HAL
open science

Formal approaches to information hiding : An analysis of interactive systems, statistical disclosure control, and refinement of specifications

Mário S. Alvim

▶ To cite this version:

Mário S. Alvim. Formal approaches to information hiding : An analysis of interactive systems, statistical disclosure control, and refinement of specifications. Cryptography and Security [cs.CR]. Ecole Polytechnique X, 2011. English. NNT : . tel-00639948v2

HAL Id: tel-00639948
https://pastel.hal.science/tel-00639948v2

Submitted on 8 Dec 2011 (v2), last revised 13 Feb 2012 (v3)

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

École Polytechnique

PhD. Thesis - *Thèse de Doctorat*
Spécialité Informatique

Formal approaches to information hiding:
An analysis of interactive systems, statistical disclosure control, and refinement of specifications

Mário S. Alvim
LIX, École Polytechnique
Palaiseau, France

Supervisor
Catuscia Palamidessi

Rapporteurs
Gilles Barthe
Michael Mislove

Examinateurs
Béatrice Bérard
Stéphanie Delaune
Loïc Hélouët
Daniel Le Métayer
Geoffrey Smith

12th of October 2011

École Polytechnique

Laboratoire d'Informatique

Équipe Comète

Centre National de la Recherche Scientifique

Délégation Générale de l'Armement

Contents

Contents i

List of Figures iv

List of Tables v

Acknowledgements vii

1 Introduction 1
- 1.1 Information hiding . . . 1
- 1.2 Qualitative and quantitative approaches to information hiding: a brief history . . . 3
  - 1.2.1 The qualitative approach . . . 4
  - 1.2.2 The quantitative approach . . . 6
- 1.3 Case studies of information hiding . . . 10
  - 1.3.1 Quantitative information flow and anonymity . . . 10
  - 1.3.2 Statistical disclosure control . . . 18
  - 1.3.3 Refining specifications into implementations . . . 20
- 1.4 Plan of the thesis and contribution . . . 22
- 1.5 Publications . . . 23

2 Preliminaries 25
- 2.1 Probability spaces . . . 25
- 2.2 Probabilistic automata . . . 27
- 2.3 CCS with internal probabilistic choice . . . 28

3 The rationale behind the use of information theory for leakage 31
- 3.1 Information theory and communication . . . 31
- 3.2 Information theory and information flow . . . 32
- 3.3 Uncertainty and leakage . . . 34
  - 3.3.1 Shannon entropy . . . 35
  - 3.3.2 Min-entropy . . . 37
  - 3.3.3 Guessing entropy . . . 40
  - 3.3.4 Marginal guesswork . . . 40

3.3.5 Comparison and discussion 41

4 Information flow in interactive systems 43
4.1 Interactive systems 45
4.2 Discrete channels with memory and feedback 48
4.2.1 The power of feedback 50
4.2.2 Directed information and capacity of channels with feedback 53
4.3 Interactive systems as channels with memory and feedback . . . 55
4.3.1 Construction of the channel associated to an IIHS . . . 58
4.3.2 Lifting the channel inputs to reaction functions 61
4.4 Leakage in interactive systems 66
4.5 An example: the Cocaine Auction protocol 72
4.5.1 Calculating the information leakage 75
4.6 Topological properties of IIHSs and their capacity 78
4.7 Related work 85
4.8 Chapter summary and discussion 86

5 Differential privacy: the trade-off between leakage and utility 89
5.1 Differential privacy 90
5.1.1 Formal definition 92
5.2 A model of utility and privacy for statistical databases 92
5.2.1 Leakage about an individual 94
5.2.2 A note on the choice of values 95
5.2.3 The questions we explore with the help of our model . . 96
5.3 Graph symmetries 96
5.4 Deriving the relation between differential privacy and quantitative information flow on the basis of the graph structure 100
5.4.1 Assumptions and notation 101
5.4.2 The matrix transformation 102
5.4.3 The bound on the a posteriori entropy of the channel . . 110
5.5 Application to leakage 112
5.5.1 Measuring the leakage about an individual 119
5.6 Application to utility 120
5.6.1 The bound on the utility 122
5.7 Related work 128
5.8 Chapter summary and discussion 129

6 Safe equivalences for security properties 131
6.1 The use of equivalences in security 132
6.2 Distributed systems and components 135
6.2.1 Tagged Probabilistic Automata 135
6.2.2 Components 136
6.2.3 Distributed systems 137

6.3 Admissible schedulers 138
6.3.1 Restricting global schedulers 139
6.3.2 Restricting local schedulers 140
6.4 Safe equivalences 141
6.4.1 Safe complete traces 141
6.4.2 Safe bisimilarity 142
6.5 Safe nondeterministic information hiding 150
6.6 Related work 151
6.7 Chapter summary and discussion 153

7 Conclusion **155**

Bibliography **159**

List of Figures

1.1 An example of the dining cryptographers protocol 16
1.2 The Crowds protocol at work 17

2.1 The semantics of CCS_p 29

4.1 Interactive system of Example 1 46
4.2 Model for discrete channel with memory and feedback 50
4.3 Scheme of secret transitions for secret-nondeterministic IIHSs . . . 56
4.4 Local transformation in an IIHS tree 58
4.5 Transformation in an IIHS tree 59
4.6 The normalized IIHS for the extended website example 61
4.7 Channel with memory and feedback model for IIHS 66
4.8 Cocaine auction example 73
4.9 Comparison between the leakage in Examples `a`, `b`, and `c` 79

5.1 Randomized function $\mathcal{K}$ 93
5.2 Leakage and utility for oblivious mechanisms 94
5.3 Some distance-regular graphs with degree 3 97
5.4 Some VT^+ graphs 98
5.5 Some $(Val^u, \sim)$ graphs 100
5.6 Venn diagram for the classes of graphs considered in this section. Here $S^* = \{\, Val^u \mid |Val| = 2, u \leq 2\}$ 100
5.7 Steps of the matrix transformation for distance-regular and VT^+ graphs 103
5.8 The relation between elements of a row i and the elements in the diagonal 105
5.9 Graphs of $Bnd(u, v, \epsilon)$ for $u = 100$ and $v = 2$ (lowest line), $v = 10$ (intermediate line), and v=100 (highest line), respectively. 115
5.10 Universe and highest min-entropy leakage matrix giving ϵ-differential privacy for Example 7. 116

6.1 Execution trees for Example 10 134
6.2 TPAs for Example 11 139

List of Tables

4.1 Channel matrix for Example 1 46
4.2 Two different channel matrices induced by two different input distributions for Example 1 47
4.3 Channel matrix for binary erasure channel 50
4.4 General form of channel matrix 52
4.5 A possible evolution of the binary channel with time, for $W = 011$ and $T = 3$ 54
4.6 Channel matrix for Example 5 68
4.7 Stochastic kernels for the Cocaine Auction example 73
4.8 Reaction functions for the cocaine auction example 74
4.9 Values of the probabilities in Figure 4.8 for Examples `a`, `b`, and `c` . 77
4.10 Values of the entropy and directed information for Examples `a`, `b`, and `c`, where $I(A^T; B^T) = H(A^T) - H(A^T|B^T)$ and $I(A^T \to B^T) = H_R - H(A^T|B^T)$ 78
4.11 The IIHSs of Example 6 and their corresponding channels 85
4.12 Summary of results 87

5.1 Mechanisms for the city with higher number of votes for candidate *cand* 127
5.2 Mechanisms for the counting query (5 voters) 128

Acknowledgements

"Praise the bridge that carried you over."
George Colman

Every piece of work is produced within a context, and naturally this thesis is no exception. I want to dedicate this space to express my gratitude to some people that have helped to create an environment of scientific, material, and emotional support, which was crucial to the development of my work over the past three years. I am deeply grateful to all these people, and the influence they have had in this work is only a small part of the influence and importance they have in my life.

First of all, I will always be deeply grateful to Catuscia Palamidessi for her outstanding work as my thesis supervisor. During these three years she has provided a stimulating and exciting scientific environment, endowed with all the material and logistic support a student could ever need. Her passion for science is contagious, and her brilliance and persistence are qualities that I can only hope to be fortunate enough to achieve someday. And not only is she a widely recognized researcher, but she is also a remarkable human being, whose kindness and ethics have set an example that I will always keep with me in academia and for life. It is with sincere joy that I can say that, besides the fruitful scientific cooperation, we were able to create a deep link of friendship, and I will do my best so that both can last for life.

Another person of fundamental importance in my path to this day is Elaine Pimentel. As the first scientific tutor I have ever had, and later as my Master's program supervisor, she was the one welcoming me to the fascinating world of academia. She guided my first steps in research, and her dedication and intelligence are remarkable. Elaine was the strongest supporter I have ever had for doing a doctoral program abroad, especially in the early times when not even my family was convinced yet it was a good idea. More than once Elaine was a thoughtful friend and a wise advisor, who helped me figure out solutions for practical problems that, in some moments, made me doubt I could get to the end of this program. Thank you, Elaine, very much for it all.

I would also like to thank the CNRS (*Centre National de la Recherche Scientifique*) and the DGA (*Direction Générale de l'Armement*) for providing

the funds for these three years of research in France. I also thank INRIA for all the financial and logistical support with respect to scientific conferences, events, and work trips.

I am grateful to the members of my jury, who kindly gave their time to go through my work and evaluate it. Thanks to Béatrice Bérard, Stéphanie Delaune, Loïc Hélouët, Daniel Le Métayer, and Geoffrey Smith. And special thanks to my *rapporteurs* Michael Mislove and Gilles Barthe, who produced the evaluation report for my thesis. I am honored to have had the opportunity to have such a high qualified jury.

I would also like to thank all the people from the Graduate School (*École Doctorale*) of École Polytechnique, especially Audrey Lémarechal for her help with the documentation regarding my stay in France, Fabrice Baronnet for his administrative work, and Christine Ferret for everything involving the thesis defense.

I feel especially fortunate for having had the opportunity to work in such a stimulating environment as is the LIX laboratory (*Laboratoire d'Informatique de l'École Polytechnique*), and in particular the Comète team. It is with a weight in my heart that I leave all these amazing people. I am deeply grateful to Frank Valencia, who gave me one of the warmest welcomes I got in my new life in Europe. Frank was not only a teacher, but a colleague, a gym companion, and a good friend. I am also grateful to his wife, Sara Södergren, and to their son, Felipe Valencia, for all the good moments shared. Thanks also to Andrés Aristizábal for his kindness and for always being ready to help; to Carlos Olarte for the help, friendship and good moments shared together (I will never forget that it was Carlos who took me on my first walk in Paris, and introduced me to the Eiffel Tower); to Sophia Knight for the shared laughs, food, jokes and complaints that make our "love-hate" friendship unique; and to Justin Dean, Sophia's husband, who is a remarkably kind and smart guy with an interesting view of life. Thanks to Dale Miller, for always having wise advice to offer when I needed it, and thanks as well to Catuscia and Dale's kids, Nadia and Alexis, for the good moments shared. I am also grateful to Christele Braun, Ehab El Salamouny, Jéremy Dubrueil, Jesus Aranda, Lili Xu, Luis Pino, Marco Giunti, Marco Stronati, Nicolás Bordenabe, Raluca Diaconu, Romain Beauxis, and Sylvain Pradalier, who, even if I did not have the opportunity to work with them directly, helped make LIX such a great environment.

I would like to thank my co-authors, with whom I have had the opportunity not only to cooperate scientifically, but also to create friendships. Thanks to Miguel E. Andrés for our fruitful collaboration, the constant good mood, and the always stimulating "joke-fights". Thanks to Konstantinos (Kostas) Chatzikokolakis for all the work we developed together, the enlightening discussions about so many subjects, and the good moments shared. Thanks to Pierpaolo Degano, with whom I had the pleasure of collaborating and learning from.

The team of administrative support at LIX was also fundamental for my work. I would like to thank Marie-Jeanne Gaffard for her remarkable competence and dedication, which have frequently saved me from a great deal of trouble. Her professional behavior is a model to be followed, and I wish I could encounter people like her everywhere I will ever work. I am also grateful to Valérie Lecomte, for the countless times she helped me, even when it was not her duty, always with the characteristic competence and sympathy. I cannot forget Corinne Poulain, who guided me through the endless administrative maze when I arrived in France. Thanks also to James Régis for the technical support; and to Isabelle Biercewicz and Lydie Fontaine for the assistance in my first years at LIX. I would also like to say a couple of words about Ryna Lam Pech, whose cheerful smile and always good mood made each coffee time in the cafeteria an even more enjoyable moment.

I am also grateful to the experienced scientists who have shared part of their vast knowledge with me, either in conferences, workshops or informal meetings, and reinforced my view that people in academia are not only brilliant, but usually good human beings as well. Thanks especially to Geoffrey Smith for sharing his expertise with me in so many insightful conversations, and for organizing the exciting workshop on information flow at Florida International University. I will not forget the hospitality he, his wife Elena, his sons Daniel and David, and the adorable Yoshi offered in Miami. Thanks also to Prakash Pananganden for the lectures at the SFM-10:QAPL summer school in Bertinoro, and also for the opportunity to participate in the workshop on quantum and classic information flow at the Bellairs Research Institute.

I cannot proceed without mentioning all the amazing friends I have in Brazil, who were fundamental in the background that brought me here. Even being far away, they are constantly in my mind, and I always count down the days to the next time I will see them again. Thanks to Aline Miranda, whom I have had the privilege of knowing and whose friendship I enjoy very much; to Aline Resende, an incredible friend, on whom I know I can always count on at any time of day or night, and with whom I have had some of the most joyful and memorable moments of all my life; to Anísio Lacerda, the talented and sensible guy whom I always enjoyed talking to about any subject (serious or not); to Deznie Lopes, who always has a smile to offer; to Katia Lage, the sweet and kind friend who is always there to help others; to Lara Coelho, the funny and practical girl, whose visit to Paris was one of the highlights of my time in the city; and to Marina Cruz, my childhood friend, the one I have known for the longest in my life and whose love always warms up my heart. I am also deeply grateful to Adriani Quatrini, who played such an important role of support and understanding during one of the darkest moments in my last three years; and to Giselle Moura, who has cared so much for me and was the main force driving the process that literally changed my face and, therefore, my life (for much better).

I had never left Brazil at all until the day I moved to Paris, and when I

arrived in Europe I did not know a single person on this side of the Atlantic Ocean. It was a big turning point in my life, and I am so glad that I decided to come, for these three years in Paris were not only a period of professional growth, but also of incredible personal learning. I have had the pleasure of meeting here some of the most remarkable human beings I have ever met, both at the professional and personal levels. In particular, our "sweet, sweet Maunoury", the building shared as home by so many foreign students at École Polytechnique, has been the stage of countless adventures, memorable moments, and deep learning. Without the companionship of the people I met there, I would not have been able to enjoy my stay in France as much, and therefore my work would not have been as productive. I would like to thank each and every one of the people I met in Maunoury for the friendship that has marked me so deeply. Also, I want to thank each one for particular things that I will keep in my memory forever. Thanks to Saddaf Shabbir for all the philosophical discussions by the lake during summer (or until late night otherwise), that have enlightened me so much in so many subjects; to Andreas Engelhardt for the constant companionship and mutual-understanding which have so many times lightened the weight of being abroad; to Nadia Vertti for the happiness and cheerfulness that could always make me smile at any time; to Keesjan de Vries for all the awesome trips shared together (*Do you wanna know why? Well...*); to Ricardo Kawahara for sharing the fun of nights out, and also the frustration of the way back home by the Noctilien 122; to Michał Zydor for the uncountable movies seen together in Paris; to Fabien Immler for being my "German little brother"; to Alex Rinke for the hospitality during the winter holidays in Berlin in 2009/2010; to Oliver Valencia for the fun moments at Bôbar; to Kalle Backlund, Anna Folke Larsen, and Uli Schneider for all the unforgettable evenings at their place in Rue Guisarde and at Chez Georges; to Steffen Lohrey and Marie Le Mouel for the nice evenings watching Audrey Hepburn movies in my room; to Chiara Altomare, Manuele Aufiero, Paolo Carozzo and Lorenzo Sponza (the "Italian mafia") for the constant cheerfulness in our beloved international kitchen; to Benjamin Mosk for the energy to never say no to a night out dancing; to Maria Rosario (Charo) Mestre for the company not only in Paris, but also in Frankfurt; to Álvaro Izquierdo for the constant company in the gym, and the fun trips together; to Amy Gilson, Anton Karrman, Davi Vasconcellos, Leland Ellison, Lysandra Alves, and Michael Martin for the unforgettable Summer of 2009; to Citlali Cabrera for her kindness in every moment, and the nice dinners she offered to me; to Igor Reshetnyak for always being ready to help in anything; to Théo Touvet for the rare example of confident and unique life choices; to Tomás Lungenstrass for the constant smile and good mood; and to François Wirion and Julia Duras for the first moments shared in the doctoral program. Thanks also to Alex Lang, Alfredo Parra, Daniel Ruiz, Federico Cárdenas, Benjamin Uekermann, Fredrik Hallgren, Henri de Belsunce, Herbert Mangesius, Ivan Moschevitin, Joe Gault, Nikita Kazarinov, Pedro Vitória, Przemysław Cho-

jecki, Sara Rome, and Seydou Traoré for all the unforgettable moments. I cannot forget Hannah Schneider and Sofia Karlsson, who have not lived in Maunoury but are part of the family, and I would like to thank them for the friendship and hospitality when I visited both Cologne and Stockholm.

It was not only on campus, however, that I met friends. Among the many amazing people I met in Paris, and all over the world, are Alexandra Silva, good company in several conferences and summer schools, whom I hope to meet often, both as a friend and as a colleague; Diogo Arbigaus, the kind and good friend who, even though he is Brazilian, I have met only in Paris; Maria Poulaki, whose refreshing company and kindness always make me feel good; and Nicolás Lopez and all the Spanish crowd, whose parties in Rue Souflot will be always in my memory.

I often say that we do not have much control over our lives, and that the best we can do is to try to be prepared enough to catch a good opportunity when it shows up. Today I can look back and be glad to say that I caught at least two life-time opportunities in the past three years. The first one was on the 1st of October 2008, when I landed in Paris to start my doctoral program at École Polytechnique. The second one was on the 19th of March 2010, when I met Trevor Ray Tisler. Meeting him was a turning point in my life, and his emotional support has paved the road so I could work with a lighter spirit. I am grateful for the patience with which he has revised my English writing so many times, the dedication he has shown to me even being overseas for over a year now, and for his love, support and presence in my life.

Finally, I would like to thank my family, of whom I am so proud, for the love and support during my whole life, and especially during the challenges these past three years have imposed on me. Thanks to my mother, Maria Angélica, who has always been a model human being for me, as a strong yet sweet woman, and who gives me strength in hard moments and shares my joy in the good ones; to my brother Marco Antônio, who has set an example for me with his dedication, ethical behavior and kindness that are a constant in everything he does; to my brother Marcus Vinícius, whose particular sense of humor and "tough" behavior are not enough to hide a kind heart and a person one can always count on; to my step-father Mario Montoya, who is a remarkable human being, and who has given me more support, understanding and love than my biological father has ever done; to my sisters in law Luciana Salomão and Débora Pires, for being like real sisters, and for the countless joyful moments shared; and to my cousin Adriana de Lima, for always being by my side and supporting me.

I apologize to the several people that played an important role in my way and who have not found their name mentioned here: I am sorry if my memory played a trick on me.

Mário S. Alvim
Paris, December 2011

Abstract

In this thesis we consider the problem of information hiding in the scenarios of interactive systems, statistical disclosure control, and refinement of specifications. We apply quantitative approaches to information flow in the first two cases, and we propose improvements for the usual solutions based on process equivalences for the third case.

In the first scenario we consider the problem of *defining the information leakage in interactive systems* where secrets and observables can alternate during the computation and influence each other. We show that the information-theoretic approach which interprets such systems as (simple) noisy channels is not valid. The principle can be recovered, however, if we consider channels of a more complicated kind, that in information theory are known as channels with memory and feedback. We show that there is a complete correspondence between interactive systems and these channels, and we propose the use of directed information from input to output as the real measure of leakage in interactive systems. We also show that our model is a proper extension of the classical one, i.e. in the absence of interactivity the model of channels with memory and feedback collapses into the model of memoryless channels without feedback.

In the second scenario we consider the problem of *statistical disclosure control*, which concerns how to reveal accurate statistics about a set of respondents while preserving the privacy of individuals. We focus on the concept of *differential privacy*, a notion that has become very popular in the database community. Roughly, the idea is that a randomized query mechanism provides sufficient privacy protection if the ratio between the probabilities that two adjacent datasets give a certain answer is bound by a constant. We observe the similarity of this goal with the main concern in the field of information flow, namely limiting the possibility of inferring the secret information from the observables. We show how to model the query system in terms of an information-theoretic channel, and we compare the notion of differential privacy with that of min-entropy leakage. We show that differential privacy implies a bound on the min-entropy leakage, and we also consider the utility of the randomization mechanism, which represents how close the randomized answers are, in average, to the real ones. Finally we show that the notion of differential privacy implies a tight bound on utility, and we propose a method that under certain conditions builds an optimal randomization mechanism.

Moving the focus away from quantitative approaches, in the third scenario we address the problem of using *process equivalences to characterize information-hiding properties* (for instance secrecy, anonymity and non-interference). In the literature, some works have used this approach, based on the principle that a protocol P with a variable x satisfies such property if and only if, for every pair of secrets s_1 and s_2, $P[^{s_1}/_x]$ is equivalent to $P[^{s_2}/_x]$. We show that, in the presence of nondeterminism, the above principle may rely on the assumption that the scheduler "works for the benefit of the protocol", and this is usually not a safe assumption. Non-safe equivalences, in this sense, include complete-trace

equivalence and bisimulation. This problem arises naturally when *refining a specification into an implementation*, since usually the former is more abstract than the latter, and the refinement process involves reducing the nondeterminism. The scheduler is, in this sense, a final product of the refinement process, after all the nondeterminism is ruled out. We present a formalism in which we can specify admissible schedulers and, correspondingly, safe versions of complete-trace equivalence and bisimulation. We prove that safe bisimulation is still a congruence. Finally, we show that safe equivalences can be used to establish information-hiding properties.

One

Introduction

"There are two mistakes one can make along the road to truth:
not going all the way, and not starting."
Gautama Siddharta

1.1 Information hiding

In the last few decades the amount of information flowing through computational systems has increased dramatically. Never before in history has a society been so dependent on such a huge amount of information being generated, transmitted and processed. It is expected that this solid trend of increase will continue in the near future, if not virtually indefinitely, reinforcing the need for efficient and safe ways to cope with this reality.

Although the efficient and broad dissemination of information is a goal in many situations, there are instances where the disclosure of information is undesirable or even unacceptable. The field of *information hiding* concerns the problem of guaranteeing that part of the information relative to an event is kept secret. In computer science, the term information hiding encompasses a large spectrum of fields. Different fields have distinct historical motivations and the resulting research followed a unique path. The variation of the subfields of information hiding depends on three main factors: (i) *what* one wants to keep secret; (ii) from *which adversary or attacker* does one want to keep it secret; and (iii) *how powerful* the adversary or attacker is.

The field of *confidentiality* (or *secrecy*) refers to the problem of keeping an action secret. One application of confidentiality is *cryptographic protocols*, where the sender and the receiver of a message can be known, but the contents of the message itself are considered to be sensitive information. Generally, we can say that confidentiality concerns *data*, while the field of *privacy* concerns *people's personal information*. When dealing with privacy, we may

be interested in protecting the information about someone (a credit card number, for instance) or the person's identity itself. *Anonymity* is the field that concerns the protection of the identities of agents involved in events. In principle, anonymity can be related to both the *active agent* (often the *sender* of a message), or to the *passive agent* (often the *receiver* of a message). For instance, in the case of a journalist receiving information from a confidential source, the identity of the sender is intended to be secret. As for the case of an intelligence agency sending a coded message to a spy, the identity of the receiver is confidential information. There is yet another kind of anonymity, sometimes referred to as *unlinkability*, where the identity of agents and actions performed are public information, but the linkage between agents and the actions performed should not be determined. One example of unlinkability is a confidential voting system, where both the voters and the final vote count are in the public domain, but the relationship between the voters' identities and the ballots cast is protected.

One application of privacy that has drawn a lot of attention in recent years is the problem of statistical databases. A statistic is a quantity computed from a sample, and the goal of *statistical disclosure control* is to enable the user of the database to learn properties of the population as a whole, while maintaining the privacy of individuals in the sample. The field of statistical databases highlights the delicate equilibrium between the benefits and the drawbacks of the spread of information. A practical example occurs in medical research, where it is desirable that a great number of individuals agree to give their personal medical information. With the information acquired, researchers or public authorities can calculate a series of statistics from the sample (such as the average age of people with a particular condition) and decide, say, how much money the health care system should spend next year in the treatment of a specific disease. It is in the interest of each individual, however, that her participation in the sample will not harm her privacy. In our example, the individuals usually do not want to have disclosed their specific status with relation to a given disease, not even to the users querying the database. Some studies, e.g. [Joi01], suggest that when individuals are guaranteed anonymity and privacy they tend to be more cooperative in giving personal information.

Another important field of information hiding is *information flow*, which concerns the leakage of classified information via public outputs in programs and systems. Consider a system that asks the users a password to grant their access to some functionality. Naturally, the password itself is intended to be secret, however an attacker trying to guess it will always get an observable reaction from the system, whether the response is an acceptance or a rejection of the entered code. In either case, the observable behavior of the system reveals some information about the password, because even if it is not guessed correctly, at least the search space is narrowed (even if, in this case, only slightly).

It is important to note that the subdivisions of information hiding are not

mutually exclusive. In a system where public outputs can reveal the identity of agents, for instance, both the problems of information flow and of anonymity are present. The classification is usually based more on the contextual motivation for the problem than on a rigid taxonomy of subfields. In fact, in recent years there has been an active line of research exploring the similarities between problems such as the foundations of anonymity and information flow, and also privacy and information flow. The result has been an increasing convergence between these fields. In this thesis we explore the similarities between information flow, statistical databases, and anonymity.

In a broader context, the importance of information hiding goes far beyond the realm of computer science, and there are a lot of subtle questions that need to be considered carefully. From a political and even philosophical perspective, the unrestricted use of privacy protection can be controversial. Even though it is broadly accepted that people should have the right to exchange e-mails privately, to vote in democratic elections anonymously, and to express their ideas on the Internet freely, there are situations where information protection policies can be argued to have serious drawbacks. The same mechanism that grants a political activist anonymity and free speech on the Internet, while living under a repressive government, also grants a pedophile anonymity to broadcast harmful material. This balance between freedom and control in the virtual media has been the subject of passionate discussion. Independently of whether one's goal is to maximize or to minimize the degree of information protection in a given situation, it is anyway desirable to measure *the extent to which* the information is protected, to define which specific *definition of protection* the information falls under, and *from whom* the information is protected.

In this thesis we avoid the controversy of deciding in which cases the application and extent of information hiding methods are justifiable. Rather, our focus is on measuring the degree of information protection offered by a system, thus making evaluation and comparison of different systems possible . Specifically, we are interested in using concepts of information theory to quantify the leakage of information.

1.2 Qualitative and quantitative approaches to information hiding: a brief history

Historically, the research on information hiding has evolved from the simple but imprecise *qualitative approach* toward the more refined, but at the same time more complex, *quantitative approach*. In the following sections we will briefly overview both. We do not intend to provide here an exhaustive study of the subject, but rather to highlight some of the most important contributions of each of these lines of research to the field of information hiding.

1.2.1 The qualitative approach

The qualitative approach emerged first in the literature of information hiding. The central idea is that, by observing the output of a system, the adversary cannot be completely sure of what the secret information is. The *principle of confusion* says that for every observable output generated by a secret input, there is another secret that could also have generated the same output. In anonymity, for instance, this corresponds to the concept of *possible innocence*, i.e. the impossibility of identifying the culprit with certainty by only observing the system's output. The principle of confusion does not take into consideration the adversary's certainty about the value of the secret: it is enough that there be an alternative hypothesis, no matter how unlikely it is. This is also known as the *possibilistic approach*.

One of the first developments in this field dates from 1976, when Bell and La Padula defined the model of *multilevel security systems* [BLP76]. In this model the components of a system are classified as either *subjects*, i.e. active entities such as users or processes, or as *objects*, i.e. passive entities such as files. The subjects are divided into *trusted* and *untrusted* entities, and the authors define restrictions on how to manage untrusted objects. The rule "no read up or write down" states that untrusted entities can read only from objects of the same or lower levels, and that they can only write into objects of the same or higher levels. This model was developed to support different levels of security, and aimed to ensure that information only flows from lower to higher levels and never in the opposite direction. Each input into and output from the system is labeled with a security level. Any pair of an input and its corresponding output is called an *event*. A *view* of a security level l corresponds to the events at level l or lower, and all the events of a higher level are *hidden* to level l.

Usually in this model only two levels are distinguished: *high* and *low*. The high level corresponds to sensitive information, which should only be available to some users with special privileges, while the low level corresponds to public information accessible to everyone. The goal of *secure information flow analysis* is, in this context, to avoid leakage from the high level to the low level.

Bell and La Padula's model, however, did not address the problem of leakage of information due to *covert channels*. A covert channel is a way of transmitting information from the high to the low environment by means not designed or intended for this purpose. Consider, for instance, a system where a low user ℓ can send a file to a high user h, and h has the power to redefine the access rights to the file. The user h can either maintain the permission of ℓ to write in the file, or she can change the policy so ℓ no longer has access to it. In this scenario, a covert channel between a corrupted high user h and low user ℓ can be established as follows. The low user sends a file to the high user, who then uses her power of deciding whether to grant or to deny ℓ further access to it to encode a message. In a later stage, ℓ tries to write in the file, and an

access failure can be interpreted as the bit 0, while a success can be interpreted as the bit 1. In this way any message can eventually be sent through the covert channel from the corrupted high user to the low one.

To cope with the threat of covert channels, Goguen and Meseguer developed the concept of *noninterference*[GM82]. A system is *noninterfering* when the actions of high users do not alter what can be seen by low users. In other words, the low outputs of the system will only reflect the values of the low inputs, independently of what the high inputs are (if any). The authors proposed a model of noninterference that separated the system from the security policies. Their model, nevertheless, was only appropriate for deterministic systems.

Noninterference, however, may be a too restrictive concept for several practical applications. It does not allow, for instance, the *summarization of data*. It is often the case where a system allows statistical (or summarizing) functions (e.g. mean, total number) to be calculated on its high inputs and then disclosed to low users, even if the high inputs themselves are supposed to be kept secret. These systems are typical in the area of statistical databases, and we will discuss this issue in more detail in Section 1.3.2. Clearly, a system that allows the summarization of high data for the low environment violates noninterference, since a change on the high input may affect the low output.

Considering this problem, in 1986 Sutherland [D.S86] proposed the concept of *nondeducibility on inputs*, which focuses not on whether the output is affected according to a change in the input, but on whether it is possible *to deduce* the input from the output. Under this definition, a system may allow summarization of data and still be secure, since the output of a statistical function does not necessarily allow the adversary to deduce what the inputs are. One drawback of the concept of nondeducibility on inputs is that it assumes that the strongest form of the principle of confusion is enough to ensure security. Notably, it relies on the assumption that "no high value can be ruled out after observing a low value". This is not a strong enough security guarantee in many real systems. In some cases, even if no high value can be ruled out as a possibility, a single value (or a small set of values) can be much more likely than the others, and in practice it makes little sense to consider the alternatives. This criticism can be seen as an early attempt to consider a quantitative approach for information flow, where it is taken into consideration "how much" an attacker learns (or does not learn) about the secret matters.

Another important issue in security systems is the problem of *compositionality*. In [McC87], McCullough pointed out the importance of *hook-up security*, i.e. the compositionality of multi-user systems. Usually, real systems are far too complex to be analyzed as a whole, especially because the task of designing and implementing a system is normally divided between teams. Each team is responsible for a number of components that, in a later stage, will be put to work together. It is highly desirable that security properties be verified in each component separately, and that this verification guarantee

that the final composite system is also secure. McCullough showed that the concepts of multilevel security systems, noninterference, and nondeducibility on inputs are not composable. As a replacement, he proposed the concept of *restrictiveness*, according to which no high level information should affect the *behavior* of the system, as seen by a low user.

In [WJ90] Wittbold and Johnson addressed the question of nondeducibility on inputs under a different perspective, showing that it is not a guarantee of absence of leakage. Consider the following algorithm, where H and L stand for the high and the low environments, respectively. Here we assume the variables x and y are binary, and the randomized command $x \leftarrow 0 \oplus_{0.5} 1$ assigns to x either the value 0 or the value 1 with 0.5 probability each.

while true **do**
  $x \leftarrow 0 \oplus_{0.5} 1$;
  output x to H;
  input y from H;
  output (x XOR y) to L;
end while

In the above algorithm, the low environment only has access to the value (x XOR y). Note, however, that the high environment H learns the value of x before having to choose the value of y, and therefore it can use this knowledge to encode a message: To transmit the bit 0, H chooses $y = x$, and to transmit the bit 1, H chooses $y = 1-x$. It is clear that there is some flow of information from the high to the low environment, even though L cannot deduce the high input y from the low output (x XOR y). Hence, satisfying nondeducibility on inputs does not guarantee a system to be secure. Wittbold and Johnson defined, then, the concept of *nondeducibility on strategies*, which means that regardless of what view L has of the machine, no strategy is excluded from being used by H.

1.2.2 The quantitative approach

The qualitative approach, although simple and easy to apply, does not reflect reality in many practical situations. In many cases some information leakage is tolerable or even intentional. Take an election protocol. After the final vote count is released, there are fewer possible hypotheses concerning who voted for whom than the hypotheses available before the votes were cast. In this example there is a natural leakage of information, since the uncertainty about the sensitive information decreases after the observation of the protocol's output. This leakage occurs, however, as a necessary functionality of the protocol.

In fact, in most real systems noninterference cannot be achieved, as typical systems will always leak some information. This does not mean, however, that all systems are equally good or bad, because the amount of leakage usually

varies from system to system. Therefore it is important to quantify *how much* leakage a system allows. Quantitative methods are useful to evaluate the extent to which a system is secure, and to compare it to other systems.

One of the first attempts to quantify information leakage was made by Denning in 1982. In [Den82] she defined the leakage from a state s to a state s' as the decrease in uncertainty about the high information in s resulting from the low information in s'. She used the concept of conditional entropy[1] $H(h_s|\ell_{s'})$, where h_s is the high information in s and $\ell_{s'}$ is the low information in s'. Her definition of leakage was:

$$M_1 = H(h_s|\ell_s) - H(h_s|\ell_{s'})$$

If the quantity M_1 is positive, then it is considered to be the leakage of information. This measure of leakage, however, does not consider the history of low inputs, a problem pointed out by Clark, Hunt and Malacaria in [CHM07]. Without the history one cannot summate the increase in knowledge (or decrease in uncertainty) that accumulates between the low states s and s'. They proposed, instead, the following measure of leakage:

$$M_2 = H(h_s|\ell_s) - H(h_s|\ell_{s'}, \ell_s)$$

Since $H(X|Y,Z) \leq H(X|Y)$ for all random variables X, Y and Z, we have $M_1 \leq M_2$. The quantity M_2 corresponds to the Shannon conditional mutual information $I(h_s; \ell_{s'}|\ell_s)$.

In 1987, Millen made a formal connection between information flow and Shannon information theory by relating noninterference and mutual information [Mil87]. In Millen's model, a computer system is seen as a channel whose input is a sequence W, possibly generated by a set of users, and whose output (after the computation is completed) is Y. The random variable X represents a subsequence of W generated by a user U, while $\overline{X}$ represents the high inputs generated by users other than U. Millen showed that in deterministic systems if X and $\overline{X}$ are independent and X is not interfering with Y, then the Shannon mutual information $I(X;Y)$ between X and Y is zero. In other words, noninterference is a sufficient condition for absence of information flow.

In 1990, Massey gave an important contribution to the field of information theory, which influenced the further development of quantitative information flow. In [Mas90] he showed that the usual definition of discrete memoryless (i.e. history-independent) channels used at that time in fact did not take into account the possibility for the use of feedback. He highlighted the conceptual

[1] The concepts of *entropy*, *conditional entropy* and *mutual information* will be defined formally in Chapter 3. For the moment it is enough to know that *entropy* is a measure of the uncertainty of a random variable; *conditional entropy* is a measure of the uncertainty of one random variable given another random variable; and *mutual information* is a measure of how much information two random variables share.

difference between causality and statistical dependence, and presented an accurate mathematical description of discrete memoryless channels that allowed feedback. Then he introduced the concept of *directed information*, which captures the idea of causality between the input and the output of a channel, and argued that in the presence of feedback, directed information is a more appropriate measure of the flow of information from input to output than mutual information.

In the same year, McLean also considered the concept of time in the description of systems by proposing his *Flow Model* [McL90]. According to this model, there is a flow of information only when a high user H assigns values to objects in a state that precedes the state in which a low user L makes her assignment. In this situation only part of the correlation between high and low information is considered as leakage. This addressed the problem of causality, but this model was too general, and relatively difficult to apply.

In [Gra91] Gray worked on bridging the gap between the overly complicated Flow Model and the more practical, yet restricted, approach of Millen. Gray used a general-purpose probabilistic (as opposed to nondeterministic) state machine that resembled Millen's model. In Gray's model, the value $\mathcal{T}(s, I, s', O)$ represents the probability of a given state s evolving into another state s', under the input I, and producing output O. The channels are partitioned into two sets, H and L, representing the channels connected to high and low processes, respectively. The high and the low environments can communicate only through their interactions with the system, as no other form of communication between them is allowed. Gray wanted to take time and causality into consideration in his definition of leakage, and he did so by allowing feedback and memory in his model. His formulation of a security guarantee was the following:

$$\begin{aligned} P(L^I \cap L^O \cap H^I \cap H^O) > 0 \qquad &\Longrightarrow \\ P(\ell | L^I \cap L^O \cap H^I \cap H^O) = P(\ell | L^I \cap L^O) \end{aligned} \tag{1.1}$$

where L^I and L^O represent the history of low inputs and outputs, respectively, and H^I and H^O represent the history of high inputs and outputs, respectively. The symbol ℓ represents the final output event channels in the low environment. The formulation (1.1) states that the probability of a low output may depend on the previous history of the low environment, but not on the previous history of the high environment.

Gray also tried to generalize the concept of capacity to the case of channels with memory and feedback. He provided a formula expressing the flow of information from the whole history of inputs and outputs (during the time period $0 \ldots t-1$) to the the low output (at time t), and conjectured that the capacity of the channel would be:

$$C \stackrel{\text{def}}{=} \lim_{n \to \infty} C_n \tag{1.2}$$

where

$$C_n \stackrel{\text{def}}{=} \max_{H,L} \frac{1}{n} \sum_{i=1}^{n} I(In_Seq_Event_{H,t}, Out_Seq_Event_{H,t}; \\ Final_Out_Event_{L,t} | In_Seq_Event_{L,t}, Out_Seq_Event_{L,t}) \tag{1.3}$$

and $In_Seq_Event_{A,t}$ is the input history at channel A (where A stands for L or H) up to time $t-1$, $Out_Seq_Event_{A,t}$ is the output history at channel A up to time $t-1$, and $Final_Out_Event_{L,t}$ is the low output event at time t. Gray showed that the absence of information flow implies that capacity as formulated in (1.2) is zero. He also conjectured that this definition of capacity would correspond to the notion of maximum transmission rate supported by the channel. As pointed out in [AAP11], however, the problem with Gray's conjecture is the following. For an output at time t, the only causal relation considered is the one with the history of inputs up to time $t-1$, while the effect that the input at time t itself may have on the output is ignored. In this way, (1.2) does not express the complete causal relation between input and output. The correct notion of capacity in the presence of memory and feedback, which corresponds to the maximum transmission rate for the channel, was proposed in 2009 by Tatikonda and Mitter [TM09], and it will be discussed later on in Chapter 4.

A similar formal approach, although with different motivations, was presented by McIver and Morgan in [MM03]. They focused on the problem of preserving security guarantees while refining specifications into implementations. The authors used an equation similar to (1.3), but in the context of sequential programing languages enriched with probabilities. Their aim was to protect the high values during the whole execution of the program, instead of the initial high values only. In other words, they wanted to assure that if the high information is not known by the low environment at the beginning of the computation, then it cannot be inferred at any later stage. They proved that, for deterministic programs, if the final values of the high objects are protected, then the initial values are protected as well. McIver and Morgan also defined the concept of *information escape* as:

$$H(h|\ell) - H(h'|\ell')$$

where $H(h|\ell)$ represents the uncertainty (conditional entropy) of the high information given the low information at the beginning of the computation, and $H(h'|\ell')$ represents the same uncertainty at the end of the computation. They defined the channel capacity as the least upper bound of information escape over all possible input distributions. In this context, a system is considered secure if it has capacity equal to zero. One advantage of this model is that it is not necessary to keep track of the whole history of the computation, but on the other hand it can be applied only in scenarios where the adversary does not have memory.

In Chapter 3 we will take up again the discussion of quantitative approaches to information flow based on information theory. For the moment we will focus on some topics related to information hiding that are of special relevance for this thesis.

1.3 Case studies of information hiding

In this section we present three case studies of information hiding that we address in this thesis.

1. The case of *quantitative information flow*, i.e. how much about the secret information an adversary can learn by observing the system's output, and by knowing how the system works. We give special attention to the broadly studied problem of anonymity, which can be seen as a particular case of the more general problem of information flow where the secret information is the identity of the agents.

2. The question of *statistical disclosure control*, which concerns the problem of allowing users of a database to obtain meaningful answers to statistical queries, while protecting the privacy of the individuals participating in the database. We focus on differential privacy, an approach to this problem that has drawn a lot of attention in recent years.

3. The problem of *preserving security guarantees while deriving implementations from specifications*. Usually specifications are more abstract than implementations, i.e. they present more nondeterminism. The task of implementing a system reduces the nondeterminism of the specification, and if it is not done carefully, an implementation may rule out possibilities allowed by specification that are essential for the security guarantees.

1.3.1 Quantitative information flow and anonymity

Anonymity is one of the most studied subjects of information hiding. The research in this area has been active in the past several years, and the advances made can be extended to the more general scenario of information flow. As briefly introduced in Section 1.1, anonymity concerns the protection of the identities of the agents involved in the events.

With the advent of the Internet, the protection of anonymity has become an issue in the daily life of millions of people around the world. The importance of anonymity is even more evident concerning the protection of freedom of speech, a situation that is particularly delicate in countries under repressive regimes.

Pfitzmann, Dresden and Hansen [PDH08] have proposed a standard terminology for anonymity concepts. In their work there are three different notions of anonymity based on the agents involved:

- *Sender anonymity*: when the identity of the originator should be protected;
- *Receiver anonymity*: when the identity of the recipient should be protected;
- *Unlinkability*: when it might be known that an agent A originated a message and an agent B received a message, yet it should not be known whether the message sent by A was actually the one received by B.

Reiter and Rubin also gave a classification of the types of adversary in an anonymity system in [RR98], where they also proposed the anonymity protocol Crowds (see Section 1.3.1). In their work, they considered that the adversary can be an eavesdropper simply observing the traffic of messages on the network, or she can be an active attacker (i.e. a collaboration between senders, between receivers, or between others taking part in the system), or even a combination of the previous two types. The authors also defined a hierarchy of anonymity degrees that a system can provide. In decreasing order of strength, the proposed scale is listed below. In this list, let s, s' denote secrets and o an observable, i.e. a particular action or output of the system that is distinguishable from the point of view of the attacker.

Strong anonymity From the attacker's point of view, the observables produced by the system do not increase her knowledge about the secret information, i.e. the identity of the individual involved in an event. Chaum also described the concept of strong anonymity in his work on the Dining Cryptographers protocol [Cha88]. It represents the ideal situation where the execution of the protocol does not give to the adversary any extra information about the secrets. The concept was formalized as follows.

$$\forall s, o \quad p(s|o) = p(s) \tag{1.4}$$

This definition is the equivalent of "probabilistic noninterference". In [CP06], Chatzikokolakis and Palamidessi showed that the condition expressed by (1.4) is equivalent to:

$$\forall s, s', o \quad p(o|s) = p(o|s') \tag{1.5}$$

i.e. the probability of the system producing an observable is the same, no matter what the secret information is. This definition is known as *equality of likelihoods* and is advantageous as it does not depend on the probability distribution on secrets.

Another definition of strong anonymity, more restrictive, was proposed by Halpern and O'Neill [HO03, HP05]. It is equivalent to each of the previous definitions ((1.4) or (1.5)) plus the assumption that the input probability is uniform. Halpern and O'Neill focused on the adversary's lack of

confidence in her guess about the secret, and defined strong anonymity as:

$$\forall s, s', o \quad p(s|o) = p(s'|o) \tag{1.6}$$

The formulation (1.6) is also known as *conditional anonymity* and corresponds to the level of anonymity called *beyond suspicion* in Reiter and Rubin's classification.

Beyond suspicion From the attacker's point of view, an agent is no more likely to be the culprit than any other agent in the system. It can be formalized as in (1.6).

Probable innocence From the attacker's point of view, an agent does not appear more likely to be involved in an event than not to be involved. Formally:

$$\forall s, o \quad p(s|o) \leq 0.5 \tag{1.7}$$

The formulation (1.7), however, is not broadly accepted as the definition of probable innocence. In [CP06], Chatzikokolakis and Palamidessi showed that the property that Reiter and Rubin indeed proved for the Crowds protocol in [RR98] was:

$$\forall s, o \quad p(o|s) \leq 0.5 \tag{1.8}$$

Possible innocence From the attacker's point of view, there is always a non-zero probability that the agent involved in the event is someone else. Formally:

$$\forall s, o. \big(p(s|o) > 0 \implies \exists s'. p(s'|o) > 0\big)$$

The above hierarchy gives a richer classification of the degree of protection offered by a system than would be possible with simpler possibilistic models.

Among the quantitative approaches to anonymity, two are of our special interest: the ones based on information-theoretic concepts and the ones based on the Bayes risk. In the following section we give a brief overview of these two approaches. These concepts will be revisited in more detail in Chapter 3.

Anonymity protocols as noisy channels

Information theoretic approaches to anonymity, and more generally to information flow, rely on concepts such as entropy and mutual information to measure the adversary's lack of information about the secret before and after observing the system's output. Typically the system is seen as a noisy channel and the concept of noninterference corresponds to the converse of the channel capacity.

There are several works in the literature that have proposed measures of degrees of anonymity in terms of the entropy and mutual information, for instance

[SD02, DSCP02, ZB05, DPW06]. In [CPP08a] Chatzikokolakis, Palamidessi and Pananganden proposed the concept of *conditional capacity* to cope with the situation where some leakage of information is intended by the system. Consider again the election protocol example. By design, the final vote counting needs to be announced and it usually increases the attacker's knowledge about the secret. In this situation, the leakage should be calculated modulo the information that is supposed to be disclosed, i.e. the vote count. In this work the authors also proposed methods to calculate the channel capacity exploiting some symmetries present in several practical systems.

Hypothesis testing and Bayes risk

In some real world situations an individual faces the following situation: she is interested in the value of some random variable $A \in \mathcal{A}$ but she has access only to the values of another random variable $O \in \mathcal{O}$. She knows that A and O are correlated by a known conditional probability distribution. This situation occurs in several fields, for instance in medicine (to make a diagnosis, the physician has access to a list of symptoms, but not to the disease itself). The attempt to infer A from O is known as the problem of *hypothesis testing*. Here we are interested in the use of hypothesis testing in the context of anonymity (and information flow). More specifically, the adversary tries to infer the secret A given that she has access to the observables O and she knows how the system works, i.e. how the probabilities of O are conditioned with relation to A.

A commonly studied approach to the problem is based on the Bayesian method and consists of assuming the a priori probability distribution on A as known, and then deriving from that and from the knowledge about how the system works, an a posteriori probability distribution after some fact has been observed. It is well known that the best strategy for the adversary is to apply the MAP rule (Maximum A posteriori Probability rule), which as the name suggests, chooses the hypothesis with the maximum probability for the given observation. Here, by "best" strategy we mean the one that induces the smallest probability of error in guessing the hypothesis, that in this case corresponds to the *Bayes risk*.

In [CPP08b] Chatzikokolakis, Palamidessi and Pananganden explored the hypothesis testing approach to anonymity, in a scenario where the adversary has one single try to guess the secret (after exactly one observation). They associated the level of anonymity to the probability of error, i.e. the probability of an attacker making a wrong guess about the secret. In order to consider the worst case scenario and to give upper bounds for the level of anonymity provided, the adversary is assumed to use the MAP rule strategy. In this case, the probability of error corresponds to the Bayes risk, and the degree of protection offered by a protocol corresponds to the Bayes risk associated with the channel matrix.

In [Smi07, Smi09] Smith also considered the scenario of one-try attacks and proposed the notion of *vulnerability*, which takes into consideration the probability that the adversary can guess the secret correctly after observing the behavior of the system only once. Smith proposed the framework of *min-entropy leakage*, which is closely related to the Bayes risk, but is different as it uses the concept of entropy (more precisely min-entropy) and formalizes leakage in information theoretic terms.

In Chapter 3 we will present a deeper discussion about the use of information theory for the formalization of information flow, including the notions of Shannon entropy, mutual information and the framework of min-entropy leakage for one-try attacks. First, however, we will review some fundamental anonymity protocols in literature.

Examples of anonymity protocols

On the Internet, every computer has a unique IP address which specifies the computer's logical location in the topology of the network. This IP address is usually sent along with any request originating from the computer. Even if the computer uses an IP address for a single session via an ISP (Internet Service Provider), the identification can be logged and retrieved later with the ISP's compliance. One common way to try to preserve anonymity is to use a *proxy*, i.e. an intermediary computer that gathers all the requests of a group of computers and serves as a unique gate for any communication with the world outside of the network. For practical purposes, it is as if all the requests originated from the proxy, and the members of the group are indistinguishable from the point of view of an outside observer. One drawback presented by the use of proxies is that it creates single points of failures, decreasing the network's robustness.

The problem illustrated above is one of the motivations for the use of communication protocols specifically designed to protect anonymity. In this section we review two of the most fundamental, and probably most famous, examples of anonymity protocols in literature: *the dining cryptographers* protocol, and the *Crowds* protocol.

The dining cryptographers The dining cryptographers protocol was proposed by Chaum in [Cha88]. It is one of the first anonymity protocols in the literature, and it is one of the few protocols that can assure strong anonymity.

The protocol is usually presented in a simplified scenario, where three cryptographers employed by the NSA (The National Security Agency of the United States) are having dinner in a restaurant. At the end of the dinner, the NSA decides whether it will pay the bill itself or whether it will assign the duty of paying to one of the cryptographers at the table. In the case the NSA decides that one of the cryptographers will pay, it announces the decision secretly to the chosen one. The goal of the protocol is to reveal whether one cryptographer

will pay the bill or not, without revealing the identity of the payer. In other words, to an external observer (and to the non-paying cryptographers as well), the only accessible information is whether the NSA is paying or not, but not the identity of the cryptographer paying (if any). We assume that the NSA does not disclose its decision to anyone but to the cryptographer it chooses (again, if any), and that the solution should be distributed, i.e. only message passing between agents is allowed, and no centralized agent coordinates the process.

The dining cryptographers protocol solves this problem as shown schematically in Figure 1.1. Each cryptographer ($Crypt_0$, $Crypt_1$ and $Crypt_2$) tosses a coin that is visible only to himself and to his right-hand neighbor. In this way every cryptographer has a shared coin with each of the other two. After all three coins (c_0, c_1 and c_2) are tossed, each cryptographer checks whether the two coins visible to him agree (both are heads or both are tails) or disagree (one is head and the other is tails). Then they announce publicly *agree* or *disagree*, according to the result they obtained with their coins. The only exception is that, if a cryptographer is paying, he will announce the opposite of what he sees, i.e. he will announce *disagree* in the case that his coins agree and *agree* if they do not. It can be proven that if the number of *disagrees* is even, then the NSA is paying, and if the number of *disagrees* is odd, then one of the cryptographers is paying. Moreover, if the coins are all fair, the protocol offers strong anonymity in the following sense: The execution of the protocol does not provide to an external observer enough evidence to change her knowledge about which cryptographer is the payer, if any. In other words the probability of any cryptographer being the payer, under the adversary's point of view, does not change after the observation of the protocol's execution.

The dining cryptographers protocol can be generalized to any number of graph nodes (i.e. cryptographers) and any type of graph connectivity (i.e. the shared coins between pairs of cryptographers). Then the same solution can be used for anonymous communication as follows. Each pair of nodes share a common secret (the value of the coin) of length n, equal to the length of the transmitted data. It is assumed that the coins are drawn uniformly from the set of possible secrets. Each node then computes the binary sum (XOR operation) of all its shared secrets and announces the result. The only exception is that the node that wants to transmit adds the datum, also of length n, to the sum it announces. It can be shown that the total sum of the announcements of all nodes is equals to the data to be transmitted, since each secret is counted twice (once by each node that can see it) and, therefore, is canceled out by the XOR operation. The protocol works under the assumption that only one node at a time tries to transmit, and if it is the case that more than one sender wants to transmit at the same time, the conflict needs to be solved by some sort of coordinator.

One drawback of the dining cryptographers protocol is its inefficiency: whenever a single node wants to transmit, all the nodes in the graph need

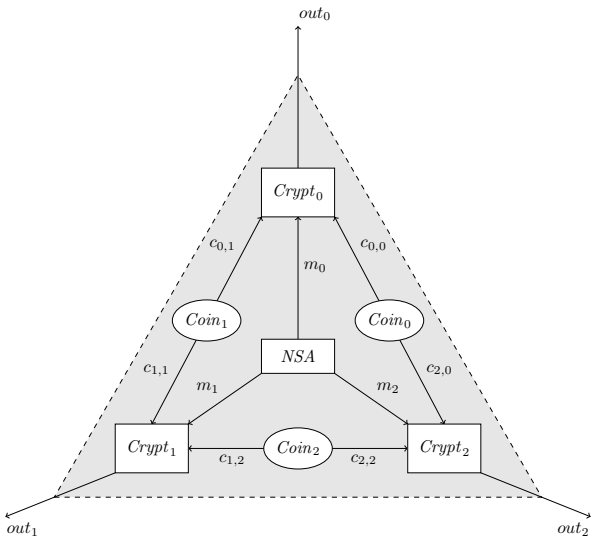


Figure 1.1: An example of the dining cryptographers protocol

to collaborate to make it happen, at the cost of a large number of message exchanges. Moreover, as previously stated, in the case where more than one node wants to transmit at the same time, a coordinator is necessary to solve the conflict.

Crowds The Crowds protocol was first presented in [RR98] and it allows Internet users to perform web transactions without revealing their identity. Usually, on the Internet, when a user communicates with a server the latter can discover the IP address of the originator. The idea behind Crowds is to gather users into a crowd and randomly redirect the request multiple times inside the group before finally letting it reach the server. In this situation, it is impossible for the server, and for any other user, to identify the initiator of the request once it receives the message: whenever someone sends a message there is a considerable probability that she is only a forwarder for someone else.

To be more precise, a *crowd* is a group of m users who participate in the protocol. It is possible that a subgroup of c users are corrupted and collaborate to disclose the identity of the original sender. Also, we assume that the protocol has a parameter $p_f \in (0, 1]$. We call *originator* or *initiator* the user who wants to make a request to the server. The originator needs to create a *path* between herself and the server in order to have her request reach the final destination, as shown in Figure 1.2.

The protocol works as follows:

- At the first step the initiator chooses, according to a uniform probability

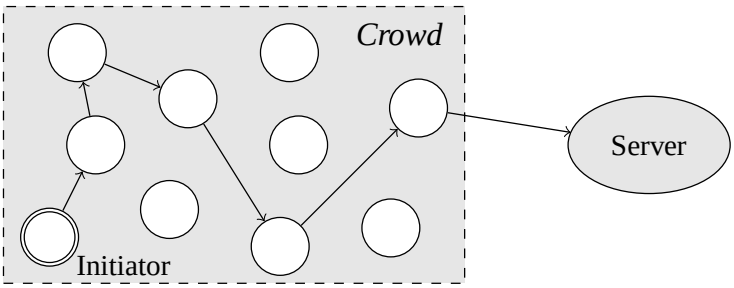


Figure 1.2: The Crowds protocol at work

distribution, another user in the crowd (possibly herself) and forwards the request to this user;

- The user who receives the message then makes a random choice. With probability p_f she forwards the message to the server, and with probability $1 - p_f$ she decides to forward the message to some user in the crowd. If this is the case, she chooses a user (possibly herself) according to a uniform probability distribution, and forwards the message to this user. This step is then repeated by the new message holder.

The response from the server to the originator follows the same path, in the opposite direction. Moreover, all the communications in a path are encrypted using a *path key*, which protects the path from threats posed by local eavesdroppers. Each user has access to the communications in which she participates, but it is assumed that a user cannot intercept messages exchanged between other users. It can be proven that the protocol is strongly anonymous with respect to the web server. Intuitively this is the case because at least one forward step is always performed, and after this step any user can be the holder of the message with equal probability. Therefore, from the server's point of view any user is equally likely to be the originator of the request.

A more interesting case is to analyze the level of anonymity ensured with respect to a corrupted user. If in the very first step of the execution of the protocol the message is forwarded to a corrupted user, she can gain more information about the possible originator than the server. A user, whether the originator or not, is said to be *detected* if she sends a message to a corrupted user. Since the originator always appears in a path, she is more likely to be detected than the rest of the users. Detecting a user (at least for the first time in a path) increases the probability that this user is the originator. Therefore, strong anonymity cannot hold with relation to corrupted users.

In [RR98] it is proven that if the number c of corrupted users is not too large, the protocol can at least ensure the level protection of probable inno-

cence. More precisely, if the number m of users in the crowd satisfies

$$m \geq \frac{p_f}{p_f - \frac{1}{2}}(c+1)$$

then the protocol ensures probable innocence in the sense of (1.8).

1.3.2 Statistical disclosure control

The field of *statistical disclosure control* concerns the problem of revealing accurate statistics about a set of respondents while preserving the privacy of individuals. In statistical databases, the data of a (large) number of participants is compiled, and users are allowed to pose statistical queries (such as average or total counting) about the sample. This kind of database is of special importance in many areas. For instance, medical databases can provide information about how a disease spreads, and a census database can help authorities to decide how to spend the next year's budget.

The data in a statistical database can be obtained in different ways. It can be collected in a census, for instance, it can be obtained opportunistically by monitoring the traffic in a network, or it can even be given by the participants by their own choice. No matter how the data is obtained, however, it is still important to ensure that the individual's participation in the database will not harm her privacy. This is not a trivial goal to achieve: the main purpose of a statistical database, in the first place, is to reveal some information about the population as a whole, i.e. to let users infer "general truths" about this population. As an example, suppose that a statistical database of individuals of a certain country indicates that, in this population, the life expectancy for women is 5 years longer than for men. Clearly this piece of information reveals something about the whole population, *even about individuals not present in the database*.

There are several approaches to dealing with the problem of preserving privacy in statistical databases. One of them is based on ensuring *large query sets*, i.e. that no query can be posed for a small set of individuals. The problem with this approach is that, even if two query sets are "large enough", their combination may not be. Consider the following two queries: "How many people have disease y?" and "How many people, not named X, have disease y?". Both queries operate on large sets, but clearly the superposition of the two queries immediately reveals sensitive information about the individual named X. Another attempt to achieve privacy is based on the *encryption of the data* in the dataset. This is not a general solution since, as we have seen, the privacy threats do not concern only the individuals in the database and, therefore, the encryption of the data will not address this issue.

Another possible solution is to apply some sort of *query auditing*: the curator of the database checks whether or not a query is possibly disclosing before deciding to provide an answer to it. This approach would cope with the

problem of the two superposing queries mentioned above, yet it presents two serious drawbacks: first, automatic tools to check every query are practically infeasible; and, second, the refusal to answer a query can be in itself a disclosing act. Another attempt to deal with the problem is by using *subsampling* of the dataset. We normally view a dataset as a collection of rows, where each row contains the data of an particular participant. The idea of subsampling is to randomly choose a subset of the rows, compute the answer to the query based on this subsample, and then report it as the final answer. If the subset is large enough, it should reflect the statistical properties of the whole database. This approach, however, protects a participant only to the extent to which it is unlikely that she is in the subsample. If being in the subsample has catastrophic results, then someone will always be seriously harmed.

The *input perturbation* approach is based on modifying either the data or the query in hope of confusing the adversary. For instance, a *randomized response* mechanism can be used at the moment the data is acquired. This modification is permanent and not even the curator knows what the original data was. The queries to the database are then made taking into consideration the randomized noise.

Yet another approach is to add randomized noise *to the answer of the query*. The idea is to compute the answer on the complete set of (the original) values in the database, and then randomize the response before reporting it to the user. If this is done naively, however, it can easily be taken care of by the adversary. Suppose that the noise is chosen to be a Gaussian additive noise with mean zero. If the query is repeated a sufficient number of times, a statistical analysis of the answers can easily estimate with high accuracy what the real answer is. Even if the curator of the database opts to record the query and always report the same answer for it, it may not solve the problem: syntactically different queries can be semantically equivalent, and if the query language is rich enough the semantic equivalence is undecidable.

In this context, it is clear that the problem of statistical disclosure control is not trivial. Yet another issue to be considered is *auxiliary (or side) information*. Auxiliary information is any piece of data about individuals that the attacker has and that does not come from the database itself. It may originate from priors, beliefs, newspapers or even other databases. Some decades ago, Dalenius [Dal77] considered the problem of auxiliary information and proposed a famous "ad omnia" privacy desideratum: nothing about an individual should be learnable from the database that could not be learned without access to the database. In other words, if the adversary has some side information and gains some knowledge about the individuals using it, by learning the response from the database this knowledge about individuals should not increase. Dalenius' property is, however, too strong to be useful in practice: Dwork showed in [Dwo06] that no useful database can satisfy it. She then proposed the notion of *differential privacy*, which is based on the idea that the presence or absence of an individual in the database, or the individual's particular value, should

not significantly change the probability of obtaining a certain answer for a given query [Dwo06, Dwo10, Dwo11, DL09].

The concept of differential privacy can be formalized as follows. Let $\mathcal{X}$ be the set of all possible databases, and $\mathcal{Z}$ be the set of possible answers to a query. Two databases $x, x' \in \mathcal{X}$ are *adjacent* (or *neighbors*), written $x \sim x'$, if they differ in the value of exactly one individual. Then, for some $\epsilon > 0$:

Definition 1 ([Dwo11]). *A randomized function $\mathcal{K}$ from $\mathcal{X}$ to $\mathcal{Z}$ satisfies ϵ-differential privacy if for all pairs $x, x' \in \mathcal{X}$, with $x \sim x'$, and all $S \subseteq \mathcal{Z}$, we have:*

$$Pr[\mathcal{K}(x) \in S] \leq e^{\epsilon} \cdot Pr[\mathcal{K}(x') \in S]$$

The concept of differential privacy has had an extraordinary impact in the database community, and we will discuss the meaning and implications of the above formulation in greater depth in Chapter 5. For the moment, it is enough to note that this definition intuitively ensures that individuals can opt in or out of the database without significantly changing the probability of any given answer to a query to be reported. In other words, it is "safe" for an individual to join (or to leave) the database. Dwork also showed that in order to ensure differential privacy it is enough to consider a Laplacian mechanism of noise [Dwo06].

Although differential privacy is a promising approach to the question of statistical disclosure control, the fact that it relies on the randomization of the query response poses some challenges with respect to the *utility* of the query mechanism. If the noise is not added with sufficient care, the reported answer can be so "different" from the real answer that the informative purpose of the database is compromised. In Chapter 5 we will come back to the question of how to apply differential privacy and, at the same time, provide maximum utility to the query mechanism.

1.3.3 Refining specifications into implementations

Deriving implementations of a system given its specification, while respecting security constraints, is a challenging problem in information hiding and, more generally, in security. A specification S is refined by an implementation P if P preserves all logically expressible properties of S. One needs to be careful, however, when refining a specification in the realm of information hiding. According to Morgan [Mor09]:

> A rigorous definition of how specifications relate to implementations, as part of reasoning, must ensure that implementations reveal no more than their specifications: they must, in effect, preserve ignorance.

By "ignorance", the author means what the user does not know about what she cannot see. This notion is closely related to the problem of information flow, i.e. determining how much about the secret behavior of a system an adversary can infer from an observation and her knowledge about how the system works.

To illustrate the problem, we will discuss the following example, adapted from the original one in [Mor09]. Consider a partition of the program states into *visible* (v) and *hidden* (h). Assume that the two variables v and h have the same domain $\mathbb{N}$ (the natural numbers), and in a specification S, after the value of h is assigned, the following is stated: *choose v from the domain* $\mathbb{N}$. Then we can ask "from the final value of v, what can the observer deduce about the value of h, given that she knows how the system works?". Of course the answer will depend on how the implementation I of the specification is done. If I is simply $v := 0$, then nothing is learned, since what the user knows about the value of h is exactly what she already knew before. If the implementation is $v := h \texttt{ mod } 2$, then she can learn h's parity. If the implementation is $v := h$, then she learns the exact value of h. Intuitively, the three implementations are in increasing order according to the loss of ignorance they induce.

It is desirable that the implementation of a specification be "ignorance preserving", in the sense that the implementation should not reveal more about the secrets than the specification does. Some works in the literature suggest that one should be careful when dealing with secure refinements if one wants to preserve information-flow security properties. In [Jac89], for instance, Jacob shows that even if an implementation is a consistent refinement with respect to a specification, it does not imply that the (information-flow) security properties of the specification are preserved in the implementation.

As pointed out in [CNP09], nondeterminism is often used in system specifications as a way of abstracting from implementation details (such as scheduler policy). Implementations are obtained from specifications by refinement algebras, which reduce nondeterminism. As we have seen in a previous example, if we assume v and h are both of type $\mathbb{N}$, then the specification *choose v from the domain* $\mathbb{N}$ can be refined to $v := h$, which is simply a reduction of nondeterminism. This is known as the "refinement paradox" [Mor09], because it does not preserve ignorance. While the specification does not tell anything about the value of h, the refinement completely reveals it.

The process of reducing nondeterminism by refinements is related to the notion of *schedulers* in nondeterministic systems: *designing an implementation of a specification involves choosing a scheduler to solve all the nondeterminism of the specification.* The scheduler is indeed a final result of the refinement process, after all the nondeterminism is ruled out.

According to this perspective, similar concerns about refinement algebras should be taken into consideration when dealing with schedulers. Indeed, it can be shown that, given a specification S and a scheduler that leads to a consistent implementation P with respect to S, it is not guaranteed that the

security properties of S are preserved in P.

In the domain of refinement of specifications, the solution proposed in [Mor09] is to apply some principles to the refinement algebra in order to assure the preservation of ignorance. These principles restrict the refinement relation, eliminating the cases that do not preserve ignorance.

A similar problem arises in the context of concurrent systems, where the scheduler that resolves the nondeterminism can violate security properties. In Chapter 6 we focus on this problem and we propose restrictions on the schedulers that also lead to ignorance-preserving refinements.

1.4 Plan of the thesis and contribution

In Chapter 2 we review some basic notions necessary for the development of this thesis, including the concepts of probability spaces, probabilistic automata and CCS_p (a probabilistic version of the process algebra of concurrent communicating processes).

In Chapter 3 we review the main approaches that have been considered to quantify the notion of information leakage using concepts of information theory. We explain concepts such as entropy, conditional entropy, mutual information and capacity. We focus on how distinct notions of entropy can model attackers with different levels of power, and we introduce the mathematical background necessary for most of this thesis. Finally we compare the main notions of uncertainty and leakage in the literature.

In Chapter 4 we consider the problem of defining the information leakage in interactive systems where secrets and observables can alternate during the computation. We show that the information-theoretic approach that interprets such systems as classic channels is not valid. The principle can be recovered, however, if we consider channels of a more complicated kind, namely channels with memory and feedback. We show that there is a complete correspondence between interactive systems and such channels. We also propose the use of directed information, as opposed to mutual information, to represent leakage in interactive systems. This proposal is based on recent results in information theory that have shown that, in channels with memory and feedback, the transmission rate does not correspond to the maximum mutual information (the standard notion of capacity), but rather to the maximum (normalized) directed information. We show that our model is a proper extension of the classical one, i.e. in the absence of interactivity the model of channels with memory and feedback collapses into the model of memoryless channels without feedback. Finally, we show that the capacity of the channels associated with interactive systems is a continuous function with respect to a pseudometric based on the Kantorovich metric.

In Chapter 5 we analyze critically the notion of differential privacy in the light of the conceptual framework provided by min-entropy leakage. We show

that there is a close relationship between differential privacy and leakage, due to the graph symmetries induced by the adjacency relation on databases. Furthermore, we consider the utility of the randomized answer, which measures its expected degree of accuracy. We focus on certain kinds of utility functions called "binary", which have a close correspondence with the notion of min-entropy leakage and the Bayes risk. Again, there can be a tight correspondence between differential privacy and utility, depending on the symmetries induced by the adjacency relation and by the query. Using these symmetries we can, in some cases, build an optimal-utility randomization mechanism while preserving the required level of differential privacy. We also provide a study of the kind of structures that can be induced by the adjacency relation and the query, and how to use them to derive bounds on the leakage and achieve the optimal utility.

In Chapter 6 we move away from the quantitative realm and focus on the problem of nondeterminism in systems specifications. In the field of security, process equivalences have been used to characterize various information-hiding properties (for instance secrecy, anonymity and noninterference) based on the principle that a protocol P with a variable x satisfies such a property if and only if, for every pair of secrets s_1 and s_2, $P[^{s_1}/_x]$ is equivalent to $P[^{s_2}/_x]$. We argue that, in the presence of nondeterminism, the above principle relies on the assumption that the scheduler "works for the benefit of the protocol", and this is usually not a safe assumption. Non-safe equivalences, in this sense, include complete-trace equivalence and bisimulation. We present a formalism in which we can specify admissible schedulers and, correspondingly, safe versions of these equivalences. We prove that safe bisimulation is still a congruence. Then we show that safe equivalences can be used to establish information-hiding properties.

Finally, in Chapter 7 we make our final observations.

1.5 Publications

Most of the results in this thesis have already been the subject of scientific publications. More precisely:

- Chapter 3 is based on the paper **Probabilistic Information Flow** [AAP10b] that appeared in the proceedings of 25th *Annual IEEE Symposium on Logic in Computer Science* (LICS 2010).

- Chapter 4 is based on the papers:

  – **Information Flow in Interactive Systems** [AAP10a] that appeared in the proceedings of the 21st *International Conference on Concurrency Theory* (CONCUR 2010);

  - **Quantitative Information Flow in Interactive Systems** [AAP11] to appear in the *Journal of Computer Security*.

- Chapter 5 is based on two complementary works:
  - The paper **On the relation between Differential Privacy and Quantitative Information Flow** [AACP11] to appear in the proceedings of the *38th International Colloquium on Automata, Languages and Programming* (ICALP 2011);
  - The technical report **Differential Privacy: on the trade-off between Utility and Information Leakage** [AAC+11].

- Chapter 6 is based on the paper **Safe Equivalences for Security Properties** [AAPvR10] that appeared in the the proceedings of the *6th IFIP International Conference on Theoretical Computer Science* (IFIP-TCS 2010).

Two

Preliminaries

"I can make just such ones if I had tools, and I could make tools if I had tools to make them with."
Eli Whitney

In this chapter we review some technical concepts from the literature that will be used throughout this thesis.

2.1 Probability spaces

In this section we recall some concepts about probability spaces.

Let Ω be a set and $\mathcal{P}(\Omega)$ represent its powerset, i.e. the collection of all subsets of Ω. A *σ-algebra* (also called *σ-field*) over Ω is a non-empty collection of sets $\mathcal{F} \subseteq \mathcal{P}(\Omega)$ that is closed under complementation and countable union. For any σ-field $\mathcal{F}$, the property $\Omega \in \mathcal{F}$ holds, and also that $\mathcal{F}$ is closed under countable intersection (by De Morgan's laws).

A *(positive) measure* on $\mathcal{F}$ is a function $\mu : \mathcal{F} \to [0, \infty)$ such that

1. $\mu(\emptyset) = 0$, and
2. $\mu(\bigcup_i C_i) = \sum_i \mu(C_i)$, where $\{C_i\}_i$ is a countable collection of pairwise disjoint sets in $\mathcal{F}$.

A *probability measure* on $\mathcal{F}$ is a measure μ on $\mathcal{F}$ such that $\mu(\Omega) = 1$. A *probability space* is a tuple $(\Omega, \mathcal{F}, \mu)$ where Ω is a non-empty set called the *sample space*, $\mathcal{F}$ is a σ-algebra on Ω called the *event space*, and μ is a probability measure on $\mathcal{F}$. In the discrete case, we have

$$\forall C \in \mathcal{F}. \quad \mu(C) = \sum_{x \in C} \mu(\{x\})$$

In this case we can construct μ from a function $p : \Omega \to [0,1]$ satisfying $\sum_{x\in\Omega} p(x) = 1$ by assigning $\mu(\{x\}) = p(x)$. The function p is called a *probability distribution* over Ω.

The set of all probability measures with sample space Ω will be denoted by $\mathcal{D}(\Omega)$. We will also denote by $\delta_x(\cdot)$ (called the *Dirac measure* on x or also a *point mass*) the probability distribution such that $\mu(\{x\}) = 1$.

If A and B are events, i.e. elements of a σ-field $\mathcal{F}$, then $A \cap B$ is also an event. If $\mu(A) > 0$ then we can define the *conditional probability* $p(B|A)$ as

$$p(B|A) = \frac{\mu(A \cap B)}{\mu(A)}$$

representing the probability of B given that A holds. Note that $p(\cdot|A)$ is a new probability measure on $\mathcal{F}$. For the scope of this thesis we are interested only in the discrete case, so it is enough to use the definition above and make sure that we never condition on an event A with zero probability.

Let $\mathcal{F}, \mathcal{F}'$ be two σ-fields on Ω, Ω' respectively. A *random variable* X is a function $X : \Omega \mapsto \Omega'$ that is *measurable*, meaning that the inverse of every element of $\mathcal{F}'$ belongs to $\mathcal{F}$:

$$\forall C \in \mathcal{F}'. \quad X^{-1}(C) \in \mathcal{F}$$

Then, given a probability measure μ on $\mathcal{F}$, X induces a probability measure μ' on $\mathcal{F}'$ as

$$\forall C \in \mathcal{F}'. \quad \mu'(C) = \mu(X^{-1}(C))$$

If μ' is a discrete probability measure then it can be constructed by a probability distribution over Ω', called *probability mass function (pmf)*, defined as

$$P([X = x]) = \mu(X^{-1}(x))$$

for each $x \in \Omega'$. The random variable in this case is called discrete. If X, Y are discrete random variables then we can define a discrete random variable (X, Y) by its pmf

$$P([X = x, Y = y]) = \mu(X^{-1}(x) \cap X^{-1}(y))$$

If X is a real-valued discrete random variable then its *expected value* (or *expectation*) is defined as

$$E(X) = \sum_i x_i \, P([X = x_i])$$

A family $\rho = \{p_v(\cdot)\}_v$ of probability measures parametrized on v (where v can range over $\{0, \ldots, n\}$ for some natural n) is called a *stochastic kernel.*[1].

[1]The general definition of stochastic kernel is more complicated (cfr. [TM09]), but it reduces to this one in the discrete case, which is what we use in this thesis.

Notation: We will use capital letters A, B, X, Y, Z to denote random variables and calligraphic letters $\mathcal{A}, \mathcal{B}, \mathcal{X}, \mathcal{Y}, \mathcal{Z}$ to denote their image. With a slight abuse of notation we will use p (and $p(x), p(y)$) to denote either

- a probability distribution, when $x, y \in \Omega$, or
- a probability measure, when $x, y \in \mathcal{F}$ are events, or
- the probability mass function $P([X = x]), P([Y = y])$ of the random variables X, Y respectively, when $x \in \mathcal{X}, y \in \mathcal{Y}$.

2.2 Probabilistic automata

Let $\mu\colon \mathcal{S} \to [0,1]$ be a discrete probability distribution on a countable set $\mathcal{S}$, and let the set of all discrete probability distributions on $\mathcal{S}$ be $\mathcal{D}(\mathcal{S})$.

A *probabilistic automaton* [Seg95] is a quadruple $M = (\mathcal{S}, \mathcal{L}, \hat{s}, \vartheta)$ where $\mathcal{S}$ is a countable set of *states*, $\mathcal{L}$ is a finite set of *labels* or *actions*, $\hat{s}$ is the *initial* state, and ϑ is a *transition function* $\vartheta : \mathcal{S} \to \mathcal{P}(\mathcal{D}(\mathcal{L} \times \mathcal{S}))$. If $\vartheta(s) = \emptyset$ then s is a *terminal* state. We write $s \to \mu$ for $\mu \in \vartheta(s)$, $s \in \mathcal{S}$. Moreover, we write $s \xrightarrow{\ell} r$ for $s, r \in \mathcal{S}$ whenever $s \to \mu$ and $\mu(\ell, r) > 0$. A *fully probabilistic automaton* is a probabilistic automaton satisfying $|\vartheta(s)| \leq 1$ for all states. In such an automaton, when $\vartheta(s) \neq \emptyset$, we overload the notation and denote by $\vartheta(s)$ the distribution outgoing from s.

A *path* in a probabilistic automaton is a sequence $\sigma = s_0 \xrightarrow{\ell_1} s_1 \xrightarrow{\ell_2} \cdots$ where $s_i \in \mathcal{S}$, $\ell_i \in \mathcal{L}$ and $s_i \xrightarrow{\ell_{i+1}} s_{i+1}$. A path can be *finite* in which case it ends with a state. A path is *complete* if it is either infinite, or finite ending in a terminal state. Given a finite path σ, $\mathit{last}(\sigma)$ denotes its last state. Let $\mathrm{Paths}_s(M)$ denote the set of all paths, $\mathrm{Paths}^\star{}_s(M)$ the set of all finite paths, and $\mathrm{CPaths}_s(M)$ the set of all complete paths of an automaton M, starting from the state s. We will omit s if $s = \hat{s}$. Paths are ordered by the prefix relation, which we denote by $\leq$. The *trace* of a path is the sequence of actions in $\mathcal{L}^* \cup \mathcal{L}^\infty$ obtained by removing the states, hence for the above σ we have $\mathit{trace}(\sigma) = l_1 l_2 \ldots$. If $\mathcal{L}' \subseteq \mathcal{L}$, then $\mathit{trace}_{\mathcal{L}'}(\sigma)$ is the projection of $\mathit{trace}(\sigma)$ on the elements of $\mathcal{L}'$.

Let $M = (\mathcal{S}, \mathcal{L}, \hat{s}, \vartheta)$ be a (fully) probabilistic automaton, $s \in \mathcal{S}$ a state, and let $\sigma \in \mathrm{Paths}^\star_s(M)$ be a finite path starting in s. The *cone* generated by σ is the set of complete paths $\langle \sigma \rangle = \{\sigma' \in \mathrm{CPaths}_s(M) \mid \sigma \leq \sigma'\}$. Given a fully probabilistic automaton $M = (\mathcal{S}, \mathcal{L}, \hat{s}, \vartheta)$ and a state s, we can calculate the *probability value* $\mathbf{P}_s(\sigma)$ of any finite path σ starting in s as follows:

$$\mathbf{P}_s(s) = 1, \text{ and}$$

$$\mathbf{P}_s(\sigma \xrightarrow{\ell} s') = \mathbf{P}_s(\sigma)\ \mu(\ell, s') \text{ where } \mathit{last}(\sigma) \to \mu$$

Let $\Omega_s \stackrel{\text{def}}{=} \text{CPaths}_s(M)$ be the sample space, and let $\mathcal{F}_s$ be the smallest σ-algebra induced by the cones generated by all the finite paths of M. Then $\mathbf{P}$ induces a unique *probability measure* on $\mathcal{F}_s$ (which we will also denote by $\mathbf{P}_s$) such that $\mathbf{P}_s(\langle\sigma\rangle) = \mathbf{P}_s(\sigma)$ for every finite path σ starting in s. For $s = \hat{s}$ we write $\mathbf{P}$ instead of $\mathbf{P}_{\hat{s}}$.

A (total) *scheduler* for a probabilistic automaton M is a function defined as $\zeta\colon \text{Paths}^\star(M) \to (\mathcal{L} \times \mathcal{D}(\mathcal{S}) \cup \{\bot\})$ such that for all finite paths σ, if $\vartheta(\mathit{last}(\sigma)) \neq \emptyset$ then $\zeta(\sigma) \in \vartheta(\mathit{last}(\sigma))$, and $\zeta(\sigma) = \bot$ otherwise. Hence, a scheduler ζ selects one of the available transitions in each state, and determines therefore a fully probabilistic automaton, obtained by pruning from M the alternatives that are not chosen by ζ. A scheduler is history dependent since it takes into account the path and not only the current state. It is possible to define partial schedulers, i.e. schedulers that may halt the execution at any time. In this thesis, however, we will consider only total schedulers, to be more in line with the standard semantics of CCS.

2.3 CCS with internal probabilistic choice

In this section we present an extension of standard CCS ([Mil89]) obtained by adding internal probabilistic choice. The resulting calculus can be seen as a simplified version of the probabilistic π-calculus presented in [HP00, PH05] and it is similar to the one considered in [DPP05]. The restriction to CCS and to internal choice is suitable for the scope of this thesis.

Let a range over a countable set of *channel names*.

The syntax of CCS_p is the following:

$\alpha ::= a \mid \bar{a} \mid \tau$	**prefixes**
$P, Q ::=$	**processes**
$\alpha.P$	prefix
$\mid P \mid Q$	parallel
$\mid P + Q$	nondeterministic choice
$\mid \sum_i p_i P_i$	internal probabilistic choice
$\mid (\nu a)P$	restriction
$\mid !P$	replication
$\mid 0$	nil

where the p_i's in the probabilistic choice should be non-negative and their sum should be 1. We will also use the notation $P_1 +_p P_2$ to represent a binary sum $\sum_i p_i P_i$ with $p_1 = p$ and $p_2 = 1 - p$.

The semantics of a CCS_p term is a probabilistic automaton defined inductively on the basis of the syntax according to the rules in Figure 2.1. We write $s \xrightarrow{a} \mu$ when (s, a, μ) is a transition of the probabilistic automaton. Given a process Q and a measure μ, we denote by $\mu \mid Q$ the measure μ' such that

$$\text{ACT}\quad \frac{}{\alpha.P \xrightarrow{\alpha} \delta(P)} \qquad \text{RES}\quad \frac{P \xrightarrow{\alpha} \mu \qquad \alpha \neq a, \overline{a}}{(\nu a)P \xrightarrow{\alpha} (\nu a)\mu}$$

$$\text{SUM1}\quad \frac{P \xrightarrow{\alpha} \mu}{P + Q \xrightarrow{\alpha} \mu} \qquad \text{SUM2}\quad \frac{Q \xrightarrow{\alpha} \mu}{P + Q \xrightarrow{\alpha} \mu}$$

$$\text{PAR1}\quad \frac{P \xrightarrow{\alpha} \mu}{P \mid Q \xrightarrow{\alpha} \mu \mid Q} \qquad \text{PAR2}\quad \frac{Q \xrightarrow{\alpha} \mu}{P \mid Q \xrightarrow{\alpha} P \mid \mu}$$

$$\text{COM}\quad \frac{P \xrightarrow{a} \delta(P') \quad Q \xrightarrow{\overline{a}} \delta(Q')}{P \mid Q \xrightarrow{\tau} \delta(P' \mid Q')} \qquad \text{PROB}\quad \frac{}{\sum_i p_i P_i \xrightarrow{\tau} \sum_i p_i\, \delta(P_i)}$$

$$\text{REP1}\quad \frac{P \xrightarrow{\alpha} \mu}{!P \xrightarrow{\alpha} \mu \mid !P} \qquad \text{REP2}\quad \frac{P \xrightarrow{a} \delta(P_1) \quad P \xrightarrow{\overline{a}} \delta(P_2)}{!P \xrightarrow{\tau} \delta(P_1 \mid P_2 \mid !P)}$$

Figure 2.1: The semantics of CCS_p

$\mu'(P \mid Q) = \mu(P)$ for all processes P and $\mu'(R) = 0$ if R is not of the form $P \mid Q$. Similarly $(\nu a)\mu = \mu'$ such that $\mu'((\nu a)P) = \mu(P)$.

A transition of the form $P \xrightarrow{a} \delta(P')$, i.e. a transition having for target a Dirac measure, corresponds to a transition of a non-probabilistic automaton (a standard labeled transition system). Note that each rule of CCS_p corresponds to one rule of CCS, except for PROB. The latter models the internal probabilistic choice: a silent τ transition is available from the sum to a measure containing all of its operands, with the corresponding probabilities.

Note that in the produced probabilistic automaton, all transitions to non-Dirac measures are silent. This is similar to the *alternating model* [HJ89], however our case is more general because the silent and non-silent transitions are not necessarily alternated. On the other hand, with respect to the simple probabilistic automata the fact that the probabilistic transitions are silent looks like a restriction. It has been proved by Bandini and Segala [BS01], however, that the simple probabilistic automata and the alternating model are essentially equivalent, so, being in between, our model is equivalent as well.

Encoding message passing into CCS_p Sometimes it is convenient to make message passing explicit in the notation of CCS_p. Namely, we enrich its syntax by allowing the prefixes to be $c(a) \mid c\langle x\rangle \mid \tau$, where c, a, x are names, and the semantic rule COM is substituted by:

$$\text{COM'}\quad \frac{P \xrightarrow{c\langle a\rangle} \delta(P') \quad Q \xrightarrow{c(x)} \delta(Q')}{P \mid Q \xrightarrow{\tau} \delta(P' \mid Q'\,[^a/_x])}$$

where $P \xrightarrow{c\langle a\rangle} \delta(P')$ denotes a process that sends the name a through channel

c and then evolves to P', and $Q \xrightarrow{c(x)} \delta(Q')$ denotes a process that receives the name x through channel c and then evolves to Q'. Here $Q'\,[^a/_x]$ is the process Q' in which every occurrence of x is replace by a.

The expressive power of CCS_p with message passing and without it is the same [Mil89]. In this thesis we will use this fact and consider explicit message passing as an alias for the corresponding encoding into the presentation of CCS_p given in Figure 2.1.

Three

The rationale behind the use of information theory for leakage

"Why, only why?"
Nadia Vertti

In this chapter we review the most important concepts related to the *information theoretic* approach to quantitative information flow. We aim at presenting these concepts in a contextualized way, discussing the intuition behind them and interpreting what they mean in terms of security.

Plan of the Chapter Section 3.1 gives a brief overview on information theory for communication. Section 3.2 introduces the information theoretic approach to information flow. Section 3.3 presents and compares several different notions based on information theory that have been used in the literature to characterize uncertainty and leakage.

3.1 Information theory and communication

The study of information theory started with Claude E. Shannon's work on the problem of coding messages to be transmitted through unreliable (or noisy) channels. A *communication channel* is a (physical) means through which information can be transmitted. The input is fed into the channel, but due to noise or any other problems that can occur during the transmission, the output of the channel may not reflect with fidelity the input. It is usual to describe the unreliable behavior of the channel in a probabilistic way. In the discrete (finite) case, if $\mathcal{A} = \{a_1, a_2, \ldots, a_n\}$ represent the possible inputs for the channel, and $\mathcal{B} = \{b_1, b_2, \ldots, b_m\}$ represent the possible outputs, the channel's probabilistic behavior can be represented as a channel matrix $M_{n\times m}$ where

each element $M_{i,j}$ ($1 \leq i \leq n$, $1 \leq j \leq m$) is defined as the probability of the channel outputting b_j when the input is a_i. In this way, we can see the input and output as two correlated random variables linked by the channel's probabilistic behavior [1].

A unique feature of information theory is its use of a numerical measure of the amount of information gained when the contents of a message are learned. More specifically, information theory reasons about the degree of uncertainty of a certain random variable, and the amount of information that it can reveal about another random variable. Among the tools provided by information theory there are concepts as *entropy*, *conditional entropy*, *mutual information* and *channel capacity*, which will be reviewed in Section 3.3.1. We consider here only the discrete case, since this is enough for the scope of this thesis.

3.2 Information theory and information flow

Several works in the literature use an information theoretic approach to model the problem of information flow and define the leakage in a quantitative way, as for example [ZB05, CHM05, Mal07, MC08, MNS03, MNCM03, CPP08a]. The idea is to model the computational system as an *information theoretic channel*. The input represents the secret, the output represents the observable, and the correlation between the input and output (*mutual information*) represents the information leakage. The worst case leakage corresponds then to the *capacity* of the channel, which is by definition the maximum mutual information that can be obtained by varying the input distribution.

In the works mentioned above, the notion of mutual information is based on *Shannon entropy*, which (because of its mathematical properties) is the most established measure of uncertainty. From the security point of view, this measure corresponds to a particular model of attack and a particular way of estimating the security threat (vulnerability of the secret). Other notions have been considered, and argued to be more appropriate for security in certain scenarios. These include: *min-entropy* [Ré61, Smi09], *Bayes risk* [CT91, CPP08b], *guessing entropy* [Mas94], and *marginal guesswork* [Pli00]. In Section 3.3 we will discuss their meaning and show how they relate (or do not relate) to each other and to Shannon entropy.

Whatever definition of uncertainty (i.e. vulnerability) we want to adopt, the notion of leakage is inherent to the system and can be expressed in a uniform way as the difference between the initial uncertainty, i.e. the degree of ignorance about the secret *before* we run the system, and the remaining uncertainty, i.e. the degree of ignorance about the secret *after* we run the system and observe its outcome. Following the principle advocated by Smith

[1] Note that we are assuming that channels are *loseless*, since the rows are probability distributions instead of sub-probability distributions.

[Smi09], and by many others:

$$\begin{aligned} \textit{information leakage} \;=\; & \textit{initial uncertainty} \\ & - \\ & \textit{remaining uncertainty} \end{aligned} \tag{3.1}$$

In (3.1), the initial uncertainty depends solely on the input distribution, aka the *a priori distribution* or *prior*. Intuitively, the more uniform it is, the less we know about the secret (in the probabilistic sense). After we run the system, if there is a probabilistic correlation between input and output, then the observation of the output should increase our knowledge of the secret. This is determined by the fact that the distribution on the input changes: in fact we can update the probability of each input with the corresponding conditional probability of the same input, given the output. The new distribution is called the *a posteriori distribution*. In case the input and output are independent, then the a priori and the a posteriori distributions coincide and the knowledge should remain the same. We will use the attributes "a priori" (or "prior") and "a posteriori" to refer to before and after the observation of the output, respectively.

The above intuitions should be reflected by any reasonable notion of uncertainty: it should be higher on more uniform distributions, and it should decrease or remain equal with the observation of related events.

If the uncertainty is expressed in terms of Shannon entropy, then the initial uncertainty is the entropy of the input, the remaining uncertainty is the conditional entropy of the input given the output, and (3.1) matches exactly the definition of mutual information. This justifies the notion of leakage adopted in the works mentioned before ([ZB05, CHM05, Mal07, MC08, MNS03, MNCM03, CPP08a]).

The analogy between information flow in a system and a (simple) channel works well when:

(i) there is no nondeterminism, i.e. either the system is deterministic, or purely probabilistic; and

(ii) there is a precise temporal relation between secrets and observables in the computations; namely, the value of the secret is chosen at the beginning of the computation, and the computation of the system produces an observable outcome with a probability that depends solely on the chosen input and on the system. Furthermore, each new run of the system is independent from the previous ones.

Restriction (i) implies that for each secret there is exactly one conditional probability distribution on the observables, where the condition is the secret value. If a system is deterministic, then under the same input each run produces always the same output, with probability 1. Therefore the matrix contains only 0's and 1's. Yet the problem of inferring the secret is interesting,

because the same output may correspond to different inputs. If the system is probabilistic, i.e. it uses some randomized mechanisms, then the matrix usually contains probabilities different from 0 and 1.

Restriction (ii) ensures that this conditional distribution depends uniquely on the system (not on the input distribution). These conditional probabilities constitute the channel matrix. Note that in a (basic) information-theoretic channel the matrix must be invariant with respect to the input distribution, which is exactly what condition (ii) guarantees.

Unfortunately, usually conditions (i) and (ii) are too restrictive for real-life systems:

- Specifications typically need to use nondeterminism in order to abstract from implementation details. This is particularly compelling in the case of concurrent and distributed systems: The order in which the various components get executed and their interactions depend on scheduling policies that may differ from implementation to implementation. Furthermore, even if the scheduling policy is fixed, there are run time circumstances that may influence the relative speed of the processes. Nondeterminism is, in practice, an unavoidable aspect of concurrency.

- Secrets and observables often alternate and interact during an execution. In particular, the choice of a new secret may depend on previous observables. Furthermore, new executions of the systems may depend on previous ones. This may be due to the way the system works, or to the presence of an active adversary that may use the knowledge derived from previous observations to try to tamper with the mechanisms of the system, with the purpose of increasing the leakage. Examples of such systems, that we call here *interactive* systems (where interaction refers to the interplay between secrets and observables), can be found in the areas of game theory, auction protocols, web servers, GUI applications, etc.

In this thesis we consider the challenges of extending the information-theoretic approach to cases where these conditions are relaxed. More specifically, Chapter 4 concerns the suppression of condition (ii), and Chapter 6 deals with the suppression of condition (i).

3.3 Uncertainty and leakage

In this section we recall various definitions of uncertainty based on information theory proposed in the literature, and we discuss the relation with security attacks and the way of measuring their success. In general we consider the kind of threats that in the model of Köpf and Basin [KB07] are called *brute-force guessing attacks*, which can be summarized as follows: The goal of the

adversary is to determine the value of a random variable. He can make a series of queries to an oracle. Each query must have a *yes*/*no* answer. In general the adversary is *adaptive*, i.e. he can choose the next query depending on the answers to the previous ones. We assume that the adversary knows the a priori probability distribution. In this section, when we talk about the meaning in security of a particular measure of uncertainty, we refer to the work in [KB07].

In the following, A, B denote two discrete random variables with finitely many values $\mathcal{A} = \{a_1, \ldots, a_n\}$, $\mathcal{B} = \{b_1, \ldots, b_m\}$, and probability distributions $p_A(\cdot)$, $p_B(\cdot)$, respectively. We will use $A \wedge B$ to represent the random variable with carrier $\mathcal{A} \times \mathcal{B}$ and joint probability distribution $p_{A \wedge B}(a, b) = p_A(a) \cdot p(b \mid A = a)$, while $A \cdot B$ will denote the random variable with carrier $\mathcal{A} \times \mathcal{B}$ and probability distribution defined as product, i.e. $p_{A \cdot B}(a, b) = p_A(a) \cdot p_B(b)$. Clearly, if A and B are independent, we have $A \wedge B = A \cdot B$. We shall omit the subscripts on the probabilities when they are clear from the context. In reference to a channel, in general A will denote the input (secret), and B the output (observable).

3.3.1 Shannon entropy

The (Shannon) *entropy* of A is defined as

$$H(A) = -\sum_{\mathcal{A}} p(a) \log p(a)$$

The entropy measures the uncertainty of A. It takes its minimum value $H(A) = 0$ when $p_A(\cdot)$ is a point mass (also called delta of Dirac). The maximum value $H(A) = \log |\mathcal{A}|$ is obtained when $p_A(\cdot)$ is the uniform distribution. Usually the base of the logarithm is set to be 2 and the entropy is measured in *bits*. Roughly speaking, m bits of entropy means that we have 2^m values to choose from, assuming a uniform distribution.

The *conditional entropy* of A given B is defined as

$$H(A \mid B) \quad = \quad \sum_{b \in \mathcal{B}} p(b)\ H(A \mid B = b) \tag{3.2}$$

where

$$H(A \mid B = b) \quad = \quad -\sum_{a \in \mathcal{A}} p(a|b) \log\ p(a|b)$$

The conditional entropy measures the uncertainty of A when B is known. It is well-known that $0 \leq H(A|B) \leq H(A)$. The minimum value, 0, is obtained when A is completely determined by B. The maximum value $H(A)$ is obtained when A and B are independent.

The *mutual information* between A and B is defined as

$$I(A; B) = H(A) - H(A|B) \tag{3.3}$$

The mutual information measures the amount of information about A that we gain by observing B. It can be shown that $I(A;B) = I(B;A)$ and $0 \leq I(A;B) \leq H(A)$. If C is a third random variable, the *conditional mutual information* between A and B given C is defined as

$$I(A;B|C) = H(A|C) - H(A|B,C)$$

The (conditional) entropy and mutual information respect the *chain rules*. Namely, given the random variables $A_1, A_2, \ldots, A_k$, B and C, we have:

$$H(A_1, A_2, \ldots, A_k|C) = \sum_{i=1}^{k} H(A_i|A_1, \ldots, A_{i-1}, C)$$

$$I(A_1, A_2, \ldots, A_k; B|C) = \sum_{i=1}^{k} I(A_i; B|A_1, \ldots, A_{i-1}, C) \tag{3.4}$$

A *discrete memoryless channel* is a tuple $(\mathcal{A}, \mathcal{B}, p(\cdot|\cdot))$, where $\mathcal{A}, \mathcal{B}$ are the sets of input and output symbols, respectively, and $p(b|a)$ is the probability of observing the output symbol b when the input symbol is a. These conditional probabilities constitute the *channel matrix*. An input distribution $p_A(\cdot)$ over $\mathcal{A}$ together with the channel determine the joint distribution $p(a,b) = p(a|b) \cdot p(a)$ and consequently $I(A;B)$. The maximum $I(A;B)$ over all possible input distributions is the channel's *capacity* C:

$$C = \max_{p_A(\cdot)} I(A;B)$$

The famous *Channel Coding Theorem* by Shannon relates the capacity of the channel to its maximum transmission rate. In brief, the channel capacity is a tight upper bound for the maximum rate by which information can be reliably transmitted using the channel. Given an acceptable probability of error ξ, there is a natural number n and a coding for which n uses of the channel will result in messages being transmitted with at most the acceptable probability of error ξ.

Meaning in security To explain what $H(A)$ represents from the security point of view, consider a partition $\{\mathcal{A}_i\}_{i \in I}$ of $\mathcal{A}$. The adversary is allowed to ask questions of the form "does $A \in \mathcal{A}_i$?" according to some strategy. Let $n(a)$ be the number of questions that are needed to determine the value of a, when $A = a$. Then $H(A)$ represents the lower bound to the expected value of $n(\cdot)$, with respect to all possible partitions and strategies of the adversary [Pli00, KB07].

3.3.2 Min-entropy

In [Rén61], Rényi introduced a one-parameter family of entropy measures, intended as a generalization of Shannon entropy. The Rényi entropy of order α ($\alpha > 0$, $\alpha \neq 1$) of a random variable A is defined as

$$H_\alpha(A) \;=\; \frac{1}{1-\alpha} \log \sum_{a \in \mathcal{A}} p(a)^\alpha$$

Rényi's motivations were of an axiomatic nature: Shannon entropy satisfies four axioms, namely symmetry, continuity, value 1 on the Bernoulli uniform distribution, and the chain rule[2]:

$$H(A \wedge B) \;=\; H(A) + H(B \,|\, A) \tag{3.5}$$

The entropy of the joint probability, $H(A \wedge B)$, is more commonly denoted by $H(A, B)$. We will use the latter notation in the following.

Shannon entropy is also the *only* function that satisfies those axioms. If we replace, however, (3.5) with a weaker property representing the additivity of entropy for independent distributions:

$$H(A \cdot B) \;=\; H(A) + H(B)$$

then there are more functions satisfying the axioms, among which are all those of Rényi's family.

Shannon entropy is obtained by taking the limit of H_α as α approaches 1. In fact we can easily prove, using l'Hôpital's rule, that

$$H_1(A) \;\stackrel{\text{def}}{=}\; \lim_{\alpha \to 1} H_\alpha(A) \;=\; -\sum_{a \in \mathcal{A}} p(a) \log p(a)$$

We are particularly interested in the limit of H_α as α approaches ∞. This is called *min-entropy*. It can easily be proven that

$$H_\infty(A) \;\stackrel{\text{def}}{=}\; \lim_{\alpha \to \infty} H_\alpha(A) \;=\; -\log \max_{a \in \mathcal{A}} p(a)$$

Rényi considered also the α-generalization of the Kullback-Liebler divergence, which is defined as (assuming that p and q are distributions on the same set $\mathcal{X}$):

$$D_{KL}(p \,\|\, q) \;=\; \sum_{x \in \mathcal{X}} p(x) \, \log \frac{p(x)}{q(x)}$$

[2] The original axiom, called the grouping axiom, does not mention the conditional entropy. It corresponds, however, to the chain rule if the conditional entropy is defined as in (3.2).

Rényi's α-generalization is:

$$D_\alpha(p \parallel q) \;=\; \frac{1}{1-\alpha} \log \sum_{x \in \mathcal{X}} p(x)^\alpha \, q(x)^{\alpha-1}$$

The standard case, i.e. the Kullback-Liebler divergence, is again obtained by taking the limit of D_α as $\alpha \to 1$.

The interest of the above for our purposes lies on the fact that Shannon mutual information can equivalently be defined in terms of the Kullback-Liebler divergence (see for instance [CT91]):

$$I(A;B) \;=\; D_{KL}(A \wedge B \parallel A \cdot B)$$

Therefore, it seems natural to define the α-generalization of the mutual information as:

$$I^*_\alpha(A;B) \;=\; D_\alpha(A \wedge B \parallel A \cdot B)$$

Other α-generalizations of the mutual information, based on the same idea, are explored in [Csi95].

As $\alpha \to \infty$, the above definition gives the following min-version of the mutual information:

$$I^*_\infty(A;B) \;\stackrel{\text{def}}{=}\; \lim_{\alpha \to \infty} I_\alpha(A;B) \;=\; \log \max_{a,b} \frac{p(a,b)}{p(a)\; p(b)} \tag{3.6}$$

Another natural way to generalize $I(A;B)$ would be to replace H by H_α in Definition 3.3. Rényi did not define, however, the α-generalization of the conditional entropy, and there is no agreement on what it should be.

Various researchers, including Cachin [Cac97], have considered the following definition, based on (3.2):

$$H^{Cachin}_\alpha(A \mid B) \;=\; \sum_{b \in \mathcal{B}} p(b)\; H_\alpha(A \mid B = b)$$

which, as $\alpha \to \infty$, becomes

$$H^{Cachin}_\infty(A \mid B) \;=\; -\sum_{b \in \mathcal{B}} p(b)\; \log \max_{a \in \mathcal{A}} p(a \mid b) \tag{3.7}$$

An alternative proposal for $H_\infty(\cdot \mid \cdot)$ came from Smith [Smi09][3]:

$$H^{Smith}_\infty(A \mid B) \;=\; -\log \textstyle\sum_{b \in \mathcal{B}} \max_{a \in \mathcal{A}}\; p(a,b) \tag{3.8}$$

Using (3.7) and 3.8), and the analogue of (3.3) we can define I^{Cachin}_∞ and I^{Smith}_∞ [4].

[3] The same formulation had been already used by Dodis et al. in [DORS04], and Smith proposed it independently. Since it is Smith's work on the subject that motivates the approach used in this thesis, we opt to refer to this formulation as Smith's.

[4] The notation I^{Smith}_∞ is ours. Smith himself opts for not adopting it, since I^{Smith}_∞ is not symmetric.

Meaning in security The min-entropy can be related to a model of adversary who is allowed to ask exactly one question, which must be of the form "is $A = a$?" (one-try attacks). More precisely, the min-entropy $H_\infty(A)$ represents the (logarithm of the inverse of the) probability of success for this kind of attack and with the best strategy, which consists, of course, in choosing the a with the maximum probability.

As for $H_\infty(A \mid B)$ and $I_\infty(A; B)$, the most interesting versions in terms of security seem to be those of Smith. In fact, in this thesis we adopt his approach to information leakage, and we will, from now on, use the following notation:

- $H_\infty(A \mid B)$ stands for $H_\infty^{Smith}(A \mid B)$ and is referred to as *conditional min-entropy*;
- $I_\infty(A; B)$ stands for $I_\infty^{Smith}(A; B)$ and is referred to as *min-entropy leakage*.

In fact, the conditional min-entropy $H_\infty(A \mid B)$ represents the log of the inverse of the (expected value of the) probability that the same kind of adversary succeeds in guessing the value of A *a posteriori*, i.e. after observing the result of B. The complement of this probability is also known as *probability of error* or *Bayes risk*. Since in general B and A are correlated, observing B increases the probability of success. In fact, we can prove formally that $H_\infty(A \mid B) \leq H_\infty(A)$, with equality if A and B are independent. The min-entropy leakage $I_\infty(A; B)$ corresponds to the *ratio* between the probabilities of success a priori and a posteriori, which is a natural notion of leakage. Here $I_\infty(A; B)$ is in the format of (3.1), but the difference becomes a ratio due to the presence of the logarithms. Note that $I_\infty(A; B) \geq 0$, which seems desirable for a good notion of leakage. It has been proven in [BCP09] that C_∞ is obtained at the uniform distribution, and that it is equal to the sum of the maxima of each column in the channel matrix, i.e. $C_\infty = \sum_{b \in \mathcal{B}} \max_{a \in \mathcal{A}} p(b \mid a)$.

The definition of $I_\infty^*(A; B)$ in (3.6) has also an interpretation in security: it represents the maximum gain in the probability of success, i.e. the maximum ratio between the a posteriori and the a priori probability. Note that also $I_\infty^*(A; B)$ is always non-negative and it is 0 if and only if A and B are independent. More generally, $D_{KL}(p \parallel q)$ and its α-extension $D_\alpha(p \parallel q)$ should represent the "inefficiency" of an adversary who bases its strategy on the distribution q, when in fact the real distribution is p. Hence $I_\alpha^*(A; B)$ defined as $D_\alpha(A \wedge B \parallel A \cdot B)$ should represent the gain of the adversary in revising his strategy according to the knowledge of the correlation between A and B.

Concerning H_α^{Cachin} and I_α^{Cachin}, they have some nice properties. For instance they enjoy weak versions of the chain rule (3.5). More precisely, the "$=$" in (3.5) becomes "$\geq$" for $\alpha < 1$, and "$\leq$" for $\alpha > 1$. There is no general relation between $H_\infty^{Cachin}(A \mid B)$ and $H_\infty(A)$, and in particular I_∞^{Cachin} is not guaranteed to be non-negative.

3.3.3 Guessing entropy

The notion of guessing entropy was introduced by Massey in [Mas94]. Let us assume, for simplicity, that the elements of $\mathcal{A}$ are ordered by decreasing probabilities, i.e. if $1 \leq i < j \leq n$ then $p(a_i) \geq p(a_j)$. Then the guessing entropy is defined as follows:

$$H_G(A) = \sum_{1 \leq i \leq |\mathcal{A}|} i\, p(a_i)$$

Massey did not define the notion of conditional guessing entropy. In some works, like [Cac97, KB07], it is defined analogously to (3.2):

$$H_G(A \mid B) = \sum_{b \in \mathcal{B}} p(b)\; H_G(A \mid B = b)$$

Meaning in security Guessing entropy represents an adversary who is allowed to ask repeatedly questions of the form "is $A = a$?". More precisely, $H_G(A)$ represents the expected number of questions that the adversary needs to ask to determine the value of A, assuming that he follows the best strategy, which consists, of course, in choosing the a's in order of decreasing probability.

$H_G(A \mid B)$ represents the expected number of questions *a posteriori*, i.e. after observing the value of B and reordering the queries according to the updated probabilities (i.e. the queries will be chosen in order of decreasing a posteriori probabilities).

Also in this case, $H_G(A \mid B)$ is not necessarily smaller than or equal to $H_G(A)$, so the corresponding notion of mutual information is not guaranteed to be non-negative[5].

3.3.4 Marginal guesswork

The marginal guesswork is a variant of guessing entropy that was proposed by Pliam [Pli00]. It is parametric in a number $\eta > 0$, and is defined as follows. Again, we assume that the elements of $\mathcal{A}$ are ordered by decreasing probabilities.

$$H_\eta(A) = \min\{j \mid \sum_{1 \leq i \leq j} p(a_i) > \eta\}$$

Pliam did not define the conditional version of marginal guesswork, but in [KB07] it is defined following (3.2):

$$H_\eta(A \mid B) = \sum_{b \in \mathcal{B}} p(b)\; H_\eta(A \mid B = b)$$

[5] This problem is inherent to the probabilistic case, and therefore it does not occur in [KB07], since that work considers only deterministic systems.

Meaning in security Consider again an adversary who is allowed to ask repeatedly questions of the form "is $A = a$?". $H_\eta(A)$ represents the minimum number of questions that the adversary needs to ask to determine the value of A with probability at least η.

$H_\eta(A \mid B)$ represents the same notion, but using the a posteriori probabilities. Again, it is not necessarily the case that $H_\eta(A \mid B) \leq H_\eta(A)$.

3.3.5 Comparison and discussion

The various notions of entropy discussed in this section have been carefully compared with Shannon entropy, to conclude that in general there is no tight relation. Fano's inequality gives a lower bound to the Bayes risk in terms of (conditional) Shannon entropy, and Rényi [Ré61], Hellman-Raviv [HR07], and Santhi-Vardi [SV06] give upper bounds as well, but all these are rather weak. Smith has shown in [Smi09] that the orderings induced on channels by the Bayes risk and by Shannon entropy are in general unrelated.

Massey has shown that the exponential of the Shannon entropy is a lower bound for the guessing entropy, and that, in case of a geometric distribution, the bound is tight. Massey has also shown that in the general case the Shannon entropy can be arbitrarily close to 0 while the guessing entropy is constant [Mas94].

As for the marginal guesswork. Pliam has shown that it is essentially unrelated with Shannon entropy [Pli00].

In this thesis we focus on the concepts of leakage based on Shannon entropy (Chapter 4) and min-entropy (Chapter 5).

Four

Information flow in interactive systems

"True interactivity is not about clicking on icons or downloading files, it's about encouraging communication."
Edwin Schlossberg

The key idea behind the information-theoretic approaches to information flow is to interpret the system as an information-theoretic channel, where the secrets are the input and the observables are the output. The channel matrix consists of the conditional probabilities $p(b \mid a)$, defined as the measure of the executions producing the observable b, relative to those which contain the secret a. The leakage is represented by the mutual information, and the worst-case leakage by the capacity of the channel (see Chapter 3 for reference).

In information theory, however, there are several different models of channels. So far the works in the literature about information theory applied to information flow have focused on the simplest kind of channels: discrete memoryless channels where the absence of feedback is implicitly assumed. This classical approach has been successfully used in scenarios where the secret value is assumed to be chosen at the beginning of the computation. In this chapter, however, we are interested in the more general scenario in which secrets can be chosen at any point. More precisely, we consider *interactive systems*, i.e. systems in which the generation of secrets and the occurrence of observables can alternate during the computation and influence each other. Examples of interactive systems include *auction protocols* like [Vic61, Sub98, SA99]. Some of these have become very popular thanks to their integration in Internet-based electronic commerce platforms [Eba, Ebi, Mer]. Other examples of interactive programs include web servers, GUI applications, and command-line programs [BPS+09].

Unfortunately, the information-theoretic approach which interprets interactive systems as classical channels is not valid. More specifically, in such systems the channel matrix is not invariant with respect to the input distri-

bution, so the channel capacity cannot be calculated in the traditional way. Therefore, the notion of maximum leakage as standard capacity is also compromised.

The goal of this chapter is to extend the classical information-theoretic approach to information flow to the more complicated scenario of interactive systems.

Contribution The main contributions of this chapter can be summarized as follows.

- We show that by considering the richer channels that support memory and feedback it is possible to retrieve the correspondence between systems and channels. We prove that there is a complete correspondence between interactive systems and channels with memory and feedback, and we show how to model the latter as the former.

- We propose the use of directed information, as opposed to mutual information, to represent leakage in interactive systems. Recent results in information theory [TM09] have shown that, in channels with memory and feedback, the transmission rate does not correspond to the maximum mutual information (the standard notion of capacity), but rather to the maximum normalized *directed information*, a concept introduced by Massey [Mas90]. We argue that in interactive channels the real leakage is due to the directed information from secrets to observables, whereas the directed information from observables to secrets (corresponding to feedback) is a characteristic of the system itself and should not be counted as leakage.

- We show that our model is a proper extension of the classical one, i.e. in the absence of interactivity the model of channels with memory and feedback collapses into the model of memoryless channels without feedback. Moreover, in that case also the concepts of mutual information and directed information from input to output coincide, the same holds for the concepts of capacity and directed capacity. We argue that in the classical approach mutual information is a good measure of leakage exactly because of this property: in the absence of feedback mutual information and directed information from input to output are the same.

- We show that the capacity of the channels associated to interactive systems is a continuous function with respect to a pseudometric based on the Kantorovich metric. The continuity of the channel capacity was also proved in [DJGP02] for simple channels, but the proof does not adapt to the case of channels with memory and feedback and we had to devise a different technique.

Plan of the Chapter This chapter is organized as follows. In Section 4.1 we introduce the concept of interactive systems and we show why channels

without memory and feedback are inadequate in this scenario. In Section 4.2 we review the notion of channels with memory and feedback, which is the core of the model we propose. We discuss the concept of directed information and also the concept of capacity in the presence of feedback. Section 4.3 contains the main contribution in this chapter: We explain how Interactive Information Hiding Systems (IIHSs) can be modeled using channels with memory and feedback. In particular we show that for any IIHS there is always a channel that simulates its probabilistic behavior. In Section 4.4 we discuss our notion of adversary and we define the quantification of information leakage as the channel's directed information from input to output, or as the directed capacity, depending on whether the input distribution is fixed or not. In Section 4.5 we apply our model to an example, the Cocaine Auction protocol. In Section 4.6 we propose a pseudometric structure on IIHSs based on the Kantorovich metric. We also show that the capacity of the channels associated to interactive systems is a continuous function with respect to this pseudometric. In Section 4.7 we present some related work, and in Section 4.8 we review and discuss the main results of the chapter, and consider future work.

4.1 Interactive systems

In this section we exemplify the problems that arise when we try to apply the classical information-theoretic approach to interactive systems. In order to derive an information-theoretic channel, at a first glance it would seem natural to define the channel matrix by using the definition of $p(b \mid a)$ in terms of the joint and marginal probabilities $p(a, b)$ and $p(b)$. Namely, the entry $p(b \mid a)$ would be defined as the measure of the traces with (secret, observable)-projection (a, b), divided by the measure of the traces with secret projection a. An approach of this kind was proposed in [DJGP02]. In the interactive case, however, this construction does not really produce an information-theoretic channel. In fact, by definition a channel should be invariant with respect to the input distribution, and this is not the case here, as shown by the following example.

Example 1. *Figure 4.1 represents a web-based interaction between one seller and two possible buyers,* rich *and* poor*. The seller can offer two different products,* cheap *and* expensive*, with given probabilities. Once the product is offered, each buyer may try to buy it, with a certain probability. For simplicity we assume that the buyers' offers are mutually exclusive. We assume that the offers are observables, in the sense that they are made public on the website, while the identity of the buyer that actually buys the product should be kept secret from an external observer. The symbols r, q_1, q_2, $\overline{r}$, $\overline{q_1}$, $\overline{q_2}$ represent probabilities, with the convention that $\overline{r} = 1 - r$ (and the same for the pairs q_1, $\overline{q_1}$ and q_2, $\overline{q_2}$).*

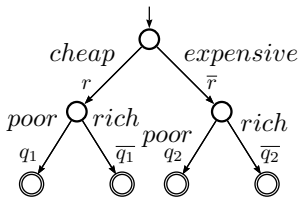


Figure 4.1: Interactive system of Example 1

Following [DJGP02] we can compute the conditional probabilities using $p(b|a) = \frac{p(a,b)}{p(a)}$, thus obtaining the matrix in Table 4.1. The matrix however is not invariant with respect to the input distribution. For instance for $r = \overline{r} = \frac{1}{2}$, $q_1 = \frac{2}{3}$, and $q_2 = \frac{1}{3}$ we obtain the matrix in Table 4.2(a). If we change the input distribution, for instance by changing the value of q_2 to be $\frac{1}{6}$, also the matrix changes. We obtain, indeed, the new matrix illustrated in Table 4.2(b).

	cheap	*expensive*
poor	$\frac{rq_1}{rq_1+\overline{r}q_2}$	$\frac{\overline{r}q_2}{rq_1+\overline{r}q_2}$
rich	$\frac{r\overline{q_1}}{r\overline{q_1}+\overline{r}\overline{q_2}}$	$\frac{\overline{r}\overline{q_2}}{r\overline{q_1}+\overline{r}\overline{q_2}}$

Table 4.1: Channel matrix for Example 1

Consequently, when the secrets occur *after* the observables and *depend on them*, we cannot consider the conditional probabilities (of the observables given the secrets) as representing a classical channel from secrets to observables, and we cannot apply the standard information-theoretic concepts. In particular, we cannot use "the capacity of the matrix" (defined by considering the matrix as a channel matrix, and taking the maximum mutual information over all possible inputs) because in general the maximum is given by a distribution different from the one that was used to define the matrix, hence the result would be unsound.

The first contribution of this chapter is to consider an extension of the theory of channels which makes the information-theoretic approach applicable also in the case of interactive systems. A richer notion of channels, known in information theory as *channels with memory and feedback*, serves our purposes. The dependence of inputs on previous outputs corresponds to feedback, and the dependence of outputs on previous inputs and outputs corresponds to memory. Recent results in information theory [TM09] have shown that, in such channels, the transmission rate does not correspond to the maximum mutual information (the standard notion of capacity), but rather to the maximum normalized *directed information*, a concept introduced by Massey [Mas90]. We propose to adopt this latter notion to represent leakage.

	cheap	*expensive*	Input distr.
poor	$\frac{2}{3}$	$\frac{1}{3}$	$p(poor) = \frac{1}{2}$
rich	$\frac{1}{3}$	$\frac{2}{3}$	$p(rich) = \frac{1}{2}$

(a) $r = \frac{1}{2}, q_1 = \frac{2}{3}, q_2 = \frac{1}{3}$

	cheap	*expensive*	Input distr.
poor	$\frac{4}{5}$	$\frac{1}{5}$	$p(poor) = \frac{5}{12}$
rich	$\frac{2}{7}$	$\frac{5}{7}$	$p(rich) = \frac{7}{12}$

(b) $r = \frac{1}{2}, q_1 = \frac{2}{3}, q_2 = \frac{1}{6}$

Table 4.2: Two different channel matrices induced by two different input distributions for Example 1

Our model of attacker is the interactive version of the attacker associated to Shannon entropy in the classification of Köpf and Basin [KB07], discussed in Chapter 3. In the case of a standard single-use channel, the invulnerability degree of the secret *before* the attacker observes the output is the entropy of the input, determined by its a priori distribution. The invulnerability degree *after* the attacker observes the output is the conditional entropy of the input given the output, determined by its a posteriori distribution. The latter is always smaller than or equal to the first. The difference between these invulnerability degrees corresponds to the mutual information, and represents the leakage of the system. In our interactive framework we consider the same scenario, but iterated. At each time step, we consider the input sequence so far; and the increase of its vulnerability caused by the observation of the new output is given by the contribution of the present step to the leakage. The sum of all these contributions represents the total leakage and, as we will see, corresponds to Massey's directed information. We will come back to the model of attacker in Section 4.4, and discuss also a variant of this interpretation.

A second contribution of our work is the proof that the channel capacity is a continuous function of a pseudometric on interactive systems based on the Kantorovich metric. The reason why we are interested in the continuity of the capacity is for computability purposes. Given a function f from a (pseudo)metric space X to a (pseudo)metric space Y the continuity of f means that, given a sequence of objects $x_1, x_2, \ldots \in \mathcal{X}$ converging to $x \in \mathcal{X}$, the sequence $f(x_1), f(x_2), \ldots \in \mathcal{Y}$ converges to $f(x) \in \mathcal{Y}$. Hence $f(x)$ can be approximated by the objects $f(x_1), f(x_2), \ldots$. The typical use of this property is in the case of execution trees generated by programs containing loops. Generally the automaton expressing the semantics of the program can be seen as the (metric) limit of the sequence of trees generated by unfolding the loop

to an increasingly deeper level. The continuity of the capacity means that we can approximate the real capacity by the capacities of these trees.

4.2 Discrete channels with memory and feedback

In this section we present the notion of channel with memory and feedback. We assume a scenario in which the channel is used repeatedly, in a finite temporal sequence of steps $1, \ldots, T$. Intuitively, memory means that the output at time $t (1 \leq t \leq T)$ depends on the input and output histories, i.e. on the inputs up to time t, and on the output up to time $t-1$. Feedback means that the input at time t depends on the outputs up to time $t-1$.

We adopt the following notation.

Convention 2. *Given sets of symbols (alphabets) $\mathcal{A} = \{a_1, \ldots, a_n\}$, $\mathcal{B} = \{b_1, \ldots, b_n\}$, we use a Greek letter (α, β, ...) to denote a sequence of symbols ordered in time. Given a sequence $\alpha = a_{i_1} a_{i_2} \ldots a_{i_m}$, the notation α_t represents the symbol at time t, i.e. a_{i_t}, while α^t represents the sequence $\alpha_{i_1} \alpha_{i_2} \ldots \alpha_{i_t}$. For instance, in the sequence $\alpha = a_3 a_7 a_5$, we have $\alpha_2 = a_7$ and $\alpha^2 = a_3 a_7$. Analogously, if X is a random variable, then X^t denotes the sequence of t consecutive instances $X_1, \ldots, X_t$ of X.*

We now define formally the concepts of memory and feedback. Consider a channel from input A to output B. The channel behavior after T uses can be fully described by the joint distribution of $A^T \times B^T$, namely by the probabilities $p(\alpha^T, \beta^T)$. Using the chain rule, we can decompose these probabilities as follows:

$$p(\alpha^T, \beta^T) = \prod_{t=1}^{T} p(\alpha_t | \alpha^{t-1}, \beta^{t-1}) p(\beta_t | \alpha^t, \beta^{t-1}) \tag{4.1}$$

Definition 3. *We say that a channel* has feedback *if, in general, $p(\alpha_t | \alpha^{t-1}, \beta^{t-1}) \neq p(\alpha_t | \alpha^{t-1})$, i.e. the probability of α_t depends not only on α^{t-1}, but also on β^{t-1}. Analogously, we say that the channel* has memory *if, in general, $p(\beta_t | \alpha^t, \beta^{t-1}) \neq p(\beta_t | \alpha_t)$, i.e. the probability of β_t depends on α^t and β^{t-1}.*

Note that in the opposite case, i.e. when $p(\alpha_t | \alpha^{t-1}, \beta^{t-1})$ coincides with $p(\alpha_t | \alpha^{t-1})$ and $p(\beta_t | \alpha^t, \beta^{t-1})$ coincides with $p(\beta_t | \alpha_t)$, we have a classical channel (memoryless, and without feedback), in which each use is independent from the previous ones. The only possible dependency on the history is the one of a_t on a^{t-1}. This is because $A_1, \ldots, A_T$ are in general correlated, due to the fact that they are produced by an encoding function. Note that in absence of

memory and feedback (4.1) reduces to:

$$\begin{aligned} p(\alpha^T, \beta^T) &= \prod_{t=1}^{T} p(\alpha_t|\alpha^{t-1})\, p(\beta_t|\alpha_t) \\ &= p(\alpha^T) \prod_{t=1}^{T} p(\beta_t|\alpha_t) \qquad \text{(by the chain rule)} \end{aligned} \tag{4.2}$$

from which we can derive the standard formula for a classical channel after T uses.

$$\begin{aligned} p(\beta^T|\alpha^T) &= \frac{p(\alpha^T, \beta^T)}{p(\alpha^T)} \\ &= \prod_{t=1}^{T} p(\beta_t|\alpha_t) \qquad \text{(by (4.2))} \end{aligned}$$

So far we have given a very abstract description of a channel with memory and feedback. We now discuss a more concrete notion following the presentation of [TM09]. Such a channel, represented in Figure 4.2, consists of a sequence of components formally defined as a family of stochastic kernels $\{p(\cdot\,|\alpha^t, \beta^{t-1})\}_{t=1}^{T}$ over $\mathcal{B}$.

The probabilities $p(\beta_t|\alpha^t, \beta^{t-1})$ represent *innermost behavior* of the channel at time t, $1 \leq t \leq T$: the internal channel takes the input α_t and, depending on the history of inputs and outputs so far, it produces an output symbol β_t. The output is then fed back to the encoder with delay one. On the input side, at time t the encoder takes the message and the past output symbols β^{t-1} and produces a channel input symbol α_t according to the code function φ_t (we will explain this concept in the next paragraph). At final time T the decoder takes all the channel outputs β^T and produces the decoded message $\hat{W}$. The order in time is the following:

$$\text{Message } W, \quad \alpha_1, \beta_1, \quad \alpha_2, \beta_2, \quad \ldots, \quad \alpha_T, \beta_T, \quad \text{Decoded Message } \hat{W}$$

Let us now explain the concept of code function. Intuitively, a code function is a strategy to encode the message into a suitable representation to be transmitted through the channel. There is a code function for each possible message, and the functions are fixed at the very beginning of the transmission (time $t = 0$). The encoding, however, can use the information provided via feedback, so each component φ_t ($1 \leq t \leq T$) of the code function takes as parameter the history of feedback β^{t-1} to generate the next input symbol α_t.

Formally, let $\mathcal{F}_t$ be the set of all measurable maps $\varphi_t : \mathcal{B}^{t-1} \to \mathcal{A}$ endowed with a probability distribution, and let F_t be the corresponding random variable. Let $\mathcal{F}^T$, F^T denote the Cartesian product on the domain and

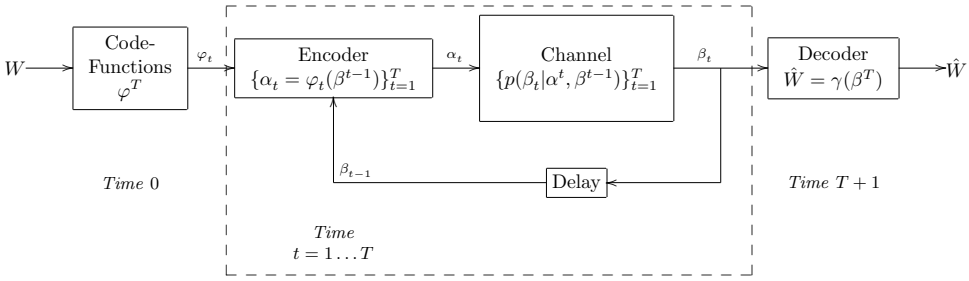


Figure 4.2: Model for discrete channel with memory and feedback

the random variable, respectively. A *channel code function* is an element $\varphi^T = (\varphi_1, \ldots, \varphi_T) \in \mathcal{F}^T$.

Note that, by the chain rule, $p(\varphi^T) = \prod_{t=1}^T p(\varphi_t|\varphi^{t-1})$. Hence the distribution on $\mathcal{F}^T$ is uniquely determined by a sequence $\{p(\varphi_t|\varphi^{t-1})\}_{t=1}^T$. The notation $\varphi^t(\beta^{t-1})$ will represent the $\mathcal{A}$-valued t-tuple $(\varphi_1, \varphi_2(\beta^1), \ldots, \varphi_t(\beta^{t-1}))$.

In Information Theory this kind of channel is used to encode and transmit messages. If $\mathcal{W}$ is a set of messages of cardinality M with typical element w, endowed with a probability distribution, a *channel code* is a set of M channel code functions $\varphi^T[w]$, interpreted as follows: for message w, if at time t the channel feedback is β^{t-1}, then the channel encoder outputs $\varphi_t[w](\beta^{t-1})$. A *channel decoder* is a map from $\mathcal{B}^T$ to $\mathcal{W}$ which attempts to reconstruct the input message after observing all the output history β^T from the channel.

4.2.1 The power of feedback

The original purpose of *communication channel* models is to represent data transmission from a source to a receiver. Shannon's Channel Coding Theorem states that for every channel there is an encoding scheme that allows a transmission rate arbitrarily close to the channel capacity with a negligible probability of error (if the number of uses of the channel is large enough). A general way to find an optimal encoding scheme that is also easy to decode has not been found yet. The use of feedback, however, can simplify the design of the encoder and of the decoder. The following example illustrates the idea.

	0	1	e
0	0.8	0	0.2
1	0	0.8	0.2

Table 4.3: Channel matrix for binary erasure channel

Example 2. *Consider a discrete memoryless binary channel* $\{\mathcal{A}, \mathcal{B}, p(.|.)\}$ *with* $\mathcal{A} = \{0, 1\}$, $\mathcal{B} = \{0, 1, \mathsf{e}\}$ *and the channel matrix of Table 4.3. This kind of*

channel is called erasure channel *because it can lose (or* erase*) bits during the transmission with a certain probability. Namely, any bit has* 0.8 *probability of being correctly transmitted, and* 0.2 *probability of being lost. On the output side the encoder is able to detect whether the bit was erased (by receiving an* e *symbol), but it cannot tell which was the actual value of the original bit. The Channel Coding Theorem guarantees that the maximum information transmission rate in this channel is (*2 *to the power of) the channel capacity, i.e.* 0.8 *bits per use of the channel.*

Following simple principles described in [CT06], an encoding that achieves the capacity can be easily obtained if the channel can be used with feedback. The idea is an adaptation of the stop-and-wait protocol [Sta06, Tan89]. Suppose that every bit received on the output end of the channel is fed back noiselessly to the source with delay 1*. Define the encoding as follows: for each bit transmitted, the encoder checks via feedback whether the bit was erased. If not, the encoder moves on to transmit the text of the message. If yes, the encoder transmits the same bit again.*

It is easy to see that with this encoding scheme the transmission rate is 0.8 *bit per usage of the channel, since in* 80% *of the cases the bit is transmitted properly, and in* 20% *it is lost and a retransmission is needed.*

We now proceed to illustrate in more detail the design and the function of the encoder and decoder.

An example illustrating the the encoder/decoder design

We proceed with the erasure channel of Example 2 to show how the enriched model of channels with memory and feedback can be used to transmit the message, and in particular how the feedback can be used to design the encoder. We assume that the set $\mathcal{W}$ of possible messages consists of all finite sequences of bits. The role of the code functions is to encode the message W into a suitable representation for the stochastic kernels within the channel. The input and output alphabets for the stochastic kernels are $\mathcal{A} = \{0, 1\}$ and $\mathcal{B} = \{0, 1, \mathtt{e}\}$, respectively. We assume that at most T uses of the channel are allowed and we use t, with $1 \leq t \leq T$, to represent the tth time step.

We consider a sort of memory that depends only on the input history and we abstract from its specific form by defining a function $\eta : \mathcal{P}(\mathcal{A}^t) \to [0, 1]$ that maps each possible input history to a correction factor to be added to (or subtracted from) a base probability value. We compute the contribution of η to the base values using arithmetic modulo 2, in such a way that the resulting values are still a probability distribution. More precisely, the stochastic kernels

are defined as follows.

$$\begin{aligned}
p(\beta_t = 0|\alpha^{t-1}0, \beta^{t-1}) &= 0.8 - \eta(\alpha^{t-1}) \\
p(\beta_t = 1|\alpha^{t-1}0, \beta^{t-1}) &= 0 \\
p(\beta_t = \mathtt{e}|\alpha^{t-1}0, \beta^{t-1}) &= 0.2 + \eta(\alpha^{t-1}) \\
p(\beta_t = 0|\alpha^{t-1}1, \beta^{t-1}) &= 0 \\
p(\beta_t = 1|\alpha^{t-1}1, \beta^{t-1}) &= 0.8 - \eta(\alpha^{t-1}) \\
p(\beta_t = \mathtt{e}|\alpha^{t-1}1, \beta^{t-1}) &= 0.2 + \eta(\alpha^{t-1})
\end{aligned} \tag{4.3}$$

Correspondingly, the general form of the channel matrix for each time $1 \leq t \leq T$ is shown in Table 4.4.

	0	1	e
$\alpha_t = 0, \beta^{t-1}$	$0.8 - \eta(\alpha^{t-1})$	0	$0.2 + \eta(\alpha^{t-1})$
$\alpha_t = 1, \beta^{t-1}$	0	$0.8 - \eta(\alpha^{t-1})$	$0.2 + \eta(\alpha^{t-1})$

Table 4.4: General form of channel matrix

The code functions are chosen at time $t = 0$, based on the message to be transmitted. For illustration purposes, let us suppose that the message is the sequence of three bits $W = 011$. The other cases of W are analogous.

At time $t = 1$, the channel is used for its first time and the feedback history so far is empty $\beta^0 = \epsilon$. The encoder selects the input symbol $\alpha_0 = 0$, as in (4.4).

$$f_1[W = 011](\beta^0 = \epsilon) = 0 \tag{4.4}$$

At time $t = 2$, the feedback history consists of only one symbol, and in principle the possibilities are either $\beta^1 = 0$, $\beta^1 = 1$ or $\beta^1 = \mathtt{e}$. In the first case, the first bit was successfully transmitted and the encoder can go on to the second bit of the message. By the way the channel is defined, the second case is not really possible, so it is not important how the reaction function is defined for this case. We will denote this indifference by attributing to the function the symbol $\mathtt{x}$ instead of a 0 or a 1. In the last case, $\beta^1 = \mathtt{e}$, the first bit was erased and the encoder tries to retransmit the bit 0. We can write it formally as below.

$$\begin{aligned}
f_2[W = 011](\beta^1 = 0) &= 1 \\
f_2[W = 011](\beta^1 = 1) &= \mathtt{x} \\
f_2[W = 011](\beta^1 = \mathtt{e}) &= 0
\end{aligned} \tag{4.5}$$

At time $t = 3$ the feedback histories allowed by the channel are $\beta^2 \in \{01, 0\mathtt{e}, \mathtt{e}0, \mathtt{ee}\}$ (the other ones have zero probability). In the first case, $\beta^2 = 01$ the two first bits of the message have been transmitted correctly and the encoder can send the third bit. If $\beta^2 = 0\mathtt{e}$, the transmission of the first bit

was successful, but the second bit was erased and needs to be resent. In the case $\beta^2 = \texttt{e0}$, the first bit was erased in the first try but was successfully transmitted in the second try, so now the encoder can move to the second bit of the message. In the last case, $\beta^2 = \texttt{ee}$, the two tries were unsuccessful and the encoder still needs to transmit the first bit of the message. Formally:

$$\begin{aligned}
&f_3[W = 011](\beta^2 = 00) = \texttt{x}\\
&f_3[W = 011](\beta^2 = 01) = 1\\
&f_3[W = 011](\beta^2 = 0\texttt{e}) = 1\\
&f_3[W = 011](\beta^2 = 10) = \texttt{x}\\
&f_3[W = 011](\beta^2 = 11) = \texttt{x}\\
&f_3[W = 011](\beta^2 = 1\texttt{e}) = \texttt{x}\\
&f_3[W = 011](\beta^2 = \texttt{e}0) = 1\\
&f_3[W = 011](\beta^2 = \texttt{e}1) = \texttt{x}\\
&f_3[W = 011](\beta^2 = \texttt{ee}) = 0
\end{aligned} \tag{4.6}$$

We can easily extend the construction of code functions f_t for $3 \leq t \leq T$ using this encoding scheme.

The decoder is very simple: once all time steps $1, \ldots, T$ have taken place, it just takes the whole output trace β^T and removes the occurrences of the erased bit symbol $\texttt{e}$ in order to recover the original message.

Table 4.5 shows a possible behavior of a binary erasure channel with memory and feedback in a scenario where the message is $W = 011$ and the channel can be used at most $T = 3$ times. Note that in this particular example the maximum number of uses of the channel is achieved before the whole message is successfully sent: the decoder can recover only the two first bits of the original message.

We can observe that the channel capacity in the above example does not increase with the addition of feedback (it is 0.8 bit per usage of the channel with or without feedback). This is because the channel is memoryless: *feedback does not increase the capacity of discrete memoryless channels* [CT06]. In general however, feedback *does* increase the capacity of channels with memory.

4.2.2 Directed information and capacity of channels with feedback

In classical Information Theory, the channel capacity, which is related to the channel's transmission rate by Shannon's Channel Coding Theorem, can be obtained as the supremum of the mutual information over all possible input distributions. In the presence of feedback, however, this correspondence no longer holds. More specifically, mutual information no longer represents the information flow from A^T to B^T. Intuitively, this is due to the fact that mutual information expresses correlation, and therefore it is increased by feedback (Example 5 in Section 4.4 depicts this fact). Yet feedback, i.e. the way the

Time t	Code functions $f_t(\beta^{t-1})$	Feedback history β^{t-1}	Encoder $\alpha_t = f_t[W](\beta^{t-1})$	Channel $p(\beta_t \vert \alpha^t, \beta^{t-1})$	Decoder $\hat{W} = \gamma(\beta^T)$
$t = 0$	Code functions for $W = 011$ are selected.	———	———	———	———
$t = 1$	As in (4.4)	ϵ	$\alpha_1 = f_1[W = 011](\epsilon) = 0$	According to $p(\beta_1 \vert 0, \epsilon)$ produces $\beta_1 = \mathsf{e}$	———
$t = 2$	As in (4.5)	e	$\alpha_2 = f_2[W = 011](\mathsf{e}) = 0$	According to $p(\beta_2 \vert 00, \mathsf{e})$ produces $\beta_2 = 0$	———
$t = 3$	As in (4.6)	$\mathsf{e}0$	$\alpha_3 = f_3[W = 011](\mathsf{e}0) = 1$	According to $p(\beta_3 \vert 001, \mathsf{e}0)$ produces $\beta_3 = 1$	———
$t = 4$	———	———	———	———	Decoded message $\hat{W} = \gamma(\beta^3 = \mathsf{e}01) = 01$

Table 4.5: A possible evolution of the binary channel with time, for $W = 011$ and $T = 3$

output influences the next input, is not part of the information to be transmitted. If we want to maintain the correspondence between the transmission rate and capacity, we need to replace the mutual information with *directed information* [Mas90].

Definition 4. *In a channel with feedback, the directed information from input A^T to output B^T is defined as*

$$I(A^T \to B^T) = \sum_{t=1}^{T} I(A^t; B_t | B^{t-1})$$

In the other direction, the directed information from B^T to A^T is defined as

$$I(B^T \to A^T) = \sum_{t=1}^{T} I(A_t; B^{t-1} | A^{t-1})$$

In Section 4.4 we will discuss the relation between directed information and mutual information, as well as the correspondence with information leakage. For the moment, we only present the extension of the concept of capacity.

Let $\mathcal{D}_T = \{p(\alpha_t | \alpha^{t-1}, \beta^{t-1})\}_{t=1}^{T}$ be the set of all input distributions in presence of feedback. For finite T, the capacity of a channel with memory and

feedback is:

$$C_T = \sup_{\mathcal{D}_T} \frac{1}{T} I(A^T \to B^T) \tag{4.7}$$

The capacity is also defined when T is infinite, see [TM09]. In this thesis, however, we only need to consider the finite case.

4.3 Interactive systems as channels with memory and feedback

Interactive Information Hiding Systems (IIHS) were introduced in [APvRS10] to represent systems where secrets (inputs) and observables (outputs) can interleave and influence each other. They are a variant of probabilistic automata in which actions are divided into secrets and observables. They can be of two kinds: *fully probabilistic*, and *secret-nondeterministic* (or *input-nondeterministic*). In the former there is no nondeterminism, while in the latter every secret choice is fully nondeterministic. In this chapter we consider *normalized* IIHSs, in which secrets and observables alternate, and the actions at the first level are secrets. We note that this is not really a restriction, because given an IIHS which is not normalized, it is always possible to transform it into a normalized IIHS which is equivalent to the former one up to a given execution level. The reader can find further below in this Section the formal definition of the transformation. Furthermore, we require that for each state s and each action ℓ there is at most one state that can be reached from s by performing an ℓ transition.

In this section we formalize the notion of IIHS and we show how to associate to an IIHS a channel with memory and feedback.

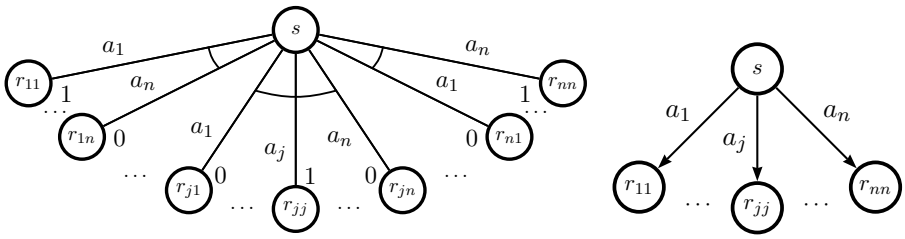
Definition 5. *A (normalized) IIHS is a triple $\mathcal{I} = (M, \mathcal{A}, \mathcal{B})$, where $\mathcal{A}$ and $\mathcal{B}$ are disjoint sets of secrets and observables respectively, M is a probabilistic automaton $(\mathcal{S}, \mathcal{L}, \hat{s}, \vartheta)$ with $\mathcal{L} = \mathcal{A} \cup \mathcal{B}$, and, for each $s \in \mathcal{S}$:*

1. *either $\vartheta(s) \subseteq \mathcal{D}(\mathcal{A} \times \mathcal{S})$ or $\vartheta(s) \subseteq \mathcal{D}(\mathcal{B} \times \mathcal{S})$. We call s a* secret state *in the first case, and an* observable state *in the second case;*
2. *if $s \xrightarrow{\ell} r$ then: if s is a secret state then r is an observable state, and if s is an observable state then r is a secret state;*
3. *$\hat{s}$ is a secret state;*
4. *if s is an observable state then $|\vartheta(s)| \leq 1$;*
5. *either:*
   - *(i) for every secret state s we have $|\vartheta(s)| \leq 1$* (fully probabilistic IIHS), *or*

(ii) *for every secret state s there exist a_i and s_i $(i = 1, \ldots, n)$ such that $\vartheta(s) = \{\delta(a_i, s_i)\}_{i=1}^n$, where $\delta(a_i, s_i)$ is the Dirac measure* (secret-nondeterministic IIHS);

6. *for every state s and action ℓ there exists a unique state r such that $s \xrightarrow{\ell} r$.*

In the rest of the chapter we will omit the adjective "normalized" for simplicity. In the above definition, Conditions 1 and 2 imply that the IIHS is alternating between secrets and observables. Moreover, all the transitions between nodes at two consecutive depths have either secret actions only, or observable actions only. Condition 3 means that the first level contains secret actions. Condition 4 means that all observable transitions are fully probabilistic. Condition 5 means that either all secret transitions are fully probabilistic, either they are all fully nondeterministic. The term "nondeterministic" is justified by the fact that the scheme of Condition 5ii represented in Figure 4.3(a), is equivalent to the one of Figure 4.3(b).



(a) Nondeterministic input using Dirac measures (b) Equivalent scheme

Figure 4.3: Scheme of secret transitions for secret-nondeterministic IIHSs

Note that we do not consider here internal nondeterminism which can arise from interleaving of concurrent processes. This means that we make a rather restricted use of probabilistic automata, but this is enough for our purposes. The nondeterminism generated by concurrency gives rise to a new set of problems (see for example [CPP08a]) which are orthogonal to those considered in this chapter.

Condition 6 means that the secret and observable actions determine the states. As a consequence, the actions are enough to retrieve the path. This is expressed by the following proposition:

Proposition 6. *Given an IIHS, consider two paths σ and σ'. If $trace_{\mathcal{A}}(\sigma) = trace_{\mathcal{A}}(\sigma')$ and $trace_{\mathcal{B}}(\sigma) = trace_{\mathcal{B}}(\sigma')$, then $\sigma = \sigma'$.*

Proof. By induction on the length of the traces. The initial state of the automaton is uniquely determined by the empty (secret and observable) traces. Assume now we are in a state s uniquely determined by secret and observable

traces α and β, respectively. If s makes a secret transition $s \xrightarrow{a} s'$, then by Condition 6 there is only one state s' reachable from s via an a-transition, and therefore s' is uniquely determined by the secret trace $\alpha' = \alpha a$ and the observable trace β. The case in which s makes an observable transition is similar. □

The normalization of IIHS trees

In this section we will address the problem of *normalizing* an IIHS, namely transforming it into a stratified automaton in which secret and observable actions alternate level by level. The process of normalization described bellow is general enough to be applied to any IIHS without loss of generality or expressive power.

Let $\mathcal{A}$ and $\mathcal{B}$ represent the secret and observable actions, respectively. Consider a general IIHS $\mathcal{I} = (M, \mathcal{A}, \mathcal{B})$ with $M = (Q, \mathcal{L}, \hat{s}, \vartheta)$, where $\mathcal{L} = \mathcal{A} \cup \mathcal{B}$. Assume that we are only interested in executions that involve up to T interactions, i.e. T uses of the system, with one secret taking place and one observable produced at each time.

In the normalization process, we unfold the automaton up to level $2T$, since there is one secret symbol and one observable symbol for each step. We also extend the secret alphabet $\mathcal{A}$ with a new symbol $a_* \notin \mathcal{A}$ and the observable alphabet $\mathcal{B}$ with a new symbol $b_* \notin \mathcal{B}$. These new symbols will be used as placeholders when we need to re-balance the tree. Let $\mathcal{A}' = \mathcal{A} \cup \{a_*\}$ and $\mathcal{B}' = \mathcal{B} \cup \{b_*\}$.

For a given level t let $\mathit{labels}(\mathcal{I}, t)$ be the set of all labels of transitions that can be performed with a non-zero probability from the states at the tth level of the automaton. Formally:

$$\mathit{labels}(\mathcal{I}, t) \equiv \{\ell \in \mathcal{L} \mid \exists \sigma, s \ . \ |\sigma| = t, \ \mathit{last}(\sigma) \xrightarrow{\ell} s\}$$

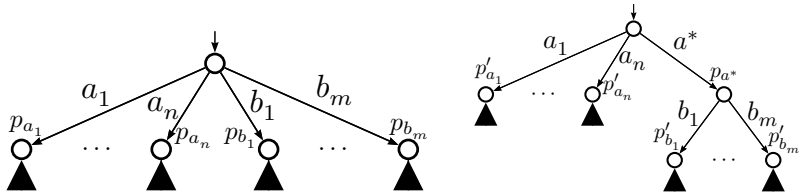
The normalization of the IIHS $\mathcal{I}$ leads to an equivalent IIHS $\mathcal{I}' = (M', \mathcal{A}', \mathcal{B}')$, where $M' = (Q', \mathcal{L}', \hat{s}', \vartheta')$ and $\mathcal{L}' = \mathcal{A}' \cup \mathcal{B}'$; and such that, for every $1 \leq t \leq 2T$:

1. $\mathit{labels}(\mathcal{I}', t) \subseteq \mathcal{A}'$ or $\mathit{labels}(\mathcal{I}', t) \subseteq \mathcal{B}'$;
2. $\mathit{labels}(\mathcal{I}', t) \subseteq \mathcal{A}'$ if and only if $\mathit{labels}(\mathcal{I}', t{+}1) \subseteq \mathcal{B}'$, for $1 \leq t \leq T{-}1$;
3. $\mathit{labels}(\mathcal{I}', 1) \subseteq \mathcal{A}'$;

Condition 1 states that each level consists of either the secret actions only, or the observable actions only. Condition 2 states that secret and observable levels alternate. Condition 3 says that the automaton starts with a secret level.

The proof is straightforward. First, the new symbols a_* and b_* are placeholders for the absence of a secret and observable symbol, respectively. If in

a given level t we want to have only secret symbols, we can postpone the occurrences of observable symbols at this level as follows: add a_* to the secret level and "move" all the observable symbols to the subtree of a_*. Figure 4.4 exemplifies the local transformations we need to make on the tree.



(a) Local nodes of the tree before the transformation

(b) Local nodes of the tree after the transformation

Figure 4.4: Local transformation in an IIHS tree

Note that in 4.4(b) the introduction of new nodes changed the probabilities of the transitions in the tree. In general, whenever we need to introduce a_* in order to postpone the observable symbols, the probabilities change as follows:

1. For every a_i, $1 \leq i \leq n$, the associated probability is maintained as $p'_{a_i} = p_{a_i}$;
2. The probability of the new symbol a_* is introduced as $p_{a_*} = \sum_{k=0}^{m} p_{b_k}$;
3. If $p_{a_*} \neq 0$, then for $1 \leq i \leq m$, the associated probability of b_j is updated to $p'_{b_j} = p_{b_j}/p_{a_*} = p_{b_j}/\sum_{k=0}^{m} p_{b_k}$. If $p_{a_*} = 0$, then $p'_{b_j} = 0$, for $1 \leq i \leq m$, and $p_{b_*} = 1$.

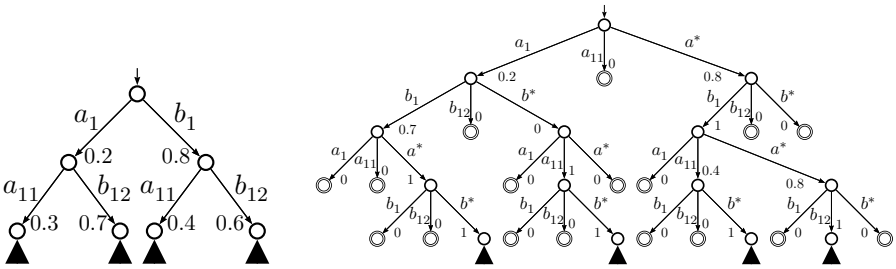
The subtrees of each node of the original tree are preserved as they are, until we apply the same transformation to them. If a node does not have a subtree (i.e. no descendants), we create a subtree by adding all the possible actions in $\mathcal{B}$ with probability 0, and the action b_* with probability 1.

If we are normalizing an observable level, the same rules apply, guarding the proper symmetry between secrets and observables. We then proceed in the same way on the deeper levels of the tree. Figure 4.5 shows an example of a full transformation on a tree (for the sake of readability, we omit the levels where only $a_* = 1$ or $b_* = 1$).

4.3.1 Construction of the channel associated to an IIHS

We now show how to associate a channel to an IIHS.

In an interactive system secrets and observables may interleave and influence each other. Considering a channel with memory and feedback is a way to capture this rich behavior. Secrets have a causal influence on observables via the channel, and, in the presence of interactivity, observables have a causal



(a) Tree before transformation

(b) Tree after transformation

Figure 4.5: Transformation in an IIHS tree

influence on secrets via feedback. This alternating mutual influence between secrets and observables can be modeled by repeated uses of the channel. Each time the channel is used it represents a different state of the computation, and the conditional probabilities of observables on secrets can depend on this state. The addition of memory to the model allows expressing the dependency of the channel matrix on such a state.

We will see that a secret-nondeterministic IIHS determines a channel as specified by its stochastic kernels, while a fully probabilistic IIHS determines, additionally, the input distribution.

In Section 4.5 we will give an extensive and detailed example of how to make such a construction for an actual security protocol.

Given a path σ of length $2t-1$, we will denote $trace_{\mathcal{A}}(\sigma)$ by α^t, and $trace_{\mathcal{B}}(\sigma)$ by β^{t-1}.

Definition 7. *Let $\mathcal{I}$ be an IIHS. For each t, the channel's stochastic kernel corresponding to $\mathcal{I}$ is defined as $p(\beta_t|\alpha^t, \beta^{t-1}) = \vartheta(s)(\beta_t, s')$, where s is the state reached from the root via the path σ whose secret and observable traces are α^t and β^{t-1} respectively.*

Note that s and s' in the previous definition are well defined: by Proposition 6, s is unique, and since the choice of β_t is fully probabilistic, s' is also unique.

The following example illustrates how to apply Definition 7, with the help of Proposition 6, to build the channel matrix of a simple example.

Example 3. *Let us consider an extended version of the website interactive system of Figure 4.1. We maintain the general definition of the system, i.e. there are two possible buyers (rich and poor, represented by rc. and pr., respectively) and two possible products (cheap and expensive, represented by chp. and exp., respectively). We still assume that offers are observable, since they are visible to everyone on the website, but the identity of buyers should be kept secret. We consider two consecutive rounds of offers and buys, which implies that, after*

normalization, $T = 3$. Figure 4.6 shows an automaton for this example in normalized form. Transitions with null probability are omitted, and the symbol a_* is used as a place holder to achieve the normalized IIHS.

To construct the stochastic kernels $\{p(\beta_t|\alpha^t, \beta^{t-1})\}_{t=1}^T$, we need to determine the conditional probability of an observable at time t given the history up to time t.

Let us take the case $t = 2$ and compute the conditional probability of the observable $\beta_2 = cheap$ given that the history of secrets up to time $t = 2$ is $\alpha^2 = a_*, poor$ and the history of observables is $\beta^1 = expensive$. Applying Definition 7, we see that $p(\beta_2 = cheap|\alpha^2 = a_*, poor, \beta^1 = expensive) = \vartheta(s)(cheap, s')$. By Proposition 6, the traces $\alpha^2 = a_*, poor, \beta^1 = expensive$ determine a unique state s in the automaton, namely, the state $s = 5$. Moreover, from the state 5 a unique transition labeled with the action cheap is possible, leading to the state $s' = 11$. Therefore, we can conclude that $p(\beta_2 = cheap|\alpha^2 = a_*, poor, \beta^1 = expensive) = \vartheta(s = 5)(cheap, s' = 11) = p_{23}$.

Similarly, with $t = 1$ and history $\alpha^1 = a_*, \beta^0 = \epsilon$, the observable symbol $\beta_1 = expensive$ can be observed with probability $p(\beta_1 = expensive|\alpha^1 = a_*, \beta^0 = \epsilon) = \vartheta(s = 0)(cheap, s' = 2) = \overline{p_1}$.

If $\mathcal{I}$ is fully probabilistic, then it determines also the input distribution and the dependency of α_t on β^{t-1} (feedback) and on α^{t-1}.

Definition 8. *Let $\mathcal{I}$ be an IIHS. If $\mathcal{I}$ is fully probabilistic, the associated channel has a conditional input distribution for each t defined as $p(\alpha_t|\alpha^{t-1}, \beta^{t-1}) = \vartheta(s)(\alpha_t, s')$, where s is the state reached from the root via the path σ whose secret and observable traces are α^{t-1} and β^{t-1} respectively.*

Example 4. *Since the system of Example 3 is fully probabilistic, we can calculate the values of the conditional probabilities $\{p(\alpha_t|\alpha^{t-1}, \beta^{t-1})\}_{t=1}^T$.*

Let us take, for instance, the case where $t = 2$ and compute the conditional probability of secret $\alpha_2 = poor$ given that the history of secrets up to time $t = 2$ is $\alpha^1 = a_*$ and the history of observables is $\beta^1 = expensive$. Applying Definition 8, we see that $p(\alpha_2 = poor|\alpha_1 = a_*, \beta^1 = expensive) = \vartheta(s)(poor, s')$. By Proposition 6, the traces $\alpha^1 = a_*, \beta^1 = expensive$ determine a unique state s in the automaton, namely, the state $s = 2$. Moreover, from the state 2 a unique transition labeled with the action poor is possible, leading to the state $s' = 5$. Therefore, we can conclude that $p(\alpha_2 = poor|\alpha_1 = a_*, \beta^1 = expensive) = \vartheta(s = 2)(poor, s' = 5) = q_{12}$.

Similarly, with $t = 3$ and history $\alpha^2 = a_*, rich, \beta^2 = cheap, expensive$, the secret symbol $\alpha_3 = rich$ can be observed with probability $p(\alpha_3 = rich|\alpha^2 = \alpha_*, rich, \beta^0 = cheap, expensive) = \vartheta(s = 10)(cheap, s' = 22) = \overline{q_{24}}$.

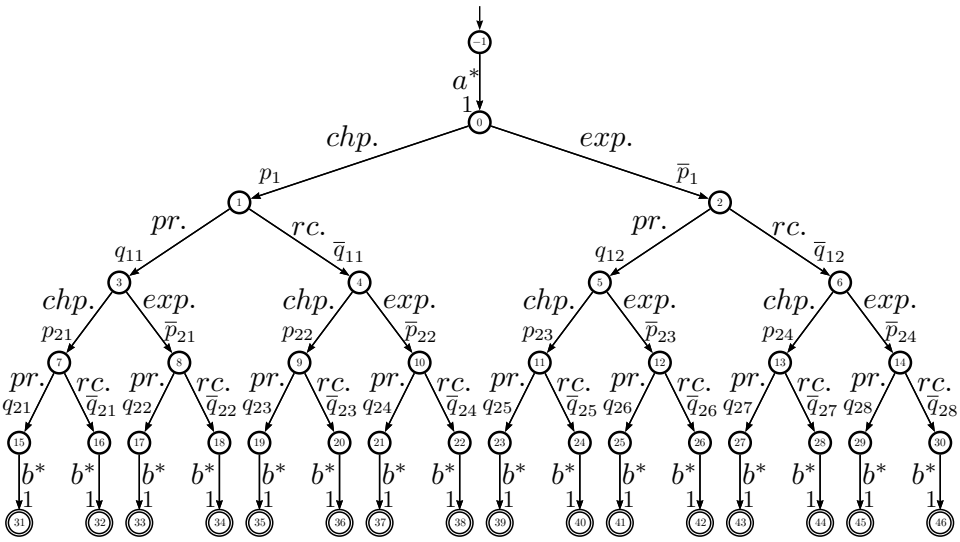


Figure 4.6: The normalized IIHS for the extended website example

4.3.2 Lifting the channel inputs to reaction functions

Taken together, Definitions 7 and 8 show how to obtain the the joint probabilities $p(\alpha^t, \beta^t)$ for a fully probabilistic IIHS. We still need to show, however, in what sense this joint probability distribution defines an information-theoretic channel.

The $\{p(\beta_t|\alpha^t, \beta^{t-1})\}_{t=1}^T$ determined by the IIHS trivially correspond to a channel's stochastic kernel. The problem resides in the conditional probabilities $\{p(\alpha_t|\alpha^{t-1}, \beta^{t-1})\}_{t=1}^T$. In an information-theoretic channel, the value of α_t is determined in the encoder by a deterministic function $\varphi_t(\beta^{t-1})$. Therefore, inside the encoder there is no possibility for a probabilistic description of α_t. The solution is to externalize this probabilistic behavior to the code functions.

As shown in [TM09], the original channel with feedback from input symbols $\mathcal{A}^T$ to output symbols $\mathcal{B}^T$ can be lifted to an equivalent channel without feedback from code functions $\mathcal{F}^T$ to output symbols $\mathcal{B}^T$. This transformation also allows us to calculate the channel capacity. Let $\{p(\varphi_t|\varphi^{t-1})\}_{t=1}^T$ be a sequence of code function stochastic kernels and let $\{p(\beta_t|\alpha^t, \beta^{t-1})\}_{t=1}^T$ be a channel with memory and feedback. The channel from F^T to B^T is constructed using a joint measure $Q(\varphi^T, \alpha^T, \beta^T)$ that respects the following constraints:

Definition 9. *A measure $Q(\varphi^T, \alpha^T, \beta^T)$ is said to be* consistent *with respect to the code function stochastic kernels $\{p(\varphi_t|\varphi^{t-1})\}_{t=1}^T$ and the channel $\{p(\beta_t|\alpha^t, \beta^{t-1})\}_{t=1}^T$ if, for each t:*

1. *There is no feedback to the code functions:*

$$Q(\varphi_t|\varphi^{t-1}, \alpha^{t-1}, \beta^{t-1}) = p(\varphi_t|\varphi^{t-1})$$

2. *The input is a function of the past outputs:*

$$Q(\alpha_t|\varphi^t, \alpha^{t-1}, \beta^{t-1}) = \delta_{\{\varphi_t(\beta^{t-1})\}}(\alpha_t)$$

   where δ is the Dirac measure;

3. *The properties of the underlying channel are preserved:*

$$Q(\beta_t|F^t = \varphi^t, A^t = \alpha^t, B^{t-1} = \beta^{t-1}) = p(\beta_t|\alpha^t, \beta^{t-1})$$

The following result states that there is only one consistent measure $Q(\varphi^T, \alpha^T, \beta^T)$.

Theorem 10 ([TM09]). *Given the probability distributions $\{p(\varphi_t|\varphi^{t-1})\}_{t=1}^T$ and a channel defined by $\{p(\beta_t|\alpha^t, \beta^{t-1})\}_{t=1}^T$, there exists only one consistent measure $Q(\varphi^T, \alpha^T, \beta^T)$. Furthermore the channel from $\mathcal{F}^T$ to $\mathcal{B}^T$ is given by:*

$$Q(\beta_t|\varphi^t, \beta^{t-1}) = p(\beta_t|\varphi^t(\beta^{t-1}), \beta^{t-1})$$

Since in our setting the concept of encoder makes little sense as there is no information to encode, we externalize the probabilistic behavior of α_t as follows. Code functions become *a single set of reaction functions $\{\varphi_t\}_{t=1}^T$ with β^{t-1} as parameter* (the message w does not play a role any more). Reaction functions can be seen as a model of how the environment reacts to given system outputs, producing new system inputs (they do not play a role of encoding a message). These reaction functions are endowed with a probability distribution that generates the probabilistic behavior of the values of α_t.

Definition 11. *A* reactor *is a distribution on reaction functions, i.e. a sequence of stochastic kernels $\{p(\varphi_t|\varphi^{t-1})\}_{t=1}^T$. A reactor R is* consistent with a fully probabilistic IIHS $\mathcal{I}$ *if it induces the compatible distribution $Q(\varphi^T, \alpha^T, \beta^T)$ such that, for every $1 \leq t \leq T$, $Q(\alpha_t|\alpha^{t-1}, \beta^{t-1}) = p(\alpha_t|\alpha^{t-1}, \beta^{t-1})$, where the latter is the probability distribution induced by $\mathcal{I}$.*

The main result of this section states that for any fully probabilistic IIHS there is a reactor that generates the probabilistic behavior of the IIHS. Before moving to this result, we need to introduce a lemma.

Lemma 12. *Let $\mathcal{X}, \mathcal{Y}$ be non-empty finite sets, and let $\tilde{x} \in \mathcal{X}, \tilde{y} \in \mathcal{Y}$. Let $p : \mathcal{X} \times \mathcal{Y} \to [0, 1]$ be a function such that, for every $x \in \mathcal{X}$, we have: $\sum_{y \in \mathcal{Y}} p(x, y) = 1$. Then:*

$$\sum_{\substack{f \in \mathcal{X} \to \mathcal{Y} \\ f(\tilde{x}) = \tilde{y}}} \prod_{x \in \mathcal{X}} p(x, f(x)) = p(\tilde{x}, \tilde{y})$$

Proof. By induction on the number of elements of $\mathcal{X}$.

Base case: $\mathcal{X} = \{\tilde{x}\}$. In this case:

$$\sum_{\substack{f \in \mathcal{X} \to \mathcal{Y} \\ f(\tilde{x}) = \tilde{y}}} \prod_{x \in \mathcal{X}} p(x, f(x)) = p(\tilde{x}, f(\tilde{x})) = p(\tilde{x}, \tilde{y})$$

Inductive case: Let $\mathcal{X} = \mathcal{X}' \cup \{\mathring{x}\}$, with $\tilde{x} \in \mathcal{X}'$ and $\mathring{x} \notin \mathcal{X}'$. Then:

$$\begin{aligned}
\sum_{\substack{f \in \mathcal{X}' \cup \{\mathring{x}\} \to \mathcal{Y} \\ f(\tilde{x}) = \tilde{y}}} \prod_{x \in \mathcal{X}' \cup \{\mathring{x}\}} p(x, f(x)) = & \quad \text{(by distributivity)} \\
\left(\sum_{\substack{f \in \mathcal{X}' \to \mathcal{Y} \\ f(\tilde{x}) = \tilde{y}}} \prod_{x \in \mathcal{X}'} p(x, f(x)) \right) \sum_{g \in \{\mathring{x}\} \to \mathcal{Y}} p(\mathring{x}, g(\mathring{x})) = & \quad \text{(by the assumption)} \\
\sum_{\substack{f \in \mathcal{X}' \to \mathcal{Y} \\ f(\tilde{x}) = \tilde{y}}} \prod_{x \in \mathcal{X}'} p(x, f(x)) = & \quad \text{(by the ind. hyp.)} \\
p(\tilde{x}, \tilde{y}) &
\end{aligned}$$

□

Theorem 13. *Let $\mathcal{I}$ be a fully probabilistic IIHS inducing the joint probability distribution $p(\alpha^t, \beta^t)$, $1 \leq t \leq T$, on secret and observable traces. It is always possible to construct a channel with memory and feedback, and an associated probability distribution $Q(\varphi^T, \alpha^T, \beta^T)$, which corresponds to $\mathcal{I}$ in the sense that, for every $1 \leq t \leq T$, α^t, β^t, the equality $Q(\alpha^t, \beta^t) = p(\alpha^t, \beta^t)$ holds.*

Proof. First note that, by laws of probability, $Q(\alpha^t, \beta^t) = \sum_{\varphi^t} Q(\varphi^t, \alpha^t, \beta^t)$. So we need to show that $\sum_{\varphi^t} Q(\varphi^t, \alpha^t, \beta^t) = p(\alpha^t, \beta^t)$ by induction on t.

Base case: $t = 1$. Let us define $Q(\varphi_1 | \epsilon) = p(\varphi_1(\epsilon))$ and $Q(\beta_1 | \alpha^1, \epsilon) =$

$p(\beta_1|\alpha_1)$. Then:

$$\begin{aligned}
\sum_{\varphi^1} Q(\varphi^1, \alpha^1, \beta^1) &= \\
\sum_{\varphi_1} Q(\varphi_1, \alpha_1, \beta_1) &= && \text{(by the chain rule)} \\
\sum_{\varphi_1} (Q(\varphi_1|\epsilon, \epsilon, \epsilon) \cdot Q(\alpha_1|\varphi_1, \epsilon, \epsilon) \cdot & \\
Q(\beta_1|\varphi_1, \alpha_1, \epsilon)) &= && \text{(by Definition 9)} \\
\sum_{\varphi_1} Q(\varphi_1|\epsilon)\delta_{\{\varphi_1(\epsilon)\}}(\alpha_1) Q(\beta_1|\alpha^1, \epsilon) &= && \text{(by construction of } Q) \\
\sum_{\varphi_1} p(\varphi_1(\epsilon))\delta_{\{\varphi_1(\epsilon)\}}(\alpha_1) p(\beta_1|\alpha_1) &= && \text{(by definition of } \delta) \\
p(\alpha_1)p(\beta_1|\alpha_1) &= \\
p(\alpha_1, \beta_1) &= \\
p(\alpha^1, \beta^1) &
\end{aligned}$$

Inductive case: Let us define $Q(\beta_t|\alpha^t, \beta^{t-1}) = p(\beta_t|\alpha^t, \beta^{t-1})$, and

$$Q(\varphi_t|\varphi^{t-1}) = \prod_{\beta^{t-1}} p(\varphi_t(\beta^{t-1})|\varphi^{t-1}(\beta^{t-2}), \beta^{t-1})$$

Note that, if we consider $\mathcal{X} = \{\beta^{t-1} \mid \beta_i \in \mathcal{B}, 1 \leq i \leq t-1\}$, $\mathcal{Y} = \mathcal{A}$, and $p(\beta^{t-1}, \alpha_t) = p(\alpha_t|\varphi^{t-1}(\beta^{t-2}), \beta^{t-1})$, then $\mathcal{X}$, $\mathcal{Y}$ and p satisfy the hypothesis of Lemma 12.

Then:

$$\begin{aligned}
\sum_{\varphi^t} Q(\varphi^t, \alpha^t, \beta^t) &= && \text{(by the chain rule)} \\
\sum_{\varphi^t} \big(Q(\varphi^{t-1}, \alpha^{t-1}, \beta^{t-1}) \cdot & \\
Q(\varphi_t|\varphi^{t-1}, \alpha^{t-1}, \beta^{t-1}) \cdot & \\
Q(\alpha_t|\varphi^t, \alpha^{t-1}, \beta^{t-1}) \cdot Q(\beta_t|\varphi^t, \alpha^t, \beta^{t-1})\big) &= && \text{(by Definition 9)} \\
\sum_{\varphi^t} \big(Q(\varphi^{t-1}, \alpha^{t-1}, \beta^{t-1}) \cdot Q(\varphi_t|\varphi^{t-1}) & \\
\delta_{\{\varphi_t(\beta^{t-1})\}}(\alpha_t) \cdot Q(\beta_t|\alpha^t, \beta^{t-1})\big) &= && \text{(by constr. of } Q) \\
\sum_{\varphi^t} \big(Q(\varphi^{t-1}, \alpha^{t-1}, \beta^{t-1}) \cdot & &&
\end{aligned}$$

$$\left(\prod_{\beta'^{t-1}} p(\varphi_t(\beta'^{t-1})|\varphi^{t-1}(\beta'^{t-2}),\beta'^{t-1})\right)\cdot$$

$$\delta_{\{\varphi_t(\beta^{t-1})\}}(\alpha_t)\cdot p(\beta_t|\alpha^t,\beta^{t-1})\Big) = \quad \text{(by definition of } \delta)$$

$$\sum_{\substack{\varphi^t\\ \varphi_t(\beta^{t-1})=\alpha_t}} \Big(Q(\varphi^{t-1},\alpha^{t-1},\beta^{t-1})\cdot$$

$$\left(\prod_{\beta'^{t-1}} p(\varphi_t(\beta'^{t-1})|\varphi^{t-1}(\beta'^{t-2}),\beta'^{t-1})\right)\cdot$$

$$p(\beta_t|\alpha^t,\beta^{t-1})\Big) =$$

$$\sum_{\varphi^{t-1}} \Big(Q(\varphi^{t-1},\alpha^{t-1},\beta^{t-1})p(\beta_t|\alpha^t,\beta^{t-1})$$

$$\sum_{\substack{\varphi_t\\ \varphi_t(\beta^{t-1})=\alpha_t}} \prod_{\beta'^{t-1}} p(\varphi_t(\beta'^{t-1})|\varphi^{t-1}(\beta'^{t-2}),\beta'^{t-1})\ \Big) = \quad \text{(by Lemma 12)}$$

$$\sum_{\varphi^{t-1}} \Big(Q(\varphi^{t-1},\alpha^{t-1},\beta^{t-1})\cdot p(\beta_t|\alpha^t,\beta^{t-1})\cdot$$

$$p(\alpha_t|\alpha^{t-1},\beta^{t-1})\ \Big) =$$

$$p(\beta_t|\alpha^t,\beta^{t-1})\cdot p(\alpha_t|\alpha^{t-1},\beta^{t-1})\cdot$$

$$\sum_{\varphi^{t-1}} Q(\varphi^{t-1},\alpha^{t-1},\beta^{t-1}) = \quad \text{(by ind. hyp.)}$$

$$p(\beta_t|\alpha^t,\beta^{t-1})\cdot p(\alpha_t|\alpha^{t-1},\beta^{t-1})\cdot p(\alpha^{t-1},\beta^{t-1}) = \quad \text{(by the chain rule)}$$

$$p(\alpha^t,\beta^t)$$

□

Corollary 14. *Let $\mathcal{I}$ be a fully probabilistic IIHS. Let $\{p(\beta_t|\alpha^t,\beta^{t-1})\}_{t=1}^T$ be a sequence of stochastic kernels and $\{p(\alpha_t|\alpha^{t-1},\beta^{t-1})\}_{t=1}^T$ a sequence of input distributions defined by $\mathcal{I}$ according to Definitions 7 and 8. Then the reactor $R=\{p(\varphi_t|\varphi^{t-1})\}_{t=1}^T$ compatible with respect to the $\mathcal{I}$ is given by:*

$$p(\varphi_1) = \qquad p(\alpha_1|\alpha^0,\beta^0) = p(\alpha_1) \tag{4.8}$$

$$p(\varphi_t|\varphi^{t-1}) = \prod_{\beta^{t-1}} p(\varphi_t(\beta^{t-1})|\varphi^{t-1}(\beta^{t-2}),\beta^{t-1}), \quad 2\le t\le T \tag{4.9}$$

Figure 4.7 depicts the model for IIHS. Note that, in relation to Figure 4.2, there are some simplifications: (1) no message W is needed; 2) the encoder

becomes an "interactor"; (3) the decoder is not used. At the beginning, a reaction function sequence φ^T is chosen and then the channel is used T times. At each usage t, the interactor produces the next input symbol α_t by applying the reaction function φ_t to the fed back output β^{t-1}. Then the channel produces an output β_t based on the stochastic kernel $p(\beta_t|\alpha^t, \beta^{t-1})$. The output is then fed back to the encoder, which uses it for producing the next input.

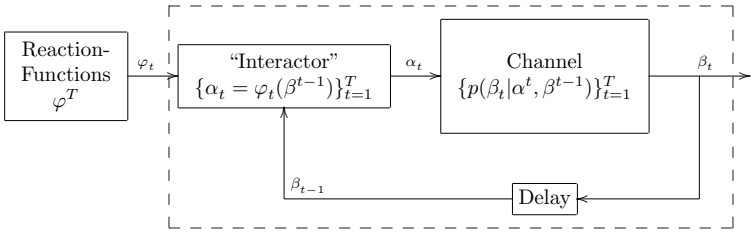


Figure 4.7: Channel with memory and feedback model for IIHS

We conclude this section by remarking on an intriguing coincidence: The notion of reaction function sequence φ^T, on the IIHSs, corresponds to the notion of deterministic scheduler [Seg95]. In fact, each reaction function φ_t selects the next step, α_t, on the basis of the β^{t-1} and α^{t-1} (generated by φ^{t-1}), and β^{t-1}, α^{t-1} represent the path up to that state.

4.4 Leakage in interactive systems

In this section we propose a definition for the notion of leakage in interactive systems. We first argue that mutual information is not the correct notion, and we propose to replace it with the directed information instead.

In the case of channels with memory and feedback, mutual information is defined as $I(A^T; B^T) = H(A^T) - H(A^T|B^T)$, and it is still symmetric (i.e. $I(A^T; B^T) = I(B^T; A^T)$). Since the roles of A^T and B^T in $I(A^T; B^T)$ are interchangeable, this concept cannot capture *causality*, in the sense that it does not imply that A^T causes B^T, nor conversely. Mutual information expresses *correlation* between the sequences of random variables A^T and B^T.

Mathematically the mutual information $I(A^T; B^T)$ for T uses of the channel can be expressed with the help of the chain rule of (3.4) in the following way.

$$I(A^T; B^T) = \sum_{t=1}^{T} I(A^T; B_t|B^{t-1})$$

In the equation above, each term of the sum is the mutual information between the random variable B_t and the whole sequence of random variables $A^T = A_1, \ldots, A_T$, given the history B^{t-1}. The equation emphasizes that at time $1 \leq t \leq T$, even though only the inputs $\alpha^t = \alpha_1, \alpha_2, \ldots, \alpha_t$ have been

fed to the channel, the whole sequence A^T, including $A_{t+1}, A_{t+2}, \ldots, A_T$, has a statistical correlation with B_t. Indeed, in the presence of feedback, B_t may influence $A_{t+1}, A_{t+2}, \ldots, A_T$.

In order to show how the concept of directed information contrasts with the above, let us recall its definition:

$$I(A^T \to B^T) = \sum_{t=1}^{T} I(A^t; B_t | B^{t-1}).$$

$$I(B^T \to A^T) = \sum_{t=1}^{T} I(A_t; B^{t-1} | A^{t-1}).$$

These notions capture the concept of *causality*, to which the definition of mutual information is indifferent. The correlation between inputs and outputs $I(A^T; B^T)$ is split into the information $I(A^T \to B^T)$ that flows from input to output through the channel and the information $I(B^T \to A^T)$ that flows from output to the input via feedback. Note that the directed information is not symmetric: the flow from A^T to B^T takes into account the correlation between A^t and B_t, while the flow from B^T to A^T takes into account the correlation between B^{t-1} and A_t.

It was proved in [TM09] that

$$I(A^T; B^T) = I(A^T \to B^T) + I(B^T \to A^T) \tag{4.10}$$

i.e. the mutual information is the sum of the directed information flow in both senses. Note that this formulation highlights the symmetry of mutual information from yet another perspective.

Once we split mutual information into directed information in the two opposite directions, it is important to understand the different roles that the information flow in each direction plays. $I(A^T \to B^T)$ represents the system behavior: via the channel the information flows from inputs to outputs according to the specification of the system, modeled by the channel stochastic kernels. This flow represents the amount of information an attacker can gain from the inputs by observing the outputs, and we argue that this is the real information leakage.

On the other hand, $I(B^T \to A^T)$ represents how the environment reacts to the system: given the system outputs, the environment produces new inputs. We argue that the information flow from outputs to inputs is independent of any particular system: it is a characteristic of the environment itself. Hence, if an attacker knows how the environment reacts to outputs (the probabilistic behavior of the reactions of the environment given the system outputs), this knowledge is part of the *a priori* knowledge of the adversary. As a further justification, observe that this is a natural extension of the classical approach, where the choice of secrets is seen as external to the system, i.e. determined by the environment. The probability distribution on the secrets constitutes the

a priori knowledge and does not count as leakage. In order to encompass the classical approach, in our extended model we should preserve this principle, and a natural way to do so is to consider the secret choices, at every stage of the computation, as external. Their probability distributions, which are now in general conditional probability distributions depending on the history of secrets and observables, should therefore be considered as part of the external knowledge, and not counted as leakage.

The following example supports our claim that, in the presence of feedback, mutual information is not a correct notion of leakage.

Example 5. *Consider the discrete memoryless channel with secret alphabet* $\mathcal{A} = \{a_1, a_2\}$ *and observable alphabet* $\mathcal{B} = \{b_1, b_2\}$ *whose matrix is represented in Table 4.6.*

	b_1	b_2
a_1	0.5	0.5
a_2	0.5	0.5

Table 4.6: Channel matrix for Example 5

Suppose that the channel is used with feedback, in such a way that, for all $1 \leq t \leq T$*, we have* $\alpha_{t+1} = a_1$ *if* $\beta_t = b_1$*, and* $\alpha_{t+1} = a_2$ *if* $\beta_t = b_2$*. It is easy to show that if* $T \geq 2$ *then* $I(A^T; B^T) \neq 0$*. Yet there is no leakage from* A^T *to* B^T*, since the rows of the matrix are all equal. We have indeed that* $I(A^T \to B^T) = 0$*, and the mutual information* $I(A^T; B^T)$ *is only due to the feedback information flow* $I(B^T \to A^T)$*.*

Having in mind the above discussion, we now propose a notion of information flow based on our model. We follow the idea of defining leakage and maximum leakage using the concepts of mutual information and capacity, making the necessary adaptations.

As discussed in Chapter 3, in the non-interactive case the definition of leakage as mutual information, for a single use of the channel, is

$$I(A; B) = H(A) - H(A|B)$$

(cfr. for instance [CPP08a, KB07]). This amounts to viewing the leakage as the difference between the a priori invulnerability and the a posteriori one. As explained in Chapter 3, these correspond to $H(A)$ and $H(A|B)$, respectively. This corresponds to the model of an attacker based on Shannon entropy discussed by Köpf and Basin in [KB07].

In the interactive case, we can extend this notion by considering the leakage at every step t as given by

$$I(A^t; B_t|B^{t-1}) = H(A^t|B^{t-1}) - H(A^t|B_t, B^{t-1})$$

The notion of attack is the same modulo the fact that we consider all the input from the beginning up to step t, and the difference in its vulnerability induced by the observation of B_t (the output at step t), taking into account the observation history B^{t-1}. It is then natural to consider as total leakage the summation of the contributions $I(A^t; B_t|B^{t-1})$ for all the steps t. This is exactly the notion of directed information (cfr. Definition 4):

$$I(A^T \to B^T) = \sum_{t=1}^{T} I(A^t; B_t|B^{t-1})$$

Definition 15. *The information leakage of a fully probabilistic IIHS is defined as the directed information $I(A^T \to B^T)$ of the associated channel with memory and feedback.*

We now show an equivalent formulation of directed information that leads to a new interpretation in terms of an attack model. First we need the following lemma.

Lemma 16. $I(B^T \to A^T) = H(A^T) - \sum_{t=1}^{T} H(A_t|A^{t-1}, B^{t-1})$

Proof.

$$\begin{aligned}
I(B^T \to A^T) &= \sum_{t=1}^{T} I(A_t; B^{t-1}|A^{t-1}) && \text{(by Definition 4)}\\
&= \sum_{t=1}^{T} \big(H(A_t|A^{t-1}) \\
&\quad -H(A_t|A^{t-1}, B^{t-1})\big) && \text{(by def. of mutual info.)}\\
&= H(A^T) - \sum_{t=1}^{T} H(A_t|A^{t-1}, B^{t-1}) && \text{(by the chain rule)}
\end{aligned}$$

□

The next proposition points out the announced alternative formulation of directed information from input to output:

Proposition 17. $I(A^T \to B^T) = \sum_{t=1}^{T} H(A_t|A^{t-1}, B^{t-1}) - H(A^T|B^T)$

Proof.

$$
\begin{aligned}
I(A^T \to B^T) &= I(A^T; B^T) - I(B^T \to A^T) && \text{(by (4.10))}\\
&= I(A^T; B^T) - H(A^T)\\
&+ \sum_{t=1}^{T} H(A_t | A^{t-1}, B^{t-1}) && \text{(by Lemma 16)}\\
&= H(A^T) - H(A^T | B^T) - H(A^T)\\
&+ \sum_{t=1}^{T} H(A_t | A^{t-1}, B^{t-1}) && \text{(by def. of mutual info.)}\\
&= \sum_{t=1}^{T} H(A_t | A^{t-1}, B^{t-1}) - H(A^T | B^T)
\end{aligned}
$$

□

We note that the term $\sum_{t=1}^{T} H(A_t | A^{t-1}, B^{t-1})$ can be seen as the entropy H_R of the reactor R, i.e. the entropy of the inputs, taking into account their dependency on the previous outputs. This brings us to an intriguing alternative interpretation of leakage.

Remark 18. *The leakage can be seen as the difference between the a priori invulnerability degree of the whole secret A^T, assuming that the attacker knows the distribution of the reactor, and the a posteriori invulnerability degree, after the adversary has observed the whole output B^T.*

In Section 4.5 we give an extensive and detailed example of how to calculate the leakage for an actual security protocol.

In the case of secret-nondeterministic IIHS, we have a stochastic kernel but no distribution on the reaction functions. In this case it seems natural to consider the worst leakage over all possible distributions on reaction functions. This is exactly the concept of capacity.

Definition 19. *The* maximum leakage *of a secret-nondeterministic IIHS is defined as the capacity C_T of the associated channel with memory and feedback (cfr. (4.7)).*

A comparison with the definition of Gray (cfr. [Gra91], Definition 5.3) is in order. As explained in the introduction, Gray's model is more complicated than ours, because it assumes that low and high variables are present at both ends of the channel. If we restrict the definition of Gray's capacity C^G to our case, by eliminating the low input and the high output, we obtain the following formula:

$$
C_T^G = \sup_{\mathcal{D}_T} \frac{1}{T} \sum_{t=1}^{T} I(A^{t-1}; B_t | B^{t-1}) \tag{4.11}
$$

By comparing (4.7), which is based on Definition 4, to (4.11), we can see that the only difference is that (4.11) considers the correlation between B_t and A^{t-1} instead of A^t. This seems to be intentional (cfr. [Gra91], discussion after Definition 4.1). We are not sure why C^G is defined in this way, our best guess is that the high values must be those of the previous time step in order to encompass the theory of McLean [McL90]. In any case, Gray's conjecture that C_T^G corresponds to the channel transmission rate does not hold. For instance, it is easy to see that for $T = 1$ we always have $C_T^G = 0$, but there obviously are channels which can transmit a non-zero amount of information even with one single use.

We conclude this section by showing that our approach to the notion of leakage generalizes the classical approach (based on mutual information) to the case of feedback. The idea is that, if a channel does not have feedback, then $I(B^T \to A^T) = 0$ and therefore $I(A^T; B^T) = I(A^T \to B^T)$. In our opinion, the fact that mutual information turns out to be a particular case of directed information helps to justify the former as a good measure of information flow, despite its symmetry: in channels without feedback it is a good measure *because it coincides with directed information* from input to output.

Lemma 20. *In absence of feedback,* $I(B^T \to A^T) = 0$

Proof. When feedback is not allowed, B^{t-1} and A_t are independent for every $1 \leq t \leq T$. Then:

$$\begin{aligned}
I(B^T \to A^T) &= \sum_{t=1}^{T} I(A_t; B^{t-1}|A^{t-1}) && \text{(by Definition 4)} \\
&= \sum_{t=1}^{T} (H(A_t|A^{t-1}) \\
&\quad - H(A_t|A^{t-1}, B^{t-1})) && \text{(by def. of mutual info.)} \\
&= \sum_{t=1}^{T} (H(A_t|A^{t-1}) \\
&\quad - H(A_t|A^{t-1})) && (B^{t-1} \text{ and } A^t \text{ are independent}) \\
&= 0
\end{aligned}$$

□

Proposition 21. *In absence of feedback, leakage can be equivalently defined as directed information or as mutual information. Similarly, in absence of feedback, the maximum leakage can be equivalently defined as directed capacity or as capacity.*

Proof. It follows directly from Lemma 20 and (4.10). □

4.5 An example: the Cocaine Auction protocol

In this section we show the application of our approach to the *Cocaine Auction Protocol* [SA99]. The formalization of this protocol in terms of IIHSs using our framework makes it possible to prove the claim in [SA99] suggesting that if the seller knows the identity of the bidders then the (strong) anonymity guaranties are no longer assured.

Let us consider a scenario in which several mobsters are gathered around a table. An auction is about to be held in which one of them offers his next shipment of cocaine to the highest bidder. The seller describes the merchandise and proposes a starting price. The others then bid increasing amounts until there are no bids for, say, 30 consecutive seconds. At that point the seller declares the auction closed and arranges a secret appointment with the winner to deliver the goods.

The basic protocol is fairly simple and is organized as a succession of rounds of bidding. Round i starts with the seller announcing the bid price b_i for that round. Buyers have t seconds to make an offer (i.e. to say yes, meaning "I'm willing to buy at the current bid price b_i"). As soon as one buyer anonymously says yes, he becomes the winner w_i of that round and a new round begins. If nobody says anything for t seconds, round i is concluded by timeout and the auction is won by the winner w_{i-1} of the previous round, if one exists. If the timeout occurs during round 0, this means that nobody made any offers at the initial price b_0, so there is no sale.

Although our framework allows the formalization of this protocol for an arbitrary number of bidders and bidding rounds, for illustration purposes we will consider the case of two bidders (*Candlemaker* and *Scarface*) and two rounds of bids. Furthermore, we assume that the initial bid is always 100 euros, so the first bid does not need to be announced by the seller. In each turn the seller can choose how much he wants to increase the current bid value. This is done by adding an increment to the last bid. There are two options of increments, namely inc_1 (100 euros) and inc_2 (200 euros). In that way, b_{i+1} is either $b_i + inc_1$ or $b_i + inc_2$. We can describe this protocol as a *normalized* IIHS $\mathcal{I} = (M, \mathcal{A}, \mathcal{B})$, where $\mathcal{A} = \{\text{Candlemaker}, \text{Scarface}, a^*\}$ is the set of secret actions, $\mathcal{B} = \{inc_1, inc_2, b_*\}$ is the set of observable actions, and the probabilistic automaton M is represented in Figure 4.8. For clarity reasons, transitions with probability 0 are not represented in the automaton. Note that the special secret action a_* represents the situation where neither *Candlemaker* nor *Scarface* bid. The special observable action b_* represents the end of the auction and it can only occur if no one has bid in the round.

Table 4.7 shows all the stochastic kernels for this example.

The next step is to construct all possible reaction functions $\{\varphi_t(\beta^{t-1})\}_{t=1}^{T}$. As seen in Section 4.3.2, the reaction functions correspond to the encoder in the channel. They take the feedback story and decide how the world will react to this situation. Table 4.8 contains the reaction functions for each time $t \leq 2$.

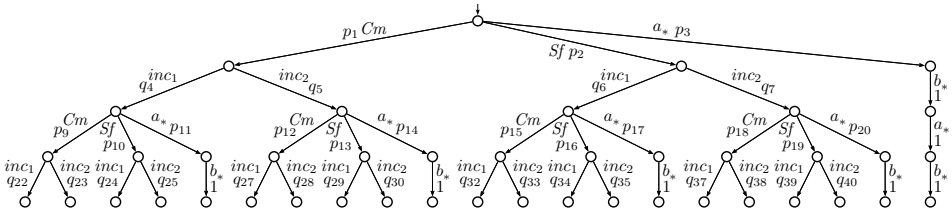


Figure 4.8: Cocaine auction example

$\alpha_1 \to \beta_1$	inc_1	inc_2	b_*
Candlemaker	q_4	q_5	0
Scarface	q_6	q_7	0
a^*	0	0	1

(a) $t=1, p(\beta_1|\alpha^1, \beta^0)$

$\alpha_1, \beta_1, \alpha_2 \to \beta_2$	inc_1	inc_2	b_*
Candlemaker,inc_1,*Candlemaker*	q_{22}	q_{23}	0
Candlemaker,inc_1,*Scarface*	q_{24}	q_{25}	0
Candlemaker,inc_1,a_*	0	0	1
Candlemaker,inc_2,*Candlemaker*	q_{27}	q_{28}	0
Candlemaker,inc_2,*Scarface*	q_{29}	q_{30}	0
Candlemaker,inc_2,a_*	0	0	1
Scarface,inc_1,*Candlemaker*	q_{32}	q_{33}	0
Scarface,inc_1,*Scarface*	q_{34}	q_{35}	0
Scarface,inc_1,a_*	0	0	1
Scarface,inc_2,*Candlemaker*	q_{37}	q_{38}	0
Scarface,inc_2,*Scarface*	q_{39}	q_{40}	0
Scarface,inc_2,a_*	0	0	1
a_*,b_*,a_*	0	0	1
All other lines	0	0	1

(b) $t = 2, p(\beta_2|\alpha^2, \beta^1)$

Table 4.7: Stochastic kernels for the Cocaine Auction example

Now we need to define the reactor, i.e. the probability distribution on reaction functions. Corollary 14 shows that we can do so by using the following equations:

$$p(\varphi_1) = p(\alpha_1|\alpha^0, \beta^0) = p(\alpha_1)$$
$$p(\varphi_t|\varphi^{t-1}) = \prod_{\beta^{t-1}} p(\varphi_t(\beta^{t-1})|\varphi^{t-1}(\beta^{t-2}), \beta^{t-1}), \quad 2 \leq t \leq T$$

For instance, $p(f_{1(1)}) = p(\mathit{Candlemaker}) = p_1$. In the same way, $p(f_{1(2)}) = p(\mathit{Scarface}) = p_2$ and $p(f_{1(3)}) = p(a_*) = p_3$.

β^0	$f_{1(1)}$	$f_{1(2)}$	$f_{1(3)}$
$\emptyset$	*Candlemaker*	*Scarface*	a_*

(a) All 3 reaction functions φ_1

β^1	$f_{2(1)}(\beta^1)$	$f_{2(2)}(\beta^1)$	$f_{2(3)}(\beta^1)$	$f_{2(4)}(\beta^1)$
inc_1	*Candlemaker*	*Candlemaker*	*Candlemaker*	*Candlemaker*
inc_2	*Candlemaker*	*Candlemaker*	*Candlemaker*	*Scarface*
b_*	*Candlemaker*	*Scarface*	a_*	*Candlemaker*
β^1	$f_{2(5)}(\beta^1)$	$f_{2(6)}(\beta^1)$	$f_{2(7)}(\beta^1)$	$f_{2(8)}(\beta^1)$
inc_1	*Candlemaker*	*Candlemaker*	*Candlemaker*	*Candlemaker*
inc_2	*Scarface*	*Scarface*	a_*	a_*
b_*	*Scarface*	a_*	*Candlemaker*	*Scarface*
β^1	$f_{2(9)}(\beta^1)$	$f_{2(10)}(\beta^1)$	$f_{2(11)}(\beta^1)$	$f_{2(12)}(\beta^1)$
inc_1	*Candlemaker*	*Scarface*	*Scarface*	*Scarface*
inc_2	a_*	*Candlemaker*	*Candlemaker*	*Candlemaker*
b_*	a_*	*Candlemaker*	*Scarface*	a_*
β^1	$f_{2(13)}(\beta^1)$	$f_{2(14)}(\beta^1)$	$f_{2(15)}(\beta^1)$	$f_{2(16)}(\beta^1)$
inc_1	*Scarface*	*Scarface*	*Scarface*	*Scarface*
inc_2	*Scarface*	*Scarface*	*Scarface*	a_*
b_*	*Candlemaker*	*Scarface*	a_*	*Candlemaker*
β^1	$f_{2(17)}(\beta^1)$	$f_{2(18)}(\beta^1)$	$f_{2(19)}(\beta^1)$	$f_{2(20)}(\beta^1)$
inc_1	*Scarface*	*Scarface*	a_*	a_*
inc_2	a_*	a_*	*Candlemaker*	*Candlemaker*
b_*	*Scarface*	a_*	*Candlemaker*	*Scarface*
β^1	$f_{2(21)}(\beta^1)$	$f_{2(22)}(\beta^1)$	$f_{2(23)}(\beta^1)$	$f_{2(24)}(\beta^1)$
inc_1	a_*	a_*	a_*	a_*
inc_2	*Candlemaker*	*Scarface*	*Scarface*	*Scarface*
b_*	a_*	*Candlemaker*	*Scarface*	a_*
β^1	$f_{2(25)}(\beta^1)$	$f_{2(26)}(\beta^1)$	$f_{2(27)}(\beta^1)$	—
inc_1	a_*	a_*	a_*	—
inc_2	a_*	a_*	a_*	—
b_*	*Candlemaker*	*Scarface*	a_*	—

(b) All 27 reaction functions $\varphi_2(\beta^1)$

Table 4.8: Reaction functions for the cocaine auction example

Let us take as an example the calculation of $p(f_{2(6)}|f_{1(1)})$:

$$\begin{aligned}
p(f_{2(6)}|f_{1(1)}) &= \prod_{\beta^1} p(f_{2(6)}(\beta^1)|\varphi_{1(1)}, \beta^1) \\
&= p(f_{2(6)}(inc_1)|\mathit{Candlemaker}, inc_1)\cdot \\
&\quad p(f_{2(6)}(inc_2)|\mathit{Candlemaker}, inc_2)\cdot \\
&\quad p(f_{2(6)}(b_*)|\mathit{Candlemaker}, b_*) \\
&= p(\mathit{Candlemaker}|\mathit{Candlemaker}, inc_1)\cdot \\
&\quad p(\mathit{Scarface}|\mathit{Candlemaker}, inc_2) \\
&\quad p(a_*|\mathit{Candlemaker}, b_*) \\
&= p_9 \cdot p_{13} \cdot 1 \\
&= p_9 p_{13}
\end{aligned}$$

Note that some reaction functions can have probability 0, which is consistent with the probabilistic automaton. For instance:

$$\begin{aligned}
p(f_{2(25)}|f_{1(3)}) &= \prod_{\beta^1} p(f_{2(25)}(\beta^1)|\varphi_{1(3)}, \beta^1) \\
&= p(f_{2(25)}(inc_1)|a_*, inc_1) \cdot p(f_{2(25)}(inc_2)|a_*, inc_2)\cdot \\
&\quad p(f_{2(25)}(b_*)|a_*, b_*) \\
&= p(a_*|a_*, inc_1) \cdot p(a_*|a_*, inc_2) \cdot p(\mathit{Candlemaker}|a_*, b_*) \\
&= 1 \cdot 1 \cdot 0 \\
&= 0
\end{aligned}$$

4.5.1 Calculating the information leakage

Let us now calculate the information leakage for this example using the concepts from Section 4.4. We will analyze three different scenarios:

Example a: There is feedback, but the probability of an observable does not depend on the history of secrets. In the auction protocol, this corresponds to a scenario where the probability of one of the mobsters to bid can depend on the increment imposed by the seller, but the history of who has previously bid in the past has no influence on how the seller chooses the bid increment in the coming turns. In other words, the seller cannot use the information of who has been bidding to change his strategy of defining the new increments. This situation corresponds to the original description of the protocol in [SA99], where the seller does

not have access to the identity of the bidder, for the sake of anonymity preservation. In general, we have $p(\beta_t|\alpha^t, \beta^{t-1}) = p(\beta_t|\beta^{t-1})$ for every $1 \leq t \leq T$. There is an exception, however: if there is no bidder, the case modeled by the secret being a_*, then the auction terminates, which is signaled by the observable b_*.

Example `b`**:** This is the most general case, without any restrictions. The presence of feedback allows the probability of the bidder to depend of the increment in the price. For instance, if *Candlemaker* is richer than *Scarface*, it is more likely that the former bids if the increment in the price is inc_2 instead of inc_1. Also, the probability of an observable can depend on the history of secrets, i.e. in general $p(\beta_t|\alpha^t, \beta^{t-1}) \neq p(\beta_t|\beta^{t-1})$ for $1 \leq t \leq T$. This scenario can represent a situation where the seller is corrupted and can use his information to affect the outcome of the auction. As an example, suppose that the seller is a friend of *Scarface* and he wants to help him in the auction. One way of doing so is to check who was the winner of the last bidding round. Whenever the winner is *Candlemaker*, the seller chooses as increment the small value inc_1, hoping that it will give *Scarface* a good chance to bid in the next round. On the other hand, whenever the seller detects that the winner is *Scarface*, he chooses as the next increment the greater value inc_2, hoping that it will minimize the chances of *Candlemaker* to bid in the next round (and therefore maximizing the chances of the auction to end up having *Scarface* as the final winner).

Example `c`**:** There is no feedback. In the cocaine auction, we can have the (perhaps unrealistic) situation in which the increment added to the bid has no influence on the probability of *Candlemaker* or *Scarface* being the bidder. Mathematically, we have $p(\alpha_t|\alpha^{t-1}, \beta^{t-1}) = p(\alpha_t|\alpha^{t-1})$ for every $1 \leq t \leq T$. As in Example `b`, however, we do not impose any restriction on $p(\beta_t|\alpha^t, \beta^{t-1})$.

For each scenario we need to fill in the values of the probabilities in the protocol tree in Figure 4.8. The probabilities for each example are listed in Table 4.9. Table 4.10 shows a comparison between some relevant values for the three cases.

In Example `a`, since the probability of observables does not depend on the history of secrets, there is (almost) no information flowing from the input to the output, and the directed information $I(A^T \to B^T)$ is close to zero, i.e. the leakage is low. The only reason why the leakage is not zero is because the end of an auction needs to be signaled. Due to presence of feedback, however, the directed information in the other sense $I(B^T \to A^T)$ is non-zero, and so is the mutual information $I(A^T; B^T)$. This is an example where the mutual information does not correspond to the real information leakage, since some (in

Probability variable	Example a value	Example b value	Example c value
p_1	0.75	0.70	0.70
p_2	0.24	0.24	0.24
p_3	0.01	0.01	0.01
q_4	0.50	0.55	0.30
q_5	0.50	0.45	0.70
q_6	0.50	0.45	0.70
q_7	0.50	0.55	0.30
p_9	0.04	0.80	0.75
p_{10}	0.95	0.19	0.20
p_{11}	0.01	0.01	0.05
p_{12}	0.95	0.19	0.75
p_{13}	0.04	0.80	0.20
p_{14}	0.01	0.01	0.05
p_{15}	0.04	0.90	0.65
p_{16}	0.95	0.09	0.35
p_{17}	0.01	0.01	0.05
p_{18}	0.95	0.09	0.65
p_{19}	0.04	0.90	0.35
p_{20}	0.01	0.01	0.05
q_{22}	0.50	0.80	0.45
q_{23}	0.50	0.20	0.55
q_{24}	0.50	0.20	0.55
q_{25}	0.50	0.80	0.45
q_{27}	0.45	0.75	0.45
q_{28}	0.55	0.25	0.55
q_{29}	0.45	0.35	0.55
q_{30}	0.55	0.65	0.45
q_{32}	0.50	0.55	0.45
q_{33}	0.50	0.45	0.55
q_{34}	0.50	0.40	0.55
q_{35}	0.50	0.60	0.45
q_{37}	0.45	0.60	0.45
q_{38}	0.55	0.40	0.55
q_{39}	0.45	0.35	0.55
q_{40}	0.55	0.55	0.45

Table 4.9: Values of the probabilities in Figure 4.8 for Examples `a`, `b`, and `c`

this case, most) of the correlation between input and output can be attributed to the feedback.

In Example `b` the information flow from input to output $I(A^T \to B^T)$ is significantly higher than zero, but still, due to feedback, the information flow from outputs to inputs $I(B^T \to A^T)$ is not zero and the mutual information $I(A^T; B^T)$ is higher than the directed information $I(A^T \to B^T)$.

Interpretation	Symbol	Example a	Example b	Example c
Input uncertainty	$H(A^T)$	1.9319	1.9054	1.9158
Reactor uncertainty	H_R	1.1911	1.5804	1.9158
A posteriori uncertainty	$H(A^T\|B^T)$	1.0303	1.2371	1.4183
Mutual information	$I(A^T;B^T)$	0.9016	0.6684	0.4975
Leakage	$I(A^T \to B^T)$	0.1608	0.3433	0.4975
Feedback information	$I(B^T \to A^T)$	0.7408	0.3250	0.0000

Table 4.10: Values of the entropy and directed information for Examples `a`, `b`, and `c`, where $I(A^T;B^T) = H(A^T) - H(A^T|B^T)$ and $I(A^T \to B^T) = H_R - H(A^T|B^T)$

In Example `c`, the absence of feedback implies that $I(B^T \to A^T)$ is zero. In that case the values of $I(A^T;B^T)$ and $I(A^T \to B^T)$ coincide, and represent the real leakage.

Finally, Figure 4.9 shows a comparison between the values of the entropy and of the directed information in the examples. The totality of the mutual information $I(A^T;B^T)$ is represented by the height of the correspondent bar, and we emphasize the contribution of the directed information in each direction by splitting the bar into two parts. This figure highlights the fact that mutual information can be misleading as a measure of leakage. The greatest mutual information is obtained in Example `a`, followed by Example `b` and then by Example `c`. The *real leakage*, however, given by $I(A^T \to B^T)$, respects exactly the inverse order, namely Example `a` presents the lowest value while Example `c` presents the highest one. Indeed, in Example `a` the value of $I(A^T \to B^T)$ represents only 18% of the mutual information, while in Example `b` it represents 51% and in Example `c` it amounts to 100%.

4.6 Topological properties of IIHSs and their capacity

In this section we show how to extend to IIHSs the notion of pseudometric defined in [DJGP02] for Concurrent Labeled Markov Chains, and we prove that the capacity of the corresponding channels is a continuous function with respect to this pseudometric. The pseudometric construction is sound for general IIHSs, but the result on capacity is only valid for secret-nondeterministic IIHSs.

Given a set of states S, a pseudometric is a function d that yields a non-negative real number for each pair of states and satisfies the following:

(i) $d(s,s) = 0$;

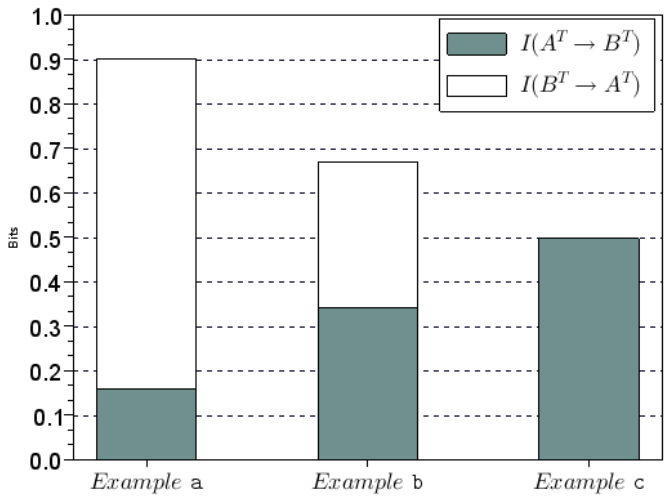


Figure 4.9: Comparison between the leakage in Examples a, b, and c

(i) $d(s,t) = d(t,s)$; and

(i) $d(s,t) \leq d(s,u) + d(u,t)$.

We say that a pseudometric d is c-bounded if $\forall s,t : d(s,t) \leq c$, where c is a positive real number.

Note that, in contrast to metrics, in pseudometrics two elements can have distance 0 without being identical. We consider pseudometrics instead of metrics because our purpose is to extend the notion of (probabilistic) bisimulation: having distance 0 will correspond to being bisimilar.

We now define a complete lattice structure on pseudometrics, in order to define the distance between IIHSs as the greatest fixpoint of a particular transformation, in line with the coinductive theory of bisimilarity. Since larger bisimulations identify more, the natural extension of the ordering to pseudometrics must shorten the distances as we go up in the lattice:

Definition 22. *$\mathcal{M}$ is the class of 1-bounded pseudometrics on states with the ordering*

$$d \preceq d' \text{ if } \forall s,s' \in S : d(s,s') \geq d'(s,s').$$

It is easy to see that $(\mathcal{M}, \preceq)$ is a complete lattice. In order to define pseudometrics on IIHSs, we now need to lift the pseudometrics on states to pseudometrics on distributions in $\mathcal{D}(\mathcal{L} \times S)$. Following standard lines [vBW01, DJGP02, DCPP06], we apply the construction based on the Kantorovich metric [Kan42].

Definition 23. *For $d \in \mathcal{M}$, and $\mu, \mu' \in \mathcal{D}(\mathcal{L} \times S)$, we define $d(\mu, \mu')$ (overloading the notation d) as*

$$d(\mu, \mu') \;=\; \max \sum_{(\ell_i, s_i) \in \mathcal{L} \times S} (\mu(\ell_i, s_i) - \mu'(\ell_i, s_i)) x_i$$

where the maximum is taken over all possible values of the x_i's, subject to the constraints $0 \leq x_i \leq 1$ and $x_i - x_j \leq \hat{d}((\ell_i, s_i), (\ell_j, s_j))$, where

$$\hat{d}((\ell_i, s_i), (\ell_j, s_j)) \;=\; \begin{cases} 1 & \text{if } \ell_i \neq \ell_j \\ d(s_i, s_j) & \text{otherwise} \end{cases}$$

It can be shown that with this definition m is a pseudometric on $\mathcal{D}(\mathcal{L} \times S)$.

Definition 24. *A pseudometric $d \in \mathcal{M}$ is a* bisimulation pseudometric [1] *if, for all $\epsilon \in [0, 1)$, $d(s, s') \leq \epsilon$ implies that if $s \to \mu$, then there exists some μ' such that $s' \to \mu'$ and $d(\mu, \mu') \leq \epsilon$.*

Note that it is not necessary to require the converse of the condition in Definition 24 to get a complete analogy with bisimulation: the converse is indeed implied by the symmetry of d as a pseudometric. Note also that we prohibit ϵ to be 1 because, throughout this chapter, 1 represents the maximum distance, which includes the case where one state may perform a transition and the other may not.

The greatest bisimulation pseudometric is

$$d_{max} = \bigsqcup \{d \in \mathcal{M} \mid d \text{ is a bisimulation pseudometric}\} \tag{4.12}$$

We now characterize d_{max} as a fixed point of a monotonic function Φ on $\mathcal{M}$. Eventually we are interested in the distance between IIHSs, and for the sake of simplicity, from now on we consider only the distance between states belonging to different IIHSs. The extension to the general case is trivial. For clarity purposes, we assume that different IIHSs have disjoint sets of states.

Definition 25. *Given two IIHSs with transition relations θ and θ' respectively, and a pseudometric d on states, define $\Phi : \mathcal{M} \to \mathcal{M}$ as:*

$$\Phi(d)(s, s') = \begin{cases} \max_i d(s_i, s'_i) & \text{if } \vartheta(s) = \{\delta_{(a_1, s_1)}, \ldots, \delta_{(a_m, s_m)}\} \\ & \text{and } \vartheta'(s') = \{\delta_{(a_1, s'_1)}, \ldots, \delta_{(a_m, s'_m)}\} \\ d(\mu, \mu') & \text{if } \vartheta(s) = \{\mu\} \text{ and } \vartheta'(s') = \{\mu'\} \\ 0 & \text{if } \vartheta(s) = \vartheta'(s') = \emptyset \\ 1 & \text{otherwise} \end{cases}$$

[1] In literature a pseudometric with this property is also known as bisimulation metric, although it is still a pseudometric.

It is easy to see that the definition of Φ is a particular case of the function F defined in [DJGP02, DCPP06], which is characterized as follows (cf. Lemma 3.8 in the full version of [DJGP02], and Definition 2.7 in [DCPP06]):

$$F(d)(s,s') = \max\{\sup_{s\to\mu}\inf_{s'\to\mu'} d(\mu,\mu')\ ,\ \sup_{s'\to\mu'}\inf_{s\to\mu} d(\mu,\mu')\}$$

Hence it can be proved, as an instance of the analogous result for F (cf. Lemma 2.8 in [DCPP06]), that $\Phi(d)$ is a pseudometric, and that the following property holds.

Lemma 26. *For $\epsilon \in [0,1)$, $\Phi(d)(s,s') \leq \epsilon$ holds if and only if whenever $s \to \mu$, there exists some μ' such that $s' \to \mu'$ and $d(\mu,\mu') \leq \epsilon$.*

From the above lemma and Definition 24 we derive (see also Lemma 2.9 in [DCPP06]):

Corollary 27. *A pseudometric d is a bisimulation pseudometric if and only if $d \preceq \Phi(d)$.*

By applying Corollary 27 to (4.12) we obtain

$$d_{max} = \bigsqcup\{d \in \mathcal{M} \mid d \preceq \Phi(d)\}$$

Furthermore, by adapting the proof of the monotonicity of F (cf. Lemma 3.9 in the full version of [DJGP02]) we can prove the following:

Lemma 28. *Φ is monotonic on $(\mathcal{M} \preceq)$.*

Thanks to Lemma 28, and using Tarski's fixed point theorem as formulated in [Tar55], we have that d_{max} is the greatest fixed point of Φ. Furthermore, by Corollary 27 we know that d_{max} is indeed a bisimulation pseudometric, and that it is the greatest bisimulation pseudometric.

In addition, the finite branching property of IIHSs ensures that the closure ordinal of Φ is ω (cf. Lemma 3.10 in the full version of [DJGP02]). Therefore we can proceed in a standard way to show that

$$d_{max} = \bigsqcap\{\Phi^i(\top) \mid i \in \mathbb{N}\},$$

where $\top$ is the greatest pseudometric (i.e. $\top(s,s') = 0$ for every s, s'), and $\Phi^0(\top) = \top$.

Given two IIHSs $\mathcal{I}$ and $\mathcal{I}'$, with initial states s and s' respectively, we define the distance between $\mathcal{I}$ and $\mathcal{I}'$ as $d(\mathcal{I},\mathcal{I}') = d_{max}(s,s')$. The following properties are auxiliary to the theorem which states the continuity of the capacity.

Lemma 29. *Consider two IIHSs $\mathcal{I}$ and $\mathcal{I}'$ with transition functions ϑ and ϑ' respectively. Given $t \geq 2$ and two sequences α^t and β^t, assume that both $\mathcal{I}(\alpha^{t-1},\beta^{t-1})$ and $\mathcal{I}'(\alpha^{t-1},\beta^{t-1})$ are defined. Assume also it is the case that $d_{max}(\mathcal{I}(\alpha^{t-1},\beta^{t-1}),\mathcal{I}'(\alpha^{t-1},\beta^{t-1})) < p(\beta_t \mid \alpha^t,\beta^{t-1})$, and $\vartheta(\mathcal{I}(\alpha^t,\beta^{t-1})) \neq \emptyset$. Then:*

1. *$\vartheta'(\mathcal{I}'(\alpha^t, \beta^{t-1})) \neq \emptyset$ holds as well,*
2. *$\mathcal{I}(\alpha^t, \beta^t)$ and $\mathcal{I}'(\alpha^t, \beta^t)$ are both defined, $p(\beta_t \mid \alpha^t, \beta^{t-1}) > 0$, and*

$$d_{max}(\mathcal{I}(\alpha^t, \beta^t), \mathcal{I}'(\alpha^t, \beta^t)) \leq \frac{d_{max}(\mathcal{I}(\alpha^{t-1}, \beta^{t-1}), \mathcal{I}'(\alpha^{t-1}, \beta^{t-1}))}{p(\beta_t \mid \alpha^t, \beta^{t-1}).}$$

Proof.

1. Assume $\vartheta(\mathcal{I}(\alpha^t, \beta^{t-1})) \neq \emptyset$ and, by contradiction, $\vartheta'(\mathcal{I}'(\alpha^t, \beta^{t-1})) = \emptyset$. Since d_{max} is a fixed point of Φ, we have $d_{max} = \Phi(d_{max})$, and therefore

$$\begin{aligned} d_{max}(\mathcal{I}(\alpha^t, \beta^{t-1}), \mathcal{I}'(\alpha^t, \beta^{t-1})) &= \Phi(d_{max})(\mathcal{I}(\alpha^t, \beta^{t-1}), \mathcal{I}'(\alpha^t, \beta^{t-1})) \\ &= 1 \\ &\geq p(\beta_t \mid \alpha^t, \beta^{t-1}), \end{aligned}$$

   which contradicts the hypothesis.

2. If $\vartheta(\mathcal{I}(\alpha^t, \beta^{t-1})) \neq \emptyset$, then, by the first point of this lemma, we have that $\vartheta'(\mathcal{I}'(\alpha^t, \beta^{t-1})) \neq \emptyset$ holds as well, and therefore both $\mathcal{I}(\alpha^t, \beta^t)$ and $\mathcal{I}'(\alpha^t, \beta^t)$ are defined. The hypothesis $d_{max}(\mathcal{I}(\alpha^{t-1}, \beta^{t-1}), \mathcal{I}'(\alpha^{t-1}, \beta^{t-1})) < p(\beta_t \mid \alpha^t, \beta^{t-1})$ ensures that $p(\beta_t \mid \alpha^t, \beta^{t-1}) \geq 0$.

   Let us now prove the bound on $d_{max}(\mathcal{I}(\alpha^t, \beta^t), \mathcal{I}'(\alpha^t, \beta^t))$. By definition of Φ, we have

$$\Phi(d_{max})(\mathcal{I}(\alpha^{t-1}, \beta^{t-1}), \mathcal{I}'(\alpha^{t-1}, \beta^{t-1})) \geq d_{max}(\mathcal{I}(\alpha^t, \beta^{t-1}), \mathcal{I}'(\alpha^t, \beta^{t-1})).$$

   Since $d_{max} = \Phi(d_{max})$, we have

$$d_{max}(\mathcal{I}(\alpha^{t-1}, \beta^{t-1}), \mathcal{I}'(\alpha^{t-1}, \beta^{t-1})) \geq d_{max}(\mathcal{I}(\alpha^t, \beta^{t-1}), \mathcal{I}'(\alpha^t, \beta^{t-1})). \tag{4.13}$$

   By definition of Φ and of the Kantorovich metric, we have

$$\begin{aligned} \Phi(d_{max})(\mathcal{I}(\alpha^t, \beta^{t-1}), \mathcal{I}'(\alpha^t, \beta^{t-1})) \geq\ & p(\beta_t \mid \alpha^t, \beta^{t-1}) \cdot \\ & d_{max}(\mathcal{I}(\alpha^t, \beta^t), \mathcal{I}'(\alpha^t, \beta^t)). \end{aligned}$$

   Using again $d_{max} = \Phi(d_{max})$, we get

$$\begin{aligned} d_{max}(\mathcal{I}(\alpha^t, \beta^{t-1}), \mathcal{I}'(\alpha^t, \beta^{t-1})) \geq\ & p(\beta_t \mid \alpha^t, \beta^{t-1}) \cdot \\ & d_{max}(\mathcal{I}(\alpha^t, \beta^t), \mathcal{I}'(\alpha^t, \beta^t)), \end{aligned}$$

   which, together with (4.13), allows us to conclude.

□

Lemma 30. *Consider two IIHSs $\mathcal{I}$ and $\mathcal{I}'$, and let $p(\cdot \mid \cdot,\cdot)$ and $p'(\cdot \mid \cdot,\cdot)$ be their distributions on the output nodes. Given $T > 0$, and two sequences α^T and β^T, assume that $p(\beta_t \mid \alpha^t, \beta^{t-1}) > 0$ for every $t < T$. Let $m = \min_{1\leq t<T} p(\beta_t \mid \alpha^t, \beta^{t-1})$ and let $\epsilon \in (0, m^{T-1})$. Assume $d(\mathcal{I}, \mathcal{I}') < \epsilon$. Then, for every $t \leq T$, we have*

$$p(\beta_t \mid \alpha^t, \beta^{t-1}) - p'(\beta_t \mid \alpha^t, \beta^{t-1}) < \frac{\epsilon}{m^{T-1}}.$$

Proof. Observe that, for every $t < T$, $\mathcal{I}(\alpha^t, \beta^t)$ must be defined, and, by repeatedly applying Lemma 29(1), we get that also $\mathcal{I}'(\alpha^t, \beta^t)$ is defined. By definition of Φ, and of the Kantorovich metric, we have

$$p(\beta_t \mid \alpha^t, \beta^{t-1}) - p'(\beta_t \mid \alpha^t, \beta^{t-1}) \leq \Phi(d_{max})(\mathcal{I}(\alpha^{t-1}, \beta^{t-1}), \mathcal{I}'(\alpha^{t-1}, \beta^{t-1})),$$

and since d_{max} is a fixed point of Φ, we get

$$p(\beta_t \mid \alpha^t, \beta^{t-1}) - p'(\beta_t \mid \alpha^t, \beta^{t-1}) \leq d_{max}(\mathcal{I}(\alpha^{t-1}, \beta^{t-1}), \mathcal{I}'(\alpha^{t-1}, \beta^{t-1})). \quad (4.14)$$

By applying Lemma 29(2) $t-1$ times, from (4.14) we get

$$\begin{aligned} p(\beta_t \mid \alpha^t, \beta^{t-1}) - p'(\beta_t \mid \alpha^t, \beta^{t-1}) &\leq \frac{d_{max}(\mathcal{I}(\alpha^0,\beta^0),\mathcal{I}'(\alpha^0,\beta^0))}{m^{t-1}} \\ &= \frac{d(\mathcal{I},\mathcal{I}')}{m^{t-1}} \\ &\leq \frac{d(\mathcal{I},\mathcal{I}')}{m^{T-1}} \\ &< \frac{\epsilon}{m^{T-1}} \end{aligned}$$

□

Note that previous lemma states a sort of continuity property of the matrices obtained from IIHSs, but not uniform continuity, because of the dependence on one of the two IIHSs. It is easy to see (from the proof of the Lemma) that uniform continuity does not hold.

The main contribution of this section, stated in the next theorem, is the continuity of the capacity with respect to the pseudometric on IIHSs. For this theorem, we assume that the IIHSs are normalized. Furthermore, it is crucial that they are secret-nondeterministic (while the definition of the pseudometric holds in general).

Theorem 31. *Consider two normalized IIHSs $\mathcal{I}$ and $\mathcal{I}'$, and fix a $T > 0$. For every $\epsilon > 0$ there exists $\nu > 0$ such that if $d(\mathcal{I}, \mathcal{I}') < \nu$ then $|C_T(\mathcal{I}) - C_T(\mathcal{I}')| < \epsilon$.*

Proof. Consider two normalized IIHSs $\mathcal{I}$ and $\mathcal{I}'$ and choose $T, \epsilon > 0$. Let $\mathcal{D}_T$ be the set of all input distributions in presence of feedback. Observe that

$$\begin{aligned} |C_T(\mathcal{I}) - C_T(\mathcal{I}')| &= |\max_{\mathcal{D}_T} \frac{1}{T} I(A^T \to B^T) - \max_{\mathcal{D}_T} \frac{1}{T} I(A'^T \to B'^T)| \\ &\leq \tfrac{1}{T} \max_{\mathcal{D}_T} |I(A^T \to B^T) - I(A'^T \to B'^T)| \end{aligned}$$

Since the directed information $I(A^T \to B^T)$ is defined by means of arithmetic operations and logarithms on the joint probabilities $p(\alpha^t, \beta^t)$ and on the conditional probabilities $p(\alpha^t, \beta^t)$, $p(\alpha^t, \beta^{t-1})$, which in turn can be obtained by means of arithmetic operations from the probabilities $p(\beta_t \mid \alpha^t, \beta^{t-1})$ and $p_F(\varphi^t)$, we have that $I(A^T \to B^T)$ is a continuous function of the distributions $p(\beta_t \mid \alpha^t, \beta^{t-1})$ and $p_F(\varphi^t)$, for every $t \leq T$. Let $p(\beta_t \mid \alpha^t, \beta^{t-1})$, $p'(\beta_t \mid \alpha^t, \beta^{t-1})$ be the distributions on the output nodes of $\mathcal{I}$ and $\mathcal{I}'$, modified in the following way: starting from level T, whenever $p(\beta_t \mid \alpha^t, \beta^{t-1}) = 0$, then we redefine the distributions at all the output nodes of the subtree rooted in $\mathcal{I}(\alpha^t, \beta^t)$ so that they coincide with the distribution of the corresponding nodes of in $\mathcal{I}'$, and analogously for $p'(\beta_t \mid \alpha^t, \beta^{t-1})$. Note that this transformation does not change the directed information, because the subtree rooted in $\mathcal{I}(\alpha^t, \beta^t)$ does not contribute to it, due to the fact that the probability of reaching any of its nodes is 0. The continuity of $I(A^T \to B^T)$ implies that there exists $\epsilon' > 0$ such that, if $|p(\beta_t \mid \alpha^t, \beta^{t-1}) - p'(\beta_t \mid \alpha^t, \beta^{t-1})| < \epsilon'$ for all $t \leq T$ and all sequences α^t, β^t, then, for any $p_F(\varphi^t)$, we have $|I(A^T \to B^T) - I(A'^T \to B'^T)| < \epsilon$. The result then follows from Lemma 30, by choosing

$$\nu = \epsilon' \cdot \min \left(\begin{array}{ll} \min\limits_{\substack{1 \leq t < T \\ p(\beta_t \mid \alpha^t, \beta^{t-1}) > 0}} p(\beta_t \mid \alpha^t, \beta^{t-1}), & \\ & \min\limits_{\substack{1 \leq t < T \\ p'(\beta_t \mid \alpha^t, \beta^{t-1}) > 0}} p'(\beta_t \mid \alpha^t, \beta^{t-1}) \end{array} \right).$$

□

We conclude this section with an example showing that the continuity result for the capacity does not hold if the construction of the channel is done starting from a system in which the secrets are endowed with a probability distribution. This is also the reason why we could not simply adopt the proof technique of the continuity result in [DJGP02] and we had to come up with different reasoning.

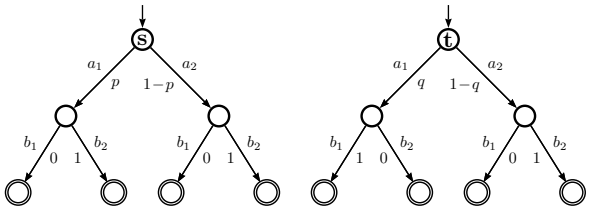
Example 6. *Consider the two following programs, where a_1, a_2 are secrets, b_1, b_2 are observable, $\|$ is the parallel operator, and $+_p$ is a binary probabilistic choice that assigns probability p to the left branch, and probability $1-p$ to the right one.*

s) $(send(a_1) +_p \ send(a_2)) \parallel receive(x).output(b_2)$

t) $(send(a_1) +_q send(a_2)) \parallel receive(x).if\, x = a_1 \ then \ output(b_1) \ else \ output(b_2)$.

Table 4.11 shows the fully probabilistic IIHSs corresponding to these programs, and their associated channels, which in this case (since the secret actions are all at the top-level) are classical channels, i.e. memoryless and without feedback. As usual for classical channels, they do not depend on p and q. It is easy to see that the capacity of the first channel is 0 and the capacity of the second one is 1. Hence their difference is 1, independently of p and q.

Let now $p = 0$ and $q = \epsilon$. It is easy to see that the distance between s and t is ϵ. Therefore (when the automata have probabilities on the secrets), the capacity is not a continuous function of the distance.



s	b_1	b_2
a_1	0	1
a_2	0	1

(a) (Channel for s

t	b_1	b_2
a_1	1	0
a_2	0	1

(b) Channel for t

Table 4.11: The IIHSs of Example 6 and their corresponding channels

4.7 Related work

Gray investigated a concept similar to directed information in [Gra91]. In contrast to our model, which is based on an eavesdropper scenario, he considered leakage in a sender-receiver model. More precisely, he considered a system based on Millen's synchronous state machine [Mil90], and connected to "low" and "high" environments via communication channels. His purpose was to measure the flow of information from the high environment to the low one, assuming that the only way for the low environment to learn about the high one (and vice versa) is through the system. To this end, he defined a notion of "quasi-directed information" by extending Gallager's formula for discrete finite state channels [Gal68]. He also conjectured a correspondence between the quasi-directed information and the transmission rate of the channel. His formulation of quasi-directed information, however, is not completely the same as directed information, and as a result the conjecture does not hold.

The continuity of the channel capacity was also proved in [DJGP02] for simple channels, but the proof does not adapt to the case of channels with memory and feedback and we had to devise a different technique.

4.8 Chapter summary and discussion

In this chapter we have investigated the problem of information leakage in interactive systems, and proved that these systems can be modeled as channels with memory and feedback. We have also proved that the channel capacity is a continuous function of a pseudometric based on the Kantorovich metric.

We have considered various kinds of automata corresponding to different combinations of nondeterministic and probabilistic choice, as summarized in Table 4.12(a). Note that in this the third row corresponds to the limit case in which the reactor is a Dirac measure, i.e. the probability is all concentrated on exactly one $\varphi^T \in \mathcal{F}$. It is easy to see that in this case $I(A^T \to B^T) = 0$ (all the entropies that constitute $I(A^T \to B^T)$ are 0), although $I(B^T \to A^T) \neq 0$. Therefore there is no leakage. In the classic case this corresponds to the situation in which the input distribution is a Dirac measure.

Table 4.12(b) summarizes the comparison between the channels with memory and feedback investigated in this chapter, and the classic channels.

Throughout this chapter we have assumed that the dependence of the secret choices on the observables is part of the external knowledge and, therefore, not considered leakage. The reader may wonder what would happen if this assumption were dropped. We argue that in this case $I(B^T \to A^T)$ *could be considered as part of the leakage.* In the cases `a` and `b` of the cocaine auction example in Section 4.5, for instance, one may want to consider the information that we can deduce about the secrets (the identities of the bidder) from the observables (the increments of the seller) as a leak due to the protocol.

In some other cases the flow of information from the observables to the secrets may even be considered as a consequence of the active attacks of an adversary, which uses the observables to modify the probability of the secrets. In this case $I(B^T \to A^T)$ could represent a measure of the effectiveness of the adversary.

As future work, we would like to provide algorithms to compute the leakage and maximum leakage of interactive systems. These are rather challenging problems given the exponential growth of reaction functions (needed to compute the leakage) and the quantification over infinitely many reactors (given by the definition of maximum leakage in terms of capacity). One possible solution is to study the relation between deterministic schedulers and sequence of reaction functions. In particular, we believe that for each sequence of reaction functions and distribution over it there exists a probabilistic scheduler for the automata representation of the secret-nondeterministic IIHS. In this way, the problem of computing the leakage and maximum leakage would reduce to a standard probabilistic model checking problem (where the challenge is to compute probabilities ranging over infinitely many schedulers).

In addition, we plan to investigate measures of leakage for interactive systems other than mutual information and capacity.

We intend to study the applicability of our framework to the area of

IIHSs as automata	**IIHSs as channels**	**Notion of leakage**
Normalized IIHSs with nondeterministic secrets and probabilistic observables	Sequence of stochastic kernels $\{p(\beta_t\|\alpha^t, \beta^{t-1})\}_{t=1}^T$	Leakage as capacity
Fully probabilistic normalized IIHSs	Sequence of stochastic kernels $\{p(\beta_t\|\alpha^t, \beta^{t-1})\}_{t=1}^T$ + reactor $\{p(\varphi_t\|\varphi^{t-1})\}_{t=1}^T$	Leakage as directed information $I(A^T \to B^T)$
Normalized IIHSs with a deterministic scheduler solving the nondeterminism	Sequence of stochastic kernels $\{p(\beta_t\|\alpha^t, \beta^{t-1})\}_{t=1}^T$ + reaction function sequence φ^T	No leakage

(a) The various models considered in this chapter

Classical channels	**Channels with memory and feedback**
The system is modeled in independent uses of the channel, often a unique use.	The system is modeled in several consecutive uses of the channel.
The channel is defined on $\mathcal{A}^T \to \mathcal{B}^T$, i.e. its input is a single string $\alpha^T = \alpha_1 \dots \alpha_T$ of secret symbols and its output is a single string $\beta^T = \beta_1 \dots \beta_T$ of observable symbols.	The channel is defined on $\mathcal{F} \to \mathcal{B}$, i.e. its input is a reaction function φ_t and its output is an observable β_t.
The channel is memoryless and in general it is implicitly assumed the absence of feedback.	The channel has memory. Despite the fact that the channel defined on $\mathcal{F} \to \mathcal{B}$ does not have feedback, the internal stochastic kernels do.
The capacity is calculated using mutual information $I(A^T; B^T)$.	The capacity is calculated using mutual directed information $I(A^T \to B^T)$.

(b) Classical channels vs. channels with memory and feedback

Table 4.12: Summary of results

game theory. In particular, the interactive nature of games such as *Prisoner Dilemma* [Pou92] and *Stag and Hunt* [Sky03] (in their iterative versions) can be modeled as channels with memory and feedback following the techniques proposed in this work. Furthermore, (probabilistic) strategies can be encoded as reaction functions. In this way, optimal strategies are attained by reaction functions maximizing the leakage of the channel.

Five

Differential privacy: the trade-off between leakage and utility

"If you have nothing to hide, then you don't have a life."
cited by Daniel J. Solove

In this chapter we consider the differential privacy approach to the problem of statistical disclosure control. In general a statistical database contains data of a group of individuals, and users can pose queries to obtain statistical information about the sample in the dataset. To preserve the privacy of the the participants in the database, it is desirable to restrict the amount of information that the system leaks about their individual values. One way of dealing with the problem is by using randomization mechanisms: to avoid leakage, the real answer is modified with some carefully added noise before being reported to the users. A very popular and studied way of doing so is based on the concept of differential privacy.

In our work we consider the relation between differential privacy and quantitative information flow. We address the problem of characterizing the protection that differential privacy provides to individuals with respect to information leakage, and the problem of the utility, i.e. the measure of how close the reported answer is to the true answer.

Contribution The main contributions of this chapter can be summarized as follows.

- We propose an information-theoretic framework to reason about both information leakage and utility.

- We explore the graph-theoretic foundations of the adjacency relation on databases[1], and we point out two types of symmetries which allow us to establish a strict link between differential privacy and information leakage.
- We prove that ϵ-differential privacy implies a tight bound on the min-entropy leakage.
- We prove that ϵ-differential privacy implies a bound on the utility, measured in terms of binary gain functions. We prove that, under certain conditions, the bound is tight.
- We identify a method that, under certain conditions, constructs randomization mechanisms that maximize utility while providing ϵ-differential privacy.

Plan of the Chapter This chapter is organized as follows. In Section 5.1 we formalize the notion of differential privacy and present an alternative interpretation for it in the special case where the adjacency relation on databases is complete (i.e. every two distinct databases are adjacent). In Section 5.2 we introduce our model to reason about leakage and utility for randomized functions in the case where the query and the randomization mechanism can be split into two distinct channels. In Section 5.3 we review some concepts from graph theory and present two special classes of graphs having symmetries that we will explore to make the connection between differential privacy and quantitative information flow. We also show that the graph structure on databases, induced by the adjacency relation and the query, presents these symmetries. In Section 5.4 we use the results of the previous section to prove a bound on the a posteriori min-entropy of the channel matrix. Then we apply this bound to derive our results for leakage in Section 5.5 and for utility in Section 5.6. Finally, in Section 5.7 we review some of the related work in the literature, and in Section 5.8 we make our final remarks and conclude this chapter.

5.1 Differential privacy

Databases are commonly used for obtaining statistical information about their participants. Simple examples of statistical queries are, for instance, the predominant disease in a certain population, or the average salary of a group of people. The fact that the answer is publicly available may, however, constitute a threat for the privacy of the individuals.

In order to illustrate the problem, consider a database that stores the values of the salaries of a set of individuals, and assume that a user can pose the query "what is the average salary of the participants in the database?". In principle

[1] The adjacency relation on databases will be defined precisely in Section 5.2.

we would like to consider the *global information* relative to the database as *public*, and the *individual information* about a participant as *private*. In this example, we would like to obtain the average salary without being able to infer the salary of any specific participant. Unfortunately this is not always possible. In particular, if the number of participants in the database is known, and an individual is removed from (or included in) the database, it is possible to infer his salary by querying again the database and calculating the influence of the removal (or inclusion) on the reported answer to the query.

Another kind of private information we may want to protect is whether a specific individual is *participating or not* in a database. If we know that a particular individual earns, say, 5.000€ a month, and all the other individuals earn less than 4.000€ a month, then learning that the average salary is greater that 4.000€ will reveal immediately the presence of our individual of interest in the database.

A common approach to this problem is to introduce some output perturbation mechanism based on randomization: instead of the exact answer, the querying mechanism reports a "noisy" answer. Namely, a randomized function is used to produce answers according to some probability distribution that depends on the database. The goal is to report this randomized answer, which ideally should be "close enough" to the real one, yet should make it harder for the user to guess the values of individual participants. For certain distributions, however, it may still be possible to guess the value of an individual with a high probability of success. The notion of *differential privacy*, due to Dwork [Dwo06, DL09, Dwo10, Dwo11], is a proposal to control the risk of violating privacy for both kinds of threats described above (value and participation). The idea is to say that a randomized function $\mathcal{K}$ satisfies ϵ-differential privacy (for some $\epsilon > 0$) if the ratio between the probabilities that two adjacent databases give a certain answer is bound by e^{ϵ}, where by "adjacent" we mean that the databases differ in only one individual (either for the value of an individual or for the presence/absence of an individual). The notion of differential privacy was developed to be independent of the *side (or auxiliary) information* the user can have about the database, and how it can affect his knowledge about the database before posing the query. This information can come from external sources (e.g. newspapers, common knowledge, etc), but does not affect the guarantees assured by differential privacy.

In this chapter we explore the similarities between differential privacy and quantitative information flow. We base our approach on the following observations: at the motivational level, the concern about privacy is akin the concern about information leakage. At the conceptual level, the randomized function $\mathcal{K}$ can be seen as an information-theoretic channel, and the limit case of $\epsilon = 0$, for which the privacy protection is total, corresponds to a 0-capacity channel, which does not allow any leakage. More specifically, we investigate the notion of differential privacy and its implications in the light of the min-entropy framework for information flow discussed in Chapter 3.

5.1.1 Formal definition

Let $\mathcal{X}$ be the set of all possible databases. Two databases $x, x' \in \mathcal{X}$ are *adjacent* (or *neighbors*), written $x \sim x'$, if they differ in the value of exactly one individual. Note that the structure $(\mathcal{X}, \sim)$ forms an undirected graph.

Intuitively, differential privacy is based on the idea that a randomized query function provides sufficient protection if the ratio between the probabilities of two adjacent databases to give a certain answer is bound by e^ϵ, for some $\epsilon > 0$. Formally:

Definition 32 ([Dwo11]). *A randomized function $\mathcal{K}$ from $\mathcal{X}$ to $\mathcal{Z}$ satisfies ϵ-differential privacy if for all pairs $x, x' \in \mathcal{X}$, with $x \sim x'$, and all $S \subseteq \mathcal{Z}$, we have:*

$$Pr[\mathcal{K}(x) \in S] \leq e^\epsilon \times Pr[\mathcal{K}(x') \in S]$$

In this thesis we consider $\mathcal{Z}$ to be finite, therefore each of its probability distributions is finite and we can rewrite the property of ϵ-differential privacy more simply. Using the notation of conditional probabilities, and considering both quotients, we can say that ϵ-differential-privacy holds in the discrete case if, for all $x, x' \in \mathcal{X}$ with $x \sim x'$, and all $z \in \mathcal{Z}$:

$$\frac{1}{e^\epsilon} \leq \frac{Pr[Z = z|X = x]}{Pr[Z = z|X = x']} \leq e^\epsilon \tag{5.1}$$

where X and Z represent the random variables associated to $\mathcal{X}$ and $\mathcal{Z}$, respectively.

Intuitively, (5.1) implies that, if a value of one single individual changes in a dataset (either by inclusion, removal or modification), the probability of the querying mechanism to report a specific answer will not "vary much". In other words, the influence of a single individual in a database is "negligible" with respect to the whole set of individuals. Of course the notion of what is meant by "much" and "negligible" depends on the value of ϵ.

5.2 A model of utility and privacy for statistical databases

In this section we present a model of statistical queries on databases, where noise is carefully added to protect the privacy of the participants in the sample, and the reported answer to a query does not need to be the real one. In this model, the notion of information leakage is to measure the amount of information that an adversary can learn about the database by posing queries and then analyzing the reported answers. Note that in principle the adversary can be a user of the database, and therefore the privacy guarantees should not depend on distinctions of who is posing the queries. Our model will also allow us to quantify the utility of the query, i.e. how much information about the

real answer can be obtained from the reported one. In our work we focus on the case in which all the values of interest are discrete.

We fix a finite set $Ind = \{0, 1, \ldots, u-1\}$ of u individuals participating in the database. In addition, we fix a finite set $Val = \{v_0, v_1, \ldots, v_{v-1}\}$, representing the set of (v different) possible values for the *sensitive attribute* of each individual (e.g. disease-name in a medical database). In the more general case where there are several sensitive attributes in the database (e.g. salary and security number in a census sample), we can think of the elements of Val as tuples. The absence of an individual in the database, if allowed, can be modeled with one special value in Val (see the discussion in Section 5.2.2). A database $D = d_0 \ldots d_{u-1}$ is a u-tuple where each $d_i \in Val$ is the value of the corresponding individual. The set of all databases is $\mathcal{X} = Val^u$. Two databases x, x' are *adjacent*, written $x \sim x'$, if and only if they differ in the value of exactly one individual. As we already pointed out, the structure $(\mathcal{X}, \sim)$ forms an undirected graph, and we call $\sim$ its *adjacency relation*.

Let $\mathcal{K}$ be a randomized function from $\mathcal{X}$ to $\mathcal{Z}$, where $\mathcal{Z} = Range(\mathcal{K})$ (see Figure 5.1). This function can be modeled by a channel $(\mathcal{X}, \mathcal{Z}, p_{Z|X}(\cdot|\cdot))$, where $\mathcal{X}$ and $\mathcal{Z}$ are the input and output alphabets, respectively, and $p_{Z|X}(\cdot|\cdot)$ is the channel matrix. The random variables modeling the input and output of the channel are denoted by X and Z, respectively. The definition of differential privacy can be directly expressed as a property of the channel: it satisfies ϵ-differential privacy if

$$p(z|x) \leq e^{\epsilon} p(z|x') \quad \text{for all } x, x' \in \mathcal{X} \text{ with } x \sim x', \text{ and all } z \in \mathcal{Z}$$

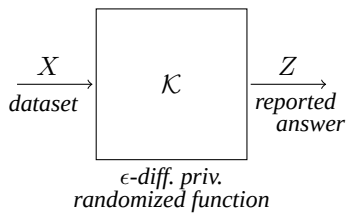


Figure 5.1: Randomized function $\mathcal{K}$

Intuitively, the correlation between X and Z measures how much information about the complete database the attacker can obtain by observing the reported answer. We will refer to this correlation as the *leakage* of the channel, denoted by $\mathcal{L}(X, Z)$. In Section 5.5 we will discuss how this leakage can be quantified using notions from information theory, and we will study the behavior of the leakage for differentially private queries.

In our model the true answer to the query f is modeled by the random variable Y ranging over $\mathcal{Y} = Range(f)$. The correlation between Y and Z measures how much we can learn about the real answer from the reported one. We will refer to this correlation as the *utility* of the channel, denoted by

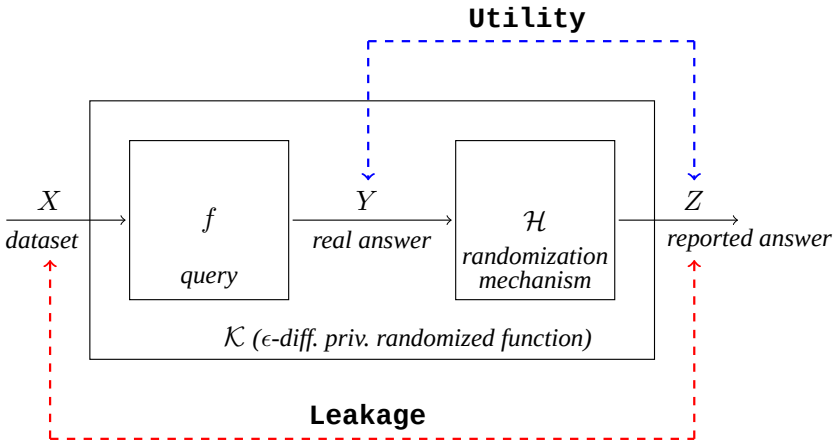


Figure 5.2: Leakage and utility for oblivious mechanisms

$\mathcal{U}(Y, Z)$. In Section 5.6 we will discuss in detail how the utility can be quantified, and we will investigate how to construct a randomization mechanism, i.e. a way of adding noise to the query outputs, so that utility is maximized while preserving differential privacy.

In practice, the randomization mechanism is often *oblivious*, meaning that the reported answer Z only depends on the real answer Y and not on the database X. In this case, the randomized function $\mathcal{K}$, seen as a channel, can be decomposed into two parts: a channel modeling the query f, and a channel modeling the oblivious randomization mechanism $\mathcal{H}$. These two channels are said to be *in cascade*, as the output of the first one is the input for the second one. The definition of utility can be then simplified as it only depends on properties of the sub-channel corresponding to $\mathcal{H}$. The leakage relating X and Y and the utility relating Y and Z for a decomposed randomized function are shown in Figure 5.2.

We capture the notion of the attacker's side information as the prior distribution on X, which is standard in information flow and also in papers on differential privacy [GRS09, KS].

5.2.1 Leakage about an individual

As already discussed, $\mathcal{L}(X, Z)$ can be used to quantify the information that the attacker can learn about the whole database. Protecting the entire database at once, however, is not the main goal of differential privacy. In fact, some information will necessarily be revealed, otherwise the query would not be useful. Instead, differential privacy aims at protecting the value of *any single individual*, even in the worst case where the values of all other individuals are known. To quantify this information leakage we can define smaller channels, where only the information of a specific individual varies. Let $x^- \in Val^{u-1}$ be

a $(u-1)$-tuple with the values of all individuals but one (the individual whose degree of protection we want to quantify). We create a channel $\mathcal{K}_{x^-}$ whose input alphabet is the set of all databases in which the $u-1$ other individuals have the same values as in x^-. Note that, since x^- is fixed, to define the input of the channel it is enough to specify the value of the individual of interest. In this way the input for the channel can be seen as a random variable V ranging over the set *Val*. Intuitively, the information leakage of this channel measures how much information about one particular individual the attacker can learn if the values of all others are known to be x^-. This leakage will be studied in Section 5.5.1.

5.2.2 A note on the choice of values

The choice of the set *Val* depends on the assumptions about the attacker's knowledge. In particular, if the attacker does not know which individuals participate in the database, a distinguished value in *Val* could be interpreted as absence (e.g. the value 0 or the special value *null*). As discussed in [Dwo11], a database x' adjacent to x can be though of either as being a superset (or subset) of x with one extra (or missing) row, or as being exactly the same database as x in all rows except for one which has a different (non-*null*) value. Our definition of $\sim$ with the possibility of *null* values covers all these cases.

At this point an important observation should be made about the choice of *Val*. Most often we are interested in protecting the *actual value* of an individual, not only his participation in the database. In this case, the definition of differential privacy (as well as the channels we are constructing) should include databases with all possible values for each individual, not just the "real" ones. In other words, to prevent the attacker from finding out the individual's value, the probability $p(z|x)$, where x contains the individual's true value, should be close to $p(z|x')$ where x' contains a hypothetical value for this individual. This might seem unnecessary at first sight, since differential privacy is often thought of as protecting the participation of an individual in a database. Hiding the participation of an individual, however, does not imply hiding his value. Consider the following example: we aim at learning the average salary of employees in a small company, and it happens that all of them have exactly the same salary s. We allow anyone to participate or not, while offering ϵ-differential privacy. If we only consider s as the value in all possible databases, then the query is always constant, so answering it any number of times without any noise should satisfy differential privacy for any $\epsilon \geq 0$. Since all reported answers are s, the attacker can deduce that the salary of all employees, including those not participating in the query, is s. Indeed, the attacker cannot find out who participated, despite the value of all individuals is revealed.

In other cases, we are only interested in hiding the identity of the participants (e.g. in a database with information about anonymous donations). Thus, *Val* should be properly selected according to the application. If who has

participated is known and we only wish to hide the values, then *Val* should contain all possible values, e.g. all possible salaries in the example above. If the values are known and participation is to be hidden, then *Val* can contain just the values 0 and 1 denoting absence and presence respectively. Finally, if both the value and the the identities of the participants are to be protected, then *Val* should contain all values plus *null*.

5.2.3 The questions we explore with the help of our model

We will use the model we just introduced to explore the following questions:

1. Does ϵ-differential privacy induce a bound on the information leakage of the randomized function $\mathcal{K}$?
2. Does ϵ-differential privacy induce a bound on the information leakage *relative to an individual*?
3. Does ϵ-differential privacy induce a bound on the utility?
4. Given a query f and a value $\epsilon > 0$, can we construct a randomized function $\mathcal{K}$ which satisfies ϵ-differential privacy and also presents maximum utility?

We will see that the answers to 1 and 2 are positive in case we take the measure of leakage to be the min-entropy leakage, and we provide bounds that are tight (i.e. for every ϵ there is a $\mathcal{K}$ whose leakage reaches the bound). For 3 we are able to give a tight bound in some cases which depend on the structure of the query, and for the same cases, we are able to construct an oblivious $\mathcal{K}$ with maximum utility (defined in terms of a binary gain function), as requested by 4.

5.3 Graph symmetries

In this section we explore some classes of graphs that will allow us to derive a strict correspondence between ϵ-differential privacy and the a posteriori entropy of the input. As we already mentioned, the input domain of databases and the adjacency relation forms an undirected graph, and this fact will be used to derive bounds on information leakage and utility. We will present two classes of graphs, distance-regular and VT^+, that will be used in the next section to transform a generic channel matrix into a matrix with a symmetric structure, while preserving the a posteriori min-entropy and the ϵ-differential privacy.

Let us first recall some basic notions. Given a graph $G = (\mathcal{V}, \sim)$, the *distance* $d(v, w)$ between two vertices $v, w \in \mathcal{V}$ is the number of edges in a shortest path connecting them. The *diameter* δ of G is the maximum distance

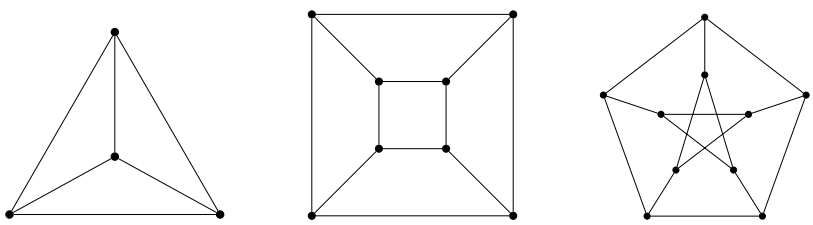
between any two vertices in $\mathcal{V}$. The *degree* of a vertex is the number of edges incident to it. G is called *regular* if every vertex has the same degree. A regular graph with vertices of degree k is called a *k-regular graph*. An *automorphism* of G is a permutation σ on the vertex set $\mathcal{V}$, such that for any pair of vertices v, w, if $v \sim w$, then $\sigma(\mathrm{v}) \sim \sigma(w)$. If σ is an automorphism, and v is a vertex, the *orbit* of v under σ is the set $\{v, \sigma(v), \ldots, \sigma^{k-1}(v)\}$ where k is the smallest positive integer such that $\sigma^k(v) = v$. Clearly, the orbits of the vertices under σ define a partition of $\mathcal{V}$. If $\mathcal{V}$ is the set of vertices of G, we denote by $\mathcal{V}_{\langle d \rangle}(\mathrm{v})$ the subset of vertices in $\mathcal{V}$ that are at distance d from the vertex v.

The following two definitions introduce the classes of graphs that we are interested in. The first class is well known in literature.

Definition 33 (Distance-regular graph)**.** *A graph $G = (\mathcal{V}, \sim)$ is called* distance-regular *if there exist integers b_d and c_d ($d \in \{0, \ldots, \delta\}$) (called* intersection numbers*) such that, for all vertices v, w at distance $d(\mathrm{v}, w) = d$, there are exactly*

- *b_d neighbors of w in $\mathcal{V}_{\langle d+1 \rangle}(\mathrm{v})$*
- *c_d neighbors of w in $\mathcal{V}_{\langle d-1 \rangle}(\mathrm{v})$*

Some examples of distance-regular graphs are illustrated in Figure 5.3.



(a) Tetrahedral graph (b) Cubical graph (c) Petersen graph

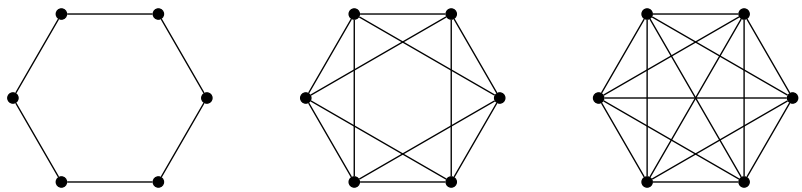
Figure 5.3: Some distance-regular graphs with degree 3

The second class we are interested in is a variant of the VT (vertex-transitive[2]) class:

Definition 34 (VT^+ graph)**.** *A graph $G = (\mathcal{V}, \sim)$ is VT^+ (*vertex-transitive +*) if there are n automorphisms σ_0, σ_1, $\ldots \sigma_{n-1}$, where $n = |\mathcal{V}|$, such that, for every vertex $\mathrm{v} \in \mathcal{V}$, we have that $\{\sigma_i(\mathrm{v}) \mid 0 \leq i \leq n-1\} = \mathcal{V}$.*

In particular, the graphs for which there exists an automorphism σ which induces only one orbit are VT^+: it is sufficient to define $\sigma_i = \sigma^i$ for all i from 0 to $n-1$. Figure 5.4 illustrates some VT^+ graphs with a single-orbit automorphism.

[2] A graph $G = (\mathcal{V}, \sim)$ is said to be *vertex-transitive* if for any pair $v, w \in \mathcal{V}$ there exists an automorphism σ such that $\sigma(\mathrm{v}) = w$.



(a) Cycle: degree 2 (b) Degree 4 (c) Clique: degree 5

Figure 5.4: Some VT^+ graphs

From graph theory we know that neither of the two classes subsumes the other. They have however a non-empty intersection, which contains in particular all the structures of the form $(Val^u, \sim)$, i.e. the database domains.

The two next propositions show that the structure $(\mathcal{X}, \sim) = (Val^u, \sim)$ is both a distance-regular graph and a VT^+ graph.

Proposition 35. *If $v \geq 2$, the graph $(Val^u, \sim)$ is a connected distance-regular graph with diameter $\delta = u$, and intersection numbers $b_d = (u-d)(v-1)$ and $c_d = d$, for all $0 \leq d \leq \delta$.*

Proof. The vertices of $(Val^u, \sim)$ are u-tuples $(v_1, \ldots, v_u)$, $v_i \in Val$ and two vertices are adjacent if and only if the differ in exactly one element v_i. It is easy to see that the distance between two vertices is the number of elements in which they differ. Let $x_1, x_2 \in Val^u$ with $d(x_1, x_2) = d$, so they differ in exactly d elements. To go at distance $d+1$ from x_1 we can select any of the remaining $u-d$ elements and change it in $v-1$ possible ways, so the total number is $(u-d)(v-1)$ and depends only on d, not on x_1, x_2. Similarly, by changing one of the differing elements of x_2 to match the value of x_1 we get a vertex at distance $d-1$, and there are d such elements. □

Proposition 36. *The graph $(Val^u, \sim)$ is a VT^+ graph.*

Proof. Recall that we assume the values in the set Val to be indexed, i.e. $Val = \{v_0, \ldots, v_j, \ldots, v_{v-1}\}$, where $v = |Val|$. Note that, for convenience, we opt to use here the indexing from 0 to $v-1$. Let us define an bijective function $\rho : Val \to Val$ as

$$\rho(v_j) = v_{j \oplus 1}$$

for every $v_j \in Val$, and where $\oplus$ represents the sum modulo v. We define the composition of ρ with itself i times as

$$\rho^i(v_j) = \underbrace{\rho \circ \rho \circ \ldots \circ \rho}_{i \text{ times}}(v_j)$$

Note that since ρ is injective, ρ^i is injective as well.

We represent a database in Val^u as $x = \mathrm{v}_{k_0} \dots \mathrm{v}_{k_\ell} \dots \mathrm{v}_{k_{u-1}}$, with $0 \leq \ell \leq u-1$ and $0 \leq k_\ell \leq v-1$. We now define a family $\{\sigma_\iota\}_{\iota=0}^{v^u-1}$ of automorphisms as follows. Given a $0 \leq \iota \leq v^u - 1$, consider the representation in base v of ι:

$$\iota = i_0 \cdot v^0 + \ldots + i_\ell \cdot v^\ell + \ldots + i_{u-1} \cdot v^{u-1} \tag{5.2}$$

where $0 \leq i_\ell \leq v-1$. Then define

$$\sigma_\iota(x) = \rho^{i_0}(\mathrm{v}_{k_0}) \dots \rho^{i_\ell}(\mathrm{v}_{k_\ell}) \dots \rho^{i_{u-1}}(\mathrm{v}_{k_{u-1}}) \tag{5.3}$$

where $x = \mathrm{v}_{k_0} \dots \mathrm{v}_{k_\ell} \dots \mathrm{v}_{k_{u-1}}$.

We have to show that:

- *σ_ι is an automorphism for all $0 \leq \iota \leq v^u - 1$.*

  First we show that σ_ι is injective. Let us consider two arbitrary databases $x = \mathrm{v}_{k_0} \dots \mathrm{v}_{k_\ell} \dots \mathrm{v}_{k_{u-1}}$ and $x' = \mathrm{v}_{k'_0} \dots \mathrm{v}_{k'_\ell} \dots \mathrm{v}_{k'_{u-1}}$, and assume $\sigma_\iota = \rho^{i_0}(\cdot) \dots \rho^{i_\ell}(\cdot) \dots \rho^{i_{u-1}}(\cdot)$. If $x \neq x'$ then $\mathrm{v}_{k_\ell} \neq \mathrm{v}_{k'_\ell}$ for some ℓ, and since an arbitrary ρ^{i_ℓ} is injective we have $\rho^{i_\ell}(\mathrm{v}_{k_\ell}) \neq \rho^{i_\ell}(\mathrm{v}_{k'_\ell})$. Therefore $\sigma_\iota(x) \neq \sigma_\iota(x')$.

  Now we show that if $x \sim x'$ then $\sigma_\iota(x) \sim \sigma_\iota(x')$. Consider an arbitrary pair of adjacent databases $x = \mathrm{v}_{k_0} \dots \mathrm{v}_{k_\ell} \dots \mathrm{v}_{k_{u-1}}$ and $x' = \mathrm{v}_{k_0} \dots \mathrm{v}_{k'_\ell} \dots \mathrm{v}_{k_{u-1}}$, where x and x' differ exactly for $\mathrm{v}_{k_\ell} \neq \mathrm{v}_{k'_\ell}$. We know that $\sigma_\iota(x) = \rho^{i_0}(\mathrm{v}_{k_0}) \dots \rho^{i_\ell}(\mathrm{v}_{k_\ell}) \dots \rho^{i_{u-1}}(\mathrm{v}_{k_{u-1}})$ and we also know that $\sigma_\iota(x') = \rho^{i_0}(\mathrm{v}_{k_0}) \dots \rho^{i_\ell}(\mathrm{v}_{k'_\ell}) \dots \rho^{i_{u-1}}(\mathrm{v}_{k_{u-1}})$. Therefore $\sigma_\iota(x)$ and $\sigma_\iota(x')$ can differ at most in $\rho^{i_\ell}(\mathrm{v}_{k_\ell})$ and $\rho^{i_\ell}(\mathrm{v}_{k'_\ell})$. Since ρ^{i_ℓ} is injective, we have $\rho^{i_\ell}(\mathrm{v}_{k_\ell}) \neq \rho^{i_\ell}(\mathrm{v}_{k'_\ell})$, and it follows that $\sigma_\iota(x) \sim \sigma_\iota(x')$.

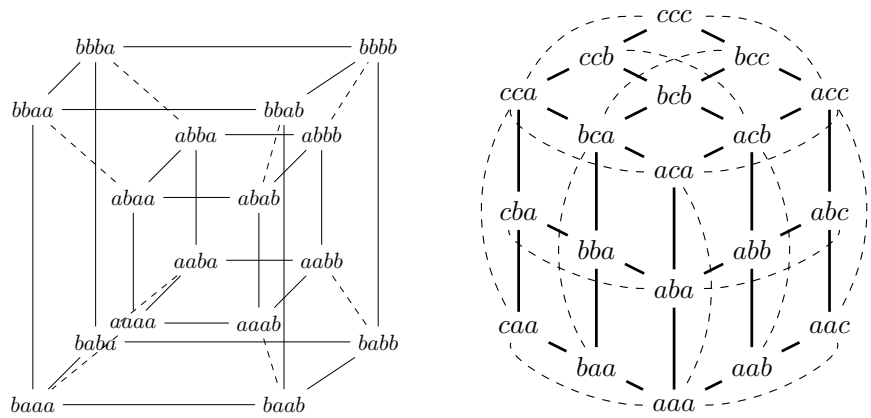
- *For every $x = \mathrm{v}_{k_0} \dots \mathrm{v}_{k_\ell} \dots \mathrm{v}_{k_{u-1}}$ in Val^u we have $\bigcup_{\iota=0}^{v^u-1} \{\sigma_\iota(x)\} = Val^u$.*

  Take an arbitrary element $x' = \mathrm{v}_{k'_0} \dots \mathrm{v}_{k'_\ell} \dots \mathrm{v}_{k'_{u-1}}$ in Val^u. Note that $\rho^{k_m}(\mathrm{v}_{k_n}) = \mathrm{v}_{k_{m \oplus n}}$ for all $0 \leq m, n \leq v-1$. Therefore the automorphism $\sigma = \rho^{k'_0 \ominus k_0}(\cdot) \dots \rho^{k'_\ell \ominus k_\ell}(\cdot) \dots \rho^{k'_{u-1} \ominus k_{u-1}}(\cdot)$, where $\ominus$ represents the subtraction modulo v, satisfies $\sigma(x) = x'$. Since $0 \leq k'_\ell \ominus k_\ell \leq v-1$ we have that $\sigma = \sigma_\iota$ for $\iota = (k'_0 \ominus k_0) \cdot v^0 + \ldots + (k'_\ell \ominus k_\ell) \cdot v^\ell + \ldots + (k'_{u-1} \ominus k_{u-1}) \cdot v^{u-1}$, and therefore σ belongs to the family $\{\sigma_\iota\}_{\iota=0}^{v^u-1}$.

□

Figure 5.5 illustrates some examples of structures $(Val^u, \sim)$. Note that when $|Val| = 2$, $(Val^u, \sim)$ is the u-dimensional hypercube.

The relation between graph structures we consider in this chapter is summarized in Figure 5.6. We remark that in general the graphs $(Val^u, \sim)$ do not have a single-orbit automorphism.



(a) $u = 4$, $Val = \{a, b\}$ (4-dimensional hypercube)

(b) $u = 3$, $Val = \{a, b, c\}$ (for readability sake we show only part of the graph)

Figure 5.5: Some $(Val^u, \sim)$ graphs

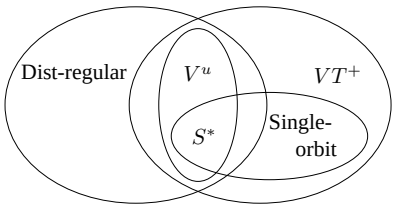


Figure 5.6: Venn diagram for the classes of graphs considered in this section. Here $S^* = \{ Val^u \mid |Val| = 2, u \leq 2\}$

5.4 Deriving the relation between differential privacy and quantitative information flow on the basis of the graph structure

In this section we present the main technical contribution of the chapter: a general technique that explores the graph structure induced by the adjacency relation $\sim$ on $\mathcal{X}$ and the query f to determine relations between ϵ-differential privacy and min-entropy leakage, and between ϵ-differential privacy and utility. We use the symmetries of the graph structure $(\mathcal{X}, \sim)$ to transform the channel matrix into an equivalent matrix with certain regularities. These regularities are the key that allow us to establish the link between ϵ-differential privacy and the a posteriori min-entropy (i.e. the conditional min-entropy associated to the channel). The establishment of bounds on the a posteriori entropy will allow us to derive bounds on leakage and utility: in Section 5.5 we will cope with leakage and in Section 5.6 we will cope with utility.

But first, in Section 5.4.2 we will present how to perform the transformation

on the channel matrix, and in Section 5.4.3 we will show how to derive a bound on the a posteriori min-entropy for the matrix obtained. It is important to note that we consider the case where *the channel input has the uniform distribution*. This is not a restriction for our bounds on the leakage: as seen in Chapter 3, the maximum min-entropy leakage is achieved in the uniform input distribution and, therefore, any bound for the uniform input distribution is also a bound for all other input distributions. In the case of utility the assumption of uniform input distribution is more restrictive, but we will see that it still provides interesting results for several practical cases.

Before we present formally our technique, let us fix some notation.

5.4.1 Assumptions and notation

In the rest of this section we consider channels (usually referred to by M, M', M'' or N) with input A and output B, with finite carriers $\mathcal{A} = \{a_0, \ldots, a_{n-1}\}$ and $\mathcal{B} = \{b_0, \ldots, b_{m-1}\}$, respectively, and we assume that the probability distribution of A is uniform. Furthermore, we assume that $|\mathcal{A}| = n \leq |\mathcal{B}| = m$. If it is the case that $n > m$, we just add to the matrix enough zero-ed columns, i.e. columns containing only 0's, so as to match the number of rows. Note that adding zero-ed columns does not change the min-entropy leakage nor the conditional min-entropy of the channel. We assume as well an adjacency relation $\sim$ on $\mathcal{A}$, i.e. that $(\mathcal{A}, \sim)$ is an undirected graph structure. With a slight abuse of notation, we will also write $i \sim h$ when i and h are associated to adjacent elements of $\mathcal{A}$, and we will write $d(i, h)$ to denote the distance between the elements of $\mathcal{A}$ associated to i and h. More generally, we may use the number i to denote the element a_i of $\mathcal{A}$ (or, equivalently, the element b_i of $\mathcal{B}$) whenever it is clear from the context.

We note that a channel matrix M satisfies ϵ-differential privacy if for each column j and for each pair of rows i and h such that $i \sim h$ we have that:

$$\frac{1}{e^\epsilon} \leq \frac{M_{i,j}}{M_{h,j}} \leq e^\epsilon.$$

The a posteriori entropy of a channel with matrix M will be denoted by $H_\infty^M(A|B)$, and its min-entropy leakage by $I_\infty^M(A;B)$.

We denote by $M[l \to k]$ the matrix obtained by "collapsing" the column l into k, i.e.

$$M[l \to k]_{i,j} = \begin{cases} M_{i,k} + M_{i,l} & \text{if } j = k, \\ 0 & \text{if } j = l, \\ M_{i,j} & \text{otherwise} \end{cases}$$

Given a partial function $\rho : \mathcal{A} \to \mathcal{B}$, the image of $\mathcal{A}$ under ρ is $\rho(\mathcal{A}) = \{\rho(a) | a \in \mathcal{A}, \rho(a) \neq \perp\}$, where $\perp$ stands for "undefined".

In the proofs we will need to use several indices, and we will typically use the letters i, j, h, k, l to range over rows and columns (usually i, h, l will range

over rows and j, k will range over columns). Given a matrix M, we denote by $\max_j^M$ the maximum value of column j over all rows i, i.e. $\max_j^M = \max_i M_{i,j}$, and by $\max^M = \max_{i,j} M_{i,j}$ the maximum element of the matrix.

Finally, given a graph $G = (\mathcal{V}, \sim)$ with diameter δ, we denote by Δ_G the set $\{0, 1, \ldots, \delta\}$. We may omit the subscript and denote the set only by Δ if the context does not allow any confusion. The notation $\mathcal{V}_{\langle d \rangle}(v)$ represents the subset of $\mathcal{V}$ of all elements w at distance d from v. For a fixed d, we define $n_d = |\mathcal{V}_{\langle d \rangle}(v)|$ as the number of vertices in $\mathcal{V}$ at distance d from v, and we intend that it will be always clear by the context to which set of vertices $\mathcal{V}$ and element v the value n_d is associated to.

5.4.2 The matrix transformation

The transformation on the channel matrices is divided into two steps, and we start this section by giving an overview of the process. Consider a channel whose matrix M has at least as many columns as rows and assume that the input distribution is uniform. First, we transform M into a matrix M' in which each of the first n columns has a maximum in the diagonal, and the remaining columns are all 0's. Second, under the assumption that the input domain is distance-regular or VT^+, we transform M' into a matrix M'' whose diagonal elements are all the same, and coincide with the maximum element $\max^{M''}$ of M''. The transformation ensures that both M' and M'' are valid channel matrices (i.e. each row is a probability distribution), also respect ϵ-differential privacy, and preserve the value of the a posteriori entropy for the uniform input distribution. A scheme of the transformation is shown in Figure 5.7, where Lemma 37 (*Step* 1) is applied on the first step of the transformation, and on the second step either Lemma 38 (*Step* 2*a*) or Lemma 39 (*Step* 2*b*) is applied, depending on whether the graph structure is distance-regular or VT^+, respectively.

We now present formally the transformation. The next Lemma is relative to the first step.

Lemma 37 (Step 1)**.** *Let M be a channel matrix of dimensions $n \times m$ with at least as many columns as rows, and assume that M satisfies ϵ-differential privacy. Then it is possible to transform M into a matrix M' satisfying the following conditions:*

(i) M' is a valid channel matrix: $\sum_{j=0}^{m-1} M'_{i,j} = 1$ for all $0 \leq i \leq n-1$;

(ii) Each of the first n columns has a maximum in the diagonal: $M'_{i,i} = \max_i^{M'}$ for all $0 \leq i \leq n-1$;

(iii) The $m-n$ last columns contain only 0's: $M'_{i,j} = 0$ for all $0 \leq i \leq n-1$ and all $n \leq j \leq m-1$;

$$M \begin{bmatrix} M_{0,0} & M_{0,1} & \dots & M_{0,m-1} \\ M_{1,0} & M_{1,1} & \dots & M_{1,m-1} \\ \vdots & \vdots & \ddots & \vdots \\ M_{n-1,0} & M_{n-1,1} & \dots & M_{n-1,m-1} \end{bmatrix}$$

Lemma Step 1
(any graph structure)

$$M' \left[\begin{array}{cccc|ccc} \max_0^{M'} & - & \dots & - & 0 & \dots & 0 \\ - & \max_1^{M'} & \dots & - & 0 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ - & - & \dots & \max_{n-1}^{M'} & 0 & \dots & 0 \end{array}\right]$$

Lemma Step 2*a*
(dist-reg)

Lemma Step 2*b*
(VT^+)

$$M'' \left[\begin{array}{cccc|ccc} \max^{M''} & - & \dots & - & 0 & \dots & 0 \\ - & \max^{M''} & \dots & - & 0 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ - & - & \dots & \max^{M''} & 0 & \dots & 0 \end{array}\right]$$

Figure 5.7: Steps of the matrix transformation for distance-regular and VT^+ graphs

(iv) *M' satisfies ϵ-differential privacy: $\frac{M'_{i,j}}{M'_{h,j}} \leq e^{\epsilon}$ for all $0 \leq i, h \leq n-1$ s.t. $i \sim h$ and all $0 \leq j \leq m-1$;*

(v) *$H_{\infty}^{M'}(A|B) = H_{\infty}^{M}(A|B)$, if A has the uniform distribution.*

Proof. We first show that there exists a matrix N of dimensions $n \times m$, and an injective total function $\rho : \mathcal{A} \to \mathcal{B}$ such that [3]:

- $N_{i,\rho(i)} = \max_{\rho(i)}^{N}$ for all $i \in \mathcal{A}$, and
- $N_{i,j} = 0$ for all $j \in \mathcal{B} \backslash \rho(\mathcal{A})$ and all $i \in \mathcal{A}$.

We iteratively construct ρ and N "column by column" via a sequence of approximating partial functions ρ_s and matrices N_s $(0 \leq s \leq m)$.

- *Initial step* $(s = 0)$

  Define $\rho_0(i) = \perp$ for all $i \in \mathcal{A}$ and $N_0 = M$.

[3] To avoid a heavy notation, here we will use the convention established in Section 5.4.1 and denote N_{a_i,b_j}, where $a_i \in \mathcal{A}$ and $b_j \in \mathcal{B}$, simply by $N_{i,j}$.

- s^{th} *step* $(1 \leq s \leq m)$

  Let j be the s-th column and let $i \in \mathcal{A}$ be one of the rows containing the maximum value of column j in M, i.e. $M_{i,j} = \max_j^M$. There are two cases:

  1. $\rho_{s-1}(i) = \perp$. We define:

  $$\rho_s = \rho_{s-1} \cup \{i \mapsto j\} \qquad \text{and}$$
  $$N_s = N_{s-1}$$

  2. $\rho_{s-1}(i) = k \in \mathcal{B}$. We "collapse" column j into column k (recall the notation introduced in Section 5.4.1):

  $$\rho_s = \rho_{s-1} \qquad \text{and}$$
  $$N_s = N_{s-1}[j \to k]$$

Since the operation of "collapsing" assigns j in ρ_s and then zeroes the column j in N_s, all unassigned columns $\mathcal{B} \setminus \rho_m(\mathcal{A})$ must be zero in N_m. We finish the construction by taking ρ to be the same as ρ_m after assigning to each unassigned row one of the columns in $\mathcal{B} \setminus \rho_m(\mathcal{A})$ (there are enough such columns since $n \leq m$). We also take $N = N_m$. Note that by construction N is a channel matrix.

Thus we get a matrix N and a function $\rho : \mathcal{A} \to \mathcal{B}$ which, by construction, is injective and satisfies $N_{i,\rho(i)} = \max_{\rho(i)}^N$ for all $i \in \mathcal{A}$, and $N_{i,j} = 0$ for all $j \in \mathcal{B} \setminus \rho(\mathcal{A})$ and all $i \in \mathcal{A}$. Furthermore, N provides ϵ-differential privacy (condition (iv)) because each column is a linear combination of columns of M. It is also easy to see that $\sum_j \max_j^N = \sum_j \max_j^M$, and from that it immediately follows that $H_\infty^N(A|B) = H_\infty^M(A|B)$ (recall that A has the uniform distribution and therefore the a posteriori entropy is a function of the sum of the maximum of each column), so condition (v) is satisfied.

Finally, we create our claimed matrix M' from N just by rearranging the columns according to ρ. Note that the order of the columns is irrelevant, since any permutation represents the same conditional probabilities and therefore the same channel [4]. The resulting matrix M' has all maxima in the diagonal $M'_{i,i}$ for $0 \leq i \leq n-1$, and every element in the columns $n \leq j \leq m-1$ are 0, which satisfies conditions (ii) and (iii). Also, since N is a valid channel matrix, so is M' and condition (i) is also satisfied.

□

[4] Note that by rearranging the columns of the channel matrix we may change the marginal probability of the outputs. This, however, does not pose a problem for our purposes, since the maximum a posteriori entropy of the channel will be maintained. If we want the marginal probability of the outputs to remain unchanged, we can just "relabel" the columns after the rearrangement so they will match the correct outputs.

The second step of the transformation depends on the graph structure of $(\mathcal{A}, \sim)$. But before we discuss this step, let us introduce a notion of distance between elements in $\mathcal{B}$, derived from the notion of distance between elements in $\mathcal{A}$. Let M be a channel matrix in which the maximum of each column is in the diagonal, as in Figure 5.8. Then we define the distance between two elements $j_1, j_2 \in \mathcal{B}$ as follows:

$$d(j_1, j_2) = \begin{cases} d(i_1, i_2) & \text{if there are } i_1, i_2 \in \mathcal{A} \text{ such that } i_1 = j_1 \text{ and } i_2 = j_2, \\ \perp & \text{otherwise.} \end{cases} \tag{5.4}$$

Note that the range of the notion of distance defined above is the set $\Delta = \{0, 1, \ldots, \delta\}$, where δ is the diameter of $(\mathcal{A}, \sim)$. Based on (5.4), we define the set $\mathcal{B}_{\langle d \rangle}(j)$ as the subset of $\mathcal{B}$ of elements at distance d from an element $j \in \mathcal{B}$. It is clear that for any $j \in \mathcal{B}$, we have $\bigcup_{d \in \Delta} \mathcal{B}_{\langle d \rangle}(j) = \mathcal{B}$.

$$\begin{array}{c}
\\
\\
\\
\text{row } i \\
\\
\\
\\
\\
\\
\end{array}
\left[\begin{array}{cccccccc}
M_{0,0} & M_{0,1} & \ldots & & & \ldots & M_{0,m-2} & M_{0,m-1} \\
M_{1,0} & \ldots & & & & \ldots & & M_{1,m-1} \\
\vdots & & & & & & & \vdots \\
\hline
M_{i,0} & \ldots & M_{i,j'} & & \ldots & M_{i,j''} & \ldots & M_{i,m-1} \\
\hline
 & & \overset{d(i,j')}{\{} \vdots & & & \vdots & & \\
 & & M_{j',j'} = \max_{j'}^{M} & & & \vdots & & \\
 & & & & & \vdots & & \\
 & & & & & \vdots & & \\
 & & & & & M_{j'',j''} = \max_{j''}^{M} & & \\
\vdots & & & & & & & \vdots \\
M_{n-2,0} & \ldots & & & & \ldots & & M_{n-2,m-1} \\
M_{n-1,0} & M_{n-1,1} & \ldots & & & \ldots & M_{n-1,m-2} & M_{n-1,m-1}
\end{array}\right]
\begin{array}{l}
\\
\\
\\
\\
\\
\\
\\
\\
\\
\\
\end{array}
\begin{array}{l}
\\
\\
\\
\\
\\
\\
\\
\\
\\
\\
\end{array}
\begin{array}{l}
\\
\\
\\
\\
\\
\\
\\
\\
\\
\\
\end{array}$$

$$\left.\begin{array}{c} \vdots \\ \vdots \\ \vdots \end{array}\right\}^{d(i,j'')}$$

Figure 5.8: The relation between elements of a row i and the elements in the diagonal

We can extend the adjacency relation $\sim$ on $\mathcal{A}$ to an adjacency relation $\sim'$ on $\mathcal{B}$ by using the notion of distance of (5.4). For any $j_1, j_2 \in \mathcal{B}$, we have $j_1 \sim' j_2$ if and only if $d(j_1, j_2) = 1$. Therefore, if $(\mathcal{A}, \sim)$ is distance-regular, so it is $(\mathcal{B}, \sim')$.

Now we are ready to present the lemma for the second step of the transformation, in the case of distance-regular graphs.

Lemma 38 (Step 2*a*)**.** *Let M' be a channel matrix of dimensions $n \times m$ with at least as many columns as rows, and assume that M' satisfies ϵ-differential privacy. Let $\sim$ be an adjacency relation on $\mathcal{A}$ such that the graph $(\mathcal{A}, \sim)$ is connected and distance-regular. Assume that the maximum value of each column is on the diagonal, that is $M_{i,i} = \max_i^M$ for all $i \in \mathcal{A}$, and that all the*

last $m-n$ columns have only zero elements, i.e. $M'_{i,j} = 0$ for all $0 \le i \le n-1$ and $n \le j \le m-1$. Then it is possible to transform M' into a matrix M'' satisfying the following conditions:

(i) M'' is a valid channel matrix: $\sum_{j=0}^{m-1} M''_{i,j} = 1$ for all $0 \le i \le n-1$;

(ii) The elements of the diagonal are all the same, and are equal to the maximum of the matrix: $M''_{i,i} = \max^{M''}$ for all $0 \le i \le n-1$;

(iii) The $m-n$ last columns contain only 0's: $M''_{i,j} = 0$ for all $0 \le i \le n-1$ and all $n \le j \le m-1$;

(iv) M'' satisfies ϵ-differential privacy: $\frac{M'_{i,j}}{M'_{h,j}} \le e^{\epsilon}$ for all $0 \le i, h \le n-1$ s.t. $i \sim h$ and all $0 \le j \le m-1$;

(v) $H_\infty^{M''}(A|B) = H_\infty^{M'}(A|B)$, if A has the uniform distribution.

Proof. Let us define $\mathcal{B}^* = \{0, 1, \ldots, n-1\}$, i.e. the subset of $\mathcal{B}$ that excludes the zero-ed columns of M' from n to $m-1$. Note that we can safely use the set $\mathcal{B}^*$ instead of $\mathcal{B}$ in this proof because the zero-ed columns do not contribute to the a posteriori entropy, and trivially respect ϵ-differential privacy.

We then define the matrix M'' as follows.

$$M''_{i,j} = \begin{cases} \frac{1}{n|\mathcal{A}_{\langle d(i,j)\rangle}(i)|} \sum_{k \in \mathcal{B}^*} \sum_{h \in \mathcal{A}_{\langle d(i,j)\rangle}(k)} M'_{h,k} & \text{if } j \in \mathcal{B}^*, \\ 0 & \text{otherwise.} \end{cases}$$

By the definition above, condition (iii) is immediately satisfied. We then show that this definition also induces a channel matrix. We have

$$\begin{aligned} \sum_{j \in \mathcal{B}^*} M''_{i,j} &= \sum_{j \in \mathcal{B}^*} \frac{1}{n|\mathcal{A}_{\langle d(i,j)\rangle}(i)|} \sum_{k \in \mathcal{B}^*} \sum_{h \in \mathcal{A}_{\langle d(i,j)\rangle}(k)} M'_{h,k} \\ &= \frac{1}{n} \sum_{k \in \mathcal{B}^*} \sum_{j \in \mathcal{B}^*} \frac{1}{|\mathcal{A}_{\langle d(i,j)\rangle}(i)|} \sum_{h \in \mathcal{A}_{\langle d(i,j)\rangle}(k)} M'_{h,k} \end{aligned}$$

Recall that $\Delta = \{0, \ldots, \delta\}$, where δ is the diameter of the graph. Note that for every i, $\mathcal{B}^* = \bigcup_{d \in \Delta} \mathcal{B}^*_{\langle d\rangle}(i)$, and for different values of d the sets $\mathcal{B}^*_{\langle d\rangle}(i)$ are disjoint. Therefore the summation over $j \in \mathcal{B}^*$ can be split as follows

$$\begin{aligned} &= \frac{1}{n} \sum_{k \in \mathcal{B}^*} \sum_{d \in \Delta} \sum_{j \in \mathcal{B}^*_{\langle d\rangle}(i)} \frac{1}{|\mathcal{A}_{\langle d\rangle}(i)|} \sum_{h \in \mathcal{A}_{\langle d\rangle}(k)} M'_{h,k} \\ &= \frac{1}{n} \sum_{k \in \mathcal{B}^*} \sum_{d \in \Delta} \sum_{h \in \mathcal{A}_{\langle d\rangle}(k)} M'_{h,k} \sum_{j \in \mathcal{B}^*_{\langle d\rangle}(i)} \frac{1}{|\mathcal{A}_{\langle d\rangle}(i)|} \end{aligned}$$

as $\sum_{j \in \mathcal{B}^*_{\langle d \rangle}(i)} \frac{1}{|\mathcal{A}_{\langle d \rangle}(i)|} = 1$, we obtain

$$= \frac{1}{n} \sum_{k \in \mathcal{B}^*} \sum_{d \in \Delta} \sum_{h \in \mathcal{A}_{\langle d \rangle}(k)} M'_{h,k}$$

and now the summations over h can be joined together

$$\begin{aligned} &= \frac{1}{n} \sum_{k \in \mathcal{B}^*} \sum_{h \in \mathcal{A}} M'_{h,k} \\ &= 1 \end{aligned}$$

which implies that condition (i) is satisfied.

We now turn our attention to the elements of the diagonal. We have

$$M''_{i,i} = \frac{1}{n} \sum_{h \in \mathcal{A}} M'_{h,h}$$

and so they are all identical. To fulfill condition (ii) we still need to show that $M''_{i,i} = \max_i^{M''}$ for all $i \in \mathcal{A}$.

$$\begin{aligned} M''_{i,j} &= \frac{1}{n|\mathcal{A}_{\langle d(i,j) \rangle}(i)|} \sum_{k \in \mathcal{B}^*} \sum_{h \in \mathcal{A}_{\langle d(i,j) \rangle}(k)} M'_{h,k} \\ &\leq \frac{1}{n|\mathcal{A}_{\langle d(i,j) \rangle}(i)|} \sum_{k \in \mathcal{B}^*} \sum_{h \in \mathcal{A}_{\langle d(i,j) \rangle}(k)} M'_{h,h} && \text{(since the biggest element is in the diagonal)} \\ &= \frac{1}{n} \sum_{k \in \mathcal{B}^*} M'_{h,h} \frac{1}{|\mathcal{A}_{\langle d(i,j) \rangle}(i)|} \sum_{h \in \mathcal{A}_{\langle d(i,j) \rangle}(k)} 1 \\ &= \frac{1}{n} \sum_{k \in \mathcal{B}^*} M'_{h,h} \frac{|\mathcal{A}_{\langle d(i,j) \rangle}(k)|}{|\mathcal{A}_{\langle d(i,j) \rangle}(i)|} \\ &= \frac{1}{n} \sum_{k \in \mathcal{B}^*} M'_{h,h} \cdot 1 && \text{(since the graph is distance-regular)} \\ &= M''_{i,i} \end{aligned}$$

Since A has the uniform distribution, $H_\infty^{M'}(A|B) = H_\infty^{M''}(A|B)$ (condition (v)) follows immediately.

It remains to show that M'' satisfies ϵ-differential privacy (condition (iv)). We need to show that

$$M''_{i,j} \leq e^\epsilon M''_{i',j} \qquad \forall j \in \mathcal{B}, i, i' \in \mathcal{A} : i \sim i'$$

From the triangular inequality we have (since $d(i, i') = 1$)

$$d(i', j) - 1 \leq d(i, j) \leq d(i', j) + 1$$

Thus, there are 3 possible cases:

1. $d(i,j) = d(i',j)$

   The result is immediate since $M''_{i,j} = M''_{i',j}$.

2. $d(i,j) = d(i',j) - 1$

   We define the set of neighbors of h "one step further away" from k:

$$\mathcal{F}_{h,k} = \{h' \sim h \mid h' \in \mathcal{A}_{\langle d(h,k)+1\rangle}(k)\}$$

   Note that $|\mathcal{F}_{h,k}| = b_{d(h,k)}$ since the graph is distance-regular. The following inequalities hold for any $h, h' \in \mathcal{A}$:

$$\begin{aligned} M'_{h,k} &\leq e^{\epsilon} M'_{h',k} && \forall h' \in \mathcal{F}_{h,k} && \text{(diff. privacy)} \Rightarrow \\ b_{d(h,k)} M'_{h,k} &\leq e^{\epsilon} \sum_{h' \in \mathcal{F}_{h,k}} M'_{h',k} && && \text{(sum of the above)} \end{aligned}$$

   we now fix a distance d and sum the above inequalities for all vertices at distance d from h:

$$\sum_{h \in \mathcal{A}_{\langle d\rangle}(k)} b_d M'_{h,k} \leq e^{\epsilon} \sum_{h \in \mathcal{A}_{\langle d\rangle}(k)} \sum_{h' \in \mathcal{F}_{h,k}} M'_{h',k}$$

   Note that each $h' \in \mathcal{A}_{\langle d+1\rangle}(k)$ is contained in $\mathcal{F}_{h,k}$ for exactly c_{d+1} different $h \in \mathcal{A}_{\langle d\rangle}(k)$. So the right-hand side above sums all vertices of $\mathcal{A}_{\langle d+1\rangle}(k)$ exactly c_{d+1} times each. Thus we get that for all $k \in \mathcal{B}^*, d \in \Delta$:

$$b_d \sum_{h \in \mathcal{A}_{\langle d\rangle}(k)} M'_{h,k} \leq e^{\epsilon}\, c_{d+1} \sum_{h \in \mathcal{A}_{\langle d+1\rangle}(k)} M'_{h,k} \tag{5.5}$$

   Finally, note that $c_{d+1}|\mathcal{A}_{\langle d+1\rangle}(i)| = b_d|\mathcal{A}_{\langle d\rangle}(i)|$ (both sides count the number of edges between a vertex at distance d and a vertex at distance $d+1$). So we have

$$\begin{aligned} M''_{i,j} &= \frac{1}{n|\mathcal{A}_{\langle d\rangle}(i)|} \sum_{k \in \mathcal{B}^*} \sum_{h \in \mathcal{A}_{\langle d\rangle}(k)} M'_{h,k} \\ &\leq e^{\epsilon} \frac{1}{n|\mathcal{A}_{\langle d\rangle}(i)|} \frac{c_{d+1}}{b_d} \sum_{k \in \mathcal{B}^*} \sum_{h \in \mathcal{A}_{\langle d+1\rangle}(k)} M'_{h,k} && \text{(from (5.5))} \\ &= e^{\epsilon} \frac{1}{n|\mathcal{A}_{\langle d+1\rangle}(i)|} \sum_{k \in \mathcal{B}^*} \sum_{h \in \mathcal{A}_{\langle d+1\rangle}(k)} M'_{h,k} \\ &= e^{\epsilon} M''_{i',j} \end{aligned}$$

3. $d(i,j) = d(i',j) + 1$

   This case is analogous to the case case where $d(i,j) = d(i',j) - 1$.

□

The next lemma is relative to the second step of the transformation, for the case of VT^+ graphs.

Lemma 39 (Step 2*b*)**.** *Consider a channel matrix M' satisfying the assumptions of Lemma 38, except for the assumption about distance-regularity, which we replace by the assumption that $(\mathcal{A}, \sim)$ is VT^+. Then it is possible to transform M' into a matrix M'' with the same properties as in Lemma 38.*

Proof. Let us define $\mathcal{B}^* = \{0, 1, \ldots, n-1\}$, i.e. the subset of $\mathcal{B}$ that excludes the zero-ed columns of M' from n to $m-1$. Note that we can safely use the set $\mathcal{B}^*$ instead of $\mathcal{B}$ in this proof because the zero-ed columns do not contribute to the a posteriori entropy, and trivially respect ϵ-differential privacy.

We then define the matrix M'' as follows.

$$M''_{i,j} = \begin{cases} \frac{1}{n} \sum_{h=0}^{n-1} M'_{\sigma_h(i),\sigma_h(j)} & \text{if } j \in \mathcal{B}^*, \\ 0 & \text{otherwise.} \end{cases}$$

By the definition above, condition (iii) is immediately satisfied. We then show that this definition also induces a channel matrix. Recall that $\{\sigma_h(j) | 0 \leq h \leq n-1\} = \mathcal{A}$ since the graph is VT^+.

$$\begin{aligned} \sum_{j=0}^{n-1} M''_{i,j} &= \sum_{j=0}^{n-1} \frac{1}{n} \sum_{h=0}^{n-1} M'_{\sigma_h(i),\sigma_h(j)} \\ &= \sum_{h=0}^{n-1} \frac{1}{n} \sum_{j=0}^{n-1} M'_{\sigma_h(i),\sigma_h(j)} \\ &= \sum_{h=0}^{n-1} \frac{1}{n} \cdot 1 && \text{(since } \sigma_h \text{ is a permutation)} \\ &= 1 \end{aligned}$$

which implies that condition (i) is satisfied.

Now we prove that the diagonal contains the maximum values of the matrix (condition (ii)), i.e. for every i, $M''_{i,i} = \max^{M''}$. It is easy to see that, by definition, the elements of the diagonal are all the same (they are the average of the diagonal elements of M'). Then we need to show that they are the

maximum of each column, from which it follows that they are the maximum of the matrix.

$$
\begin{aligned}
M''_{i,i} &= \frac{1}{n}\sum_{h=0}^{n-1} M'_{\sigma_h(i),\sigma_h(i)} \\
&\geq \frac{1}{n}\sum_{h=0}^{n-1} M'_{\sigma_h(i),\sigma_h(j)} && \text{(since } M'_{\sigma_h(j),\sigma_h(j)} = \max^{M'}_{\sigma_i(j)}\text{)} \\
&= M''_{i,j}
\end{aligned}
$$

We now prove that M'' provides ϵ-differential privacy (condition (iv)). For every pair $i \sim i'$ and every j:

$$
\begin{aligned}
M''_{i,j} &= \frac{1}{n}\sum_{h=0}^{n-1} M'_{\sigma_h(i),\sigma_h(j)} \\
&\leq \frac{1}{n}\sum_{h=0}^{n-1} e^{\epsilon} M'_{\sigma_h(i'),\sigma_h(j)} && \text{(by } \epsilon\text{-diff. privacy, for some } i' \\
& && \text{s.t. } \sigma_h(i') = \sigma_h(j)\text{)} \\
&= e^{\epsilon} M''_{i',j}
\end{aligned}
$$

Finally, we prove condition (v):

$$
\begin{aligned}
H_{\infty}^{M''}(A|B) &= \frac{1}{n}\sum_{i=0}^{n-1} M'_{h,h} \\
&= \frac{1}{n}\sum_{i=0}^{n-1}\frac{1}{n}\sum_{h=0}^{n-1} M'_{\sigma_h(i),\sigma_h(i)} \\
&= \frac{1}{n}\sum_{i=0}^{n-1} H_{\infty}^{M'}(A|B) && \text{(since } M'_{\sigma_h(i),\sigma_h(i)} = \max^{M'}_{\sigma_i(i)}\text{)} \\
&= H_{\infty}^{M'}(A|B)
\end{aligned}
$$

□

5.4.3 The bound on the a posteriori entropy of the channel

Once the transformation presented in the previous section has been applied, and the channel matrix respects the properties of M'', we can use again the

graph structure of $(\mathcal{A}, \sim)$ to determine a bound on the a posteriori entropy $H_\infty^{M''}(A|B)$ of M''. Recall that our matrix transformation preserves the value of the a posteriori conditional entropy, so the bound we find is also valid for the original channel matrix we started with.

It is a known result in literature (cfr. [BCP09]) that, if the distribution on A is uniform, then the a posteriori entropy of the channel M is given by

$$H_\infty^M(A|B) = -\log_2 \frac{1}{n} \sum_{j \in \mathcal{B}} \max_j^M$$

Hence, under our assumption that the input distribution A is uniform, and knowing that matrix the M'' the diagonal elements are all equal to the maximum $\max^{M''}$, we have

$$H_\infty^{M''}(A|B) = -\log_2 \max^{M''} \tag{5.6}$$

Therefore to find a bound on the a posteriori entropy of the channel M'' it is enough to find a bound on $\max^{M''}$. This is exactly what we do in this section.

We proceed by noting that the property of ϵ-differential privacy induces a relation between the ratio of elements at any distance:

Remark 40. *Let M be a matrix satisfying ϵ-differential privacy. Then, for any column j, and any pair of rows i and h we have that:*

$$\frac{1}{e^{\epsilon\, d(i,h)}} \leq \frac{M_{i,j}}{M_{h,j}} \leq e^{\epsilon\, d(i,h)}$$

In particular, as we know that the diagonal elements of M are equal to the maximum element $\max^M$, then for each element $M_{i,j}$ we have that:

$$M_{i,j} \geq \frac{\max^M}{e^{\epsilon\, d(i,j)}} \tag{5.7}$$

which motivates the next proposition.

Proposition 41. *Let M be a channel matrix satisfying ϵ-differential privacy where the diagonal elements are the maximum element max^M of the matrix. Then:*

$$\max^M \leq \frac{1}{\sum_{d \in \Delta} \frac{n_d}{e^{\epsilon d}}}$$

where $\Delta = \{0, 1, \ldots, \delta\}$, δ is the diameter of the graph $(\mathcal{A}, \sim)$, and $n_d = \mathcal{A}_{\langle d \rangle}(j)$ is the number of elements $M_{i,j}$ that are at distance d from the corresponding diagonal element $M_{j,j}$, i.e. such that $d(i,j) = d$.

Proof. The elements of any given row i of M represent a probability distribution, therefore they sum to 1.

$$\sum_j M_{i,j} = 1$$

By substituting (5.7) in the equation above we obtain:

$$\sum_j \left(\frac{\max^M}{e^{\epsilon d(i,j)}} \right) \leq 1$$
$$\sum_d \left(\frac{n_d}{e^{\epsilon d}} \max^M \right) \leq 1$$

and therefore

$$\max^M \leq \frac{1}{\sum_d \frac{n_d}{e^{\epsilon d}}}$$

□

Putting together all the steps of this section, we obtain our main result.

Theorem 42. *Consider a channel matrix M satisfying ϵ-differential privacy for some $\epsilon > 0$, and assume that $(\mathcal{A}, \sim)$ is either distance-regular or VT^+. Then we have:*

$$H_\infty^M(A|B) \geq -\log_2 \frac{1}{\sum_d \frac{n_d}{e^{\epsilon\, d}}} \tag{5.8}$$

where $n_d = |\mathcal{A}_{\langle d \rangle}(i)|$ is the number of nodes $j \in \mathcal{A}$ at distance d from $i \in \mathcal{A}$.

Moreover, this bound it tight, in the sense that we can build a matrix for which (5.8) holds with equality.

Proof. The inequality follows directly from (5.6) and Proposition 41. To prove that the bound is tight, it is sufficient to define each element $M_{i,j}$ according to (5.7) with equality instead of inequality. □

In the next sections we will see how to use this theorem for establishing a bound on the leakage and on the utility.

5.5 Application to leakage

As discussed in the Section 5.2, the correlation $\mathcal{L}(X, Z)$ between X and Z measures the information that the attacker can learn about the database by observing the reported answers. In this section we consider the min-entropy leakage as a measure of this information, that is $\mathcal{L}(X, Z) = I_\infty(X; Z)$. We then investigate bounds on information leakage imposed by differential privacy.

Before we continue, let us make a very important observation about the results we obtain in this section.

Remark 43. *The bounds on the min-entropy leakage we present in this section (Theorem 44, Proposition 47, and Proposition 48) are derived under the assumption that the input distribution X for the channel is uniform. As seen in Chapter 3, we know from the literature [BCP09, Smi09] that the min-entropy leakage $I_\infty^M(X;Z)$ of a given matrix M is maximum when input distribution is uniform (even though it may not be the only case). Therefore the bounds we present in this section, although based on the assumption that X has the uniform distribution, are valid for every possible input distribution. As we model side information as input distributions, and as we provide bounds on the leakage for any possible input distribution, it follows that our bounds on the min-entropy leakage are valid for any possible side information the attacker may have.*

Our first result shows that the min-entropy leakage of a randomized function $\mathcal{K}$ is bounded by a quantity depending on ϵ, and on the numbers $u = |Ind|$ and $v = |Val|$ of individuals and values respectively. We assume that $v \geq 2$.

As seen in Section 5.2, $\mathcal{K}$ can be modeled as a channel with input X and output Z. From Propositions 35 and 36 we know that $(\mathcal{X}, \sim)$ is both distance-regular and VT^+, and therefore we can apply Theorem 42. Then, by (5.7) we know that for $j \in \mathcal{X}_{\langle d \rangle}(x)$ (i.e. every j in $\mathcal{X}$ at distance d from a given x) it is the case that $M_{x,j} \geq \frac{\max^M}{e^{\epsilon d}}$. Furthermore we note that each element j at distance d from x can be obtained by changing the value of d individuals in the u-tuple representing i. We can choose those d individuals in $\binom{u}{d}$ possible ways, and for each of these individuals we can change the value (with respect to the one in x) in $v-1$ possible ways. Therefore $|\mathcal{X}_{\langle d \rangle}(x)| = \binom{u}{d}(v-1)^d$, and we obtain that the number of databases at distance d from x is

$$n_d = |\mathcal{X}_{\langle d \rangle}(x)| = \left(\begin{array}{c} u \\ d \end{array} \right) (v-1)^d \tag{5.9}$$

In fact, recall that x can be represented as a u-tuple with values in V. We need to select d individuals in the u-tuple and then change their values, and each of them can be changed in $v-1$ different ways.

Using the value of n_d from (5.9) in Theorem 42 we obtain the following result.

Theorem 44. *If $\mathcal{K}$ satisfies ϵ-differential privacy, then the information leakage is bound from above as follows:*

$$I_\infty(X;Z) \leq u \log_2 \frac{v\, e^\epsilon}{v - 1 + e^\epsilon} = Bnd(u, v, \epsilon)$$

Proof. For this proof we need a matrix with all column maxima on the diagonal, and all equal. We obtain such a matrix by transforming the matrix associated to $\mathcal{K}$ as follows: first we apply Lemma 37 to it (with $A = X$ and $B = Z$), and then we apply either Lemma 38 or Lemma 39 (we can choose

either of them, since $(\mathcal{X}, \sim)$ is both distance-regular and VT^+). The final matrix M has all non-zero elements on its $n \times n$ submatrix, with $n = |\mathcal{X}| = Val^u$, provides ϵ-differential privacy, and for every row i we have that $M_{i,i} = \max^M$. Furthermore, $I_\infty^M(X;Z)$ is equal to the min-entropy leakage of $\mathcal{K}$, assuming a uniform distribution on X.

Then we can derive:

$$\begin{aligned}\sum_{j=1}^{n} M_{i,j} &\geq \sum_{d=0}^{u} n_d \frac{\max^M}{(e^\epsilon)^d} \\ &= \sum_{d=0}^{u} \binom{u}{d} (v-1)^d \frac{\max^M}{(e^\epsilon)^d} && \text{(by (5.9))}\end{aligned}$$

Since each row represents a probability distribution, the elements of row i must sum up to 1:

$$\sum_{d=0}^{u} \binom{u}{d} (v-1)^d \frac{\max^M}{(e^\epsilon)^d} \leq 1$$

and by multiplying both sides of the inequality by $e^{\epsilon u}$ we get

$$\max^M \textstyle\sum_{d=0}^{u} \binom{u}{d} (v-1)^d e^{\epsilon(u-d)} \leq e^{\epsilon u}$$

Since by the binomial expansion $\displaystyle\sum_{d=0}^{u} \binom{u}{d} (v-1)^d (e^\epsilon)^{u-d} = (v-1+e^\epsilon)^u$, we obtain:

$$\max^M \leq \left(\tfrac{e^\epsilon}{v-1+e^\epsilon}\right)^u \tag{5.10}$$

Therefore:

$$\begin{aligned}I_\infty^M(X;Y) &= H_\infty(X) - H_\infty^M(X|Y) && \text{(by definition)} \\ &= \log_2 Val^u + \log_2 \max^M && \text{(by (5.6))} \\ &\leq \log_2 Val^u + \log_2 \left(\frac{e^\epsilon}{v-1+e^\epsilon}\right)^u && \text{(by (5.10))} \\ &= u \log_2 \frac{v\, e^\epsilon}{v-1+e^\epsilon}\end{aligned}$$

To conclude our proof we recall that, since the above bound on $I_\infty^M(X;Y)$ is valid for the case where X has the uniform distribution, it is also valid for any distribution on X.

□

Note that the bound $Bnd(u,v,\epsilon) = u \log_2 \frac{v\, e^\epsilon}{(v-1+e^\epsilon)}$ is a continuous function in ϵ, has value 0 when $\epsilon = 0$, and converges to $u \log_2 v$ as ϵ approaches infinity.

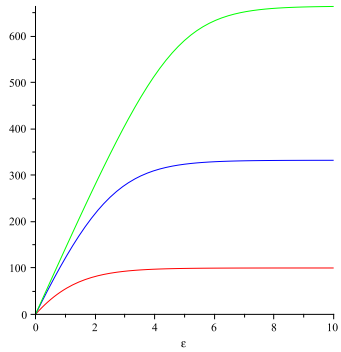


Figure 5.9: Graphs of $Bnd(u, v, \epsilon)$ for u=100 and v=2 (lowest line), v=10 (intermediate line), and v=100 (highest line), respectively.

Figure 5.9 shows the growth of $Bnd(u, v, \epsilon)$ along with ϵ, for various fixed values of u and v.

The next proposition shows that the bound obtained in previous theorem is tight.

Proposition 45. *For every u, v, and ϵ there exists a randomized function $\mathcal{K}$ which provides ϵ-differential privacy and whose min-entropy leakage, for the uniform input distribution, is $I_\infty(X; Z) = Bnd(u, v, \epsilon)$.*

Proof. The adjacency relation in $\mathcal{X}$ determines a graph structure $G_\mathcal{X}$. Set $\mathcal{Z} = \mathcal{X}$ and define the matrix of $\mathcal{K}$ as follows:

$$p_\mathcal{K}(z|x) = \frac{Bnd(u, v, \epsilon)}{(e^\epsilon)^d} \tag{5.11}$$

where d is the distance between x and z in $G_\mathcal{X}$.

We need to show that $p_\mathcal{K}(\cdot|x)$ is a probability distribution for every x:

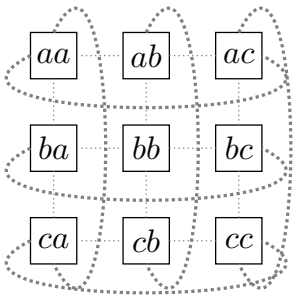
$$\begin{aligned}
\sum_{z\in\mathcal{Z}} \frac{Bnd(u, v, \epsilon)}{(e^\epsilon)^d} &= Bnd(u, v, \epsilon) \sum_{z\in\mathcal{Z}} \frac{1}{(e^\epsilon)^d} \\
&= Bnd(u, v, \epsilon) \sum_{d} \frac{n_d}{(e^\epsilon)^d} \\
&= Bnd(u, v, \epsilon) \frac{1}{\max^M} && \text{by Proposition 41} \\
&= Bnd(u, v, \epsilon) \frac{1}{Bnd(u, v, \epsilon)} && \text{take } d = 0 \text{ in (5.11)} \\
&= 1
\end{aligned}$$

To see that $\mathcal{K}$ provides ϵ-differential privacy, just take $d = 1$ in (5.11), and to see that $I_\infty(X; Z) = Bnd(u, v, \epsilon)$ take $d = 0$ in the same equation.

□

We now give an example of the use of $Bnd(u, v, \epsilon)$ as a bound for the min-entropy leakage.

Example 7. *Assume that we are interested in the eye color of a certain population* $Ind = \{Alice, Bob\}$. *Let* $Val = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$ *where* $\mathtt{a}$ *stands for absent (i.e. the* null *value),* $\mathtt{b}$ *stands for blue, and* $\mathtt{c}$ *stands for coalblack. We can represent each dataset as a tuple* $d_0 d_1$, *where* $d_0 \in Val$ *represents the eye color of Alice (cases* $d_0 = b$ *and* $d_0 = c$*), or that Alice is not in the dataset (case* $d_0 = a$*).* d_1 *provides the same kind of information for Bob. Note that* $v = 3$. *Fig 5.10(a) represents the set* $\mathcal{X}$ *of all possible datasets and its adjacency relation. Fig 5.10(b) represents the matrix with input* $\mathcal{X}$ *which provides* ϵ*-differential privacy and has the highest min-entropy leakage. In the representation of the matrix, the generic entry* α *stands for* $\frac{\max^M}{e^\epsilon \alpha}$, *where* $\max^M$ *is the highest value in the matrix, i.e.* $\max^M = \frac{e^\epsilon}{(v-1+e^\epsilon)} = \frac{e^\epsilon}{(2+e^\epsilon)}$.



(a) The datasets and their adjacency relation

	aa	*ab*	*ac*	*ba*	*ca*	*bb*	*bc*	*cb*	*cc*
aa	0	1	1	1	1	2	2	2	2
ab	1	0	1	2	2	1	2	1	2
ac	1	1	0	2	2	2	1	2	1
ba	1	2	2	0	1	1	2	1	2
ca	1	2	2	1	0	2	2	1	1
bb	2	1	2	1	2	0	1	1	2
bc	2	2	1	1	2	1	0	2	1
cb	2	1	2	2	1	1	2	0	1
cc	2	2	1	2	1	2	1	1	0

(b) The representation of the matrix

Figure 5.10: Universe and highest min-entropy leakage matrix giving ϵ-differential privacy for Example 7.

Note that the bound $Bnd(u, v, \epsilon)$ is guaranteed to be reached with the uniform input distribution. The construction of the matrix for Proposition 45 gives a square matrix of dimension $Val^u \times Val^u$. Often, however, the range of $\mathcal{K}$ is fixed, as it is usually related to the possible answers to the query f. Hence it is natural to consider the scenario in which we are given a number $r < Val^u$, and want to consider only those $\mathcal{K}$'s whose range has cardinality at most r. Proposition 47 shows that in n this restricted setting we can find a better bound than the one given by Theorem 44. But first we need the following lemma.

Lemma 46. *Let* $\mathcal{K}$ *be a randomized function with input* X, *where* $\mathcal{X} = Val^u$, *providing* ϵ*-differential privacy. Assume that* $r = |Range(\mathcal{K})| = v^\ell$, *for some* $\ell < u$. *Let* M *be the matrix associated to* $\mathcal{K}$. *Then it is possible to build a square matrix* M' *of size* $v^\ell \times v^\ell$, *with row and column indices in* $\mathcal{A} \subseteq \mathcal{X}$, *and*

a binary relation $\sim' \subseteq \mathcal{A} \times \mathcal{A}$ *such that* $(\mathcal{A}, \sim')$ *is isomorphic to* $(Val^\ell, \sim_\ell)$*, and such that:*

(i) M' *is a valid channel matrix:* $\sum_{j=0}^{m-1} M'_{i,j} = 1$ *for all* $0 \leq i \leq n-1$*;*

(ii) $M'_{i,j} \leq (e^\epsilon)^{u-l+d} M'_{h,j}$ *for all* $i, h \in \mathcal{X}$ *and* $j \in \mathcal{Y}$*, where* d *is the* $\sim'$*-distance between* i *and* h*;*

(iii) The elements of the diagonal are all equal to the maximum element of the matrix: $M'_{i,i} = \max^{M'}$ *for all* $i \in \mathcal{X}$*;*

(iv) $H_\infty^{M'}(X|Y) = H_\infty^{M}(X|Y)$*, if* X *has the uniform distribution.*

Proof. We first apply a procedure similar to that of Lemma 37 to construct a square matrix of size $v^\ell \times v^\ell$ which has the maximum values of each column in the diagonal. (In this case we construct an injection from the columns to rows containing their maximum value, and we eliminate the rows that at the end are not associated to any column.) Then define $\sim'$ as the projection of $\sim_u$ on Val^ℓ. It is easy to see that condition (ii) in is satisfied by this definition of $\sim'$. Finally, apply the procedure in Lemma 38, or equivalently the procedure in Lemma 39, on the structure $(\mathcal{X}, \sim')$ to make all elements in the diagonal equal to the maximum element of the matrix (condition (iii)). Note that this procedure preserves the property of condition (ii), and conditional min-entropy ((iv)). Also the matrix obtained is a valid channel matrix (condition (i)). □

Now we are ready to prove the proposition.

Proposition 47. *Let* $\mathcal{K}$ *be a randomized function with associated channel matrix* M*, and let* $r = |Range(\mathcal{K})|$*. If* $\mathcal{K}$ *provides* ϵ-differential privacy *then the min-entropy leakage associated to* $\mathcal{K}$ *is bounded from above as follows:*

$$I_\infty^M(X;Z) \leq \log_2 \frac{r\,(e^\epsilon)^u}{(v-1+e^\epsilon)^\ell - (e^\epsilon)^\ell + (e^\epsilon)^u}$$

where $\ell = \lfloor \log_v r \rfloor$*.*

Proof. Assume first that r is of the form v^ℓ. We transform the matrix M associated to $\mathcal{K}$ by applying Lemma 46, and let M' be the resulting matrix. Let us denote by $\max^{M'}$ the value of every element in the diagonal of M', i.e. $\max^{M'} = M'_{i,i}$ for every row i, and let us denote by $\mathcal{A}'_{\langle d \rangle}(i)$ the set of elements whose $\sim'$-distance from i is d. Note that for every $j \in \mathcal{A}'_{\langle d \rangle}(i)$ we have that $M'_{j,j} \leq M'_{i,j}(e^\epsilon)^{u-\ell+d}$, hence

$$M'_{i,j} \geq \frac{\max^M}{(e^\epsilon)^{u-\ell+d}}$$

Furthermore each element j at $\sim'$-distance d from i can be obtained by changing the value of d individuals in the ℓ-tuple representing i (remember that $(\mathcal{A}, \sim')$ is isomorphic to $(Val^\ell, \sim_\ell)$). We can choose those d individuals in $\binom{\ell}{d}$ possible ways, and for each of these individuals we can change the value (with respect to the one in i) in $v-1$ possible ways. Therefore

$$|\mathcal{A}'_{\langle d \rangle}(i)| = \binom{\ell}{d}(v-1)^d$$

Taking into account that for $M'_{i,i}$ we do not need to divide by $(e^\epsilon)^{u-\ell+d}$, we obtain:

$$\max^M + \sum_{d=1}^{\ell} \binom{\ell}{d}(v-1)^d \frac{\max^M}{(e^\epsilon)^{u-\ell+d}} \leq \sum_j M'_{i,j}$$

Since each row represents a probability distribution, the elements of row i must sum up to 1. Hence:

$$\max^M + \sum_{d=1}^{u} \binom{u}{d}(v-1)^d \frac{\max^M}{(e^\epsilon)^{u-\ell+d}} \leq 1 \tag{5.12}$$

By performing some simple calculations, similar to those of the proof of Theorem 44, we obtain:

$$\max^M \leq \frac{(e^\epsilon)^u}{(v-1+e^\epsilon)^\ell - (e^\epsilon)^\ell + (e^\epsilon)^u}$$

Therefore:

$$I_\infty^{M'}(X;Z) = H_\infty(X) - H_\infty^{M'}(X|Z) \qquad \text{(by definition)} \tag{5.13}$$

$$= \log_2 v^u + \log_2 \sum_{j=1}^{v^\ell} \max^M \tfrac{1}{v^u} \tag{5.14}$$

$$= \log_2 v^u + \log_2 \frac{1}{v^u} + \log_2(v^\ell \max^M) \tag{5.15}$$

$$\leq \log_2 \frac{v^\ell\,(e^\epsilon)^u}{(v-1+e^\epsilon)^\ell - (e^\epsilon)^\ell + (e^\epsilon)^u} \qquad \text{(by (5.12))} \tag{5.16}$$

Consider now the case in which r is not of the form v^ℓ. Let ℓ be the maximum integer such that $v^\ell < r$, and let $m = r - v^\ell$. We transform the matrix M associated to $\mathcal{K}$ by collapsing the m columns with the smallest maxima into the m columns with highest maxima. Namely, let $j_1, j_2, \ldots, j_m$ the indices of the columns which have smallest maxima values, i.e. $\max_{j_t}^M \leq \max_j^M$ for every column $j \neq j_1, j_2, \ldots, j_m$. Similarly, let $k_1, k_2, \ldots, k_m$ be the indexes of the columns which have maxima values. Then, define

$$N = M[j_1 \to k_1][j_2 \to k_2] \ldots [j_m \to k_m]$$

Finally, eliminate the m zero-ed columns to obtain a matrix with exactly v^ℓ columns. It is easy to show that

$$I_\infty^M(X;Z) \leq I_\infty^N(X;Z)\frac{r}{v^\ell}$$

After transforming N into a matrix M' with the same min-entropy leakage as described in the first part of this proof, from (5.13) we conclude

$$I_\infty^M(X;Z) \leq I_\infty^{M'}(X;Z)\frac{r}{v^\ell} \leq \log_2 \frac{r\,(e^\epsilon)^u}{(v-1+e^\epsilon)^\ell - (e^\epsilon)^\ell + (e^\epsilon)^u}$$

□

Note that this bound can be much smaller than the one provided by Theorem 44. For instance, if $r = v$ this bound becomes:

$$\log_2 \frac{v\,(e^\epsilon)^u}{v-1+(e^\epsilon)^u}$$

which for large values of u is much smaller than $Bnd(u, v, \epsilon)$.

Let us clarify that there is no contradiction with the fact that the bound $Bnd(u, v, \epsilon)$ is strict: in fact it is strict when we are free to choose the range, but here we fix the dimension of the range.

5.5.1 Measuring the leakage about an individual

As discussed in Section 5.2, the main goal of differential privacy is not to protect information about the complete database, but about each of its individual participants. To capture the leakage about a particular individual, we start from a tuple $x^- \in Val^{u-1}$ containing the given (and known) values of all other $u-1$ individuals. Then we create a channel whose input V ranges over the values in Val and represents the value of our individual of interest. Note that this means that we take into consideration all possible input databases where the values of the other individuals are exactly those of x^- and only the value of the selected individual varies. Intuitively, $I_\infty^{x^-}(V;Z)$ measures the leakage about the individual's value where all other values are known to be as in x^-. (Similarly, $H_\infty^{x^-}(V|Z)$ represents the conditional entropy of V given Z for a fixed database where all other values are x^-.) As all these databases are adjacent, differential privacy provides a stronger bound for this leakage.

Therefore, the *leakage for a single individual* can be characterized as follows.

Proposition 48. *Assume that $\mathcal{K}$ satisfies ϵ-differential privacy. Then the information leakage for an individual is bound from above by:*

$$I_\infty^{x^-}(V;B) \leq \log_2 \frac{v\,e^\epsilon}{v-1+e^\epsilon}$$

Proof. Let us fix a database x, and a particular individual i in Ind. The possible ways in which we can change the value of i in x are $v-1$. All the new databases obtained in this way are adjacent to each other, i.e. the graph structure associated to the input is a clique of v nodes. Recall that n_d is the

number of elements of the input at distance d from a given element x. In this case we have

$$n_d = \begin{cases} 1 & \text{for } d = 0, \\ v - 1 & \text{for } d = 1, \\ 0 & \text{otherwise.} \end{cases}$$

By substituting this value of n_d in Theorem 42, we get

$$\begin{aligned} H_\infty^{x^-}(V|Z) &\geq -\log_2 \frac{1}{1 + \frac{v-1}{e^\epsilon}} \\ &= -\log_2 \frac{e^\epsilon}{v - 1 + e^\epsilon} \end{aligned}$$

The particular individual can present v different values, and thus in the case the input distribution is uniform its min-entropy is $H_\infty^{x^-}(V) = \log_2 v$.

$$\begin{aligned} I_\infty^{x^-}(V;Z) &= H_\infty^{x^-}(V) - H_\infty^{x^-}(V|Y) && \text{(by definition)} \\ &= \log_2 v + \log_2 \frac{e^\epsilon}{v - 1 + e^\epsilon} && \text{(by the derivations above)} \\ &= \log_2 \frac{v\, e^\epsilon}{v - 1 + e^\epsilon} \end{aligned}$$

Since the min-entropy leakage is maximum in the case of the uniform input distribution, the result follows.

□

Note that the bound on the leakage for an individual does not depend on the size u of *Ind*, nor on the database x^- that we fix.

5.6 Application to utility

As discussed in Section 5.2, the utility of a randomized function $\mathcal{K}$ is the correlation between the real answers Y for a query and the reported answers Z.

For our analysis we assume an oblivious randomization mechanism. As discussed in Section 5.2, in this case the system can be decomposed into the cascade of two channels, and the utility becomes a property of the channel associated to the randomization mechanism $\mathcal{H}$ which maps the real answer $y \in \mathcal{Y}$ into a reported answer $z \in \mathcal{Z}$ according to given probability distributions $p_{Z|Y}(\cdot|\cdot)$. The user, however, does not necessarily take z as her guess for the real answer, since she can use some Bayesian post-processing to maximize the probability of success, i.e. a right guess. Thus for each reported answer z the

user can remap her guess to a value $y' \in \mathcal{Y}$ according to some strategy that maximizes her expected gain.

The standard way to define utility is by means of *gain* functions (see for instance [BS94]). We define $gain : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$ and the value $gain(y, y')$ represents the reward for guessing the answer y' when the correct answer is y.

It is natural to define the global utility of the mechanism $\mathcal{H}$ as the expected gain:

$$\mathcal{U}(Y, Z) = \sum_{y} p(y) \sum_{y'} p(y'|y) gain(y, y') \tag{5.17}$$

where $p(y)$ is the prior probability of real answer y, and $p(y'|y)$ is the probability of the user guessing y' when the real answer is y.

Assuming that the user uses a remapping function $guess : \mathcal{Z} \to \mathcal{Y}$, we can derive the following characterization of the utility. Recall that $\delta_x(\cdot)$ represents the probability distribution which has value 1 on x and 0 elsewhere.

$$\begin{aligned}
\mathcal{U}(Y, Z) &= \sum_{y} p(y) \sum_{y'} p(y'|y) gain(y, y') && \text{(by (5.17))} \\
&= \sum_{y} p(y) \sum_{y'} \left(\sum_{z} p(z|y) p(y'|z) \right) gain(y, y') \\
&= \sum_{y} p(y) \sum_{y'} \left(\sum_{z} p(z|y) \delta_{y'}(guess(z)) \right) gain(y, y') && (y' = guess(z)) \\
&= \sum_{y} p(y) \sum_{z} p(z|y) \sum_{y'} \delta_{y'}(guess(z)) gain(y, y') \\
&= \sum_{y,z} p(y, z) \sum_{y'} \delta_{y'}(guess(z)) gain(y, y') \\
&= \sum_{y,z} p(y, z) gain(y, guess(z)) && (5.18)
\end{aligned}$$

We focus here on the so-called *binary* gain function, which is defined as

$$gain_{bin}(y, y') = \begin{cases} 1 & \text{if } y = y', \\ 0 & \text{otherwise.} \end{cases}$$

Note that in the above equation the value y' represents the user's guess after the observed answer z. Therefore we have

$$gain_{bin} = \delta_y(guess(z))$$

This kind of function represents the case in which there is no reason to prefer one answer over another, except if it is the *correct* answer. More precisely, we obtain some gain if and only if we guess the right answer. Note that if the

answer domain is equipped with a notion of distance (i.e. even if two answers are wrong, one of them may be "closer" to the correct one than the other) then the gain function could take into account the proximity of the reported answer to the real one. In this case a "close" answer, even if wrong, is considered better than a distant one. We do not assume here a notion of distance, and therefore we will focus on the binary case. The use of binary gain functions in the context of differential privacy was also investigated in [GRS09][5].

By substituting *gain* with $gain_{bin}$ in (5.18) we obtain:

$$\mathcal{U}(Y,Z) = \sum_{y,z} p(y,z)\delta_y(guess(z)) \tag{5.19}$$

which tells us that the expected utility is the greatest when $guess(z) = y$ is chosen to maximize $p(y,z)$. Assuming that the user chooses such a maximizing remapping, we have:

$$\begin{aligned}\mathcal{U}(Y,Z) &= \sum_z \max_y p(y,z) \\ &= \sum_z \max_y (p(y)\,p(z|y)) \qquad \text{(by the Bayes law)}\end{aligned} \tag{5.20}$$

If the gain function is binary, and the function *guess* is chosen to optimize utility (i.e. it represents the user's best strategy), then there is a well-known correspondence between $\mathcal{U}$ and the Bayes risk / the a posteriori min-entropy. This correspondence is expressed by the following proposition:

Proposition 49. *Assume that function gain is binary and the function guess is optimal. Then:*

$$\mathcal{U}(Y,Z) = \sum_z \max_y (p(y)\,p(z|y)) = 2^{-H_\infty(Y|Z)}$$

Proof. Just substitute (5.20) in the definition of conditional min-entropy: $H_\infty(Z \mid Y) = -\log_2 \sum_z \max_y((p(y)\,p(z|y))$. □

5.6.1 The bound on the utility

In this section we show that,in some special cases, the fact that $\mathcal{K}$ provides ϵ-differential privacy induces a bound on the utility as defined in terms of a binary gain function. We start by extending the adjacency relation $\sim$ from the datasets $\mathcal{X}$ to the real answers $\mathcal{Y}$, in such a way that two values in $\mathcal{Y}$ are adjacent if they have pre-images that are adjacent. Intuitively, the function f associated to the query determines a partition on the set of all databases ($\mathcal{X}$, i.e. Val^u), and we say that two classes are adjacent if they contain an adjacent pair. More formally:

[5] The authors of [GRS09] used the dual notion of *loss functions* instead of gain functions, but the final result is equivalent.

Definition 50. *Given $y, y' \in \mathcal{Y}$, with $y \neq y'$, we say that y and y' are adjacent (notation $y \sim y'$), if and only if there exist $x, x' \in Val^u$ with $x \sim x'$ such that $y = f(x)$ and $y' = f(x')$.*

Since $\sim$ is symmetric on databases, it is also symmetric on $\mathcal{Y}$, therefore also $(\mathcal{Y}, \sim)$ forms an undirected graph.

Using the above concept of neighborhood for the inputs of the randomization mechanism $\mathcal{H}$, we can show that in an oblivious mechanisms (see Figure 5.2) if the query f is deterministic, then the randomized function $\mathcal{K}$ provides ϵ-differential privacy with respect to neighbor databases if and only if $\mathcal{H}$ respects ϵ-differential privacy with respect to neighbor answers. Intuitively, this result follows from the fact that a deterministic query f remaps every database $x \in \mathcal{X}$ to a sole answer $y \in \mathcal{Y}$, working as a sort of "relabeling" that substitutes databases for answers in the adjacency graph structure, and therefore preserving ϵ-differential privacy. Note also that if $\mathcal{K}$ is oblivious, the probability of any reported answer $z \in \mathcal{Z}$ does not depend on the database, but solely on the real answer y. Therefore under a deterministic f, two databases x and x' can be mapped to same value of y only if, for all z, $\mathcal{K}(z|x) = \mathcal{K}(z|x')$.

Proposition 51. *If the query function f is deterministic, then the randomized function $\mathcal{K}$ satisfies ϵ-differential privacy with respect to every pair of neighbor databases $x, x' \in \mathcal{X}$ if and only if the randomization mechanism $\mathcal{H}$ satisfies ϵ-differential privacy with respect to every pair of neighbor answers $y, y' \in \mathcal{Y}$.*

Proof. Since the matrix $\mathcal{K}$ can be obtained by the product of the two matrices corresponding to f and $\mathcal{H}$, we can derive that, for every pair of neighbor databases x and x' and for all reported answer z:

$$
\begin{aligned}
\frac{\mathcal{K}(z|x)}{\mathcal{K}(z|x')} &= \frac{Pr[Z = z|X = x]}{Pr[Z = z|X = x']} \\
&= \frac{\sum_y Pr[Y = y|X = x]Pr[Z = z|Y = y]}{\sum_y Pr[Y = y|X = x']Pr[Z = z|Y = y]} && \text{(matrix multiplication)} \\
&= \frac{\sum_y \delta_{f(x)}(y)Pr[Z = z|Y = y]}{\sum_y \delta_{f(x')}(y)Pr[Z = z|Y = y]} && \text{(since } f \text{ is deterministic)} \\
&= \frac{Pr[Z = z|Y = f(x)]}{Pr[Z = z|Y = f(x')]} && \text{(applying the Dirac } \delta\text{)} \\
&= \frac{\mathcal{H}(z|f(x)]}{\mathcal{H}(z|f(x')]}
\end{aligned}
$$

Therefore it follows immediately that $\frac{\mathcal{K}(z|x)}{\mathcal{K}(z|x')} \leq e^\epsilon$ if and only if $\frac{\mathcal{H}(z|f(x))}{\mathcal{H}(z|f(x'))} \leq e^\epsilon$.

□

The link the above proposition establishes between the randomized function $\mathcal{K}$ and the randomization mechanism $\mathcal{H}$ will help us find determine a bound on the utility of $\mathcal{H}$, since, in the case the query f is deterministic, requiring $\mathcal{K}$ to respect ϵ-differential privacy is equivalent to requiring that $\mathcal{H}$ does.

Theorem 52. *Consider a randomized mechanism $\mathcal{H}$, and let y be an element of $\mathcal{Y}$. Assume that the distribution of Y is uniform and that $(\mathcal{Y}, \sim)$ is either distance-regular or VT^+ and that $\mathcal{H}$ satisfies ϵ-differential privacy. For each distance $d \in \{0, 1, \ldots, \delta\}$, where δ is the diameter of $(\mathcal{Y}, \sim)$, we have that:*

$$\mathcal{U}(Y, Z) \leq \frac{1}{\sum_d \frac{n_d}{e^{\epsilon\, d}}} \tag{5.21}$$

where n_d is the number of nodes $y' \in \mathcal{Y}$ at distance d from y.

Proof. Since $(\mathcal{Y}, \sim)$ is distance-regular or VT^+, we can apply Theorem 42 to derive that $H_\infty^M(Z|Y) \geq -\log_2 \frac{1}{\sum_d \frac{n_d}{e^{\epsilon\, d}}}$. Then we just substitute this result in Proposition 49. □

The above bound is tight, in the sense that (provided $(\mathcal{Y}, \sim)$ is distance-regular or VT^+) we can construct a mechanism $\mathcal{H}$ which satisfies (5.21) with equality. More precisely, for $0 \leq i \leq n-1$ and $0 \leq j \leq n-1$, we define $\mathcal{H}$ (here identified with its channel matrix for simplicity) as follows:

$$\mathcal{H}_{i,j} = \frac{\gamma}{e^{\epsilon\, d(i,j)}} \tag{5.22}$$

where

$$\gamma = \frac{1}{\sum_d \frac{n_d}{e^{\epsilon\, d}}} \tag{5.23}$$

Note that $\mathcal{H}$ is a square matrix of dimension $n \times n$, where $n = |\mathcal{X}|$. This is not a problem because since we assume $(\mathcal{Y}, \sim)$ to be either distance-regular or VT^+, via Theorem 42 we can transform the channel matrix into an equivalent one such that all non zero elements are in the submatrix of dimensions $n \times n$. Let us introduce now $\mathcal{Z}^* = \{0, 1, \ldots, n-1\}$, i.e. the subset of $\mathcal{Z}$ that excludes the zero-ed columns of the channel matrix from n to $m-1$. Note that for the following result we can safely use the set $\mathcal{Z}^*$ instead of $\mathcal{Z}$ because the zero-ed columns do not contribute to the a posteriori entropy, and trivially respect ϵ-differential privacy.

Theorem 53. *Assume $(\mathcal{Y}, \sim)$ is distance-regular or VT^+ and that the distribution of Y is uniform. Then the matrix $\mathcal{H}$ defined in (5.22) satisfies ϵ-differential privacy and has maximal utility:*

$$\mathcal{U}(Y, Z) = \frac{1}{\sum_d \frac{n_d}{e^{\epsilon\, d}}}$$

Proof. First we prove that the matrix as defined in (5.22) is a channel matrix, i.e. that each row is a probability distribution.

$$\begin{aligned}
\sum_{j \in \mathcal{Z}^*} \mathcal{H}_{i,j} &= \sum_{j \in \mathcal{Z}^*} \frac{\gamma}{e^{\epsilon d(i,j)}} \\
&= \gamma \sum_{j \in \mathcal{Z}^*} \frac{1}{e^{\epsilon d(i,j)}} \\
&= \gamma \sum_d \frac{n_d}{e^{\epsilon d}} && \text{by (5.23)} \\
&= \gamma \frac{1}{\gamma} \\
&= 1
\end{aligned}$$

Now we show that the utility is maximum.

$$\begin{aligned}
\mathcal{U}(Y, Z) &= \sum_{z \in \mathcal{Z}^*} \max_y (p(y)\, \mathcal{H}(z|y)) && \text{by (5.20)} \\
&= \sum_{z \in \mathcal{Z}^*} \max_y \frac{1}{|\mathcal{Y}|} \mathcal{H}(z|y) && \text{since } Y \text{ is uniform} \\
&= \frac{1}{|\mathcal{Y}|} \sum_{z \in \mathcal{Z}^*} \max_y \frac{\gamma}{\max_d e^{\epsilon d(i,j)}} && \text{by (5.22)} \\
&= \frac{1}{|\mathcal{Y}|} \sum_{z \in \mathcal{Z}^*} \gamma && \text{maximum is } d = 0 \\
&= \frac{1}{|\mathcal{Y}|} \cdot |\mathcal{Z}^*| \gamma \\
&= \gamma && \text{since } |\mathcal{Y}| = |\mathcal{Z}^*| = n
\end{aligned}$$

□

Therefore we can always define $\mathcal{H}$ as in (5.22): the matrix so defined will be a legal channel matrix, and it will satisfy ϵ-differential privacy. If $(\mathcal{Y}, \sim)$ is neither distance-regular nor VT^+, then the utility of such $\mathcal{H}$ is not necessarily optimal.

The conditions for the construction of the optimal matrix are strong, but there are some interesting scenarios in which they are satisfied. Depending on

the degree of connectivity c of the graph $(\mathcal{Y}, \sim)$, we can have $\lfloor \frac{|\mathcal{Y}|}{2} \rfloor - 1$ different cases (note that the case of $c = 1$ is not possible because the datasets are fully connected via their adjacency relation), whose extremes are:

- $(\mathcal{Y}, \sim)$ is a *clique*, i.e. every element has exactly $|\mathcal{Y}|-1$ adjacent elements.
- $(\mathcal{Y}, \sim)$ is a *ring*, i.e. every element has exactly two adjacent elements. This is similar to the case of the counting queries considered in [GRS09], with the difference that our "counting" is in arithmetic modulo $|\mathcal{Y}|$.

Remark 54. *Note that our method can be applied also when the conditions of Theorem 53 are not met: We can always add "artificial" adjacencies to the graph structure so as to meet those conditions. Namely, for computing the distance in (5.22) we use, instead of $(\mathcal{Y}, \sim)$, a structure $(\mathcal{Y}, \sim')$ which satisfies the conditions of Theorem 53, and such that $\sim \subseteq \sim'$. Naturally, the matrix constructed in this way provides ϵ-differential privacy, but in general is not optimal. It is clear that, in general, the smaller $\sim'$ is, the higher is the utility.*

The matrices generated by (5.22) can be very different, depending on the value of c. The next two examples illustrate queries that give rise to the clique and to the ring structures, and show the corresponding matrices.

Example 8. *Consider a database with electoral information where each entry corresponds to a voter and contains the following three fields:*

- Id*: a unique (anonymized) identifier assigned to each voter;*
- City*: the name of the city where the user voted;*
- Candidate*: the name of the candidate the user voted for.*

Consider the query "What is the city with the greatest number of votes for a given candidate *cand*?". *For such a query the binary utility function could be taken as the natural choice: from the user's point of view, only the right city could give some gain, and all wrong answers would be equally bad. It is easy to see that every two answers are neighbors, i.e. the graph structure of the answers is a clique.*

Let us consider the scenario where City $= \{A, B, C, D, E, F\}$ *and assume for simplicity that there is a unique answer for the query, i.e. there are no two cities with exactly the same number of individuals voting for candidate cand. Table 5.1 shows two alternative mechanisms providing ϵ-differential privacy (with $\epsilon = \ln 2$). The first one, M_1, is based on the truncated geometric mechanism method used in [GRS09] for counting queries (here extended to the case where every two distinct answers are neighbors). The second mechanism, M_2, is obtained by applying the definition of (5.22). From Theorem 53 we know that for the uniform input distribution M_2 gives optimal utility.*

In/Out	A	B	C	D	E	F
A	0.535	0.060	0.052	0.046	0.040	0.267
B	0.465	0.069	0.060	0.053	0.046	0.307
C	0.405	0.060	0.069	0.060	0.053	0.353
D	0.353	0.053	0.060	0.069	0.060	0.405
E	0.307	0.046	0.053	0.060	0.069	0.465
F	0.267	0.040	0.046	0.052	0.060	0.535

(a) M_1: truncated geometric mechanism

In/Out	A	B	C	D	E	F
A	2/7	1/7	1/7	1/7	1/7	1/7
B	1/7	2/7	1/7	1/7	1/7	1/7
C	1/7	1/7	2/7	1/7	1/7	1/7
D	1/7	1/7	1/7	2/7	1/7	1/7
E	1/7	1/7	1/7	1/7	2/7	1/7
F	1/7	1/7	1/7	1/7	1/7	2/7

(b) M_2: our mechanism

Table 5.1: Mechanisms for the city with higher number of votes for candidate *cand*

For the uniform input distribution, it is easy to see that $\mathcal{U}(M_1) = 0.2242 < 0.2857 = \mathcal{U}(M_2)$. *Even for non-uniform distributions, our mechanism still provides better utility. For instance, for* $p(A) = p(F) = 1/10$ *and* $p(B) = p(C) = p(D) = P(E) = 1/5$, *we have* $\mathcal{U}(M_1) = 0.2412 < 0.2857 = \mathcal{U}(M_2)$. *This is not too surprising: the geometric mechanism, as well as the Laplacian mechanism proposed by Dwork, perform very well when the domain of answers is provided with a metric and the utility function is not binary*[6]. *It also works well when* $(\mathcal{Y}, \sim)$ *has low connectivity, in particular in the cases of a ring and of a line. But in this example, we are not in these cases, because we are considering* binary gain functions *and* high connectivity.

Example 9. *Let us consider the same database as the previous example, but now assume a counting query of the form* "What is the number of votes for candidate *cand*?". *It is easy to see that each answer has at most two neighbors. More precisely, the graph structure on the answers is a line. For illustration purposes, let us assume that only* 5 *individuals have participated in the election. Table 5.2 shows two alternative mechanisms providing* ϵ*-differential privacy* ($\epsilon = \log 2$)*: the truncated geometric mechanism* M_1 *proposed in [GRS09] and the mechanism we propose* M_2. *Note that in order to apply our method we have first to apply Remark 54 to transform the graph structure from a line into a ring.*

[6] As we mentioned before, in the metric case the gain function can take into account the proximity of the reported answer to the real one, the idea being that a close answer, even if wrong, is better than a distant one.

Let us consider the uniform prior distribution. We see that the utility of M_1 is higher than the utility of M_2, in fact the first is 4/9 and the second is 8/21. This does not contradict our theorem, because our matrix is guaranteed to be optimal only in the case of a ring structure, not a line as we have in this example. If the structure were a ring, i.e. if the last row were adjacent to the first one, then M_1 would not provide ϵ-differential privacy. In case of a line as in this example, the truncated geometric mechanism has been proved optimal [GRS09].

In/Out	0	1	2	3	4	5
0	2/3	1/6	1/12	1/24	1/48	1/48
1	1/3	1/3	1/6	1/12	1/24	1/24
2	1/6	1/6	1/3	1/6	1/12	1/12
3	1/12	1/12	1/6	1/3	1/6	1/6
4	1/24	1/24	1/12	1/6	1/3	1/3
5	1/48	1/48	1/24	1/12	1/6	2/3

(a) M_1: truncated $\frac{1}{2}$-geom. mechanism

In/Out	0	1	2	3	4	5
0	8/21	4/21	2/21	1/21	2/21	4/21
1	4/21	8/21	4/21	2/21	1/21	2/21
2	2/21	4/21	8/21	4/21	2/21	1/21
3	1/21	2/21	4/21	8/21	4/21	2/21
8	2/21	1/21	2/21	4/21	8/21	4/21
5	4/21	2/21	1/21	2/21	4/21	8/21

(b) M_2: our mechanism

Table 5.2: Mechanisms for the counting query (5 voters)

5.7 Related work

To the best of our knowledge, the first work to investigate the relation between differential privacy and information-theoretic leakage *for an individual* was [ACDP10]. In this work, the definition of channel was relative to a given database x, and the channel inputs were all possible databases adjacent to x. Two bounds on leakage were presented, one for the min-entropy, and one for Shannon entropy. Our bound in Proposition 48 is an improvement with respect to the (min-entropy) bound in [ACDP10].

Barthe and Köpf [BK11] were the first to investigate the (more challenging) connection between differential privacy and the min-entropy leakage *for the entire universe of possible databases.* They considered the "end-to-end differentially private mechanisms", which correspond to what we call the randomized function $\mathcal{K}$ in this chapter, and proposed, like we do, to interpret them as information-theoretic channels. They provided a bound for the leakage, but

pointed out that it was not tight in general. They also showed that there cannot be a domain-independent bound, by proving that for any number of individuals u the optimal bound must be at least a certain expression $f(u, \epsilon)$. Finally, they showed that the question of providing optimal upper bounds for the leakage of ϵ-differentially private randomized functions in terms of rational functions of ϵ is decidable, and left the actual function as an open question. In our work we used rather different techniques and found (independently) the same function $f(u, \epsilon)$ (the bound in Theorem 42), but we actually proved that $f(u, \epsilon)$ is the optimal bound[7]. Another difference between their work and ours is that [BK11] captures the case in which the focus of differential privacy is on hiding *participation* of individuals in a database, whereas we consider both the participation and the *values* of the participants.

Clarkson and Schneider also considered differential privacy as a case study of their proposal for quantification of integrity [CS11]. There, the authors analyzed database privacy conditions from the literature (such as differential privacy, k-anonymity, and l-diversity) using their framework for utility quantification. In particular, they studied the relationship between differential privacy and a notion of leakage (which is different from ours - in particular their definition is based on Shannon entropy) and they provided a tight bound on leakage.

Heusser and Malacaria [HM09] were among the first to explore the application of information-theoretic concepts to databases queries. They proposed to model database queries as programs, which allows for statistical analysis of the information leaked by the query. [HM09], however, did not attempt to relate information leakage to differential privacy.

In [GRS09] the authors aimed at obtaining optimal-utility randomization mechanisms while preserving differential privacy. The authors proposed adding noise to the output of the query according to the geometric mechanism. Their framework is very interesting in the sense it provides a general definition of utility for a mechanism M that captures any possible side information and preference (defined as a loss function) the users of M may have. They proved that the geometric mechanism is optimal in the particular case of counting queries. Our results in Section 5.6 do not restrict to counting queries, but on the other hand we only consider the case of binary loss function.

5.8 Chapter summary and discussion

In this chapter we have investigated the relation between ϵ-differential privacy and leakage, and between ϵ-differential privacy and utility. Our main contribution was the development of a general technique for determining these relations depending on the graph structure of the input domain, induced by

[7] When discussing our result with Barthe and Köpf, they said that they also conjectured that $f(u, \epsilon)$ is the optimal bound.

the adjacency relation and by the query. We have considered two particular structures, the distance-regular graphs, and the VT^+ graphs, which allowed us to obtain tight bounds on the leakage and on the utility. We also constructed an optimal randomization mechanism satisfying ϵ-differential privacy for some special cases.

As future work, we plan to extend our result to other kinds of utility functions. In particular, we are interested in the case in which the the answer domain is provided with a metric, and we are interested in taking into account the degree of accuracy of the inferred answer.

Six

Safe equivalences for security properties

"Too much may be the equivalent of none at all."
Lee Loevinger

In the field of Security, process equivalences have been used to characterize various information-hiding properties (for instance secrecy, anonymity and noninterference) based on the principle that a protocol P with a variable x satisfies such a property if and only if, for every pair of secrets s_1 and s_2, $P[^{s_1}/_x]$ is equivalent to $P[^{s_2}/_x]$. We argue that, in the presence of nondeterminism, the above principle may rely on the assumption that the scheduler "works for the benefit of the protocol", and this usually is not a safe assumption. Non-safe equivalences, in this sense, include complete-trace equivalence and bisimulation.

The goal of this chapter is to present a formalism in which we can specify admissible schedulers and, correspondingly, safe versions of these equivalences. Then we are able to show that safe equivalences can be used to establish information-hiding properties.

Contribution The main contributions of this chapter can be summarized as follows.

- We propose a formalism for concurrent distributed systems which accounts for both probabilistic and nondeterministic behavior, and in which the latter is of two kinds: *global* and *local*. The global nondeterminism represents the possible interleavings produced by the parallel components, which may be influenced by the attacker. The local nondeterminism is associated to the possible internal choices of each component,

which may depend on the secrets or other unknown parameters, not controlled by the attacker. Correspondingly, we split the scheduler into two constituents: a global one and a local one. The latter is actually a tuple of local schedulers, one for each component of the system.

- We propose a notion of *admissible scheduler* for the above systems, in which the global constituent is not allowed to see the secrets, and each local constituent is not allowed to see any information about the other components. We then generalize the standard definition of strong (probabilistic) information hiding (such as noninterference and strong anonymity) to the case in which also nondeterminism is present, under the assumption that the schedulers are admissible.

- We use admissible schedulers to define safe versions of complete-trace[1] equivalence and bisimilarity which are specially tuned for security. This means that we account for the possibility that the global constituent of the scheduler is in collusion with the attacker, and therefore does not necessarily help the system to obfuscate the secret. We show that the bisimilarity is still a congruence, as in the classical case.

- We finally show that our notions of safe complete-trace equivalence and bisimilarity imply strong information hiding in the sense discussed above.

Plan of the Chapter This chapter is organized as follows. In Section 6.1 we review the role equivalences traditionally play in formalizing security properties. In Section 6.2 we formalize the notions of distributed systems and components used in this chapter. In Section 6.3 we focus on restricting the discerning power of global and local schedulers, and in Section 6.4 we present our proposal for safe equivalences, namely safe complete-traces and safe bisimilarity. In Section 6.5 we define the notion of information hiding under the novel assumption that nondeterminism is handled partly in a demonic way and partly in an angelic way. Finally, in Section 6.6 we review the related bibliography, and in Section 6.7 we summarize the chapter and outline some future work.

6.1 The use of equivalences in security

As we have seen in Chapter 1, one technique used to prevent an attacker of inferring the secret from the observables is to create *noise*, namely to make sure that for every execution in which a given secret produces a certain observable, there is at least another execution in which a different secret produces the same observable. In practice this is often done by using randomization.

[1] In this chapter we may refer to "complete traces" simply as "traces".

In the literature about the foundations of computer security, however, the quantitative aspects are often abstracted away, and probabilistic behavior is replaced by nondeterministic behavior. Correspondingly, there have been various approaches in which information-hiding properties are expressed in terms of equivalences based on nondeterminism, especially in a concurrent setting. For instance, [SS96] defines *anonymity* as follows[2]: A protocol S is anonymous if, for every pair of culprits a and b, $S[^a/_x]$ and $S[^b/_x]$ produce the same observable traces. A similar definition is given in [AG99] for *secrecy*, with the difference that $S[^a/_x]$ and $S[^b/_x]$ are required to be bisimilar. In [DKR09], an electoral system S preserves the *confidentiality of the vote* if for any voters v and w, the observable behavior of S is the same if we swap the votes of v and w, i.e. if $S[^a/_v \mid^b /_w]$ is bisimilar to $S[^b/_v \mid^a /_w]$.

These proposals are based on the implicit assumption that *all the nondeterministic executions present in the specification of S will always be possible under every implementation of S*. Or at least, that the adversary will believe so. In concurrency, however, as argued in [CNP09], nondeterminism has a rather different meaning: if a specification S contains some nondeterministic alternatives, typically it is because we want to abstract from specific implementations, such as the scheduling policy. A specification is considered correct, with respect to some property, if every alternative satisfies the property. Correspondingly, an implementation is considered correct if all executions are among those possible in the specification, i.e. if the implementation is a refinement of the specification. There is no expectation that the implementation will actually make possible all the alternatives indicated by the specification.

We argue that the use of nondeterminism in concurrency corresponds to a *demonic* view: the scheduler, i.e. the entity that will decide which alternative to select, may try to choose the "worst" alternative. Hence we need to make sure that all alternatives are "good", in the sense that they satisfy the intended property. In the approaches to formalize security properties mentioned above, on the contrary, the interpretation of nondeterminism is *angelic*: the scheduler is expected to actually help the protocol to confuse the adversary and thus protect the secret information.

There is another issue, orthogonal to the angelic/demonic dichotomy, but relevant for the achievement of security properties: the scheduler *should not be able to make its choices dependent on the secret*, or else nearly every protocol would be insecure, i.e. the scheduler would always be able to leak the secret to an external observer (for instance by producing different interleavings of the observables, depending on the secret). This remark has been made several times already, and several approaches have been proposed to cope with the problem of full-information schedulers (aka almighty, omniscient, clairvoyant, etc.), see for example [CCK+06a, CCK+06b, CP, CNP09, APvRS].

The risk of a naive use of nondeterminism to specify a security property is

[2]The actual definition of [SS96] is more complicated, but the spirit is the same.

not only that it may rely on an implicit assumption that the scheduler behaves angelically, but also that it is clairvoyant (fully-informed), i.e. that it peeks at the secrets (that it is not supposed to be able to see) to achieve its angelic strategy.

Example 10. *Consider the following system, presented in a CCS-like syntax:* $S \stackrel{def}{=} (c)(A \parallel H_1 \parallel H_2 \parallel Corr)$, *with* $A \stackrel{def}{=} \overline{c}\langle sec\rangle$, $H_1 \stackrel{def}{=} c(s).\overline{out}\langle a\rangle$, $H_2 \stackrel{def}{=} c(s).\overline{out}\langle b\rangle$, $Corr \stackrel{def}{=} c(s).\overline{out}\langle s\rangle$. *The name sec represents a secret.*

It is easy to see that we have $S\,[^a/_{sec}] \sim S\,[^b/_{sec}]$, *as shown in the execution tress in Figure 6.1. Note that, in order to simulate the rightmost branch in* $S\,[^a/_{sec}]$, *the process* $S\,[^b/_{sec}]$ *needs to follow its leftmost branch. Vice-versa, in order to simulate the rightmost branch in* $S\,[^b/_{sec}]$, *the process* $S\,[^a/_{sec}]$ *needs to follow its middle branch. This means that, in order to achieve bisimulation, the scheduler needs to know the secret, and change its choice accordingly.*

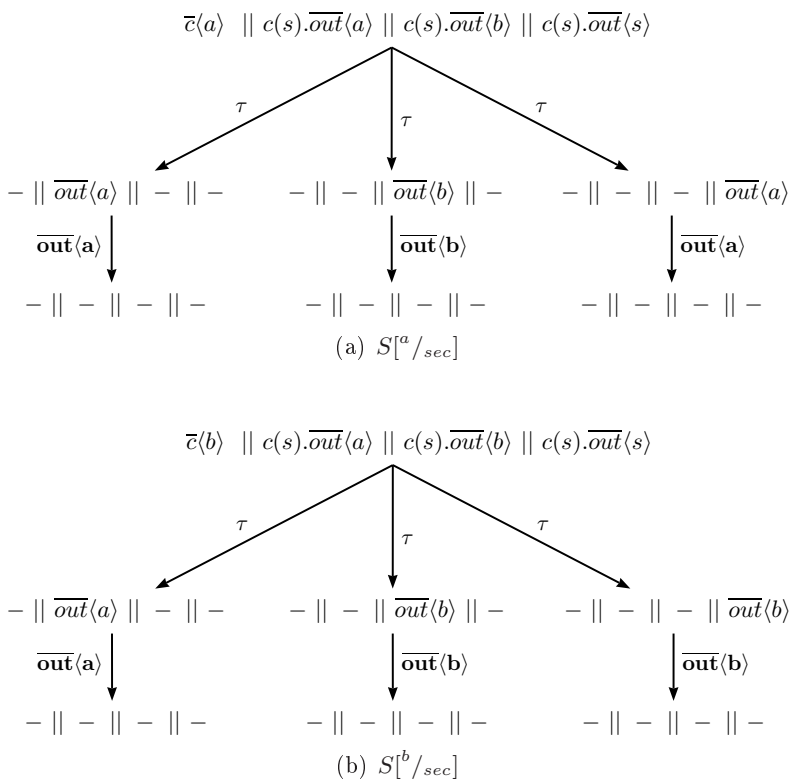


Figure 6.1: Execution trees for Example 10

This example shows a distributed system that intuitively is not secure, because one of its components, *Corr*, reveals whatever secret it receives. According to the equivalence-based notions of security discussed above, however, *it is secure*. But it is considered secure thanks to a scheduler that:

(i) angelically helps the system to protect the secret; and

(ii) does so by making its choices dependent on the secret.

We consider these assumptions on the scheduler to be excessively strong.

Here we do not claim, however, that we should rule out the use of angelic nondeterminism in security: on the contrary, angelic nondeterminism can be a powerful specification concept. We only advocate a cautious use of this notion. In particular, it should not be used in a context in which the scheduler may be in collusion with the attacker. The goal of this chapter is to define a framework in which we can combine both angelic and demonic nondeterminism in a setting in which also probabilistic behavior may be present, and in a context in which the scheduler is restricted (i.e. not fully-informed). We define "safe" variant of typical equivalence relations (complete traces and bisimulation), and we show how to use them to characterize information-hiding properties.

6.2 Distributed systems and components

In this section we describe the kind of distributed systems we are dealing with. We start by introducing a variant of probabilistic automata, that we call *Tagged Probabilistic Automata* (TPA). These systems are parallel compositions of probabilistic processes, called *components*. Each component is equipped with a unique identifier, called *tag*. Whenever a component (or a pair of components in case of synchronization) makes a step, the corresponding transition will be decorated with the associated tag (or pair of tags).

Similar systems have been already introduced in [APvRS]. The main differences are that here the components may contain nondeterminism

6.2.1 Tagged Probabilistic Automata

We now formalize the notion of TPA.

Definition 55. *A* Tagged Probabilistic Automaton *(or* TPA*) is a tuple* $(\mathcal{Q}, \mathcal{T}, \mathcal{L}, \hat{q}, \vartheta)$*, where* $\mathcal{Q}$ *is a set of* states*,* $\mathcal{T}$ *is a set of* tags*,* $\mathcal{L}$ *is a set of* actions*,* $\hat{q} \in \mathcal{Q}$ *is the* initial state*, and* $\vartheta \colon \mathcal{Q} \to \mathcal{P}(\mathcal{T} \times \mathcal{L} \times \mathcal{D}(\mathcal{Q}))$ *is a* transition function*.*

In the following we write $q \xrightarrow{t_g:a} \mu$ for $(t_g, a, \mu) \in \vartheta(q)$, and we use $enab(q)$ to denote the tags of the components that are enabled to make a transition. More formally:

$$enab(q) \stackrel{\text{def}}{=} \{t_g \in \mathcal{T} \mid \textit{there exists } a \in \mathcal{L}, \mu \in \mathcal{D}(\mathcal{Q}) \textit{ such that } q \xrightarrow{t_g:a} \mu\}$$

In these systems, we can decompose the scheduler into two: a *global scheduler*, which, via tags, decides which component or pair of components makes the

next move, and a *local scheduler*, which, also via tags, solves the internal nondeterminism of the selected component.

We assume that the local scheduler can only select enabled transitions, and that the global scheduler can only select enabled components. This means that the execution does not stop unless all components are blocked. This is in line with the tradition of process algebra and of Markov Decision Processes, but contrasts with that of Probabilistic Automata [SL95]. The results in this chapter, however, do not depend on this assumption.

Definition 56. *Let $M = (\mathcal{Q}, \mathcal{T}, \mathcal{L}, \hat{q}, \vartheta)$ be a TPA. Then:*

- *A global scheduler for M is a function $\zeta\colon \mathrm{Paths}^\star(M) \to (\mathcal{T} \cup \{\bot\})$ such that for all finite paths σ, if $enab(last(\sigma)) \neq \emptyset$ then $\zeta(\sigma) \in enab(last(\sigma))$, and $\zeta(\sigma) = \bot$ otherwise.*
- *A local scheduler for M is a function $\xi\colon \mathrm{Paths}^\star(M) \to (\mathcal{T} \times \mathcal{L} \times \mathcal{D}(\mathcal{Q}) \cup \{\bot\})$ such that, for all finite paths σ, if $\vartheta(last(\sigma)) \neq \emptyset$ then $\xi(\sigma) \in \vartheta(last(\sigma))$, and $\xi(\sigma) = \bot$ otherwise.*
- *A global scheduler ζ and a local scheduler ξ for M are* compatible *if, for all finite paths σ, $\xi(\sigma) = (t_g, a, \mu)$ implies $\zeta(\sigma) = t_g$, and $\xi(\sigma) = \bot$ implies $\zeta(\sigma) = \bot$.*
- *A scheduler is a pair (ζ, ξ) of compatible global and local schedulers.*

6.2.2 Components

We will use a simple probabilistic process calculus, very close to the CCS_p we introduced in Chapter 2, to specify the components.

We assume a set of *actions* or *channel names* $\mathcal{L}$ with elements $a, a_1, a_2, \cdots$, including the special symbol τ denoting a *silent step*. Except for τ, each action a has a co-action $\bar{a} \in \mathcal{L}$ and we assume $\bar{\bar{a}} = a$. Components are specified by the following grammar:

$$q \quad ::= \quad 0 \quad | \quad a.q \quad | \quad q_1 + q_2 \quad | \quad \sum_i p_i : q_i \quad | \quad q_1 | q_2 \quad | \quad (a)q \quad | \quad Q$$

The constructs 0, $a.q$, $q_1 + q_2$, $q_1|q_2$ and $(a)q$ represent termination, prefixing, nondeterministic choice, parallel composition, and the restriction operator, respectively. $\sum_i p_i : q_i$ is a probabilistic choice, where p_i represents the probability of the i-th branch and must satisfy $0 \leq p_i \leq 1$ and $\sum_i p_i = 1$. The process call Q is a simple process identifier. For each identifier, we assume a corresponding unique process declaration of the form $Q \stackrel{def}{=} q$. The idea is that, whenever Q is executed, it triggers the execution of q. Note that q can contain Q or another process identifier, which means that our language allows (mutual) recursion. We will denote by $fn(q)$ the *free channel names* occurring in q, i.e. the channel names not bound by a restriction operator.

Components' semantics: The operational semantics consists of probabilistic transitions of the form $q \xrightarrow{a} \mu$ where $q \in \mathcal{Q}$ is a process, $a \in \mathcal{L}$ is an action and $\mu \in \mathcal{D}(\mathcal{Q})$ is a distribution on processes. They are specified by the following rules:

$$\text{PRF}\quad \frac{}{a.q \xrightarrow{a} \delta_q} \qquad \text{NDT}\quad \frac{q_1 \xrightarrow{a} \mu}{q_1 + q_2 \xrightarrow{a} \mu}$$

$$\text{PRB}\quad \frac{}{\sum_i p_i : q_i \xrightarrow{\tau} \sum\!\!\!\!\!\!\int_i p_i \cdot \delta_{q_i}} \qquad \text{PAR}\quad \frac{q_1 \xrightarrow{a} \mu}{q_1 \mid q_2 \xrightarrow{a} \mu \mid q_2}$$

$$\text{CALL}\quad \frac{q \xrightarrow{a} \mu}{A \xrightarrow{a} \mu}\ \text{if } A \stackrel{\text{def}}{=} q \qquad \text{COM}\quad \frac{q_1 \xrightarrow{a} \delta_{r_1} \quad q_2 \xrightarrow{\bar{a}} \delta_{r_2}}{q_1 \mid q_2 \xrightarrow{\tau} \delta_{r_1 \mid r_2}}$$

$$\text{RST}\quad \frac{q \xrightarrow{a} \mu}{(b)q \xrightarrow{a} (b)\mu}\ {\scriptstyle a,\bar{a} \neq b}$$

We assume also the symmetric versions of the rules NDT, PAR and COM. Recall that the symbol δ_q is the delta of Dirac, which assigns probability 1 to q and 0 to all other processes. The symbol $\sum\!\!\!\!\!\!\int_i$ is the summation on distributions. Namely, $\sum\!\!\!\!\!\!\int_i p_i \cdot \mu_i$ is the distribution μ such that $\mu(x) = \sum_i p_i \cdot \mu_i(x)$. The notation $\mu \mid q$ represents the distribution μ' such that $\mu'(r) = \mu(q')$ if $r = q' \mid q$, and $\mu'(r) = 0$ otherwise. Similarly, $(b)\mu$ represents the distribution μ' such that $\mu'(q) = \mu(q')$ if $q = (b)q'$, and $\mu'(q) = 0$ otherwise.

Remark 57. *In some of the examples in this chapter we use an extension of our process calculus that allows message passing (cfr. Chapter 2). Since the expressive power of our calculus with message passing or without it is the same, we consider explicit message passing simply as an alias for the correspondent encoding into the presentation of the calculus given above.*

6.2.3 Distributed systems

A distributed system has the form $(A)\ q_1 \parallel q_2 \parallel \cdots \parallel q_n$, where the q_i's are components and $A \subseteq \mathcal{L}$. The restriction on A enforces synchronization on the channel names belonging to A, in accordance with the CCS spirit.

Systems' semantics The semantics of a system gives rise to a TPA, where the states are terms representing systems during their evolution. A transition now is of the form $q \xrightarrow{t_g : a} \mu$ where $a \in \mathcal{L}$, $\mu \in \mathcal{D}(\mathcal{Q})$, and $t_g \in \mathcal{T}$ is either the tag of the component which makes the move, or a (unordered) pair of tags representing the two partners of a synchronization. We can simply define $\mathcal{T}$ as $\mathcal{T} = I \cup I^2$ where $I = \{1, 2, \ldots, n\}$ is the set of components' identifiers.

$$\text{Interleaving}\quad \frac{q_i \xrightarrow{a} \sum_k p_k \cdot \delta_{q_{ik}}}{(A)\ q_1 \parallel \cdots \parallel q_i \parallel \cdots \parallel q_n \xrightarrow{i:a} \sum_k p_k \cdot \delta_{(A)q_1\parallel\cdots\parallel q_{ik}\parallel\cdots\parallel q_n}}\ a \notin A$$

where i is the tag indicating that the component i is making the step. Note that we assume that probabilistic choices are finite. This implies that every transition $q \xrightarrow{t_g:a} \mu$ can be written $q \xrightarrow{t_g:a} \sum_k p_k \cdot \delta_{q_k}$, and justifies the notation used in the interleaving rule.

$$\text{Synch.}\quad \frac{q_i \xrightarrow{a} \delta_{q'_i} \qquad q_j \xrightarrow{\bar{a}} \delta_{q'_j}}{(A)\ q_1 \parallel \cdots \parallel q_i \parallel \cdots \parallel q_j \parallel \cdots \parallel q_n \xrightarrow{\{i,j\}:\tau} \delta_{(A)q_1\parallel\cdots\parallel q'_i\parallel\cdots\parallel q'_j\parallel\cdots\parallel q_n}}$$

here $\{i, j\}$ is the tag indicating that the components making the step are i and j. Note that it is an unordered pair. Sometimes we will write i, j instead of $\{i, j\}$, for simplicity.

Example 11. *Consider again the systems of Example 10. Figures 6.2(a) and 6.2(b) show the TPAs for $S\,[^a/_{sec}]$ and for $S\,\left[^b/_{sec}\right]$ respectively. For simplicity we do not write the restriction on channels c and out, nor the termination symbol* 0*. We use '$-$' to denote a component that is stuck. The corresponding tags are indicated in the figure with numbers above the components.*

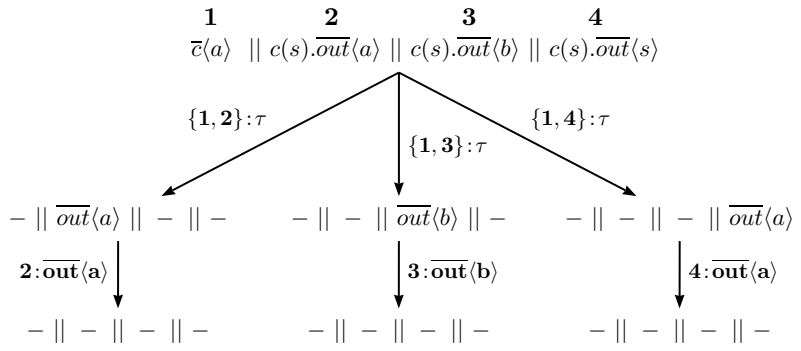
The set of enabled transitions should be clear from the figures. For instance, we have $enab(S\,\left[^b/_{sec}\right]) = \{\{1,2\},\{1,3\},\{1,4\}\}$ and $enab(\ -\ \|\ \overline{out}\langle a\rangle\ \|\ -\ \|\ -\ ) = \{2\}$. The scheduler ζ defined as

$$\zeta(\sigma) \stackrel{def}{=} \begin{cases} \{1,4\} & if\ \sigma = S\,[^a/_{sec}]\,, \\ 2 & if\ \sigma = S\,[^a/_{sec}] \xrightarrow{1,2:\tau} (\ -\ \|\ \overline{out}\langle a\rangle\ \|\ -\ \|\ -\ ), \\ 3 & if\ \sigma = S\,[^a/_{sec}] \xrightarrow{1,3:\tau} (\ -\ \|\ -\ \|\ \overline{out}\langle b\rangle\ \|\ -\ ), \\ 4 & if\ \sigma = S\,[^a/_{sec}] \xrightarrow{1,4:\tau} (\ -\ \|\ -\ \|\ -\ \|\ \overline{out}\langle a\rangle\ ), \\ \bot & otherwise, \end{cases}$$

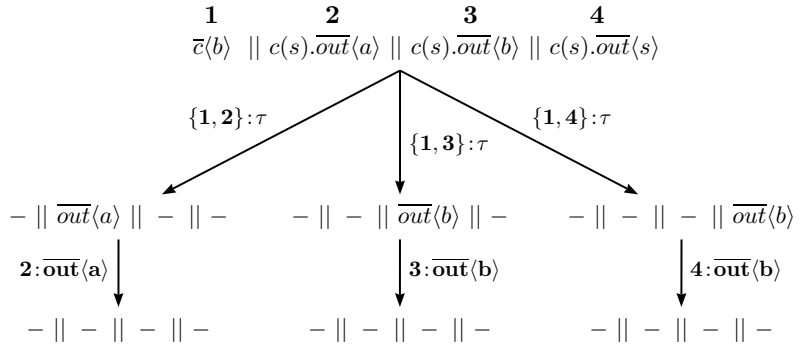
is a global scheduler for $S\,[^a/_{sec}]$.

6.3 Admissible schedulers

In this section we restrict the discerning power of the global and local schedulers in order to avoid the problem of the information leakage induced by



(a) $S[^{a}/_{sec}]$



(b) $S[^{b}/_{sec}]$

Figure 6.2: TPAs for Example 11

clairvoyant schedulers. We impose two kinds of restrictions: For the global scheduler, following [APvRS], we assume that it can only see, and keep memory of, the observable actions and the components that are enabled, but not the secret actions. As for the local scheduler, we assume that the local nondeterminism of each component is solved on the basis of the view of the history local to that component, i.e. the projection of the history of the system on that component. In other words, each component has to make decisions based only on the history of its own execution; it cannot see anything of the other components.

6.3.1 Restricting global schedulers

We assume that the set of actions $\mathcal{L}$ is divided in two disjoint sets, the *secret actions* $\mathcal{S}$ and the *observable actions* $\mathcal{O}$, such that $\mathcal{S} \cup \mathcal{O} = \mathcal{L}$. The secret actions are supposed to be invisible to the global scheduler. Formally, this can

be achieved using a function *sift* with

$$sift(a) = \begin{cases} \tau & \text{if } a \in \mathcal{S}, \\ a & \text{otherwise.} \end{cases}$$

Then, we restrict the power of the global scheduler by forcing it to make the same decisions on paths he cannot tell apart.

Definition 58. *Given a TPA M, a global scheduler ζ for M is* admissible *if for all paths σ_1 and σ_2 we have $view(\sigma_1) = view(\sigma_2)$ implies $\zeta(\sigma_1) = \zeta(\sigma_2)$, where*

$$view\left(\hat{q} \xrightarrow{t_{g1}:a_1} q_1 \xrightarrow{t_{g2}:a_2} \cdots \xrightarrow{t_{gn}:a_n} q_{n+1}\right) \stackrel{def}{=} (enab(\hat{q}), sift(a_1), t_{g1})$$
$$(enab(q_1), sift(a_2), t_{g2}) \cdots (enab(q_n), sift(a_n), t_{gn})$$

The idea is that *view* sifts the information of the path that the scheduler can see. Since *sift* "hides" the secrets, the scheduler cannot take different decisions based on them.

6.3.2 Restricting local schedulers

The restriction on local schedulers is based on the idea that a step of the component i of a system can only be based on the view that i has of the history, i.e. its own history. In order to formalize this restriction, it is convenient to introduce the concept of i-view of a path σ, or *projection* of σ on i, which we will denote by $\sigma_{\restriction i}$. We define it inductively:

$$(\sigma \xrightarrow{t_g:a} \mu)_{\restriction i} = \begin{cases} \sigma_{\restriction i} \xrightarrow{i:b} \delta_{q_i} & \text{if } t_g = \{i,j\} \text{ and } \mu = \delta_{(A)\ q_1\|\ldots\|q_i\|\ldots\|q_j\|\ldots\|q_n} \\ \sigma_{\restriction i} \xrightarrow{i:a} \mu & \text{if } t_g = i \\ \sigma_{\restriction i} & \text{otherwise} \end{cases}$$

In the above definition, the first line represents the case of a synchronization step involving the component i, where we assume that the premise for i is of the form $q'_i \xrightarrow{b} \delta_{q_i}$. The second line represents an interleaving step in which i is the active component. The third line represents step in which the component i is idle.

The restriction to the local scheduler can now be expressed as follows:

Definition 59. *Given a TPA M and a local scheduler ξ for M, we say that ξ is* admissible *if for all paths σ and σ', if whenever $\xi(\sigma) = (t_g, a, \mu)$, and $\xi(\sigma') = (t'_g, a', \mu')$ we have:*

- *if $t_g = t'_g = i$ and $\sigma_{\restriction i} = \sigma'_{\restriction i}$, then $\xi(\sigma) = \xi(\sigma')$,*
- *if $t_g = t'_g = \{i,j\}$, $\sigma_{\restriction i} = \sigma'_{\restriction i}$, and $\sigma_{\restriction j} = \sigma'_{\restriction j}$ then $\xi(\sigma) = \xi(\sigma')$.*

A pair of compatible schedulers (ζ, ξ) is called *admissible* if ζ and ξ are admissible.

6.4 Safe equivalences

In this section we revise process equivalence notions to make them safe for security.

6.4.1 Safe complete traces

We define here a safe version of complete-trace semantics. The idea is that we compare two processes based not only on their traces, but also on the choices that the global scheduler makes at every step. We do this by recording explicitly the tags in the traces.

Definition 60. *Here we define the notion of safe complete traces.*

- *Given a TPA $M = (\mathcal{Q}, \mathcal{T}, \mathcal{L}, \hat{q}, \vartheta)$, the (complete) safe traces of M, denoted here by $Traces_s$, are defined as the probabilities of sequences of tags and actions corresponding to all possible complete executions, i.e.*

$$\begin{aligned} Traces_s(M) = \{ & f : (\mathcal{T} \times \mathcal{L})^\infty \to [0,1] \mid \\ & \textit{there exists an admissible scheduler}(\zeta, \xi) \textit{ s.t.} \\ & \forall t \in (\mathcal{T} \times \mathcal{L})^\infty \\ & f(t) = \mathbf{P}_{M,\zeta,\xi}(\{\sigma \in \mathrm{CPaths}(M) \mid trace_{ta}(\sigma) = t\}) \ \} \end{aligned}$$

  where $\mathbf{P}_{M,\zeta,\xi}$ is the probability measure in M under (ζ, ξ), and $trace_{ta}$ extracts from a path the sequence of tags and actions, i.e.

$$\begin{aligned} trace_{ta}(\epsilon) &= \epsilon \\ trace_{ta}(q \xrightarrow{t_g:a} \sigma) &= t_g : a \cdot trace_{ta}(\sigma) \end{aligned}$$

- *We denote by $Traces_s(q)$ the safe traces of the automaton associated to a system q.*

- *Two systems q_1 and q_2 are safe-trace equivalent, denoted by $q_1 \simeq_s q_2$, if and only if $Traces_s(q_1) = Traces_s(q_2)$.*

The following example points out the difference between $\simeq_s$ and the standard (complete) trace equivalence.

Example 12. *Consider the TPAs of Example 11. The two TPAs have the same complete traces. In fact we have*

$$Traces(S\,[^a/_{sec}]) \quad = \quad \{\tau \cdot \overline{out}\langle a\rangle \ , \ \tau \cdot \overline{out}\langle b\rangle\} \quad = \quad Traces(S\left[^b/_{sec}\right])$$

But on the other hand, we have

$$Traces_s(S\,[^a/_{sec}]) = \{f_1, f_2, f_3\} \quad \neq \quad \{f_1, f_2, f_4\} = Traces_s(S\,[^a/_{sec}])$$

where

$$f_1(t) = \begin{cases} 1 & \textit{if } t = \{1,2\} : \tau \cdot 2 : \overline{out}\langle a \rangle, \\ 0 & \textit{for all other values of } t \in (\mathcal{T} \times \mathcal{L})^\infty. \end{cases}$$

$$f_2(t) = \begin{cases} 1 & \textit{if } t = \{1,3\} : \tau \cdot 3 : \overline{out}\langle b \rangle, \\ 0 & \textit{for all other values of } t \in (\mathcal{T} \times \mathcal{L})^\infty. \end{cases}$$

$$f_3(t) = \begin{cases} 1 & \textit{if } t = \{1,4\} : \tau \cdot 4 : \overline{out}\langle a \rangle, \\ 0 & \textit{for all other values of } t \in (\mathcal{T} \times \mathcal{L})^\infty. \end{cases}$$

$$f_4(t) = \begin{cases} 1 & \textit{if } t = \{1,4\} : \tau \cdot 4 : \overline{out}\langle b \rangle, \\ 0 & \textit{for all other values of } t \in (\mathcal{T} \times \mathcal{L})^\infty. \end{cases}$$

6.4.2 Safe bisimilarity

In this section we propose a security-safe version of strong bisimulation, that we call *safe bisimulation*. This is an equivalence relation stricter than safe-trace equivalence, with the advantage of being a congruence. Since in this chapter we assume that schedulers can always observe which component is making a step (even a silent step), it does not seem natural to consider weak bisimulation.

We start with some notation. Given a TPA $M = (\mathcal{Q}, \mathcal{T}, \mathcal{L}, \hat{q}, \vartheta)$, and a global scheduler ζ, we write $q \xrightarrow{a}_\zeta \mu$ if there exists $\sigma \in \text{Paths}^\star(M)$ such that $\zeta(\sigma) \neq \bot$, $(\zeta(\sigma), a, \mu) \in \vartheta(q)$, and $q = \textit{last}(\sigma)$. Note that the restriction to ζ still allows nondeterminism, i.e. there may be μ_1, μ_2, such that $q \xrightarrow{a_1}_\zeta \mu_1$ and $q \xrightarrow{a_2}_\zeta \mu_2$ (with either $a_1 = a_2$ or $a_1 \neq a_2$).

We now define the notion of safe bisimulation. The idea is that, if q_1 and q_2 are bisimilar states, then every move from q_1 should be mimicked by a move from q_2 *using the same (admissible) scheduler.*

Definition 61. *Given a TPA $M = (\mathcal{Q}, \mathcal{T}, \mathcal{L}, \hat{q}, \vartheta)$, we say that a relation $\mathcal{R} \subseteq \mathcal{Q} \times \mathcal{Q}$ is a safe bisimulation if and only if, whenever $q_1 \,\mathcal{R}\, q_2$:*

1. *$enab(q_1) = enab(q_2)$, and*
2. *for all admissible global schedulers ζ for M such that $\zeta(\sigma_1) \,\mathcal{R}\, \zeta(\sigma_2)$ whenever $last(\sigma_1) = q_1$ and $last(\sigma_2) = q_2$:*
   - *if $q_1 \xrightarrow{a}_\zeta \mu_1$, then there exists μ_2 such that $q_2 \xrightarrow{a}_\zeta \mu_2$ and $\mu_1 \,\mathcal{R}\, \mu_2$, and*
   - *if $q_2 \xrightarrow{a}_\zeta \mu_2$, then there exists μ_1 such that $q_1 \xrightarrow{a}_\zeta \mu_1$ and $\mu_1 \,\mathcal{R}\, \mu_2$,*

where $\mu_1 \,\mathcal{R}\, \mu_2$ means that for all equivalence classes $X \in \mathcal{Q}_{\hat{\mathcal{R}}}$, we have $\mu_1(X) = \mu_2(X)$, where $\hat{\mathcal{R}}$ is the smallest equivalence class induced by $\mathcal{R}$.

It is possible to simplify Definition 61, restricting the schedulers to be history-independent. In other words, to show that two distributed systems are bisimilar, it suffices to consider one-step computations and show that two states are equivalent by using only history-independent schedulers. The lemma bellow justifies this claim.

Lemma 62. *Let $M = (\mathcal{Q}, \mathcal{T}, \mathcal{L}, \hat{q}, \vartheta)$ be a TPA, and let $\mathcal{R}$ be an equivalence relation on the set of states $\mathcal{Q}$. Consider ζ to be a global scheduler for M such that, for every pair of states $q_1, q_2 \in \mathcal{Q}$, if $q_1 = last(\sigma_1)\mathcal{R}\, last(\sigma_2) = q_2$ then $\zeta(\sigma_1) = \zeta(\sigma_2)$. In that case ζ is history-independent, i.e. it depends only on the last state of a path σ.*

Proof. It is easy to see that the relation of having the same last state is an equivalence relation on paths, and therefore it determines a partition on the set of paths. Since the above q_1 and q_2 may be identical, the scheduler must give the same value on equivalent paths and it is, therefore, history-independent. □

Using the lemma above, in the following results about safe bisimulation we will usually write $\zeta(q)$ where q is a state. Note however that this does not mean that in the computations of safely bisimilar systems the schedulers are necessarily history-independent: at each step of the computation we may change scheduler, and therefore we may change alternative when we pass by the same state q at a later time.

The following result is analogous to the case of standard bisimulation. It implies that largest safe bisimulation exists, and coincides with the union of all safe bisimulations. We call it *safe bisimilarity*, and we denote it by $\sim_s$.

Proposition 63. *The union of all the safe bisimulations is still a safe bisimulation.*

Proof. Assume that $q_1 \sim_s q_2$. Then $q_1 \mathcal{R}\, q_2$ holds, for some safe bisimulation $\mathcal{R}$. Hence we have $enab(q_1) = enab(q_2)$, and for every global scheduler ζ, if $\zeta(q_1) = \zeta(q_2)$, and $q_1 \xrightarrow{a}_\zeta \mu_1$, then there exists μ_2 such that $q_2 \xrightarrow{a}_\zeta \mu_2$, and $\mu_1 \mathcal{R}\, \mu_2$. This implies that $\mu_1 \sim_s \mu_2$. In fact $\hat{\mathcal{R}}$ (the smallest equivalence class induced by $\mathcal{R}$) is a finer relation than $\hat{\sim}_s$, i.e. $q_1 \hat{\mathcal{R}}\, q_2$ implies $q_1 \hat{\sim}_s q_2$. Also, $\hat{\mathcal{R}}$ is an equivalence relation, and therefore it induces a partition on each of the equivalence classes $X \in \mathcal{Q}_{\hat{\sim}_s}$. Hence we have, for each $X \in \mathcal{Q}_{\hat{\sim}_s}$, $\mu_1(X) = \sum_{Y \in X_{\hat{\mathcal{R}}}} \mu_1(Y) = \sum_{Y \in X_{\hat{\mathcal{R}}}} \mu_2(Y) = \mu_2(X)$.

We proceed analogously to show that, if $q_2 \xrightarrow{a}_\zeta \mu_2$, then there exists μ_1 such that $q_1 \xrightarrow{a}_\zeta \mu_1$ and $\mu_1 \sim_s \mu_2$.

□

Given two TPAs $M_1 = (\mathcal{Q}_1, \mathcal{T}, \mathcal{L}, \hat{q}_1, \vartheta_1)$ and $M_2 = (\mathcal{Q}_2, \mathcal{T}, \mathcal{L}, \hat{q}_2, \vartheta_2)$ sharing the same set of tags $\mathcal{T}$ and actions $\mathcal{L}$, we can define bisimulation and

bisimilarity across their states, i.e. as relations on $(\mathcal{Q}_1 \cup \mathcal{Q}_2)$, in the obvious way, by constructing the TPA M with a new initial state $\hat{q}$ with transitions to $\delta_{\hat{q}_1}$ and to $\delta_{\hat{q}_2}$, respectively.

Given two components or systems q_1 and q_2, we will say that q_1 and q_2 are safely bisimilar, denoted by $q_1 \sim_s q_2$, if the initial states of the corresponding TPAs are safely bisimilar. Note that $q_1 \sim_s q_2$ is possible only if q_1 and q_2 have the same number of active components, where "active", for a component, means that during the execution of the system it will make at least one step. Note that in the case of components, or of systems constituted by one component only, safe bisimulation and safe bisimilarity coincide with standard bisimulation and bisimilarity (denoted by $\sim$), respectively. This is not the case for systems, as shown by the following example:

Example 13. *Consider again the TPAs of Example 11. As pointed out earlier in this chapter, we have $S\left[{}^{a}/_{sec}\right] \sim S\left[{}^{b}/_{sec}\right]$. Yet $S\left[{}^{a}/_{sec}\right] \not\sim_s S\left[{}^{b}/_{sec}\right]$. To show this, let us construct a new TPA (as described before) with initial state $\hat{q}$ such that $\hat{q} \xrightarrow{t_g:\tau} S\left[{}^{a}/_{sec}\right]$ and $\hat{q} \xrightarrow{t_g:\tau} S\left[{}^{b}/_{sec}\right]$. Now consider the (admissible) global scheduler ζ such that*

$$\zeta(\sigma) \stackrel{def}{=} \begin{cases} t_g & if\ \sigma = \hat{q}, \\ \{1,4\} & if\ \sigma = \hat{q} \xrightarrow{t_g:\tau} S\left[{}^{a}/_{sec}\right], \\ 2 & if\ \sigma = \hat{q} \xrightarrow{t_g:\tau} S\left[{}^{a}/_{sec}\right] \xrightarrow{1,2:\tau} (\ - \parallel \overline{out}\langle a\rangle \parallel\ - \parallel - \ ), \\ 3 & if\ \sigma = \hat{q} \xrightarrow{t_g:\tau} S\left[{}^{a}/_{sec}\right] \xrightarrow{1,3:\tau} (\ - \parallel\ - \parallel \overline{out}\langle b\rangle \parallel - \ ), \\ 4 & if\ \sigma = \hat{q} \xrightarrow{t_g:\tau} S\left[{}^{a}/_{sec}\right] \xrightarrow{1,4:\tau} (\ - \parallel\ - \parallel\ - \parallel \overline{out}\langle a\rangle\ ), \\ \{1,4\} & if\ \sigma = \hat{q} \xrightarrow{t_g:\tau} S\left[{}^{b}/_{sec}\right], \\ 2 & if\ \sigma = \hat{q} \xrightarrow{t_g:\tau} S\left[{}^{b}/_{sec}\right] \xrightarrow{1,2:\tau} (\ - \parallel \overline{out}\langle a\rangle \parallel\ - \parallel - \ ), \\ 3 & if\ \sigma = \hat{q} \xrightarrow{t_g:\tau} S\left[{}^{b}/_{sec}\right] \xrightarrow{1,3:\tau} (\ - \parallel\ - \parallel \overline{out}\langle b\rangle \parallel - \ ), \\ 4 & if\ \sigma = \hat{q} \xrightarrow{t_g:\tau} S\left[{}^{b}/_{sec}\right] \xrightarrow{1,4:\tau} (\ - \parallel\ - \parallel\ - \parallel \overline{out}\langle b\rangle\ ), \\ \perp & otherwise. \end{cases}$$

It is easy to see that $S\left[{}^{b}/_{sec}\right]$ cannot mimic the transition $4:\overline{out}\langle a\rangle$ produced by $S\left[{}^{a}/_{sec}\right]$ using the same scheduler ζ.

We now show that safe bisimulation is a congruence with respect to all the operators of our language. In the following theorem, statements 2a and 2b are just the standard compositionality result for probabilistic bisimulation.

Theorem 64.

1. *$\sim_s$ is an equivalence relation.*
2. *Let $a \in \mathcal{L}$ be an action and $A, B, B' \subseteq \mathcal{L}$ be sets of restrictions. Let $p_1, \ldots, p_n$ be probability values, and let $q, q_1, q_2, \ldots, q_n$, $q'_1, q'_2, \ldots, q'_n$ be components.*

a) *If* $q_1 \sim_s q_2$, *then* $a.q_1 \sim_s a.q_2$, $q_1 + q \sim_s q_2 + q$, $(a)q_1 \sim_s (a)q_2$, *and* $q_1 \mid q \sim_s q_2 \mid q$.

b) *If* $q_1 \sim_s q'_1, \ldots, q_n \sim_s q'_n$, *then* $\sum_i p_i : q_i \sim_s \sum_i p_i : q'_i$.

c) *If* $(B)\ q_1 \parallel \ldots \parallel q_n \ \sim_s \ (B')\ q'_1 \parallel \ldots \parallel q'_n$, *and* $fn(q) \notin B \cup B'$, *then*

$$(A \cup B)\ q_1 \parallel \ldots \parallel q \parallel \ldots \parallel q_n \ \sim_s \ (A \cup B')\ q'_1 \parallel \ldots \parallel q \parallel \ldots \parallel q'_n.$$

Proof.

1. Although safe bisimulations are not equivalence relations in general, their union, i.e. safe bisimilarity, is an equivalence. In fact:

   - It is easy to see that, if $\mathcal{R}$ is a safe bisimulation, then the smallest equivalence that includes $\mathcal{R}$, namely $\hat{\mathcal{R}}$, is also a safe bisimulation.
   - From Proposition 63 we know that $\sim_s$ is a safe bisimulation.
   - Hence we derive that $\hat{\sim}_s$ is a safe bisimulation, and therefore $\hat{\sim}_s \subseteq \sim_s$. But since obviously $\sim_s \subseteq \hat{\sim}_s$, we conclude that $\sim_s = \hat{\sim}_s$, which means that $\sim_s$ is already an equivalence relation.

2. Assume that a, $A, B, B', p_1, \ldots, p_n, q, q_1, q_2, \ldots, q_n$, $q'_1, q'_2, \ldots, q'_n$ are of the type prescribed by the hypothesis of the theorem.

   a) Assume $q_1 \sim_s q_2$.

   - Let
   $$\mathcal{R} = \{(a.q_1, a.q_2)\} \cup \sim_s .$$
   We show that $\mathcal{R}$ is a safe bisimulation, which is sufficient to prove that $a.q_1 \sim_s a.q_2$. Note that, since there is only one component in each of those states, and it is enabled, we have $enab(a.q_1) = enab(a.q_2) = \{1\}$, and $\zeta(a.q_1) = \zeta(a.q_2) = 1$ for any global scheduler ζ. Given a global scheduler ζ, there is exactly one transition from each of $a.q_1$ and $a.q_2$: these are $a.q_1 \xrightarrow{a}_\zeta \delta_{q_1}$ and $a.q_2 \xrightarrow{a}_\zeta \delta_{q_2}$, respectively, which mimic each other in the action a. Finally, since $q_1 \sim_s q_2$, we have $\delta_{q_1} \sim_s \delta_{q_2}$ and therefore $\delta_{q_1} \mathcal{R} \delta_{q_2}$.

   - Let
   $$\mathcal{R} = \{(q_1 + q, q_2 + q)\} \cup \sim_s .$$
   We show that $\mathcal{R}$ is a safe bisimulation, which is sufficient to prove that $q_1 + q \sim_s q_2 + q$. We have that $enab(q_1 + q) = enab(q_1) \cup enab(q) = enab(q_2) \cup enab(q) = enab(q_2 + q)$, in fact $enab(q_1) = enab(q_2)$ since $q_1 \sim_s q_2$. Correspondingly, given a

global scheduler ζ, we have either $\zeta(q_1 + q) = \zeta(q_2 + q) = 1$ or $\zeta(q_1 + q) = \zeta(q_2 + q) = \perp$, since there is only one component. Assume $q_1 + q \xrightarrow{a}_\zeta \mu_1$. We have two cases: either $q_1 \xrightarrow{a}_\zeta \mu_1$, or $q \xrightarrow{a}_\zeta \mu_1$. The second case is obvious. In the first case, since $q_1 \sim_s q_2$, we have that also $q_2 \xrightarrow{a}_\zeta \mu_2$, with $\mu_1 \sim_s \mu_2$. We derive that $\mu_1 \mathcal{R} \mu_2$. For the transitions from $q_2 + q$ we proceed in the analogous way.

- Let

$$\mathcal{R} = \{((a)q_1, (a)q_2) \mid q_1 \sim_s q_2\}.$$

We show that $\mathcal{R}$ is a safe bisimulation, which is sufficient to prove that, if $q_1 \sim_s q_2$, then $(a)q_1 \sim_s (a)q_2$. First observe that $enab((a)q_1) = enab(q_1) = \{1\}$ if q_1 can make a transition with a label different from a, otherwise $enab((a)q_1) = \emptyset$. The same holds for $(a)q_2$. Since $q_1 \sim_s q_2$, we derive that $enab((a)q_1) = enab((a)q_2)$. Accordingly, given a global scheduler ζ, we have that either $\zeta((a)q_1) = \zeta((a)q_2) = 1$, or $\zeta((a)q_1) = \zeta((a)q_2) = \perp$. Assume $(a)q_1 \xrightarrow{b}_\zeta \mu_1$. Then we must have $b \neq a$ and $\mu_1 = (a)\mu_1'$, where $q_1 \xrightarrow{b}_\zeta \mu_1'$. Since $q_1 \sim_s q_2$, we have also $q_2 \xrightarrow{b}_\zeta \mu_2'$, with $\mu_1' \sim_s \mu_2'$. We derive $(a)q_2 \xrightarrow{b}_\zeta (a)\mu_2'$, and $(a)\mu_1' \mathcal{R} (a)\mu_2'$. We proceed in an analogous way for the transitions from $(a)q_2$.

- The case of the parallel operator in components is similar to the case of the parallel operator on systems (see the last item of this proof).

b) Assume $q_1 \sim_s q_1', \ldots, q_n \sim_s q_n'$. Let

$$\mathcal{R} = \{(\sum_i p_i : q_i, \sum_i p_i : q_i')\} \cup \sim_s .$$

We show that $\mathcal{R}$ is a safe bisimulation, which is sufficient to prove that $\sum_i p_i : q_i \sim_s \sum_i p_i : q_i'$. Observe that both $\sum_i p_i : q_i$ and $\sum_i p_i : q_i'$ are enabled, and, since there is only one component, $enab(\sum_i p_i : q_i) = enab(\sum_i p_i : q_i') = \{1\}$. Accordingly, if ζ is a global scheduler, we have $enab(\sum_i p_i : q_i) = enab(\sum_i p_i : q_i') = 1$. Given a global scheduler ζ, the only transitions from $\sum_i p_i : q_i$ and $\sum_i p_i : q_i'$ are $\sum_i p_i : q_i \xrightarrow{\tau}_\zeta \sum_i p_i \cdot \delta_{q_i}$ and $\sum_i p_i : q_i' \xrightarrow{\tau}_\zeta \sum_i p_i \cdot \delta_{q_i'}$ respectively, which mimic each other in the action τ. It is easy to see that we have $(\sum_i p_i : q_i) \sim_s (\sum_i p_i : q_i')$, and therefore $(\sum_i p_i : q_i) \mathcal{R} (\sum_i p_i : q_i')$.

c) Let

$$\mathcal{R} = \left\{ \begin{array}{l} ((A \cup B)\ q_1 \parallel \ldots \parallel q \parallel \ldots \parallel q_n, \\ (A \cup B')\ q'_1 \parallel \ldots \parallel q \parallel \ldots \parallel q'_n) \mid \\ (B)\ q_1 \parallel \ldots \parallel q_n \ \sim_s \ (B')\ q'_1 \parallel \ldots \parallel q'_n \end{array} \right\}$$

We show that $\mathcal{R}$ is a safe bisimulation, which is sufficient to prove that, if

$$(B)\ q_1 \parallel \ldots \parallel q_n \ \sim_s \ (B')\ q'_1 \parallel \ldots \parallel q'_n \ ,$$

then

$$(A \cup B)\ q_1 \parallel \ldots \parallel q \parallel \ldots \parallel q_n \ \sim_s \ (A \cup B')\ q'_1 \parallel \ldots \parallel q \parallel \ldots \parallel q'_n \ .$$

Observe first that

$$\begin{array}{c} enab((A \cup B)\ q_1 \parallel \ldots \parallel q \parallel \ldots \parallel q_n) = \\ enab((A \cup B')\ q'_1 \parallel \ldots \parallel q \parallel \ldots \parallel q'_n) \end{array}$$

In fact the enabled components are the same as those of $(B)\ q_1 \parallel \ldots \parallel q_n$ and of $(B')\ q'_1 \parallel \ldots \parallel q'_n$ (modulo the index shift), which are equal by the bisimilarity hypothesis, plus possibly the component q, plus possibly the synchronizations with q, which again are equal by the bisimilarity hypothesis, minus the transitions with labels in A. Note that the hypothesis $fn(q) \notin B \cup B'$ is essential here to guarantee that the component q is enabled (or disabled) in both sides.

Let us consider the synchronization case; the interleaving case is just a simplified variant. Given a global scheduler ζ, assume

$$\zeta((A \cup B)\ q_1 \parallel \ldots \parallel q \parallel \ldots \parallel q_n) = \zeta((A \cup B')\ q'_1 \parallel \ldots \parallel q \parallel \ldots \parallel q'_n).$$

Consider a move from the system in the left-hand side:

$$(A \cup B)\ q_1 \parallel \cdots \parallel q_i \parallel \cdots \parallel q_j \parallel \cdots \parallel q_n \xrightarrow{i,j:\tau} \delta_{(A)q_1 \parallel \cdots \parallel r_i \parallel \cdots \parallel r_j \parallel \cdots \parallel q_n}.$$

Then we must have

$$q_i \xrightarrow{a} \delta_{r_i} \quad , \quad q_j \xrightarrow{\bar{a}} \delta_{r_j} \quad ,$$

where one of the q_i, q_j could be q, and

$$\zeta((A \cup B)\ q_1 \parallel \cdots \parallel q_i \parallel \cdots \parallel q_j \parallel \cdots \parallel q_n) = \{i, j\}.$$

Since $q_i \sim_s q'_i$ and $q_j \sim_s q'_j$ (in case $q_i = q$ then $q'_i = q$ and therefore $q_i \sim_s q'_i$ because $\sim_q$ is reflexive, and analogously for q_j), we must have

$$q'_i \xrightarrow{a} \delta_{r'_i} \quad , \quad q'_j \xrightarrow{\bar{a}} \delta_{r'_j} \quad ,$$

for some r'_i, r'_j such that $\delta_{r_i} \sim_s \delta_{r'_i}$ and $\delta_{r_j} \sim_s \delta_{r'_j}$. We derive that

$$(A \cup B)\ q'_1 \parallel \cdots \parallel q'_i \parallel \cdots \parallel q'_j \parallel \cdots \parallel q'_n \xrightarrow{i,j:\tau} \delta_{(A)q'_1\|\cdots\|r'_i\|\cdots\|r'_j\|\cdots\|q'_n}\ ,$$

and, since $\delta_{r_i} \sim_s \delta_{r'_i}$, $\delta_{r_j} \sim_s \delta_{r'_j}$ imply $r_i \sim_s r'_i$, $r_j \sim_s r'_j$, and by the definition of $\mathcal{R}$, we conclude

$$(\delta_{(A)q_1\|\cdots\|r_i\|\cdots\|r_j\|\cdots\|q_n})\ \mathcal{R}\ (\delta_{(A)q'_1\|\cdots\|r'_i\|\cdots\|r'_j\|\cdots\|q'_n}).$$

We proceed in an analogous way for the transitions from the right-hand side.

□

The following property shows that bisimulation is stronger than safe-trace equivalence, like in the standard case.

Proposition 65. *If $q_1 \sim_s q_2$ then $q_1 \simeq_s q_2$.*

Proof. For this proof, it is convenient to consider a coinductive approximation of safe-trace equivalence. We start with a coinductive characterization of the safe traces. This in itself is not a key notion of the proof, but will help understanding the definition of the approximation.

Given a TPA $M = (\mathcal{Q}, \mathcal{T}, \mathcal{L}, \hat{q}, \vartheta)$, consider the operator

$$\mathcal{T}_{\mathrm{T}r} : (\mathcal{Q} \to \mathcal{P}(\mathrm{CPaths}(M) \to [0,1])) \to (\mathcal{Q} \to \mathcal{P}(\mathrm{CPaths}(M) \to [0,1]))$$

defined as:

$$\begin{aligned}
\mathcal{T}_{\mathrm{T}r}(F)(q) = \ \{\ & f : (\mathcal{T} \times \mathcal{L})^\infty \to [0,1] \mid \\
& \text{if } q \not\to \text{ then } f(\epsilon) = 1, \text{ else } f(\epsilon) = 0 \text{ and,} \\
& \text{for all } t_g \in \mathcal{T}, a \in \mathcal{L}, \\
& \bullet \text{ if there exists } \mu \text{ s.t. } q \xrightarrow{t_g:a} \mu, \text{ then for each } q' \in \mathcal{Q} \\
& \text{there exists } f'_{q'} \in F(q') \text{ s.t. for every } t \in (\mathcal{T} \times \mathcal{L})^\infty, \\
& f(t_g : a \cdot t) = \textstyle\sum_{q'} \mu(q') f'_{q'}(t) \\
& \bullet \text{ if } q \not\xrightarrow{t_g:a}, \text{ then } f(q)(t_g : a \cdot t) = 0 \quad \}
\end{aligned}$$

where $q \not\to$ means that for all $t_g \in \mathcal{T}, a \in \mathcal{L}$, we have $q \not\xrightarrow{t_g:a}$.

Consider the ordering $\sqsubseteq$ on $\mathcal{Q} \to \mathcal{P}(\mathrm{CPaths}(M) \to [0,1])$ given by

$$F \sqsubseteq F' \qquad \text{if and only if} \qquad \text{for all } q \in \mathcal{Q}, F(q) \subseteq F'(q)$$

Clearly $(\mathrm{CPaths}(M) \to [0,1]), \sqsubseteq)$ is a complete lattice and $\mathcal{T}_{\mathrm{T}r}$ is monotonic, so by the theorem of Knaster-Tarski it has a greatest fixed point, which coincides with Traces_s.

Following the definition of $\mathcal{T}_{\mathrm{T}r}$, we now give a coinductive approximation of the equivalence relation induced by Traces_s. Given a TPA $M = (\mathcal{Q}, \mathcal{T}, \mathcal{L}, \hat{q}, \vartheta)$, consider the operator

$$\mathcal{T}_{\mathrm{T}req} : (\mathrm{CPaths}(M) \to \mathcal{Q} \times \mathcal{Q}) \to (\mathrm{CPaths}(M) \to \mathcal{Q} \times \mathcal{Q})$$

defined as:

$$q_1 \ \mathcal{T}_{\mathrm{T}req}(\mathcal{R})(\epsilon) \ q_2 \quad \overset{\mathrm{def}}{\Leftrightarrow} \quad (q_1 \not\to \ \Leftrightarrow \ q_2 \not\to)$$

and

$$q_1 \ \mathcal{T}_{\mathrm{T}req}(\mathcal{R})(t_g : a \cdot t) \ q_2 \quad \overset{\mathrm{def}}{\Leftrightarrow} \quad \left(\begin{array}{c} q_1 \xrightarrow{t_g:a} \mu_1 \Rightarrow \exists \mu_2.(q_2 \xrightarrow{t_g:a} \mu_2 \ \wedge \ \mu_1 \ \mathcal{R}(t) \ \mu_2) \\ \wedge \\ q_2 \xrightarrow{t_g:a} \mu_2 \Rightarrow \exists \mu_1.(q_1 \xrightarrow{t_g:a} \mu_1 \ \wedge \ \mu_1 \ \mathcal{R}(t) \ \mu_2) \end{array} \right)$$

Consider the ordering $\preceq$ on $\mathrm{CPaths}(M) \to \mathcal{Q} \times \mathcal{Q}$ given by

$$\mathcal{R} \preceq \mathcal{R}' \qquad \text{if and only if} \qquad \text{for all } t \in \mathrm{CPaths}(M), \ \mathcal{R}(t) \subseteq \mathcal{R}'(t)$$

Clearly $(\mathrm{CPaths}(M) \to \mathcal{Q} \times \mathcal{Q}, \preceq)$ is a complete lattice and $\mathcal{T}_{\mathrm{T}req}$ is monotonic, hence by the Knaster-Tarski theorem it has a greatest fixed point, which also coincides with the greatest pre-fixed point, i.e. the greatest relation $\mathcal{R}$ such that $\mathcal{R} \preceq \mathcal{T}_{\mathrm{T}req}(\mathcal{R})$. Using the definition of $\mathcal{T}_{\mathrm{T}r}$ it is easy to see that, if $\mathcal{R}$ is a pre-fixed point, and $q_1 \ \mathcal{R}(t) \ q_2$ for all $t \in \mathrm{CPaths}(M)$, then $\mathit{Traces}_s(q_1) = \mathit{Traces}_s(q_2)$, i.e. $q_1 \simeq_s q_2$. In fact, if $F(q_1) = F(q_2)$, and $q_1 \ \mathcal{R}(t) \ q_2$ for all $t \in \mathrm{CPaths}(M)$, and $\mathcal{R}$ is a pre-fixed point of $\mathcal{T}_{\mathrm{T}req}$, then $\mathcal{T}_{\mathrm{T}r}(F)(q_1) = \mathcal{T}_{\mathrm{T}r}(F)(q_2)$[3]. Consider now a safe bisimulation $\mathcal{R}$, and let us lift it to a constant function $\mathcal{R} : \mathrm{CPaths}(M) \to \mathcal{Q} \times \mathcal{Q}$ defined as $\mathcal{R}(t) = \mathcal{R}$. It is easy to see that $\mathcal{R}$ is a pre-fixed point of $\mathcal{T}_{\mathrm{T}req}$[4].

Assume now $q_1 \ \mathcal{R} \ q_2$. We trivially derive that $q_1 \ \mathcal{R}(t) \ q_2$ for all $t \in \mathrm{CPaths}(M)$, from which we conclude $q_1 \simeq q_2$.

□

Like in the standard case, the vice-versa does not hold, and safe-trace equivalence is not a congruence[5].

[3]Note that the condition is only sufficient, because $\sum_{q'} \mu_1(q') f'_{q'1}(t) = \sum_{q'} \mu_2(q') f'_{q'2}(t)$ may hold even if μ_1 and μ_2 assign different probability to some equivalence class of $\hat{\mathcal{R}(t)}$.

[4]Note that the converse does not hold, i.e. $\mathcal{R}$ could be a pre-fixpoint of $\mathcal{T}_{\mathrm{T}req}$ even if $\mathcal{R}$ is not a bisimulation. This is because $\mathcal{R}$ is sensitive to the (nondeterministic) branching structure, while $\mathcal{R}$ is not.

[5]This is because we are considering the *complete* traces.

6.5 Safe nondeterministic information hiding

In this section we define the notion of information hiding under the most general hypothesis that the nondeterminism is handled partly in a demonic way and partly in an angelic way. We assume that the demonic part is in the realm of the global scheduler, while the angelic part is controlled by the local scheduler. The motivation is that in a protocol the local components can be thought of as programs running locally in a single machine, and locally predictable and controllable, while the network can be subject to attacks that make the interactions unpredictable.

We recall that, in a purely probabilistic setting, the absence of leakage, such as noninterference and strong anonymity, is expressed as follows (see for instance [BP]). Given a purely probabilistic automaton M, and a sequence $\tilde{a} = a_1 a_2 \ldots a_n$, let $\mathbf{P}_M([\tilde{a}])$ represent the probability measure of all complete paths with trace $\tilde{a}$ in M. Let S be a protocol containing a variable action $secr$, and let s be secret actions. Let M_s be the automaton corresponding to $S[^s/_{secr}]$. Define $Pr(\tilde{a} \mid s)$ as $\mathbf{P}_{M_s}([\tilde{a}])$. Then S is leakage-free if for every observable trace $\tilde{a}$, and for every secret s_1 and s_2, we have $Pr(\tilde{a} \mid s_1) = Pr(\tilde{a} \mid s_2)$.

In a purely nondeterministic setting, on the other hand, the absence of leakage has been characterized in the literature by the property $S[^{s_1}/_{secr}] \cong S[^{s_2}/_{secr}]$, where $\cong$ is an equivalence relation like trace equivalence, or bisimulation. As we have argued in the introduction, this definition assumes an angelic interpretation of nondeterminism.

We want to combine the above notions so to cope with both probability and nondeterminism. Furthermore, we want to extend it to the case in which part of the nondeterminism is interpreted demonically. Let us first introduce some notation.

Let S be a system containing a variable action $secr$. Let s be a secret action. Let M_s be the TPA associated to $S[^s/_{secr}]$ and let (ζ, ξ) be a compatible pair of global and local schedulers for M_s. The probability of an observable trace $\tilde{a}$ given s is defined as $Pr_{\zeta,\xi}(\tilde{a} \mid s) = \mathbf{P}_{M_s,\zeta,\xi}([\tilde{a}])$.

The global nondeterminism is interpreted demonically, and therefore we need to ensure that the conditional of an observable, given the two secrets, are calculated with respect to the same global scheduler. On the other hand, the local scheduler is interpreted angelically, and therefore we can compare the conditional probabilities generated by the two secrets as sets under different schedulers. In other words, we have the freedom to match conditional probability from the first set with one of the other set, without requiring the local scheduler to be the same.

Either angelic or demonic, we want to avoid the clairvoyant schedulers, i.e. a scheduler should not be able to use the secret information to achieve its goals. For this purpose, we require both the global and the local scheduler to be admissible.

Definition 66. *A system is leakage-free if, for every pair of secrets s_1 and s_2, every admissible global scheduler ζ, and every observable trace $\tilde{a}$,*

$$\{Pr_{\zeta,\xi}(\tilde{a} \mid s_1) \mid \xi \text{ is admissible and compatible with } \zeta\} = \\ \{Pr_{\zeta,\xi}(\tilde{a} \mid s_2) \mid \xi \text{ is admissible and compatible with } \zeta\}$$

The safe equivalences defined in Section 6.4 imply the absence of leakage:

Theorem 67. *Let S be a system with a variable action secr and assume $S[^{s_1}/_{secr}] \simeq_s S[^{s_2}/_{secr}]$ for every pair of secrets s_1 and s_2. Then S is leakage-free.*

Proof. Consider the abstraction operator β from safe traces to pairs of the form (tagged observable trace, probability) defined as:

$$(\tilde{a}, p) \in \beta(F) \quad \overset{def}{\Leftrightarrow} \quad p \;=\; \sum_{\substack{f \in F \\ t_{\restriction \mathcal{T}\times\mathcal{O}} = \tilde{a}}} f(t)$$

It is easy to see that β is an abstraction, i.e. if $F_1 = F_2$ then $\beta(F_1) = \beta(F_2)$. Therefore, $S[^{s_1}/_{secr}] \simeq_s S[^{s_2}/_{secr}]$ implies $\beta(Traces_s(S[^{s_1}/_{secr}]) = \beta(Traces_s(S[^{s_2}/_{secr}])$. Finally, the latter holds (for every pair of secrets s_1, s_2) if and only if S is leakage-free.

□

Note that the vice versa is not true, i.e. it is not the case that the leakage-freedom of S implies $S[^{s_1}/_{secr}] \simeq_s S[^{s_2}/_{secr}]$. This is because in the definition of safe trace equivalence we compare the set of probability functions (determined by the schedulers) on traces, while in the definition of leakage-freedom we compare the set of probabilities of each trace, which may come from different functions. This additional degree of freedom generated by the local scheduler helps the system to obfuscate the secret, and provides further justification for the adjective "angelic" for the local nondeterminism.

From the above theorem and from Proposition 65, we also have the following corollary (with the same premises as the previous theorem):

Corollary 68. *If $S[^{s_1}/_{secr}] \sim_s S[^{s_2}/_{secr}]$ for every pair of secrets s_1 and s_2, then S is leakage-free.*

6.6 Related work

The problem of deriving correct implementations from secrecy specifications has received a lot of attention already. One of the first works to address the problem was [Jac89], which showed that the fact that an implementation is a consistent refinement with respect to a specification does not imply that

the (information-flow) security properties are preserved. More recently, [AZ06] has proposed a notion of secrecy-preserving refinement, and a simulation-based technique for proving that a system is the refinement of another. [CS08] argues that important classes of security policies such as noninterference and average response time cannot be expressed by traditional notion of *properties*, which consist of sets of traces, and proposes to use *hyperproperties* (sets of properties) instead. [DDM10] addresses the problem of supervisory control, i.e. given a critical system G that may leak confidential information, how to design a controller C so that the system $G|C$ dos not leak. An effective algorithm is presented to compute the most permissible controller such that the system is still opaque with respect to a secret.

Concerning angelic and demonic nondeterminism, there are various works which investigate their relation and possible combination. In [BvW92] it is shown that angelic and demonic nondeterminism are dual. [MCR07] uses multi-relations to express specifications involving both angelic and demonic nondeterminism. There are two kinds of agents, demonic and angelic ones, and there is the point of view of the internal system and the one of the external adversary.

[Mor09] considers the problem of refining specifications while preserving ignorance. While the focus is on the reduction of demonic nondeterminism of the specification, the hidden values are treated essentially in a angelic way.

The problem of the leakage caused by full-information schedulers has also been investigated in the literature. [CCK+06a] and [CCK+06b] work in the framework of probabilistic automata and introduce a restriction on the scheduler to the purpose of making them suitable to applications in security protocols. Their approach is based on dividing the actions of each component of the system in equivalence classes (*tasks*). The order of execution of different tasks is decided in advance by a so-called *task scheduler*, which is history-independent and therefore much more restricted than our notion of global scheduler. [APvRS] proposes a notion of system and admissible scheduler very similar to our notion of system and admissible global scheduler. The main difference is that in that work the components are deterministic and therefore there is no notion of local scheduler.

The work in [CP, CNP09] is similar to ours in spirit, but in a sense *dual* from a technical point of view. Instead of defining a restriction on the class of schedulers, the authors a way to specify that a choice is transparent to the scheduler. They achieve this by introducing labels in process terms, used to represent both the states of the execution tree and the next action or step to be scheduled. They make two states indistinguishable to schedulers, and hence the choice between them private, by associating to them the same label. We believe that every scheduler in our formalism can be expressed in theirs, too. In [CNP09] the authors consider the problem of defining a safe version of bisimulation for expressing security properties. They call it *demonic bisimulation*. The main difference with our work is that we consider a combination

of angelic and demonic nondeterminism, and this affects also the definition of bisimulation. Similarly, our definition of leakage-freedom reflects this combination. In [CNP09] the aspect of angelicity is not considered, although they may be able to simulate it with an appropriate labeling.

The fact that full-information schedulers are unrealistic has also been observed in fields other than security. First attempts used restricted schedulers in order to obtain rules for compositional reasoning [dAHJ01]. The justification for those restricted schedulers is the same as for ours, namely, that not all information is available to all entities in the system. That work considers a synchronous parallel composition, however, so the setting is rather different from ours. Later on, it was shown that model checking is unfeasible in its general form for the restricted schedulers in [dAHJ01] (see [GD07] and, more recently, [Gir09]). Despite of undecidability, not all results concerning such schedulers have been negative as, for instance, the technique of partial-order reduction can be improved by assuming that schedulers can only use partial information [GDF09].

6.7 Chapter summary and discussion

In this chapter we have observed that some definitions of security properties based on process equivalences may be too naive, in the sense that they assume the scheduler to be angelic, and, worse yet, to achieve its angelic strategy by peeking at the secrets. We have presented a formalism allowing us to specify a demonic constituent of the scheduler, possibly in collusion with the attacker, and an angelic one, under the control of the system. We have also considered restrictions on the schedulers to limit the power of what they can see, and extended to our nondeterministic framework the (probabilistic) information-hiding properties like non interference and strong anonymity. We then have defined "safe" equivalences. In particular we have defined the notions of safe trace equivalence and safe bisimilarity, and we have shown that the latter is still a congruence. Finally, we have shown that the safe equivalences can be used to prove information-hiding properties.

For the future, we plan to extend our framework to quantitative notions of information leakage, possibly based on information theory. We also plan to implement model checking techniques to verify information hiding properties for our kind of systems.

Seven

Conclusion

"To succeed, jump as quickly at opportunities as you do at conclusions."
Benjamin Franklin

In this thesis we concentrated on the problem of information hiding in the scenarios of interactive systems, statistical disclosure control, and the refinement of specifications. We started by giving a general overview of the field of information hiding, including a brief description of its historical development. We then discussed the main differences between the qualitative and the quantitative approaches to information hiding, and we introduced the background for the three main topics covered in this thesis: information flow (exemplified by anonymity), statistical disclosure control, and the refinement of specifications into implementations.

Having adopted the quantitative approach, we then continued to discuss the rationale of the use of information theory for quantitative information flow. We reviewed several formulations of entropy, with a special focus on Shannon entropy and min-entropy, and the related concept of mutual information and its interpretation in terms of attacks and information leakage.

We then proceeded to present the technical contributions of the thesis. We started with the scenario of interactive systems, i.e systems where secrets and observables can alternate and influence each other during the computation. In this type of systems the traditional information theoretical approach that makes use of classic memoryless channels, and the related concepts of mutual information and classical capacity, no longer works. We proposed to model interactive systems with a richer notion of channels, namely channels with memory and feedback. In this more general model it is possible to split the statistical correlation between secrets and observables (that correspond to the input and the output of the channel, respectively) into two causal components: the *directed information from input to output* represents the flow of information through the channel, and the *directed information from output to input*

corresponds to the way the input is influenced by the output via feedback. We showed that the directed information is the correct measure of leakage in interactive systems, and so is the concept of directed capacity if we are interested in the worst case leakage. We also proved that our model is a proper extension of the classic one: in the absence of feedback (i.e interaction) our model collapses into the simpler classic model. Finally, we showed that the capacity of channels with memory and feedback is a continuous function of a pseudometric based on the Kantorovich metric.

With respect to interactive systems, as future work we want to explore algorithms to calculate the leakage and the maximum leakage using our model. This is a rather challenging problem, given the exponential growth of reaction functions (a technical aspect of our model) and the quantification of possibly infinite many reactors (also another technicality of our model). We also want to explore other notions of entropy as a measure of leakage, as for instance the min-entropy and the corresponding notion of one-try attack.

In the sequence we moved to the problem of statistical disclosure control. We considered the problem of preserving the privacy of individuals participating in a database that allows statistical queries to be posed by users. Using differential privacy, databases that are similar, i.e differ by the contents of at most one row, should give statistically "similar" answers to the same query. This is achieved by introducing noise in the query mechanism to blur the link between the reported answer and the data about individuals. We proposed a model where the differential privacy mechanism can be split into two channels in cascade, in the case the randomization mechanism is oblivious (i.e it only depends on the real answer to the query, and not on the database itself). The first channel corresponds to the query, and it maps the database to the real answer to the query. The second channel corresponds to the oblivious randomization mechanism, and it takes the real answer and maps it to a randomized answer to be reported to the user. In this scenario we see the *leakage* as the correlation between the reported answer and the database, and the *utility* as the correlation between the real answer and the reported one. We used this model to derive bounds for the leakage and utility based on the level of differential privacy designed for the system (namely the parameter ϵ). As a measure of leakage we adopted the min-entropy leakage, and for utility we used the notion of gain functions, focusing on the binary gain function, which is strictly related to min-entropy leakage and Bayes risk. We used the graph structure on the input domain derived from the adjacency relation on databases to derive bounds for the maximum min-entropy leakage of channels. We showed that if the graph structure is distance-regular or VT^+ (which is always the case for the database domain), then we can derive bounds for the maximum min-entropy leakage associated to the channel. Finally, we found a way of constructing a utility-maximizing randomization function that respects differential privacy for a special class of graph structures.

In relation to statistical databases, as future work we intend to extend our

results to other types of gain functions than the binary one, namely gain functions that take into consideration a notion of distance between answers. We also want to investigate whether or not non-oblivious randomization mechanisms can be used to improve utility while still preserving differential privacy.

The last scenario we investigated in the thesis was the use of equivalence relations to specify security guarantees, which is a common approach when refining implementations into specifications. Under this perspective, two systems (e.g a specification and its implementation) are considered equivalently secure if they respect some equivalence relation defined to capture the intended security guarantee. Such equivalences include, for instance, trace-equivalence and bisimilarity. We showed that a naive use of these equivalences can lead to unrealistic assumptions about the scheduler: (i) that the scheduler is angelic, i.e that it will help to keep the secret information from the attacker; and (ii) that the scheduler can peek at the secrets to make its choices. Those assumptions are not safe in practical cases and, therefore, we proposed a model that deals with the problem. We introduced a formalism that explicitly separates the demonic and angelic parts of the scheduler, and we imposed restrictions to limit the power of the scheduler with respect to what it can see. Namely, the scheduler cannot peek at the secrets to make its choices. We then defined notions of safe-equivalences (safe trace equivalence and safe bisimilarity) and we showed that the latter is a congruence. Finally, we showed that safe equivalences can be used to prove information hiding properties.

As future work regarding safe equivalences, we want to extend our model to quantitative notions based on information theory, and we want to use model checking to certify information hiding properties for our systems.

As final remark, we believe that information hiding is a very promising field of research, and we are excited and thrilled by the promising challenges that lie ahead.

Bibliography

[AAC+11] Mário S. Alvim, Miguel E. Andrés, Konstantinos Chatzikokolakis, Pierpaolo Degano, and Catuscia Palamidessi. Differential privacy: on the trade-off between utility and information leakage. Technical report, INRIA, 2011. http://hal.inria.fr/inria-00580122/en/.

[AACP11] Mário S. Alvim, Miguel E. Andrés, Konstantinos Chatzikokolakis, and Catuscia Palamidessi. On the relation between differential privacy and quantitative information flow. In *Proceedings of the 38th International Colloquium on Automata, Languages and Programming (ICALP 2011), Zürich, Switzerland, July 4th-8th 2011*, 2011. to appear.

[AAP10a] Mário S. Alvim, Miguel E. Andrés, and Catuscia Palamidessi. Information Flow in Interactive Systems. In Paul Gastin and François Laroussinie, editors, *Proceedings of the 21th International Conference on Concurrency Theory (CONCUR 2010), Paris, France, August 31-September 3*, volume 6269 of *Lecture Notes in Computer Science*, pages 102–116. Springer, 2010.

[AAP10b] Mário S. Alvim, Miguel E. Andrés, and Catuscia Palamidessi. Probabilistic information flow. In *Proceedings of the 25th Annual IEEE Symposium on Logic in Computer Science (LICS 2010)*, pages 314–321. IEEE Computer Society, 2010.

[AAP11] Mário S. Alvim, Miguel E. Andrés, and Catuscia Palamidessi. Quantitative information flow in interactive systems. *Journal of Computer Security*, 2011. To appear.

[AAPvR10] Mário S. Alvim, Miguel E. Andrés, Catuscia Palamidessi, and Peter van Rossum. Safe Equivalences for Security Properties. In Cristian S. Calude and Vladimiro Sassone, editors, *Proceedings of the 6th IFIP International Conference on Theoretical Computer Science (TCS 2010)*, volume 323 of *IFIP Advances in Information and Communication Technology*, pages 55–70. Springer, 2010.

[ACDP10] Mário S. Alvim, Konstantinos Chatzikokolakis, Pierpaolo Degano, and Catuscia Palamidessi. Differential privacy versus quantitative information flow. Technical report, 2010.

[AG99] Martín Abadi and Andrew D. Gordon. A calculus for cryptographic protocols: The spi calculus. *Information and Computation*, 148(1):1–70, 10January 1999.

[APvRS] Miguel E. Andrés, Catuscia Palamidessi, Peter van Rossum, and Ana Sokolova. Information hiding in probabilistic concurrent systems. www.cs.ru.nl/M.Andres/downloads/SAuN.pdf.

[APvRS10] Miguel E. Andrés, Catuscia Palamidessi, Peter van Rossum, and Geoffrey Smith. Computing the leakage of information-hiding systems. In Javier Esparza and Rupak Majumdar, editors, *Proceedings of the 16th International Conference on Tools and Algorithms for the Construction and Analysis of Systems (TACAS 2010)*, volume 6015 of *Lecture Notes in Computer Science*, pages 373–389. Springer, 2010.

[AZ06] Rajeev Alur and Steve Zdancewic. Preserving secrecy under refinement. In *Proc. of the 33rd International Colloquium on Automata, Languages and Programming (ICALP '06), volume 4052 of Lecture Notes in Computer Science*, number 4052 in Lecture Notes in Computer Science, pages 107–118. Springer-Verlag, 2006.

[BCP09] Christelle Braun, Konstantinos Chatzikokolakis, and Catuscia Palamidessi. Quantitative notions of leakage for one-try attacks. In *Proceedings of the 25th Conf. on Mathematical Foundations of Programming Semantics*, volume 249 of *Electronic Notes in Theoretical Computer Science*, pages 75–91. Elsevier B.V., 2009.

[BK11] Gilles Barthe and Boris Köpf. Information-theoretic bounds for differentially private mechanisms. In *Proceedings of CSF*, 2011. To appear.

[BLP76] E. D. Bell and J. L. La Padula. Secure computer system: Unified exposition and multics interpretation, 1976.

[BP] Mohit Bhargava and Catuscia Palamidessi. Probabilistic anonymity. In Martín Abadi and Luca de Alfaro, editors, *Proceedings of CONCUR*, Lecture Notes in Computer Science, pages 171–185. Springer.

[BPS+09] Aaron Bohannon, Benjamin C. Pierce, Vilhelm Sjöberg, Stephanie Weirich, and Steve Zdancewic. Reactive noninterference. In Ehab Al-Shaer, Somesh Jha, and Angelos D. Keromytis,

editors, *Proceedings of the 2009 ACM Conference on Computer and Communications Security, CCS 2009, Chicago, Illinois, USA, November 9-13, 2009*, pages 79–90. ACM, 2009.

[BS94] Jose M. Bernardo and Adrian F. M. Smith. *Bayesian Theory*. John Wiley & Sons, Inc., 1994.

[BS01] Emanuele Bandini and Roberto Segala. Axiomatizations for probabilistic bisimulation. In *Proceedings of the 28th International Colloquium on Automata, Languages and Programming*, volume 2076 of *Lecture Notes in Computer Science*, pages 370–381. Springer, 2001.

[BvW92] R. J. R. Back and J. von Wright. Combining angels, demons and miracles in program specifications. *Theoretical Computer Science*, 100(2):365–383, 1992.

[Cac97] Christian Cachin. *Entropy Measures and Unconditional Security in Cryptography*. PhD thesis, Zürich, Switzerland, 1997.

[CCK+06a] Ran Canetti, Ling Cheung, Dilsun Kaynar, Moses Liskov, Nancy Lynch, Olivier Pereira, and Roberto Segala. Task-structured probabilistic i/o automata. In *Proceedings the 8th International Workshop on Discrete Event Systems (WODES'06)*, Ann Arbor, Michigan, 2006.

[CCK+06b] Ran Canetti, Ling Cheung, Dilsun Kirli Kaynar, Moses Liskov, Nancy A. Lynch, Olivier Pereira, and Roberto Segala. Time-bounded task-PIOAs: A framework for analyzing security protocols. In Shlomi Dolev, editor, *Proceedings of the 20th International Symposium in Distributed Computing (DISC '06)*, volume 4167 of *Lecture Notes in Computer Science*, pages 238–253. Springer, 2006.

[Cha88] D. Chaum. The dining cryptographers problem: unconditional sender and recipient untraceability. *J. Cryptol.*, 1:65–75, March 1988.

[CHM05] David Clark, Sebastian Hunt, and Pasquale Malacaria. Quantitative information flow, relations and polymorphic types. *J. of Logic and Computation*, 18(2):181–199, 2005.

[CHM07] David Clark, Sebastian Hunt, and Pasquale Malacaria. A static analysis for quantifying information flow in a simple imperative language. *J. Comput. Secur.*, 15:321–371, August 2007.

[CNP09] Konstantinon Chatzikokolakis, Gethin Norman, and David Parker. Bisimulation for demonic schedulers. In Luca de Alfaro, editor, *Proc. of the Twelfth International Conference on Foundations of Software Science and Computation Structures (FOSSACS 2009)*, volume 5504 of *Lecture Notes in Computer Science*, pages 318–332, York, UK, March 2009 2009. Springer.

[CP] Konstantinos Chatzikokolakis and Catuscia Palamidessi. Making random choices invisible to the scheduler. In Luís Caires and Vasco Thudichum Vasconcelos, editors, *Proceedings of the 18th International Conference on Concurrency Theory (CONCUR 2007)*, Lecture Notes in Computer Science, pages 42–58. Springer.

[CP06] Konstantinos Chatzikokolakis and Catuscia Palamidessi. Probable innocence revisited. *Theoretical Computer Science*, 367(1-2):123–138, 2006.

[CPP08a] Konstantinos Chatzikokolakis, Catuscia Palamidessi, and Prakash Panangaden. Anonymity protocols as noisy channels. *Inf. and Comp.*, 206(2–4):378–401, 2008.

[CPP08b] Konstantinos Chatzikokolakis, Catuscia Palamidessi, and Prakash Panangaden. On the bayes risk in information-hiding protocols. *J. Comput. Secur.*, 16:531–571, December 2008.

[CS08] Michael R. Clarkson and Fred B. Schneider. Hyperproperties. In *Computer Security Foundations Symposium*, pages 51–65, Los Alamitos, CA, USA, 2008. IEEE Computer Society.

[CS11] M. R. Clarkson and F. B. Schneider. Quantification of integrity, 2011. Tech. Rep.. http://hdl.handle.net/1813/22012.

[Csi95] Imre Csiszár. Generalized cutoff rates and Rényi's information measures. *Transactions on Information Theory*, 41(1):26–34, 1995.

[CT91] Thomas M. Cover and Joy A. Thomas. *Elements of Information Theory*. John Wiley & Sons, Inc., 1991.

[CT06] Thomas M. Cover and Joy A. Thomas. *Elements of Information Theory*. John Wiley & Sons, Inc., second edition, 2006.

[dAHJ01] Luca de Alfaro, Thomas A. Henzinger, and Ranjit Jhala. Compositional methods for probabilistic systems. In Kim Guldstrand Larsen and Mogens Nielsen, editors, *Proceedings of the 12th International Conference on Concurrency Theory (CONCUR 2001)*, volume 2154 of *Lecture Notes in Computer Science*. Springer, 2001.

[Dal77] Tore Dalenius. Towards a methodology for statistical disclosure control. *Statistik Tidskrift*, 15:429 — 444, 1977.

[DCPP06] Yuxin Deng, Tom Chothia, Catuscia Palamidessi, and Jun Pang. Metrics for action-labelled quantitative transition systems. In *Proceedings of the Third Workshop on Quantitative Aspects of Programming Languages (QAPL 2005)*, volume 153 of *Electronic Notes in Theoretical Computer Science*, pages 79–96. Elsevier Science Publishers, 2006.

[DDM10] J. Dubreil, P. Darondeau, and H. Marchand. Supervisory control for opacity. *IEEE Transactions on Automatic Control*, 55(5):1089 –1100, May 2010.

[Den82] Dorothy E. Denning. Cryptography and data security, 1982.

[DJGP02] Josee Desharnais, Radha Jagadeesan, Vineet Gupta, and Prakash Panangaden. The metric analogue of weak bisimulation for probabilistic processes. In *Proceedings of the 17th Annual IEEE Symposium on Logic in Computer Science*, pages 413–422. IEEE Computer Society, 2002.

[DKR09] Stéphanie Delaune, Steve Kremer, and Mark Ryan. Verifying privacy-type properties of electronic voting protocols. *Journal of Computer Security*, 17(4):435–487, 2009.

[DL09] Cynthia Dwork and Jing Lei. Differential privacy and robust statistics. In *Proc. of the 41st Annual ACM Symposium on Theory of Computing, STOC 2009, Bethesda, MD, USA, May 31 - June 2, 2009*, pages 371–380. ACM, 2009.

[DORS04] Yevgeniy Dodis, Rafail Ostrovsky, Leonid Reyzin, and Adam Smith. Fuzzy extractors: How to generate strong keys from biometrics and other noisy data. technical report 2003/235, cryptology eprint archive, http://eprint.iacr.org, 2006. previous version appeared at eurocrypt 2004. In *34 [DRS07] [DS05] [EHMS00] [FJ01] Yevgeniy Dodis, Leonid Reyzin, and Adam*, pages 79–100. Springer-Verlag, 2004.

[DPP05] Yuxin Deng, Catuscia Palamidessi, and Jun Pang. Compositional reasoning for probabilistic finite-state behaviors. In Aart Middeldorp, Vincent van Oostrom, Femke van Raamsdonk, and Roel C. de Vrijer, editors, *Processes, Terms and Cycles: Steps on the Road to Infinity*, volume 3838 of *Lecture Notes in Computer Science*, pages 309–337. Springer, 2005.

[DPW06] Yuxin Deng, Jun Pang, and Peng Wu. Measuring anonymity with relative entropy. In T. Dimitrakos, F. Martinelli, P. Y. A. Ryan, and S. A. Schneider, editors, *Proc. of the of the 4th Int. Worshop on Formal Aspects in Security and Trust*, volume 4691 of *LNCS*, pages 65–79. Springer, 2006.

[D.S86] D.Sutherland. A model of information. In *Proceedings of the 9th National Computer Security Conference*, 1986.

[DSCP02] Claudia Díaz, Stefaan Seys, Joris Claessens, and Bart Preneel. Towards measuring anonymity. In Roger Dingledine and Paul F. Syverson, editors, *Proceedings of the workshop on Privacy Enhancing Technologies (PET) 2002*, volume 2482 of *Lecture Notes in Computer Science*, pages 54–68. Springer, 2002.

[Dwo06] Cynthia Dwork. Differential privacy. In *Automata, Languages and Programming, 33rd Int. Colloquium, ICALP 2006, Venice, Italy, July 10-14, 2006, Proc., Part II*, volume 4052 of *LNCS*, pages 1–12. Springer, 2006.

[Dwo10] Cynthia Dwork. Differential privacy in new settings. In *Proc. of the Twenty-First Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2010, Austin, Texas, USA, January 17-19, 2010*, pages 174–183. SIAM, 2010.

[Dwo11] Cynthia Dwork. A firm foundation for private data analysis. *Communications of the ACM*, 54(1):86–96, 2011.

[Eba] Ebay website. http://www.ebay.com/.

[Ebi] The ebid website. http://www.ebid.net/.

[Gal68] Robert G. Gallager. *Information Theory and Reliable Communication*. John Wiley & Sons, New York, NY, 1968.

[GD07] Sergio Giro and Pedro R. D'Argenio. Quantitative model checking revisited: Neither decidable nor approximable. In Jean-Fraï¿½ois Raskin and P. S. Thiagarajan, editors, *Proceedings of the 5th International Conference on Formal Modeling and Analysis of Timed Systems (FORMATS)*, volume 4763 of *Lecture Notes in Computer Science*, pages 179–194. Springer, 2007.

[GDF09] Sergio Giro, Pedro R. D'Argenio, and Luis María Ferrer Fioriti. Partial order reduction for probabilistic systems: A revision for distributed schedulers. In Mario Bravetti and Gianluigi Zavattaro, editors, *Proceedings of the 20th International Conference on Concurrency Theory*, volume 5710 of *Lecture Notes in Computer Science*, pages 338–353. Springer, 2009.

[Gir09] Sergio Giro. Undecidability results for distributed probabilistic systems. In Marcel Vinicius Medeiros Oliveira and Jim Woodcock, editors, *12th Brazilian Symposium on Foundations and Applications of Formal Methods (SBMF)*, volume 5902 of *Lecture Notes in Computer Science*, pages 220–235. Springer, 2009.

[GM82] Joseph A. Goguen and José Meseguer. Security policies and security models. In *IEEE Symposium on Security and Privacy*, pages 11–20, 1982.

[Gra91] J. W. Gray, III. Toward a mathematical foundation for information flow security. In *Proceedings of the 1991 IEEE Computer Society Symposium on Research in Security and Privacy (SSP '91)*, pages 21–35, Washington - Brussels - Tokyo, May 1991. IEEE.

[GRS09] Arpita Ghosh, Tim Roughgarden, and Mukund Sundararajan. Universally utility-maximizing privacy mechanisms. In *Proceedings of the 41st annual ACM symposium on Theory of computing*, STOC '09, pages 351–360, New York, NY, USA, 2009. ACM.

[HJ89] H. Hansson and B. Jonsson. A framework for reasoning about time and reliability. In *Proceedings of the 10th IEEE Symposium on Real-Time Systems*, pages 102–111, Santa Monica, California, USA, 1989. IEEE Computer Society Press.

[HM09] Jonathan Heusser and Pasquale Malacaria. Applied quantitative information flow and statistical databases. In *Formal Aspects in Security and Trust*, pages 96–110, 2009.

[HO03] Joseph Y. Halpern and Kevin R. O'Neill. Anonymity and information hiding in multiagent systems. In *Proc. of the 16th IEEE Computer Security Foundations Workshop*, pages 75–88, 2003.

[HP00] Oltea Mihaela Herescu and Catuscia Palamidessi. Probabilistic asynchronous π-calculus. In Jerzy Tiuryn, editor, *Proceedings of FOSSACS 2000 (Part of ETAPS 2000)*, volume 1784 of *Lecture Notes in Computer Science*, pages 146–160. Springer, 2000.

[HP05] Joseph Y. Halpern and Riccardo Pucella. Probabilistic algorithmic knowledge. *Journal of Logical Methods in Computer Science*, 3(1), 2005.

[HR07] M.E. Hellman and J. Raviv. Probability of error, equivocation, and the Chernoff bound. *IEEE Trans. on Information Theory*, IT–16:368–372, 2007.

[Jac89] Jeremy Jacob. On the derivation of secure components. In *Proc. of the 1989 IEEE Symposium on Security and Privacy, S&P'89*, pages 242–247, Oakland, CA, USA, 1989. IEEE Comput. Soc. Press.

[Joi01] Adam N. Joinson. Self-disclosure in computer-mediated communication: The role of self-awareness and visual anonymity. *Eur. J. Soc. Psychol.*, 31(2):177–192, 2001.

[Kan42] Leonid Kantorovich. On the transfer of masses (in Russian). *Doklady Akademii Nauk*, 5(1):1–4, 1942. Translated in *M*anagement Science, 5(1):1–4, 1958.

[KB07] Boris Köpf and David A. Basin. An information-theoretic model for adaptive side-channel attacks. In Peng Ning, Sabrina De Capitani di Vimercati, and Paul F. Syverson, editors, *Proceedings of the 2007 ACM Conference on Computer and Communications Security, CCS 2007, Alexandria, Virginia, USA, October 28-31, 2007*, pages 286–296. ACM, 2007.

[KS] Shiva Prasad Kasiviswanathan and Adam Smith. A note on differential privacy: Defining resistance to arbitrary side information. *CoRR*.

[Mal07] Pasquale Malacaria. Assessing security threats of looping constructs. In Martin Hofmann and Matthias Felleisen, editors, *Proceedings of the 34th ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages, POPL 2007, Nice, France, January 17-19, 2007*, pages 225–235. ACM, 2007.

[Mas90] James L. Massey. Causality, feedback and directed information. In *Proc. of the 1990 Intl. Symposium on Information Theory and its Applications*, November 1990.

[Mas94] Massey. Guessing and entropy. In *Proceedings of the IEEE International Symposium on Information Theory*, page 204. IEEE, 1994.

[MC08] Pasquale Malacaria and Han Chen. Lagrange multipliers and maximum information leakage in different observational models. In Úlfar Erlingsson and Marco Pistoia, editor, *Proceedings of the 2008 Workshop on Programming Languages and Analysis for Security (PLAS 2008)*, pages 135–146, Tucson, AZ, USA, June 2008. ACM.

[McC87] Daryl McCullough. Specifications for multi-level security and a hook-up. *Security and Privacy, IEEE Symposium on*, 0:161, 1987.

[McL90] John McLean. Security models and information flow. In *SSP'90*, pages 180–189. IEEE, 1990.

[MCR07] C. E. Martin, S. A. Curtis, and I. Rewitzky. Modelling angelic and demonic nondeterminism with multirelations. *Science of Computer Programming*, 65(2):140–158, 2007.

[Mer] Mercadolibre website. http://www.mercadolibre.com/.

[Mil87] Jonathan K. Millen. Covert channel capacity. In *IEEE Symposium on Security and Privacy*, pages 60–66, 1987.

[Mil89] R. Milner. *Communication and Concurrency*. International Series in Computer Science. Prentice Hall, 1989.

[Mil90] Jonathan K. Millen. Hookup security for synchronous machines. In *Proceedings of the 3rd IEEE Computer Security Foundations Workshop (CSFW)*, pages 84–90, 1990.

[MM03] Annabelle McIver and Carroll Morgan. *A probabilistic approach to information hiding*, pages 441–460. Springer-Verlag New York, Inc., New York, NY, USA, 2003.

[MNCM03] Ira S. Moskowitz, Richard E. Newman, Daniel P. Crepeau, and Allen R. Miller. Covert channels and anonymizing networks. In Sushil Jajodia, Pierangela Samarati, and Paul F. Syverson, editors, *Workshop on Privacy in the Electronic Society 2003*, pages 79–88. ACM, 2003.

[MNS03] Ira S. Moskowitz, Richard E. Newman, and Paul F. Syverson. Quasi-anonymous channels. In *Proc. of CNIS*, pages 126–131. IASTED, 2003.

[Mor09] Carroll Morgan. The shadow knows: Refinement and security in sequential programs. *Science of Computer Programming*, 74(8):629–653, 2009.

[PDH08] Andreas Pfitzmann, Tu Dresden, and Marit Hansen. Anonymity, unlinkability, undetectability, unobservability, pseudonymity, and identity management: A consolidated proposal for terminology, 2008.

[PH05] Catuscia Palamidessi and Oltea M. Herescu. A randomized encoding of the π-calculus with mixed choice. *Theoretical Computer Science*, 335(2-3):373–404, 2005.

[Pli00] Pliam. On the incomparability of entropy and marginal guesswork in brute-force attacks. In *Proceedings of INDOCRYPT: International Conference in Cryptology in India*, number 1977 in Lecture Notes in Computer Science, pages 67–79. Springer-Verlag, 2000.

[Pou92] William Poundstone. *Prisoners Dilemma*. Doubleday NY, 1992.

[Ré61] Alfréd Rényi. On Measures of Entropy and Information. In *Proceedings of the 4th Berkeley Symposium on Mathematics, Statistics, and Probability*, pages 547–561, 1961.

[RR98] Michael K. Reiter and Aviel D. Rubin. Crowds: anonymity for Web transactions. *ACM Transactions on Information and System Security*, 1(1):66–92, 1998.

[SA99] Frank Stajano and Ross J. Anderson. The cocaine auction protocol: On the power of anonymous broadcast. In *Information Hiding*, pages 434–447, 1999.

[SD02] Andrei Serjantov and George Danezis. Towards an information theoretic metric for anonymity. In Roger Dingledine and Paul F. Syverson, editors, *Proceedings of the workshop on Privacy Enhancing Technologies (PET) 2002*, volume 2482 of *Lecture Notes in Computer Science*, pages 41–53. Springer, 2002.

[Seg95] Roberto Segala. *Modeling and Verification of Randomized Distributed Real-Time Systems*. PhD thesis, June 1995. Tech. Rep. MIT/LCS/TR-676.

[Sky03] Brian Skyrms. *The Stag Hunt and the Evolution of Social Structure*. Cambridge University Press, 2003.

[SL95] Roberto Segala and Nancy Lynch. Probabilistic simulations for probabilistic processes. *Nordic Journal of Computing*, 2(2):250–273, 1995. An extended abstract appeared in *Proceedings of CONCUR '94*, LNCS 836: 481-496.

[Smi07] Geoffrey Smith. Adversaries and information leaks (tutorial). In Gilles Barthe and Cédric Fournet, editors, *Proceedings of the Third Symposium on Trustworthy Global Computing*, volume 4912 of *Lecture Notes in Computer Science*, pages 383–400. Springer, 2007.

[Smi09] Geoffrey Smith. On the foundations of quantitative information flow. In Luca de Alfaro, editor, *Proc. of the 12th Int. Conf. on Foundations of Software Science and Computation Structures*, volume 5504 of *LNCS*, pages 288–302, York, UK, 2009. Springer.

[SS96] Steve Schneider and Abraham Sidiropoulos. CSP and anonymity. In *Proc. of the European Symposium on Research in Computer Security (ESORICS)*, volume 1146 of *Lecture Notes in Computer Science*, pages 198–218. Springer, 1996.

[Sta06] William Stallings. *Data and Computer Communications*. Prentice Hall, eigth edition, 2006.

[Sub98] Srividhya Subramanian. Design and verification of a secure electronic auction protocol. In *Proceedings of the 17th IEEE Symposium on Reliable Distributed Systems*, pages 204–210, Los Alamitos, CA, USA, 1998. IEEE Computer Society.

[SV06] Nandakishore Santhi and Alexander Vardy. On an improvement over Rényi's equivocation bound, 2006. Presented at the 44-th Annual Allerton Conference on Communication, Control, and Computing, September 2006. Available at http://arxiv.org/abs/cs/0608087.

[Tan89] Andrew Tanenbaum. *Computer Networks*. Prentice Hall, second edition, 1989.

[Tar55] Alfred Tarski. A lattice-theoretical fixpoint theorem and its applications. *Pacific Journal of Mathematics*, 5(2):285—309, 1955.

[TM09] Sekhar Tatikonda and Sanjoy K. Mitter. The capacity of channels with feedback. *IEEE Transactions on Information Theory*, 55(1):323–349, 2009.

[vBW01] Franck van Breugel and James Worrell. Towards quantitative verification of probabilistic transition systems. In Fernando Orejas, Paul G. Spirakis, and Jan van Leeuwen, editors, *Proceedings of the 28th International Colloquium on Automata, Languages and Programming (ICALP)*, volume 2076 of *Lecture Notes in Computer Science*, pages 421–432. Springer, 2001.

[Vic61] William Vickrey. Counterspeculation, Auctions, and Competitive Sealed Tenders. *The Journal of Finance*, 16(1):8–37, 1961.

[WJ90] J. Todd Wittbold and Dale M. Johnson. Information flow in nondeterministic systems. In *IEEE Symposium on Security and Privacy*, pages 144–161, 1990.

[ZB05] Ye Zhu and Riccardo Bettati. Anonymity vs. information leakage in anonymity systems. In *Proc. of ICDCS*, pages 514–524. IEEE Computer Society, 2005.