeed, we already knew that min-entropy leakage is a g-leakage, but we also proved that Shannon leakage and guessing entropy leakage can be expressed as *additive* g-leakages if we extend the gain functions considered in [ACPS12] to allow uncountably infinite sets of guesses and gain functions that take values in the range $[-\infty, \infty)$.

Finally, we remarked that abstract channels are not compositional, that is, the abstract channel of a composition of channels cannot be determined based only on the abstract channels of its constituents.

5.9 Credits

The first three sections of this chapter are based on our publication titled *Abstract Channels and their Robust Information-Leakage Ordering*, that will appear in the proceedings of the 3rd Conference on Principles of Security and Trust (POST 2014).

This paper is co-authored with Annabelle McIver, Carroll Morgan, Geoffrey Smith, and Larissa Meinicke.

The subsequent three sections are joint work with my advisor Geoffrey Smith.

The proof of Theorem 5.3.2 is due to Annabelle McIver, Carroll Morgan, and Larissa Meinicke.

The proof of Theorem 5.2.10 is my own work, although it was nicely split into multiple parts including Lemma 5.2.8 by Annabelle McIver and Carroll Morgan. An initial proof of this theorem (not presented here) that used an alternative geometric argument was due to Geoffrey Smith.

CHAPTER 6

CHANNEL MATRIX FACTORIZATION

In Chapter 4, we established upper bounds on the min-entropy leakage of a cascade of channels based on the leakage of its first link. Moreover, we proved in Chapter 5 that composition refinement is associated to a robust leakage ordering of channels, and that it is in fact the only way for a channel to never leak more information than another with respect to g-leakage. In light of the significance of composition refinement and cascading, in this chapter we set out to study techniques for decomposing a channel matrix into the cascade of two channel matrices. Note that if channel A can be decomposed into the cascade BR of channels B and R, we know that B composition refines A, and that the leakage of B is an upper bound for the leakage of A. Therefore, given a channel A, such techniques can be applied to find a channel B such $A \sqsubseteq_\circ B$, or, in general, in the analysis and design of secure systems.

Decomposing a channel into a cascade of two channels amounts to finding a *factorization* of its channel matrix into the product of two channel matrices. Hence, we are interested in solving a matrix factorization problem subject to the restriction that both the input matrix and the resulting matrix factors are row-stochastic, that is, their rows are non-negative and add up to one. At first glance, it may seem that this problem can be solved using classic numerical analysis methods for matrix decomposition. But the constraints imposed by row-stochastic matrix factorization make such methods inapplicable. Fortunately, and quite surprisingly, it turns out that this problem can be characterized as a non-negative matrix factorization problem (NMF), where the intent is to *approximate* a non-negative matrix as the product of two non-negative ones. This observation was first pointed out by Ho and Van Dooren [HvD08], who studied a method for NMF that preserves the row and column sums of the input matrix. Therefore, in order to factor a channel matrix into the product of channel matrices, we can take full advantage of the existing body of knowledge for solving NMF.

The contributions of this chapter include showing how NMF algorithms can be used to solve the problem of decomposing a channel into the cascade of two channels. We also describe a procedure that uses basic matrix multiplication properties to obtain a factorization of a channel matrix A into the product of two channel matrices, given a non-negative factorization of A. This procedure is directly derived from the proof from Ho and Van Dooren [HvD08] that any non-negative factorization of a row-stochastic matrix A is associated to a row-stochastic factorization of A. Finally, we suggest scenarios where channel matrix factorization can be applied to the analysis and design of secure systems.

Throughout this chapter we use the following notation. For any matrix X we denote the entry at the i-th row and j-th column with $X[i, j]$. Similarly, for any vector p we denote its i-th entry as $p[i]$. We also use the notation $A^{(i)}$ to refer to the i-th row of matrix A. In order to specify the dimensions of a matrix X of m rows and n columns we will write $X(m \times n)$. Finally, $\arg\min_x f(x)$ denotes the function that returns the values of x that minimize function f.

For a review of the linear-algebraic concepts we use in this chapter we recommend referring to Section 2.5.

The rest of this chapter is organized as follows. Section 6.1 gives a formal treatment of the channel matrix factorization problem. Section 6.2 describes a procedure to obtain a channel matrix factorization given a non-negative matrix factorization, and proposes a solution to the channel matrix factorization problem using non-negative matrix factorization with the generalized Kullback-Leibler (KL) divergence. Section 6.3 is a survey of the existing algorithms for solving NMF with the generalized KL divergence. Section 6.4 explains how to apply channel matrix factorization to the analysis and design of secure systems. Finally, Section 6.5 presents related work, and Section 6.6 summarizes the chapter.

6.1 Channel Matrix Factorization

We consider the problem of factoring a channel matrix into the product of two channel matrices. This is a matrix decomposition problem subject to the restriction that both the input matrix and the resulting factors are row-stochastic. A matrix $A(m \times n)$ is row-stochastic if its rows are non-negative and add up to one, that is, for any row index i,

$$\sum_{j=1}^{n} A^{(i)}[j] = 1.$$

We remark that, in general, an exact factorization of a channel matrix $A(m \times n)$ into channel matrix factors $B(m \times r)$ and $R(r \times n)$ might not exist for a specified factorization rank r, particularly if $r \leq min(m, n)$. To understand this, it is convenient to view matrix multiplication from a row-wise perspective: if $A = BR$, then row i of matrix A is a linear combination of the rows of R with coefficients in row i of B. Formally,

$$A^{(i)} = B^{(i)} R = \sum_{k=1}^{r} B^{(i)}[k] R^{(k)}.$$

But the rows of matrix B are stochastic vectors and, therefore, constitute sets of convex coefficients. Hence, the rows of A are convex combinations of the rows of R. Geometrically, this means that when solving a channel matrix factorization problem we are interested in finding the vertices of a convex polytope (rows of R) that falls within the space of probability distributions (so that the rows of R are stochastic vectors) and that, at the same time, contains all of our original data points (rows of A). Clearly, if $r < min(m, n)$, such polytope might not exist.

Therefore, we formulate the channel matrix factorization problem as that of *approximately* factoring a channel matrix A into the product of two channel matrix factors B and R. It is also important in our formulation to specify the *rank* of the factorization, which is the number of rows of the second matrix factor R, or equivalently, the number of vertices of the convex polytope that we want to find. We will also refer to the rows of matrix R as the basis vectors of the factorization. Formally, we define the channel matrix factorization problem as follows:

Problem 6.1.1 (Channel matrix factorization)**.** *Given a channel matrix $A(m \times n)$ find channel matrix factors $B(m \times r)$ and $R(r \times n)$ such that $r \leq \min(m,n)$ and:*

$$A \approx BR.$$

Notice that because the rows of both B and R are stochastic vectors, any approximate solution $\tilde{A} = BR$ is also necessarily row-stochastic.

The problem of channel matrix factorization can also be expressed as an optimization problem where we want to minimize some measure of dissimilarity between our original data matrix A and the resulting decomposition BR. Let f be our dissimilarity function. Then, we can formally define the optimization problem as:

Problem 6.1.2 (Channel matrix factorization as an optimization problem)**.** *Given a channel matrix $A(m \times n)$ find channel matrix factors $B(m \times r)$ and $R(r \times n)$ such that $r \leq \min(m,n)$ and $f(A, BR)$ is minimized with respect to B and R.*

Classic methods for matrix factorization such as LU, QR, or singular value decomposition [GVL12] cannot be applied to solve row-stochastic matrix factorization because they do not guarantee that the resulting factors will be non-negative, much less row-stochastic.

Our problem is closer to that of non-negative matrix factorization (NMF) [LS01] [PT94], where both the input matrix and the factors are bound to be non-negative. Formally, NMF is defined as:

Problem 6.1.3 (Non-Negative Natrix Factorization)**.** *Given a non-negative matrix* $X(m \times n)$ *find non-negative matrix factors* $W(m \times r)$ *and* $H(r \times n)$ *such that* $r \leq \min(m, n)$ *and:*

$$X \approx WH.$$

Similarly, NMF can be expressed as an optimization problem, by choosing a dissimilarity function f as follows:

Problem 6.1.4 (NMF as an optimization problem)**.** *Given a non-negative matrix* $X(m \times n)$ *find non-negative matrix factors* $W(m \times r)$ *and* $H(r \times n)$ *such that* $r \leq \min(m, n)$ *and* $f(X, WH)$ *is minimized with respect to* W *and* H*.*

At first glance, it may seem that the additional condition of row-stochasticity imposed by channel matrix factorization results in a much harder problem than NMF. However, in the next section we will show that, contrary to our intuition, given a non negative matrix factorization of a channel matrix A it is easy to obtain also a channel matrix factorization. This will allow us to take advantage of existing numerical optimization algorithms for solving NMF.

6.2 Non-Negative Matrix Factorization Characterization

Ho and Van Dooren [HvD08] proved that given a factorization of a row-stochastic matrix A into non-negative—but not necessarily row-stochastic—factors W and H it is possible to obtain row-stochastic factors B and R so that $A = BR$. Their proof is part of a more general theorem stating that any non-negative factorization WH of a matrix A has the form $WH = PDR$ where P is column-stochastic, D is diagonal non-negative, and R is row-stochastic. Furthermore, if A is row-stochastic, then matrix PD is also row-stochastic. Similarly, if A is column-stochastic, then matrix DR is also column-stochastic.

Based on their proof, in Algorithm 1 we present a procedure for transforming a non-negative matrix factorization of a channel matrix into a channel matrix factorization. Given a channel matrix A, and a non-negative factorization $A = WH$, the algorithm first normalizes the rows of H to obtain a channel matrix R. Then, it pushes the normalization constants into matrix W to obtain a channel matrix B. The output matrices B and R then satisfy $BR = WH$.

Algorithm 1 Convert NMF to channel matrix factorization

```
Require: non-negative matrices $W(m \times r)$, $H(r \times n)$ such that $WH$ is row-stochastic
Ensure: channel matrices $B(m \times r)$, $R(r \times n)$ such that $BR = WH$
$B \leftarrow 0(m \times r)$
$D \leftarrow 0(r \times r)$
$R \leftarrow 0(r \times n)$
{ Build diagonal matrix $D$ with row sums of $H$ }
for $k = 1$ to $r$ do
    $D[k,k] = \sum_{j=1}^{n} H^{(k)}[j]$
end for
{ Normalize rows of $H$ to obtain $R$ }
for $k = 1$ to $r$ do
    for $j = 1$ to $n$ do
        $R^{(k)}[j] = \frac{H^{(k)}[j]}{D[k,k]}$
    end for
end for
{ Push row sums of $H$ into $W$ to obtain $B$ }
$B \leftarrow WD$
return $B, R$
```

The following proof of the correctness of Algorithm 1 is a more leisurely written version of Ho and Van Dooren's original proof, simplified for the particular case of row-stochastic matrices.

Proof. After the loop in line 5, D is a diagonal matrix with the row sums of H on its diagonal. Then, the loop in line 9 normalizes the rows of H by dividing each row by the row sums stored in matrix D. Since multiplying on the left by a diagonal matrix scales the rows of a matrix by the factors in the diagonal, at this point W, D, and R satisfy:

$$WH = W(DR).$$

Then, on line 15 we set B to be the product of matrices W and D. Consequently, after this step, the matrix product BR is equal to the input matrix WH:

$$WH = W(DR) = (WD)R = BR.$$

So far we have proved that R is row stochastic and that $BR = WH$. Furthermore, we know that B must be non-negative, because that is the case for both matrices W and D. We now need to show that matrix B is also row-stochastic. We can see this by considering that both $WH = BR$ and R are row-stochastic. Then, it follows that the rows of B must add up to one:

$$\sum_{k=1}^{r} B[i,k]$$

$$= \qquad \ll \text{the rows of } R \text{ sum to one} \gg$$

$$\sum_{k=1}^{r} B[i,k] \sum_{j=1}^{n} R[k,j]$$

$$= \qquad \ll \text{distributive law} \gg$$

$$\sum_{k=1}^{r} \sum_{j=1}^{n} B[i,k]R[k,j]$$

$$= \qquad \ll \text{associativity} \gg$$

$$\sum_{j=1}^{n} \left(\sum_{k=1}^{r} B[i,k]R[k,j] \right)$$

$$= \qquad \ll \text{definition of matrix multiplication} \gg$$

$$\sum_{j=1}^{n} (BR)[i,j]$$

$$= \qquad \ll \text{the rows of } BR \text{ sum to one} \gg$$

$$1$$

□

We now present an example of how to transform a non-negative matrix factorization of a channel matrix, into a channel matrix factorization.

Example 6.2.1. *This example illustrates how to obtain a channel matrix factorization of a channel matrix A given a non-negative factorization $A = WH$. We follow the steps in Algorithm 1.*

$$\begin{array}{c|ccc} A & y_1 & y_2 & y_3 \\ \hline x_1 & 7/10 & 0 & 3/10 \\ x_2 & 0 & 5/8 & 3/8 \\ x_3 & 7/15 & 5/24 & 13/40 \end{array} = \begin{array}{c|cc} W & z_1 & z_2 \\ \hline x_1 & 0 & 3/5 \\ x_2 & 3/4 & 0 \\ x_3 & 1/4 & 2/5 \end{array} \cdot \begin{array}{c|ccc} H & y_1 & y_2 & y_3 \\ \hline z_1 & 0 & 5/6 & 1/2 \\ z_2 & 7/6 & 0 & 1/2 \end{array}$$

$$= \begin{array}{c|cc} W & z_1 & z_2 \\ \hline x_1 & 0 & 3/5 \\ x_2 & 3/4 & 0 \\ x_3 & 1/4 & 2/5 \end{array} \cdot \left(\begin{array}{c|cc} D & z_1 & z_2 \\ \hline z_1 & 4/3 & 0 \\ z_2 & 0 & 5/3 \end{array} \cdot \begin{array}{c|ccc} R & y_1 & y_2 & y_3 \\ \hline z_1 & 0 & 5/8 & 3/8 \\ z_2 & 7/10 & 0 & 3/10 \end{array} \right)$$

$$= \left(\begin{array}{c|cc} W & z_1 & z_2 \\ \hline x_1 & 0 & 3/5 \\ x_2 & 3/4 & 0 \\ x_3 & 1/4 & 2/5 \end{array} \cdot \begin{array}{c|cc} D & z_1 & z_2 \\ \hline z_1 & 4/3 & 0 \\ z_2 & 0 & 5/3 \end{array} \right) \cdot \begin{array}{c|ccc} R & y_1 & y_2 & y_3 \\ \hline z_1 & 0 & 5/8 & 3/8 \\ z_2 & 7/10 & 0 & 3/10 \end{array}$$

$$= \begin{array}{c|cc} B & z_1 & z_2 \\ \hline x_1 & 0 & 1 \\ x_2 & 1 & 0 \\ x_3 & 1/3 & 2/3 \end{array} \cdot \begin{array}{c|ccc} R & y_1 & y_2 & y_3 \\ \hline z_1 & 0 & 5/8 & 3/8 \\ z_2 & 7/10 & 0 & 3/10 \end{array}$$

□

Hence, in order to obtain a channel matrix factorization of a channel matrix A we can use NMF techniques to first find a non-negative factorization $A = WH$, and then follow Algorithm 1 to finally obtain channel matrix factors B and R. There is of course the challenge that NMF techniques generally return approximate factorizations of the original input matrix, so the approximation WH is not guaranteed to be row-stochastic. However, Ho and Van Dooren [HvD08] also showed that when the *generalized KL divergence*, defined below, is used as a dissimilarity function for NMF, the outputs of NMF algorithms preserve the row sums and column sums of the input matrix. In particular, this implies that if matrix A is row-stochastic, the approximation factors W and H are also row-stochastic. Therefore, in order to solve row-stochastic matrix factorization, we can leverage the existing literature of NMF with the generalized KL divergence.

The generalized KL divergence [CZPA09], also known as I-divergence, from matrix A to matrix B is defined as:

$$D(A||B) = \sum_{i,j}(A[i,j]log\frac{A[i,j]}{B[i,j]} - A[i,j] + B[i,j]).$$

$D(A||B)$ is a measure of the bits of information lost when B is used to approximate A. Some properties of $D(A||B)$ include that it is non-negative and that it is zero if and only if $A = B$. The generalized KL divergence is also not symmetric in A and B in the sense that $D(A||B) \neq D(B||A)$.

The generalized KL divergence belongs the class of Csiszár f-divergences, to the class of Amari alpha-divergences, and to the class of beta-divergences. It is also a Bregman divergence, and it reduces to the KL divergence D_{KL} when the entries in both A and B are non-negative and sum to 1.

The problem of NMF with the generalized KL divergence can be formalized as follows:

Problem 6.2.2 (NMF with the generalized KL divergence)**.** *Given a non-negative matrix $X(m \times n)$ find non-negative matrix factors $W(m \times r)$ and $H(r \times n)$ such that $r \leq \min(m,n)$ and $D(X\|WH)$ is minimized with respect to W and H. That is, find W and H that are a solution to:*

$$\arg\min_{W \geq 0, H \geq 0} D(X\|WH).$$

Let KL-NMF be some algorithm for solving problem 6.2.2. Using KL-NMF we can construct a procedure for solving the channel matrix factorization problem when the generalized KL divergence is the dissimilarity function. We present this procedure in algorithm 2.

Algorithm 2 Channel matrix factorization with the generalized KL divergence

```
Require: channel matrix $A(m \times n)$, factorization rank $r \leq \min(m,n)$
Ensure: channel matrices $B(m \times r)$, $R(r \times n)$ that are local minimizers of $D(A\|BR)$
$W, H \leftarrow$ KL-NMF$(A, r)$
Convert $WH$ to channel matrix factorization $BR$ using Algorithm 1
return $B, R$
```

We should remark that problem 6.2.2 is not convex in both W and H at the same time [LS01], so numerical optimization methods are not guaranteed to find a global minimizer of the dissimilarity function. Nevertheless, the dissimilarity function becomes convex if we fix either W or H. Existing algorithms for solving NMF with the generalized KL divergence take advantage of this property but can only guarantee to return a local minimizer of the dissimilarity function. Fortunately, these local minimizers are, in fact, the set of matrices that Ho and Van Dooren [HvD08] showed that preserve the row sums of the input matrix.

Another aspect that adds to the complexity of problem 6.2.2 is the possibility of multiple global minimizers of the dissimilarity function. However, as described by

Cichocki et al. [CZPA09, section 1.3.2], this problem is mitigated (except for scaling and permutation ambiguities) when the input matrix X is normalized. Such is the case of row-stochastic matrices.

In the next section we present a brief survey of the existing techniques for solving NMF with the generalized KL divergence.

6.3 Algorithms for Solving NMF with the Generalized KL Divergence

Since Lee and Seung proposed the first algorithm for solving NMF with the generalized KL divergence [LS01], researchers have come up with a variety of numerical approximation methods to solve the problem. Such methods can be classified based on the technique used to optimize the dissimilarity function. The most widespread techniques include multiplicative update rules, projected gradients, and quasi-Newton optimization [CZPA09].

Algorithms that use multiplicative update rules follow an iterative procedure where one of the matrices W or H is fixed while the other one is optimized by multiplying each of its entries by some factor. Lee and Seung's original algorithm [LS01], which uses this technique, can be understood as following a diagonally scaled gradient descent approach where we alternate between optimizing W and H. Another algorithm that falls into this category include the alpha SMART algorithm [CAZ+06, CZA06], which is based on the more general Amari alpha-divergence [CZPA09]. This algorithm optimizes the parameters using exponentiated gradient descent and provides a better convergence rate and efficiency than the method of Lee and Seung. Also, a multiplicative algorithm was recently proposed by Févotte and Idier [FI10] that handles the general case of the beta-divergences.

In contrast to multiplicative update rule algorithms, projected gradient algorithms use additive update rules to optimize the matrices W and H. Such algorithms alternate between optimizing W and H using the gradient descent method. Then, after each update step, all negative elements in either W or H are set to zero. The projected gradients method was first suggested by Lin [Lin07], but only developed for the case of the squared Euclidean distance. More recently, Yang et al. [YZYO11] proposed a version of the projected gradients method for optimizing the original KL divergence $D_{KL}(X||WH)$. They argue that when the input matrix has been normalized—as in the case of row-stochastic matrices—we can replace the generalized KL divergence with the original KL divergence as the dissimilarity function for NMF. Their paper also discusses some experiments where the proposed algorithm results in better approximations and faster running times than previously studied algorithms.

Quasi-Newton optimization methods are second order optimization methods, that is, they not only rely on the gradient of the dissimilarity function—which indicates the direction of the steepest descent—but also on its curvature. This additional consideration is aimed to help the algorithm find better solutions and improve the convergence rates. An example of this kind of method is the NMF with projected quasi-Newton optimization for alpha-divergences [ZC06]. At each optimization step, this algorithm sets the negative entries in the factors W and H to zero in order to enforce the non-negativity constraint.

Some challenges in the implementation of the previous algorithms include initializing the matrix factors W and H, choosing the learning rate for gradient descent based algorithms, and specifying stopping criteria. Due to the non-convex nature of the problem, NMF algorithms tend to be sensitive with respect to these parameters. See [CZPA09] for a survey of techniques for handling these problems.

Researchers have also made available software packages with implementations of NMF algorithms. For instance, *nmfpack* by Hoyer [Hoy06] and *NMFLAB* by Cichocki and Zdunek [CZ06] provide MATLAB® implementations of a variety of NMF algorithms.

6.4 Applicability

The channel matrix factorization technique from Algorithm 2 can be used to find channels that composition refine a given channel. Even though Algorithm 2 only guarantees to return a factorization that locally minimizes the dissimilarity function, this procedure may still find an exact factorization if it does exist.

To illustrate this, we implemented Algorithm 2 using the multiplicative update rules NMF algorithm from Lee and Seung [LS01]. Using this implementation, we were able to find the following exact factorization of matrix A from Example 6.2.1, by choosing $r = 2$:

$$\begin{array}{|c|ccc|} \hline A & y_1 & y_2 & y_3 \\ \hline x_1 & 7/10 & 0 & 3/10 \\ x_2 & 0 & 5/8 & 3/8 \\ x_3 & 7/15 & 5/24 & 13/40 \\ \hline \end{array} = \begin{array}{|c|cc|} \hline B & z_1 & z_2 \\ \hline x_1 & 1 & 0 \\ x_2 & 0 & 1 \\ x_3 & 2/3 & 1/3 \\ \hline \end{array} \cdot \begin{array}{|c|ccc|} \hline R & y_1 & y_2 & y_3 \\ \hline z_1 & 7/10 & 0 & 3/10 \\ z_2 & 0 & 5/8 & 3/8 \\ \hline \end{array}$$

Of course, we already knew that matrix A above has an exact factorization with factorization rank 2. But it is generally unknown which factorization rank would result in an exact factorization. One approach for addressing this challenge is to execute multiple runs of the algorithm with different factorization ranks. For example, we executed our implementation of Algorithm 2 for each possible factorization rank of a channel matrix $A(20 \times 20)$ generated by multiplying a channel matrix $B(20 \times 5)$ and a channel matrix $R(5 \times 20)$. Figure 6.1 shows the resulting generalized KL diver-

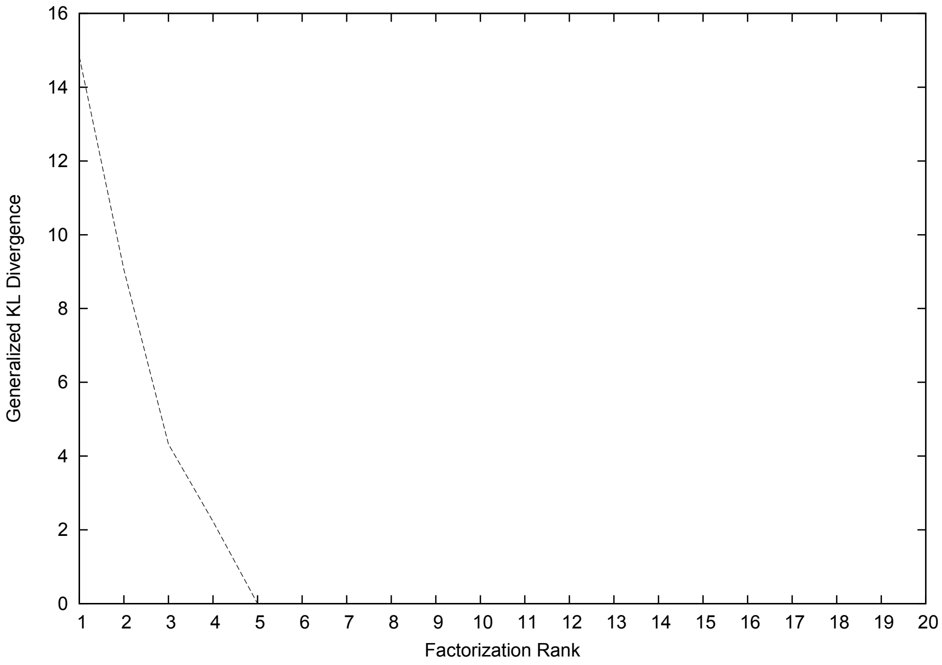

Figure 6.1: $D(A||BR)$ for each factorization rank of a channel matrix $A(20 \times 20)$.

gence for each run. As expected, we can observe that the divergence is (practically) zero starting at a rank of 5.

We can motivate another possible usage scenario of channel matrix factorization by recalling that the min-capacity of a cascade of channels $A = BR$ is upper bounded by the (logarithm of the) number outputs of channel B. Hence, if we can approximately factor A into the channel matrix product $\tilde{A} = BR$ with factorization rank r, the min-capacity of $\tilde{A}$ is at most $\log r$. Based on this observation, a potential application of Algorithm 2 is in statistical disclosure control [Geh10], where the intent is to reveal accurate statistics about a set of individuals while preserving their privacy. The idea would be to model a sensitive query as a channel matrix A, and then find an approximate channel matrix factorization $\tilde{A}$ of A with a factorization rank r chosen according to the maximum amount of min-entropy leakage desired.

Then, the approximation $\tilde{A}$ can be used instead of A in order to provide formal information flow security guarantees.

6.5 Related Work

An NMF problem that appears to be similar to that of channel matrix factorization is Convex-NMF [DLJ10], a method that is used for data clustering. The goal in Convex-NMF is to find a decomposition of a non-negative matrix X into non-negative matrix factors W and H, such that the rows of H are convex combinations of the rows of X. Therefore, with Convex-NMF, the rows of H fall within the convex hull of the rows of X. But this is an unnecessary restriction for channel matrix factorization, where the intent is actually the opposite: to find a channel matrix H such that the rows of X fall within the convex hull of the rows of H. Note that, in general, attempts to solve channel matrix factorization using data clustering algorithms results in the same issue.

We also considered using principal component analysis (PCA) [Shl05] to solve the problem of channel matrix factorization. PCA is a statistical procedure for dimensionality reduction of a data set. Given a set of points in $\mathbb{R}^n$, PCA finds an orthonormal basis that minimizes the approximation errors that result from projecting the original data points on the lower-dimensional subspace spanned by the first k basis vectors. The vectors of this basis are referred to as the principal components. PCA lets the first principal component lay on the axis where the projections of the points exhibit the highest variance. The subsequent principal components are chosen in decreasing order of variance.

In order to factor a channel matrix A into the channel matrix product BR we could think of using principal component analysis to first reduce the dimensionality of the rows of matrix A based on the desired factorization rank, and then use a convex hull algorithm to find the vertices of a convex polytope that contains the lower-dimensionality data set. However, both the lower-dimensionality data set, and the vertices of the convex polytope may fall outside of the space of stochastic vectors. Furthermore, restricting the convex hull algorithm to choose only stochastic vectors may result in a greater loss of precision.

6.6 Summary

We have described a general method for approximately factoring a channel matrix into the product of two channel matrices. Such method relies on existing algorithms for solving the NMF problem with the generalized KL divergence, and is based on the proof from Ho and Van Dooren [HvD08] that any non-negative factorization of a row-stochastic matrix is associated to a row-stochastic factorization. We also presented a brief survey of the existing algorithms for solving NMF with the generalized KL divergence, and classified them according to the numerical optimization method used for minimizing the dissimilarity function. Finally, we pointed out some readily available libraries for solving NMF with the generalized KL divergence, and identified some scenarios where channel matrix factorization can be applied to the analysis and design of secure systems.

6.7 Credits

The findings that I have presented in this chapter are the result of a paper I wrote for the Data Mining class I took during Fall 2011 under the supervision of Prof. Tao Li.

The characterization of channel matrix factorization using principal component analysis discussed in the related works section is the result of an earlier paper that I wrote for the Topics in Algorithms class I took during Spring 2010 under the supervision of Prof. Giri Narasimhan.

CHAPTER 7

CONCLUSION

In this thesis we have addressed four research problems in the area of quantitative information flow: (1) exploring the perspective that secrecy can be viewed as a resource that is gradually consumed by a system, (2) analyzing the information flow of combined channels, (3) determining the conditions that give rise to a robust leakage ordering of channels, and (4) studying techniques for factoring a channel matrix into the product of channel matrices.

We began our study by considering the consumption of secrecy in a system. After choosing min-entropy as our measure of secrecy, we looked into three different models for its consumption: a new dynamic model of min-entropy leakage that quantifies the information flow in a single run of the system, a new worst-case run model, and the generally discussed average-case model. We found that min-entropy does not behave as a reasonable resource in the dynamic model, since an adversary that tries to guess the value of the secret may find that its level of secrecy has increased after observing the output of the channel. We also remarked that the dynamic model makes policy enforcement difficult, in that stopping a potential leak during the execution of the system may actually reveal information about the secret. We then moved on to study the worst-case model, and found that min-entropy does behave as a resource in this case. However, a drawback of focusing on the leakage of the worst-case output of the channel is that the measurements end up being overly sensitive to bad outputs of the system that are possibly highly unlikely, as is the case of an adversary that guesses the correct password for a particular user. We thus looked back at the average-case model of min-entropy leakage, which apart from conforming to the viewpoint of secrecy as a resource, does not exhibit any of the drawbacks of the previous models. Even so, the worst-case model may still be of interest depending on the scenario being studied, as highlighted by the information flow analysis of the Crowds anonymity protocol which we discussed. Moreover, it might be useful to keep in mind that the worst-case min-entropy leakage gives an upper bound for the average min-entropy leakage and, therefore, for the min-capacity of a channel.

Having considered the min-entropy consumption within a system, we moved on to study its consumption when multiple systems are combined. We showed that min-entropy leakage satisfies a number of compositionality results that allow the leakage of a complex system to be bounded by the leakage of its constituents. First of all, we proved that the min-entropy leakage of a cascade of two channels is upper bounded by the min-entropy leakage of the first channel. However, such upper bound does not hold with respect to the second channel of the cascade. But when we turned our attention to min-capacity, we found that both channels of the cascade behave as bottlenecks to the information flow of the combined channel. Moving on to other channel composition operators, we studied the information flow when repeated independent runs of a channel are allowed, showing that the min-capacity of the combined channel grows logarithmically with respect to the number of runs, a result that was first proved by Köpf and Smith within the context of timing attacks against a cryptosystem. Regarding the min-entropy leakage in an adaptive composition of channels, we reviewed the results from Barthe and Köpf who showed that the min-capacity of the combined channel is upper bounded by the sum of the min-capacities of both channels. Going further, we analyzed the case of non-adaptive composition where the second channel ignores the output from first one, extended the upper bound from Barthe and Köpf to the general case of n adaptive and non-adaptive compositions, and showed that these more general bounds cannot be strengthened.

The leakage bounds for combined channels that we have presented here could possibly facilitate the development of compositional analysis and design techniques for secure systems. In fact, upper bounds on the capacity of a cascade and on the capacity of a repeated independent runs channel had already been used by Köpf and Smith to establish formal bounds on the leakage of a timing side channel attack against a cryptosystem that is protected by blinding and bucketing. Future work should study leakage bounds of repeated independent runs channels and adaptive compositions of channels with respect to other leakage measures, besides min-entropy leakage.

The third research problem that we set out to solve could also be useful in the development of secure software. Indeed, knowing that a channel is always more secure than another regardless of prior distribution or leakage measure is necessary if we aim to develop secure software through stepwise refinement. In this thesis we showed that whenever channel A is the cascade of channels B and R for some channel R, that is, whenever A is composition refined by B, then the leakage of B is an upper bound for the leakage of A for any of the usual leakage measures. Note that this property is a generalized data-processing inequality. Moreover, we proved that composition refinement is in fact a partial order on abstract channels, which are formed by quotienting away the redundant structure of channels as far as information leakage is concerned. These results are further complemented by the proof that composition refinement is the only way for channels to satisfy a leakage ordering with respect to g-leakage regardless of the prior distribution or the choice of gain function. It is then clear that cascading plays a crucial role in establishing the leakage ordering of channels.

The significance of composition refinement is further highlighted by noticing its connection to partition refinement. Recall that, composition refinement coincides with partition refinement on deterministic channels. Furthermore, like partition refinement, it is associated to a robust leakage ordering on channels; although partition refinement is limited to the realm of deterministic channels. Therefore, composition refinement can be seen as a generalization of partition refinement from deterministic to probabilistic channels. However, there are some discrepancies. First of all, leakage ordering regardless of the prior with respect to min-entropy leakage alone is not sufficient to guarantee composition refinement. Similarly, it is likely that this is also the case for Shannon leakage. Finally, it should be noted that preliminary investigations suggest that composition refinement is not a lattice, as is the case with partition refinement. Future work should confirm this conjecture and analyze its implications for the quantitative information flow analysis of programs.

Another interesting result that we encountered along the way is that both guessing entropy leakage and Shannon leakage can be expressed as *additive* g-leakages if we allow uncountably infinite sets of guesses and gain functions that take values in the range $[-\infty, \infty)$. Recall that additive g-leakage measures leakage as the difference between the posterior and prior g-vulnerabilities, rather than the logarithm of their ratio. Relevant future work is then about the mathematical properties of additive g-leakage and the implication of relaxing the restrictions originally imposed on the gain functions. It would also be interesting to determine whether min-entropy leakage can be expressed as an additive g-leakage.

Finally, we remarked that the abstract channel of a composition of channels cannot be determined based only on the abstract channels of its constituents, a result that discourages the usage of abstract channels for the purpose of compositional program analysis.

The last of our four research problems, channel matrix factorization, was motivated by the key roles that cascading and composition refinement play in the foundations of quantitative information flow. Our study of channel matrix factorization resulted in proposing a general procedure for approximately factoring a channel matrix into the product of two channel matrices. We found that, as described in the data mining literature, any exact non-negative factorization of a row-stochastic matrix is associated to a row-stochastic factorization. Furthermore, we learned that whenever the generalized KL divergence is used as a dissimilarity function for non-negative matrix factorization of a row-stochastic matrix, the solutions are also row-stochastic. Hence, our procedure for channel matrix factorization relies on existing algorithms for solving the non-negative matrix factorization problem with the generalized KL divergence.

Channel matrix factorization can be applied in the analysis and design of secure systems. For instance, we discussed its application to the problem of finding a channel that composition refines a particular channel. We also suggested its usage in the area of statistical disclosure control, where the goal is to reveal accurate statistics about a set of individuals while preserving their privacy.

BIBLIOGRAPHY

[AAC⁺11] Mário Alvim, Miguel Andrés, Kostas Chatzikokolakis, Pierpaolo Degano, and Catuscia Palamidessi. Differential privacy: on the trade-off between utility and information leakage. In G. Barthe, A. Datta, and S. Etalle, editors, *Proc. Formal Aspects of Security and Trust (FAST 2011)*, Lecture Notes in Computer Science, 2011. To appear.

[AAP10a] Mário Alvim, Miguel Andrés, and Catuscia Palamidessi. Information flow in interactive systems. In *Proc. 21st International Conference on Concurrency Theory (CONCUR 2010)*, pages 102–116, 2010.

[AAP10b] Mario Alvim, Miguel Andrés, and Catuscia Palamidessi. Probabilistic information flow. In *Proc. 25th IEEE Symposium on Logic in Computer Science (LICS 2010)*, pages 314–321, 2010.

[Abr63] Norman Abramson. *Information Theory and Coding.* McGraw-Hill, 1963.

[ACPS12] Mário S. Alvim, Kostas Chatzikokolakis, Catuscia Palamidessi, and Geoffrey Smith. Measuring information leakage using generalized gain functions. In *Proc. 25th IEEE Computer Security Foundations Symposium (CSF 2012)*, June 2012.

[APvRS10] Miguel Andrés, Catuscia Palamidessi, Peter van Rossum, and Geoffrey Smith. Computing the leakage of information-hiding systems. In Javier Esparza and Rupak Majumdar, editors, *Tools and Algorithms for the Construction and Analysis of Systems (TACAS '10)*, volume 6015 of *Lecture Notes in Computer Science*, pages 373–389, 2010.

[BBJ13] Frederic Besson, Nataliia Bielova, and Thomas Jensen. Hybrid information flow monitoring against web tracking. In *Computer Security Foundations Symposium (CSF), 2013 IEEE 26th*, pages 240–254. IEEE, 2013.

[BCP09] Christelle Braun, Konstantinos Chatzikokolakis, and Catuscia Palamidessi. Quantitative notions of leakage for one-try attacks. In *Proc. 25th Conference on Mathematical Foundations of Programming Semantics (MFPS 2009)*, volume 249 of *ENTCS*, pages 75–91, 2009.

[BK11] Gilles Barthe and Boris Köpf. Information-theoretic bounds for differentially private mechanisms. In *Proc. 24th IEEE Computer Security Foundations Symposium (CSF 2011)*, pages 191–204, 2011.

[BKR09] Michael Backes, Boris Köpf, and Andrey Rybalchenko. Automatic discovery and quantification of information leaks. In *Proc. 30th IEEE Symposium on Security and Privacy*, pages 141–153, 2009.

[Bor06] Michele Boreale. Quantifying information leakage in process calculi. In *Proc. ICALP '06*, pages 119–131, 2006.

[BPP11] Michele Boreale, Francesca Pampaloni, and Michela Paolini. Asymptotic information leakage under one-try attacks. In *Proc. FOSSACS '11*, pages 396–410, 2011.

[CAZ$^+$06] Andrzej Cichocki, Shun-ichi Amari, Rafal Zdunek, Raul Kompass, Gen Hori, and Zhaohui He. Extended smart algorithms for non-negative matrix factorization. In Leszek Rutkowski, Ryszard Tadeusiewicz, Lotfi Zadeh, and Jacek Zurada, editors, *Artificial Intelligence and Soft Computing ICAISC 2006*, volume 4029 of *Lecture Notes in Computer Science*, pages 548–562. Springer Berlin / Heidelberg, 2006.

[CHM01] David Clark, Sebastian Hunt, and Pasquale Malacaria. Quantitative analysis of the leakage of confidential data. In *Proc. Workshop on Quantitative Aspects of Programming Languages*, volume 59 (3) of *Electr. Notes Theor. Comput. Sci*, pages 238–251, 2001.

[CMS05] Michael Clarkson, Andrew Myers, and Fred Schneider. Belief in information flow. In *Proc. 18th IEEE Computer Security Foundations Workshop (CSFW '05)*, pages 31–45, 2005.

[CPP08a] Konstantinos Chatzikokolakis, Catuscia Palamidessi, and Prakash Panangaden. Anonymity protocols as noisy channels. *Information and Computation*, 206:378–401, 2008.

[CPP08b] Konstantinos Chatzikokolakis, Catuscia Palamidessi, and Prakash Panangaden. On the Bayes risk in information-hiding protocols. *Journal of Computer Security*, 16(5):531–571, 2008.

[CT06] Thomas M. Cover and Joy A. Thomas. *Elements of Information Theory.* John Wiley & Sons, Inc., second edition, 2006.

[CZ06] Andrzej Cichocki and Rafal Zdunek. *NMFLAB for signal and image processing*, 2006.

[CZA06] Andrzej Cichocki, Rafal Zdunek, and Shun-ichi Amari. Csiszár's divergences for non-negative matrix factorization: Family of new algorithms. In Justinian Rosca, Deniz Erdogmus, Jos Prncipe, and Simon Haykin, editors, *Independent Component Analysis and Blind Signal Separation*, volume 3889 of *Lecture Notes in Computer Science*, pages 32–39. Springer Berlin / Heidelberg, 2006.

[CZPA09] Andrzej Cichocki, Rafal Zdunek, Anh Huy Phan, and Shun-ichi Amari. *Nonnegative Matrix and Tensor Factorizations: Applications to Exploratory Multi-way Data Analysis and Blind Source Separation*. Wiley Publishing, 2009.

[Dav02] Brian A Davey. *Introduction to lattices and order*. Cambridge university press, 2002.

[Des53] Charles A. Desoer. *Communication through channels in cascade*. PhD thesis, Massachusetts Institute of Technology, 1953.

[DLJ10] C.H.Q. Ding, Tao Li, and M.I. Jordan. Convex and semi-nonnegative matrix factorizations. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 32(1):45 –55, jan. 2010.

[Dwo11] Cynthia Dwork. A firm foundation for private data analysis. *Communications of the ACM*, 54(1), 2011.

[ES78] A. B. El-Sayed. Cascaded channels and the equivocation inequality. *Metrika*, 25:193 – 208, 1978.

[ES12] Barbara Espinoza and Geoffrey Smith. Min-entropy leakage of channels in cascade. In *Formal Aspects of Security and Trust*, pages 70–84. Springer, 2012.

[ES13] Barbara Espinoza and Geoffrey Smith. Min-entropy as a resource. *Information and Computation*, 226:57–75, 2013.

[Fel68] William Feller. *An Introduction to Probability Theory and Its Applications*, volume I. John Wiley & Sons, Inc., third edition, 1968.

[FI10] Cédric Févotte and Jérôme Idier. Algorithms for nonnegative matrix factorization with the beta-divergence. *Neural Computation*, 13(3):1–24, 2010.

[Gal68] Robert G. Gallager. *Information Theory and Reliable Communication.* John Wiley & Sons, Inc., 1968.

[Geh10] Johannes Gehrke. Programming with differential privacy: technical persepctive. *Communications of the ACM*, 53(9):88–88, 2010.

[GVL12] Gene H Golub and Charles F Van Loan. *Matrix computations*, volume 3. JHU Press, 2012.

[HM10] Jonathan Heusser and Pasquale Malacaria. Quantifying information leaks in software. In *Proc. ACSAC '10*, 2010.

[Hoy06] Patrik O. Hoyer. *nmfpack: MATLAB® code for performing NMF and its various extensions*, 2006.

[HSP10] Sardaouna Hamadou, Vladimiro Sassone, and Catuscia Palamidessi. Reconciling belief and vulnerability in information flow. In *Proc. 31st IEEE Symposium on Security and Privacy*, pages 79–92, 2010.

[HvD08] Ngoc-Diep Ho and Paul van Dooren. Non-negative matrix factorization with fixed row and column sums. *Linear Algebra and its Applications*, 429(5–6):1020–1025, 2008.

[KB07] Boris Köpf and David Basin. An information-theoretic model for adaptive side-channel attacks. In *Proc. 14th ACM Conference on Computer and Communications Security (CCS '07)*, pages 286–296, 2007.

[KC93] Aaron B. Kiely and John T. Coffey. On the capacity of a cascade of channels. *IEEE Transactions on Information Theory*, 39(4):1310– 1321, 1993.

[Koc96] Paul Kocher. Timing attacks on implementations of Diffie-Hellman, RSA, DSS, and other systems. In *Proc. Advances in Cryptology (CRYPTO 1996)*, volume 1109 of *Lecture Notes in Computer Science*, pages 104–113. Springer-Verlag, 1996.

[KR10] Boris Köpf and Andrey Rybalchenko. Approximation and randomization for quantitative information-flow analysis. In *Proc. 23nd IEEE Computer Security Foundations Symposium (CSF '10)*, pages 3–14, 2010.

[KS10] Boris Köpf and Geoffrey Smith. Vulnerability bounds and leakage resilience of blinded cryptography under timing attacks. In *Proc. 23nd IEEE Computer Security Foundations Symposium (CSF '10)*, pages 44–56, 2010.

[Lin07] Chih-Jen Lin. Projected gradient methods for nonnegative matrix factorization. *Neural Comput.*, 19:2756–2779, October 2007.

[LR93] Jaisook Landauer and Timothy Redmond. A lattice of information. In *Proc. Computer Security Foundations Workshop VI*, pages 65 –70, June 1993.

[LS01] Daniel D. Lee and H. Sebastian Seung. Algorithms for non-negative matrix factorization. *Advances in Neural Information Processing*, pages 556–562, 2001.

[Mac03] David J.C. MacKay. *Information Theory, Inference, and Learning Algorithms*. Cambridge University Press, 2003.

[Mal07] Pasquale Malacaria. Assessing security threats of looping constructs. In *Proc. 34th Symposium on Principles of Programming Languages (POPL '07)*, pages 225–235, 2007.

[Mal11] Pasquale Malacaria. Algebraic foundations for information theoretical, probabilistic and guessability measures of information flow. *CoRR*, abs/1101.3453, 2011.

[Mas94] James L. Massey. Guessing and entropy. In *Proc. 1994 IEEE International Symposium on Information Theory*, page 204, 1994.

[MEM$^+$13] Annabelle McIver, Barbara Espinoza, Larissa Meinicke, Carroll Morgan, and Geoffrey Smith. Abstract channels, gain functions and the information order. In *Abstract presented at the Workshop on Foundations of Computer Security*, 2013.

[MMHS11] Piotr Mardziel, Stephen Magill, Michael Hicks, and Mudhakar Srivatsa. Dynamic enforcement of knowledge-based security policies. In *Proceed-*

ings of the Computer Security Foundations Symposium (CSF '11), pages 114–128, June 2011.

[MMM10] Annabelle McIver, Larissa Meinicke, and Carroll Morgan. Compositional closure for Bayes risk in probabilistic noninterference. In *Proc. ICALP'10*, pages 223–235, 2010.

[MMM12] Annabelle McIver, Larissa Meinicke, and Carroll Morgan. Draft proof of the coriaceous conjecture. *Dagstuhl Seminar*, 11 2012.

[MMS+14] Annabelle McIver, Carroll Morgan, Geoffrey Smith, Barbara Espinoza, and Larissa Meinicke. Abstract channels and their robust information-leakage ordering. In *Proc. 3rd Conference on Principles of Security and Trust (POST 2014)*, 2014. To appear.

[MS11] Ziyuan Meng and Geoffrey Smith. Calculating bounds on information leakage using two-bit patterns. In *Proc. Sixth Workshop on Programming Languages and Analysis for Security (PLAS '11)*, 2011.

[NMS09] James Newsome, Stephen McCamant, and Dawn Song. Measuring channel capacity to distinguish undue influence. In *Proc. Fourth Workshop on Programming Languages and Analysis for Security (PLAS '09)*, pages 73–85, 2009.

[Pli00] John O Pliam. On the incomparability of entropy and marginal guesswork in brute-force attacks. In *Progress in CryptologyINDOCRYPT 2000*, pages 67–79. Springer, 2000.

[PT94] Pentti Paatero and Unto Tapper. Positive matrix factorization: A non-negative factor model with optimal utilization of error estimates of data values. *Environmetrics*, 5(2):111–126, 1994.

[Rén61] Alfréd Rényi. On measures of entropy and information. In *Proc. 4th Berkeley Symposium on Mathematics, Statistics and Probability 1960*, pages 547–561, 1961.

[Rén70] Alfréd Rényi. *Foundations of Probability.* Holden-Day, Inc., 1970.

[Rom08] Steven Roman. *Advanced Linear Algebra.* Springer New York, Irvine, California, third edition, 2008.

[RR98] Michael K. Reiter and Aviel D. Rubin. Crowds: Anonymity for web transactions. *ACM Transactions on Information Systems Security*, 1(1):66–92, 1998.

[Sha48] Claude E. Shannon. A mathematical theory of communication. *Bell System Technical Journal*, 27:379–423, 623–656, 1948.

[Shl05] Jonathon Shlens. A tutorial on principal component analysis. *Systems Neurobiology Laboratory, University of California at San Diego*, 82, 2005.

[Smi09] Geoffrey Smith. On the foundations of quantitative information flow. In Luca de Alfaro, editor, *Proc. 12th International Conference on Foundations of Software Science and Computational Structures (FoSSaCS '09)*, volume 5504 of *Lecture Notes in Computer Science*, pages 288–302, 2009.

[Smi11] Geoffrey Smith. Quantifying information flow using min-entropy. In *Proc. QEST 2011: 8th International Conference on Quantitative Evaluation of SysTems*, pages 159–167, 2011.

[SS01] Andrei Sabelfeld and David Sands. A per model of secure information flow in sequential programs. *Higher-order and symbolic computation*, 14(1):59–91, 2001.

[SV06] Nandakishore Santhi and Alexander Vardy. On an improvement over Rényi's equivocation bound. In *44th Annual Allerton Conference on Communication, Control, and Computing*, 2006.

[Tru71] Kathleen Trustrum. *Linear Programming. Library of Mathematics.* Routledge and Kegan Paul, 1971.

[YT10] Hirotoshi Yasuoka and Tachio Terauchi. Quantitative information flow — verification hardness and possibilities. In *Proc. 23nd IEEE Computer Security Foundations Symposium (CSF '10)*, pages 15–27, 2010.

[YZYO11] Zhirong Yang, He Zhang, Zhijian Yuan, and Erkki Oja. Kullback-leibler divergence for nonnegative matrix factorization. In Timo Honkela, Wlodzislaw Duch, Mark Girolami, and Samuel Kaski, editors, *Artificial Neural Networks and Machine Learning ICANN 2011*, volume 6791 of *Lecture Notes in Computer Science*, pages 250–257. Springer Berlin / Heidelberg, 2011.

[ZC06] Rafal Zdunek and Andrzej Cichocki. Non-negative matrix factorization with quasi-newton optimization. In Leszek Rutkowski, Ryszard Tadeusiewicz, Lotfi Zadeh, and Jacek Zurada, editors, *Artificial Intelligence and Soft Computing ICAISC 2006*, volume 4029 of *Lecture Notes in Computer Science*, pages 870–879. Springer Berlin / Heidelberg, 2006.

VITA

BARBARA ESPINOZA BECERRA

2006	B.Eng., Computer Science Universidad Simón Bolívar Caracas, Venezuela
2013	M.S., Computer Science Florida International University Miami, Florida

PUBLICATIONS AND PRESENTATIONS

Yingbo Wang, Yali Wu, Andrew Allen, Barbara Espinoza, Peter J. Clarke and Yi Deng. *Towards the Operational Semantics to Support the Rapid Realization of User-Centric Communication Models*. In proceedings of the *33rd Annual IEEE International conference on Computers Software and Applications (COMPSAC 2009)*, pages 254–262. Seattle, Washington, July 2009.

Barbara Espinoza. *Min-entropy Leakage and Channel Composition (Presentation). IFF Workshop on Quantitative Information Flow*. Florida International University, Miami, Florida, April 18, 2011.

Barbara Espinoza and Geoffrey Smith. *Min-entropy Leakage of Channels in Cascade*. In Gilles Barthe, Anupam Datta, and Sandro Etalle, editors, *Formal Aspects of Security and Trust*, volume 7140 of *Lecture Notes in Computer Science*, pages 70-84. Springer Berlin Heidelberg, 2012.

Barbara Espinoza. *Min-entropy Leakage of Channels in Cascade (Presentation). Graduate Student Seminar*. Florida International University, School of Computing and Information Sciences, Miami, Florida, November 29, 2012.

Barbara Espinoza and Geoffrey Smith. *Min-entropy as a Resource*. In *Special Issue: Information Security as a Resource*, volume 226 of *Information and Computation*, pages 57-75. May 2013.

Barbara Espinoza and Geoffrey Smith. *Channels and the Information Order (Poster)*. Presented at the *34th IEEE Symposium on Security and Privacy (Oakland 2013)*. San Francisco, California, May 20, 2013.

Barbara Espinoza, Annabelle McIver, Larissa Meinicke, Carroll Morgan and Geoffrey Smith. *Abstract channels, Gain Functions and the Information Order (Abstract).* Presented at the *Workshop on Foundations of Computer Security.* Tulane University, New Orleans, Louisiana, June 29, 2013.

Annabelle McIver, Carroll Morgan, Geoffrey Smith, Barbara Espinoza, and Larissa Meinicke. *Abstract Channels and their Robust Information-Leakage Ordering.* To appear in proceedings of the *3rd Conference on Principles of Security and Trust (POST 2014).* Grenoble, France, April 2014.