Information Security and Cryptography

More information about this series at http://www.springer.com/series/4752

Mário S. Alvim • Konstantinos Chatzikokolakis
Annabelle McIver • Carroll Morgan
Catuscia Palamidessi • Geoffrey Smith

The Science of Quantitative Information Flow

Springer

Mário S. Alvim
Computer Science Department
Universidade Federal de Minas Gerais
Belo Horizonte, Brazil

Annabelle McIver
Department of Computing
Macquarie University
Sydney, NSW, Australia

Catuscia Palamidessi
Inria Saclay and LIX
École Polytechnique
Institut Polytechnique de Paris
Palaiseau, France

Konstantinos Chatzikokolakis
Department of Informatics
and Telecommunications
University of Athens
Athens, Greece

Carroll Morgan
School of Computer Science
& Engineering
University of New South Wales
Trustworthy Systems, Data61
CSIRO
Sydney, NSW, Australia

Geoffrey Smith
School of Computing
& Information Sciences
Florida International University
Miami, FL, USA

ISSN 1619-7100 ISSN 2197-845X (electronic)
Information Security and Cryptography
ISBN 978-3-319-96129-3 ISBN 978-3-319-96131-6 (eBook)
https://doi.org/10.1007/978-3-319-96131-6

This Springer imprint is published by the registered company Springer Nature Switzerland AG
The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland

The authors dedicate this book as follows:

Mário S. Alvim to his mother, Maria Angélica, his stepfather, Mario, his brothers, Marco Antônio and Marcus Vinícius, and his husband, Trevor.

Kostas Chatzikokolakis to his father, Thymios.

Annabelle McIver to her daughter, Eleanor, and her parents, Anne and Ted.

Carroll Morgan to the policy of diversity and tolerance deliberately instituted and actively sustained at Data61's Trustworthy Systems Group.

Catuscia Palamidessi to her husband, Dale Miller, and their children, Alexis and Nadia Miller.

Geoffrey Smith to his parents, Marilynn and Seward, his wife, Elena, his sons, Daniel and David, and his cockatiel, Yoshi.

Cockatiel *Yoshi* as a probabilistic channel C that maps a top-secret document X to a (randomly generated) pile of shredded paper Y

Preface

Information Flow is the transfer of information from a source (who knows the information) to a target (who does not yet know it). In history, that topic has sometimes been studied in order to *impede* flow (e.g. Caesar's Cipher from millennia ago), and sometimes to *facilitate* it (e.g. Shannon's work in the 1940's). Usually, however, the aims are a careful mixture of the two: to let information flow to those who need to know it, but to keep it from those who must not have it. That is the focus of our contemporary perspective –facilitate some flows, impede others– and our main (but not exclusive) concern here is computer systems.

But *first*: what is so special about now? Information-flow security is a critical problem today because of recent technological developments and their –largely uncontrolled– spread to many hands: all the way from everyday home users to super-skilled hackers, and all over the earth. Data is being collected more than ever before (smart phones, surveillance cameras, "loyalty" cards); networks then enable its transmission to unknown (or unintended) destinations; and powerful corporate and governmental agents gain financial and/or political benefits by collecting and analyzing that data. And, of course, there are the criminals.

Because so much is flowing, and so many have access to it, and we know so little specifically about who they are, we can no longer protect our information by relying on the people through whose hands it passes. Thus the standard technologies like *access control* and *encryption* are insufficient, because there we require the entities granted access to our data to handle it appropriately, and that implied trust might well be misplaced: a smartphone app could legitimately need access to our location, for example, but then leak that information to some other party, perhaps maliciously — but also perhaps just by accident.

Thus instead we must try to generate, process, and transfer our data with systems that protect *themselves*, that are safe no matter who accesses them or how they might abuse that access. It demands a fundamental, rigorous approach; and that fundamental rigor is exactly the *science* that we are striving for.

Thus, *second*: how can it be done? Early rigorous work in information-flow security (since the 1970's) suggested ways in which programs could be analyzed to see whether the program variables an adversary "could see" might depend on variables that were not supposed to be seen: our secrets. If there was no dependence, then the program was secure; but if there was *any dependence at all*, then the program was deemed insecure. That "depends or not" criterion was later realized to be too coarse, however: even a password-checking program, no matter how carefully constructed, would be deemed insecure, because `Access Denied` still unavoidably exhibits a dependence — on what the password *is not*.

Quantitative information flow solves the "depends or doesn't", the "black or white" problem by relativizing information leaks, recognizing that it's not really that clear-cut — some leaks are more important than others, and thus some are tolerable (e.g. leaking what a password isn't, provided it's only infrequently). A typical quantitative approach is to use Shannon's information theory to measure the "entropy" of a secret (roughly, how hard it is to guess) before a system is run, and then to determine what the entropy would become after the program is run (by analyzing the source code, which we assume is available to our adversary). The difference between the two entropies, before minus after, is then how many bits have flowed from the system (escaped, if that flow is not desirable) and –again roughly– if it's a small proportion of the bits that should remain secret, then the actual impact might be considered to be quite limited. Further, because the flow is quantified, the impact can actually be reasoned about rather than merely regretted. That technique realizes a powerful insight, and it works well in many situations: quantifying secrecy in the Shannon style (via entropy) provides the needed nuance to escape the earlier "all or nothing" judgments. For example, if the amount of entropy leaked by a failed login is indeed very small, it is exactly there that quantitative reasoning allows us to calculate with "very small" and "how often" and compare the result to "tolerable".

But much more recently still, it was suggested that Shannon's approach could be generalized, taken further, because in some situations also *it* turned out to be too inflexible: were the numbers it produced, how many bits escaped, really the numbers we needed to know? The generalization was to allow a *selection* of entropies –many more than just Shannon's alone– whose characteristics were derived empirically from a study of the possible adversaries' motivations and capabilities. Which secrets do they really want, and which ones would they not bother to steal? What exactly can they do with their knowledge about the secret? That last step –the generalized entropies– completes the conceptual trajectory from "Does information flow at all?" (simple dependence) through "How many bits of information flow?" (Shannon leakage) to finally (at least for the moment) "What is the *value to the adversary* of the information that flows?" or, dually, "What damage to us is caused by that flow, and how much would we spend (or should we have spent) to prevent it?" Generalized entropies (of which Shannon entropy is a special case) are captured by what we call "loss functions"; dually, we also consider generalized "vulnerabilities", captured by "gain functions". Furthermore, loss- and gain functions enable a connection with the science of program *development*, where specification programs are "refined" into implementation programs that satisfy those specifications both in terms of functionality and security. (Shannon-entropy leakage is not usually a compositional criterion; and yet compositionality is essential for reliable program construction. The use of generalized entropies, however, *is* compositional.)

For all of those reasons, our study of *the science of quantitative information flow* aims to understand fundamentally how sensitive information "flows" as it is processed by an authorized entity (e.g. our computer program), and to ensure that those flows are acceptable to us in terms of the quantified damage they might cause. And here –as we will emphasize– it is important to understand "flows" in a very broad sense: indeed flow occurs whenever sensitive information is *correlated* with observable outputs, allowing an adversary to make *inferences* about the sensitive information. Such correlations can be blatant, as when a sensitive file is copied to some publicly observable place, but they can also be subtle, as when a medical database outputs a patient's country as "United States" if the patient has diabetes and as "USA" if not: in that case the patient's diabetes status "flows" to the country output in a way that probably was not intended.

Extant studies of information flow encompass a variety of domains –such as non-interference, anonymity, unlinkability, secure multi-party computation, differential privacy, statistical databases, side channels, voting, and anonymous communication and publishing– and we have tried to do the same. Something that makes those studies challenging, and our study as well, is that perfection is often unachievable, because *some* undesirable flows cannot be helped. Publishing statistics about a database of medical records necessarily involves revealing some information about the individual records: keeping those records completely private is not an option in that case. Indeed there are many practical reasons for accepting flows that –in a perfect world– we would prefer not to have:

- Sometimes a flow is *intentional*: we *want* to learn something from our statistical database.
- Sometimes a flow is due to *side channels* that are hard or impossible to control fully.
- Sometimes a flow is in exchange for a *service*, one which for example might need our location.
- Sometimes a flow is in exchange for *efficiency*, as when a weaker but more efficient anonymous communication system is used instead of a stronger but less efficient protocol.

All of those support our belief that we must not (only) ask *whether* there is an information flow, and not even (only) *how many* bits of Shannon entropy might flow. We try to study instead *how much damage* an information flow would cause; and because of the generality of that approach, the earlier two are special cases.

The six authors of this book come from a number of distinct research domains, including process calculi, privacy, type systems for secure information flow, and programming-language semantics and refinement. As we all came to understand information flow better, we recognized that our efforts shared deep commonalities; and so, merging our earlier specialties, we have been working intensively as a group together since about 2010. This book is our comprehensive treatment of *quantitative information flow* (*QIF*) as we currently understand it — and we hope that it will lead to further and wider collaboration with those who might read it.

Much of what we present here is based on material already published, but by no means all of it — it is not at all merely "a collection of papers". Instead we have tried hard to write a unified and self-contained text, hoping as we did that to find better terminology and notation than we might have used before, and then in some cases even rewriting whole presentations from scratch to take advantage of it. As well, in many cases we have also replaced earlier mathematical proofs with new ones that are clearer and more self-contained.

Finally, while this book is mainly focused on the systematic development of the theory of quantitative information flow, we also demonstrate the theory's practical utility by including (in Part V) case studies showing how quantitative–information-flow analysis can be applied to a number of interesting realistic scenarios.

Intended readership

Our intended reader is anyone interested in the mathematical foundations of computer security. As far as the required technical background is concerned, we have tried to make the main story understandable to anyone with just a basic knowledge of discrete probability, though sometimes deeper concepts are used. But, in those cases, we have tried to minimize the need for prior familiarity by presenting the necessary material within our text.

It is worth clarifying however that this book is not aimed at readers interested in the legal, ethical, or sociological aspects of information flow. While it is clear that some information flows are beneficial and others are harmful, we make no effort to address the question of which are which.

And finally, we recognize that information flow is in fact a *general phenomenon* with relevance beyond security. So while the theory developed here has largely been motivated by the question of how to limit the leakage of sensitive information, that same theory can no doubt be applied fruitfully in diverse contexts such as machine learning, recommendation systems, and robotics. (Interestingly, in those contexts information flow would typically be seen as a *good* thing.) For this reason, readers outside the field of security may also profit from reading this book.

Organization and structure

We now briefly describe the overall structure of the book.

In Part I, we motivate the study of quantitative information flow, and we give an informal overview of some of its important concepts by discussing information leakage in a very simple context.

In Part II, we begin our detailed development by explaining what a *secret* X actually is, or at least what we consider it to be: a *probability distribution* π that specifies the adversary's knowledge about the likelihood of X's possible values. We also consider how π can be used in quantifying either X's *vulnerability* or (complementarily) the adversary's *uncertainty* about X, observing that there are *many* reasonable ways to do that, depending on the operational scenario, and showing that a single framework, based on "gain functions" (or dually "loss functions"), can encompass them all.

In Part III, we move from secrets to *systems*, modeled as information-theoretic channels that process secret information and possibly leak some of it to their public outputs. We develop a rich family of gain-function–leakage measures to quantify the damage a channel's leakage might cause, carefully considering the *operational significance* of such measures and developing theory that supports *robust* judgments about leakage.

In Part IV, we consider a more detailed model of systems as *programs* written in a simple probabilistic imperative programming language, enabling *compositional reasoning* about information leakage. Here, with assignment statements to program variables we can treat secrets that *change over time*. For that we introduce a mathematical technique that generalizes both channels (which leak secrets) and assignments (which update them). The technique is based on *Hidden Markov Models*.

Finally, in Part V we present a number of *case studies* showing how one can apply quantitative–information-flow analysis to many interesting realistic scenarios — including anonymity protocols, side-channel attacks on cryptography, voting protocols, and even differential privacy in statistical databases. Those chapters are intended to be somewhat self-contained, and readers interested in applications might wish to browse through them early.

Details of presentation

> We sometimes format a definition, theorem, or paragraph in a box to give it greater visual prominence, as we have done in this paragraph. Our intent in doing that is to express our judgments, necessarily subjective, about which things are particularly significant or interesting.

The main text has been kept essentially free of literature citations and historical remarks — instead they are collected in a final section "Chapter Notes" for each chapter. The bibliography is, similarly, organized chapter by chapter.

Cited authors can be found alphabetically in the index, where they appear within square brackets, for example "[Claude E. Shannon]". A glossary appears just before the index, and its entries are in order of first occurrence in the main text. The entry usually reads "*see* something", without a page number, in the hope that the something on its own will be enough to jog the memory. If it isn't, the index entry for "something" itself should be consulted to get a page reference.

Possible usage as a textbook

We have used draft chapters from Parts I, II, and III in a master's-level course on the foundations of cybersecurity that also included extensive coverage of cryptography.

For a full-semester course, we envisage that a course based on Parts I, II, and III and selected chapters from Part V could be taught at both the advanced undergraduate and master's levels. Part IV is more advanced mathematically, and is probably more suitable for doctoral students.

To facilitate the use of the book as a course textbook, we have included a section of *Exercises* at the end of most chapters. Solutions to these exercises are available to qualified instructors.

Language issues

Turning finally to questions of language: we come from six different countries (Brazil, Greece, the United Kingdom, Australia, Italy, and the United States) — which had the advantage that the sun never set on this book's preparation: at all times at least one of us could be found hard at work on it. But such diversity also raises issues of spelling and usage. For the sake of consistency we have made an essentially arbitrary choice to follow American conventions throughout.

Also, with respect to the thorny question of personal pronouns, we have chosen to refer to the *defender* (i.e. the person or entity trying to protect sensitive information) as "he" or "him", to the *adversary* as "she" or "her", and to the *authors* and *readers* of this book as "we" or "us". When there are several points of view, for example in multi-party protocols, we will occasionally use the neuter "it". While assigning genders to the defender and adversary is of course arbitrary (and some readers might indeed prefer the opposite assignment), it has the advantages of avoiding the syntactic awkwardness of "he or she" and, more importantly, of enabling us to write with greater clarity and precision.

Acknowledgments

Our many collaborators have made profound contributions to our understanding of quantitative information flow — and we are particularly grateful to Arthur Américo, Miguel Andrés, Nicolás Bordenabe, Chris Chen, Michael R. Clarkson, Pierpaolo Degano, Kai Engelhardt, Barbara Espinoza, Natasha Fernandes, Jeremy Gibbons, Michael Hicks, Yusuke Kawamoto, Boris Köpf, Piotr Mardziel, Larissa Meinicke, Ziyuan Meng, Tahiry Rabehaja, Andre Scedrov, Fred B. Schneider, Tom Schrijvers, David M. Smith, Marco Stronati, and Roland Wen.

The authors are grateful for support from Digiteo and the Inria équipe associée Princess. Also, Mário S. Alvim was supported by the Computer Science Department at Universidade Federal de Minas Gerais (DCC/UFMG), by the National Council for Scientific and Technological Development (CNPq), by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES), and by the Fundação de Amparo à Pesquisa de Minas Gerais (FAPEMIG). Konstantinos Chatzikokolakis was supported by the Centre national de la recherche scientifique (CNRS), by the Institut national de recherche en sciences et technologies du numérique (Inria), and by the Department of Informatics and Telecommunications of the National and Kapodistrian University of Athens. Annabelle McIver was supported by the Department of Computing at Macquarie University and the Optus Macquarie Cyber Security Hub, Carroll Morgan by the Trustworthy Systems Group of CSIRO's Data61 and the School of Engineering and Computer Science at the University of New South Wales, and both of them by the Australian Research Council and the Information Security Group at ETH Zürich. Catuscia Palamidessi was supported by the Institut national de recherche en sciences et technologies du numérique (Inria), by her ERC grant HYPATIA and by the ANR project REPAS. Geoffrey Smith was supported by the School of Computing and Information Sciences at Florida International University and by the National Science Foundation under grant CNS-1116318.

Belo Horizonte	*Mário S. Alvim*
Athens	*Konstantinos Chatzikokolakis*
Sydney	*Annabelle McIver*
Sydney	*Carroll Morgan*
Paris	*Catuscia Palamidessi*
Miami	*Geoffrey Smith*

April 2020

Contents

List of definitions, theorems, examples, *etc*.

List of figures

List of exercises