IMT Institute for Advanced Studies, Lucca

Lucca, Italy

Information-Theoretic Models of Confidentiality and Privacy

PhD Program in Computer Science and Engineering

XXVI Cycle

By

Michela Paolini

2014

The dissertation of Michela Paolini is approved.

Program Coordinator: Prof. Rocco De Nicola, IMT Institute for Advanced Studies, Lucca

Supervisor: Prof. Michele Boreale, University of Florence

Tutor: Dr. Francesco Tiezzi, IMT Institute for Advanced Studies, Lucca

The dissertation of Michela Paolini has been reviewed by:

Dr. Boris Köpf, IMDEA Software Institute

Prof. Geoffrey Smith, Florida International University

IMT Institute for Advanced Studies, Lucca

2014

Contents

List of Figures

List of Tables

Acknowledgements

At the end of any experience, we often feel in a cloud of sensations, that contains teaching and richness of the journey and that ranges from joy for the reached destination to uncertainty about the future. It is, therefore, natural, stop for a moment and look back towards the path, consisting, as all paths, of curves, climbs and descents, but, in my case, made up mostly of faces and people.

At first I would like to express my sincere gratitude to Prof. Michele Boreale for his continuous patience and loyalty, for his guidance, advices and help in these years and, moreover, for his support in every situations, even in my (several) downs.

My sincere thanks also goes to Prof.ssa Catuscia Palamidessi and to both Comète and Parsifal groups, that offered me the opportunity to meet new friends and to take part on very interesting projects. I thank Dr. Boris Köpf and Prof. Geoffrey Smith for their careful reading and suggestions about the present thesis. I thank also Prof. Fabio Corradi for his advices.

The results presented in this thesis are based on joint works with Prof. Michele Boreale, University of Florence, and Francesca Pampaloni, PhD awarded at IMT Institute for Advanced Studies, Lucca. In particular, Chapter 3 is based on (BPP11a; BPPar; BPP11b), joint works with Prof. Michele Boreale and Francesca Pampaloni. Chapter 4 is based on (BP12; BPb), joint works with Prof. Michele Boreale. Finally, Chapter 5 is based on the results presented in (BPa), a joint work with Prof. Michele Boreale.

Moving to people that shared with me everyday life, I spent just a few words (not because I have nothing to say, but because,

maybe, this is not the most appropriate forum). At first, I think with love to Giovanni, for being my living "miracle" (you know what I mean) and giving me hope and strength, even when I have not.

I refer also to my family and, in particular, to my parents, able, nevertheless, to make me feel always surrounded by that warmth, which is, at the same time, unconditional love and fruitful comparison. Many thank to Benni, that, with his joyful exuberance, always provides me affection and walks.

An affectionate thought goes to Francesca, my traveling companion, for her closeness and support, for our phone calls and mails, that, I think, we miss to both.

Last, but not the least, I would like to thank my guides in Assisi and the other students and friends, that have shared with me part of the street.

Vita

August 12, 1986	Viareggio (LU), Italy
October 2005 - October 2010	Bachelor and Master Degree in Mathematics University of Florence, Italy Final mark: 110/110 cum laude
March 2011 - Present	PhD Student IMT Lucca, Italy
October 2012 - December 2012	Visiting Student INRIA, École Polythecnique, Palaiseau, France

Publications

Journal papers:

1. M. Boreale, F. Pampaloni, M. Paolini. Asymptotic information leakage under one-try attacks (full version). *MSCS*, in press.
2. M. Boreale, M. Paolini. Worst- and average-case privacy breaches in randomization mechanisms (full version). Submitted.

Conference papers:

3. M. Boreale, F. Pampaloni, M. Paolini. Asymptotic information leakage under one-try attacks. *Proc. of FoSSaCS 2011, LNCS* 6604: pp. 396-410, 2011.
4. M. Boreale, F. Pampaloni, M. Paolini. Quantitative information flow, with a view. *Proc. of ESORICS 2011, LNCS* 6879: pp. 588-606, 2011
5. M. Boreale, M. Paolini. Worst- and Average-Case Privacy Breaches in Randomization Mechanisms. *IFIP TCS* 2012: pp. 72-86, 2012.
6. M. Boreale, M. Paolini. On formally bounding information leakage by statistical estimation. Submitted.

Presentations

Conference talks:

1. Worst- and Average-Case Privacy Breaches in Randomization Mechanisms, IFIP Theoretical Computer Science 2012, CWI Amsterdam, Holland, September 2012.

Meeting and project talks:

1. Worst- and Average-Case Privacy Breaches in Randomization Mechanisms. CINA Kick-off Meeting, Pisa, February 2013.
2. Formal bounds for information leakage by statistical estimation. CINA Second General Meeting, Bologna, February 2014.
3. On formally bounding information leakage by statistical estimation. ASCENS Meeting, Modena, March 2014.

Abstract

In a variety of contexts, involving execution of software or hardware manipulating sensitive data, one would like that public observables do not leak information that should be kept secret. Since it is practically impossible to avoid leakage entirely, there is a growing interest in the *quantitative* aspects of information flow analysis. This is also related to *privacy*, that is to protection of sensitive information concerning individuals.

In this context, we first study methods to measure the *average* amount of leakage due to system execution, quantifying the possibility of inferring the secret information from observables. We assume an attacker that can make a single guess after observing a certain number of independent executions of the program. We study the asymptotic behaviour of information leakage and the dichotomy between protection of the *whole secret* and of a *property of the secret*.

An average measure of leakage may not be adequate when privacy of individuals is at stake. To study this issue, we introduce a strong semantic notion of security, that expresses absence of any privacy breach above a given level of seriousness, irrespective of any background information. We then analyze this notion according to two dimensions: worst vs. average case, single vs. repeated observations, and clarify its relation to differential privacy.

Finally, motivated by the complexity of exact computation of quantitative information leakage, we study statistical approaches to its estimation, when only a black-box access to the system is provided, and little is known about the input generation mechanism.

Chapter 1

Introduction

In this chapter we first survey certain aspects of sensitive information protection that have motivated our studies; at the same time, we provide a literature review of the consolidated results related to our work, mainly in the area of Quantitative Information Flow (QIF). Finally, we present an overview of our contributions and the structure of the thesis.

1.1 Motivations and literature review

The problem of preventing the leakage of secret information is a paramount concern in the design and analysis of all kinds of computational systems. Ideally, the goal is to ensure *Noninterference* (GM82), which means complete absence of leakage. That is by observing the output of a system, one learns nothing about the corresponding input. It soon became evident, though, that non-interference is a too strong requirement for real systems. Indeed, sometimes one *needs* or *wants* to reveal information that depends on the secret input. For example, in a password checker, one needs to reject a wrong password, although this reveals information about what the secret password is not. And in medical research, one wants to collect personal medical information of a large population, while maintaining the privacy of single individuals. These and other examples have prompted researchers to consider weaker notions of security. By now, a

popular approach is *Quantitative* Information Flow (QIF): the aim is not providing a yes-or-no answer about system security, but rather trying to quantify the amount of leakage using techniques from Information Theory, see (CHM01; CPP08a; CPP08b; Bor09; Smi09; BCP09; KS10; Mal10; BPP11a; BPP11b). This idea lays the basis for analyses that are far more flexible than the rigid safe/unsafe classification provided by the classical Noninterference approach.

A general situation is that of a program, protocol or device carrying out computations that depend probabilistically on a secret piece of information, such as a password, the identity of a user or a private key. We collectively designate these as *information hiding systems*, following a terminology established in (CPP08a). During the computation, some observable information related to the secret may be disclosed. This might happen either by design, e.g. if the output of the system is directly related to the secret (think of a password checker denying access), or for reasons depending on the implementation. In the latter case, the observable information may even take the form of physical quantities, such as the execution time or the power consumption of the device: think of timing and power attacks on smart cards (Koc96; KJJ99). The observable information can be exploited by an eavesdropper to reconstruct the secret, or at least to limit the search space. The goal of QIF is to quantify how much, if any, information about the input is leaked by the program in the output, namely how much information about the input can be deduced observing the corresponding output, tolerating, possibly, small leakages. More precisely, we would like to measure the amount of information in the secret (adversary's prior uncertainty), the amount of leaked information (leakage) and the amount of *un*leaked information (adversary's posterior uncertainty). Intuitively, these quantities satisfy the following equation:

$$\text{leakage} = \text{prior uncertainty} - \text{posterior uncertainty}. \tag{1.1}$$

Example 1.1.1 *The following example is found in (Smi09). Assume x is a 32 bit integer input variable. The program P_1 is a constant function, with just one possible output. In this case, each input is mapped in the same output and, looking at it, we are not able to infer anything about the secret. Hence there is no leakage and Noninterference holds. The program P_2 is the identity function: here each*

Program	Initial uncertainty	Remaining uncertainty	Leakage
$P_1(x) = 0$	32	32	0
$P_2(x) = x$	32	0	32
$P_3(x) = x \,\&\, 037_{16}$	32	27	5

Table 1: Three different programs and the corresponding leakage.

input is mapped to itself and the leakage is total. P_3 is an intermediate case, where the program copies the last 5 bits of input to the output: P_3 leaks 5 out of the 32 bits of input (in Table 1 & is the bitwise-`and`* operator and 037 is a hexadecimal constant).*

Note that, in the case of *deterministic* programs, once the input is fixed, the output is determined. Therefore, even if the adversary is allowed to run a program several times, while keeping the secret input fixed, he/she does not benefit from the collection of these repeated observations. However, in security applications, programs are often *probabilistic*, either due to the presence of noise or by design. Indeed, in a variety of contexts, randomization is regarded as an effective technique to conceal sensitive information. For example, anonymity protocols like Crowds (RR98) or the Dining Cryptographers (Cha88) rely on randomization to "confound" the adversary as to the true actions undertaken by each participant. In a probabilistic scenario, the analysis of repeated observations may be crucial. Indeed, in real-world situations, re-execution may happen either forced by the attacker (as in the case of an adversary querying several times a smart card), or by design (as in the case of routing paths established, repeatedly, between a sender and a receiver in anonymity protocols like Crowds (RR98)). Intuitively, if the program can be ran just once, the information leaked is not much, but, if one is allowed with repeated observations, the amount of leaked information increases. This intuition finds application in statistical attacks against secrecy, anonymity, privacy and other confidential properties in systems that handle sensitive data. In these attacks, the adversary gets to know a sample of observations of a target system – such as timing or power traces of a smart-card (Koc96), attribute values in a dataset (NS08a), etc. – and, exploiting some form of

correlation existing between the secret and the observables, tries to infer the secret – the private key, the identity of an individual, etc.

When the sensitive information meant to be protected concerns individuals, we enter the realm of *privacy*. Here, considering an average notion of leakage may not be adequate. Indeed, one may argue that there are systems that on average appear to be secure, but from the point of view of specific users are not, since those users may be easily exposed to eavesdropping. For instance, an anonymity protocol which groups users into a small number of "indistinguishability" classes is considered as secure on average. However, it might well be the case that, while the vast majority of users belong to large classes, few individual users belong to singleton classes, hence being exposed to eavesdropping. These issues are well-known in the area of databases, where often the goal is to learn properties of the population as a whole, while maintaining the privacy of individuals. In medical research, it is desirable to collect personal medical information of a large number of individuals. Researchers or public authorities can calculate a series of statistics from the sample and decide, for instance, how much money the health care system should spend next year in the treatment of a specific disease. It is desirable that the participation in the sample will not damage the privacy of any individual: usually people do not want to have disclosed their specific status with respect to diseases, or other personal matters. The fact that the answer is publicly available constitutes a threat for the privacy of the individuals. For instance, assume that we are interested in the query "what is the percentage of individuals with a given disease?". The addition of an individual to the database will, in general, modify the percentage, and reveal whether the individual has the disease or not to anyone who is informed about the insertion of the individual in the database.

A proposed solution to the above problem is to introduce some output perturbation mechanism based on *randomization*: instead of the exact answer to the query, a *noisy* answer is reported. For example, in the field of Data Mining, techniques have been proposed by which datasets containing personal information that are released for business or research purposes are often perturbed with noise, so as to prevent an adversary from re-

identifying individuals or learning sensitive information about them (see e.g. (EGS03)).

The notion of *differential privacy*, due to Dwork and her collaborators (DMNS06; Dwo06), is a proposal for controlling the risk of violating privacy. The idea is that a randomization mechanism satisfies ϵ-differential privacy (for some $\epsilon > 0$) if the ratio between the probability of obtaining any answer is bounded by e^{ϵ} "in two adjacent databases". Here by "adjacent" we mean that the databases differ for only one individual. Often we will abbreviate "ϵ-differential privacy" as ϵ-DP. Note that the smaller is ϵ, the greater is the degree of privacy protection. In particular, when ϵ is close to 0 the output of the randomization mechanism is nearly independent from the input. Unfortunately, such mechanism is practically useless. The *utility*, i.e. the capability to providing accurate information in the reported answers, is the other important feature of a randomization mechanism. It is clear that there is a trade-off between utility and privacy. These two notions are not exactly one against the other: utility concerns the relation between the reported answer and the real answer to the query, while privacy concerns the relation between the reported answer and the information in the database. This makes the problem of finding a good compromise between the two aspects very interesting.

At this point, we can make an analogy between the area of differential privacy (DP) and the one of quantitative information flow (QIF), or, more precisely, between the concept of DP and the quantitative notions of information-flow. First of all, note that a randomized mechanism can be seen as an information-theoretic channel, and that the limit case of $\epsilon = 0$, for which the privacy protection is total, corresponds to the case in which the answer to the query does not reveal any information about the input distribution, and therefore the leakage is 0. Furthermore while information-theoretic leakage, analyzed with QIF techniques, is mainly concerned with quantifying the degree of protection offered against an adversary trying to guess the *whole secret*, DP is rather concerned with the protection of *individual bits of the secret*, possibly in the presence of background information, like knowledge of the remaining bits (BK11; AACP11b). The investigation about the relation between the quantitative notion of leakage and DP is

not yet concluded, but our studies suggest that another important aspect concerns the difference between worst- and average-case privacy breaches. Indeed, as discussed above, when privacy is at stake, an average analysis may not guarantee protection of single individuals.

Another important aspect of the QIF area is that of computational methods for actually computing information leakage. Recent developments show that the problem of exactly computing information leakage of programs in general, or even that of giving nontrivial bounds that hold with certainty, turns out to be computationally intractable, even in the *deterministic* case (YT11). For this reason, there has been recently much interest towards methods for calculation of *approximate* information leakage (CCG10; CG11; CKN13; CK13; KR13). In many fields of Computer Science, a viable alternative to static analysis is represented by simulation. In the case of QIF, this prompts interesting research issues: to what extent, can one dispense with structural properties and adopt a black box approach, in conjunction with statistical techniques? What kind of formal guarantees can such an approach provide? And under what circumstances, if any, is it effective?

1.2 Terminology and structure of the thesis

Since there is no universal agreement on the exact meaning of terms like confidentiality, leakage and so on, in order to avoid confusion on the usage of such terms, we fix some terminology that we will use throughout the thesis. Next we outline the main contributions and the structure of the subsequent chapters.

Terminology. Information flow is concerned with the leakage of secret information through computer systems. QIF is a class of techniques that analyzes the dissemination of this secret information and the attribute "quantitative" refers to the fact that we are interested in measuring the amount of leakage, not only in detecting its presence. Confidentiality and privacy are two security properties that we can summarize as follows: confidentiality requires that no secret information is given to unauthorized

parties, while privacy is a special case of confidentiality, where the secret information is related to individuals. Hence both leakage (with respect to some notion of entropy) and DP are instances of confidentiality properties. Finally, we use the term security with a very general meaning, referred, according to the context, to both confidentiality and privacy issues. In the case where confusion might arise, the particular context will be explicitly specified.

Structure of the thesis. This thesis deals with issues concerning protection of *confidentiality* and *privacy* of sensitive information and, more specifically, with methodologies to analyze the information flow of the systems that process this information. We consider programs whose outputs probabilistically depend on a secret piece of information and the case of a *passive* attacker, that can observe repeatedly the outcomes of system executions and attempt statistical attacks.

The thesis is organized as follows.

- In Chapter 2 we introduce a simple model for QIF and the main concepts from Information Theory that we will use throughout the subsequent chapters. In particular, we compare the effectiveness of Shannon entropy and Rényi min-entropy as information leakage measures. Furthermore we review the main tools from Information Theory that we will use in the subsequent chapters.

- In Chapter 3, we propose a general analysis for assessment of system security against passive eavesdroppers under repeated, independent observations. We consider two attack scenarios: in the first case the adversary targets the secrets, while in the second one he/she targets some property related to it (named *view*). In both situations, we explore a security metric based on min-entropy and analyze its asymptotic behaviour as the number of observations collected by the attacker increases. We provide simple and tight bounds, showing that the convergence rate is exponential. The second scenario allows one to consider, not only *how much* information is leaked, but also *what* is leaked. We note that certain known statistical attacks can be

viewed as a direct application of our model. In particular, viewing privacy properties as a special case of views, we consider statistical attacks against privacy in sparse datasets. This chapter is based on (BPP11a; BPPar; BPP11b).

- In Chapter 4 we study privacy. We provide a semantic notion of security, that expresses the absence of any privacy breach above a given level of seriousness. We examine this notion according to two dimensions: worst vs. average case, single vs. repeated observations. In each case we characterize the security level achievable in a simple fashion. We next clarify the relation between our worst-case security notion and differential privacy, showing that, while the former is in general stronger, the two coincide in important special cases. Finally, we turn our attention to expected utility in the case of repeated independent observations. We characterize the exponential growth rate of any reasonable utility function. In the particular case of mechanisms providing differential privacy, we study the relation of the utility rate with the security level: we offer either exact expressions or upper-bounds for utility rate that apply to practically interesting cases, such as the (truncated) geometric mechanism. This chapter is based on (BP12; BPb).

- In Chapter 5, we study the problem of giving formal bounds on information leakage, when only a black-box access to the system is provided, and little is known about the input generation mechanism. After introducing a formal notion of information leakage *estimator*, we prove that, in the absence of significant a priori information about the output distribution, no such estimator can in fact exist that does significantly better than exhaustive search. Moreover, we show that the difficulty mostly lies in obtaining accurate *upper* bounds. This motivates us to consider a relaxed scenario, where the analyst is given some control over the input distribution: an estimator is introduced that, with high probability, gives lower bounds irrespective of the underlying distribution, and accurate upper bounds if the input distribution induces a "close to uniform" output distribution.

We then define sampling methods that ideally compute one such input distribution, and discuss a practical methodology based on those. We finally demonstrate the proposed methodology with a few simulations, which give encouraging results. This chapter is based on (BPa).

- In Chapter 6 we draw some concluding remarks.
- Appendix A contains some additional technical material.

Chapter 2

Preliminaries on QIF and Information Theory

In this chapter we introduce a simple model for QIF and review a few important notions from Information Theory. In particular, we present measures for quantifying information leakage and discuss their actual usage. Next, we review some tools from Information Theory that will be used throughout the subsequent chapters.

2.1 Information hiding systems and randomization mechanisms

We consider a program or protocol that receives some secret inputs and produces public outputs. An adversary, observing the output, can infer some information about the secret. We assume an input set $\mathcal{X} = \{x_1, x_2, ...\}$ and an output set $\mathcal{Y} = \{y_1, y_2, ...\}$, which are both finite and nonempty. The goal of QIF is to quantify the security threat to the secret input caused by disclosing the corresponding output. Starting from (1.1), one can formalize this intuition both in the case of deterministic and probabilistic programs.

Deterministic programs. We view a deterministic program P simply as a function $P : \mathcal{X} \to \mathcal{Y}$. For simplicity, we restrict our attention to termi-

nating programs, or, equivalently, assume termination is an observable value in $\mathcal{Y}$. We model the program's input as a random variable X taking values in $\mathcal{X}$ and distributed according to some distribution p, written $X \sim p(x)$. Once fed to a program P, X induce an output $Y = P(X)$. We let $p(x|y) \stackrel{\triangle}{=} \Pr(X = x|Y = y)$ denote the a posteriori input distribution after observation of output.

Note that P induces an equivalence relation $\sim$ on $\mathcal{X}$

$$x \sim x' \quad \text{iff} \quad P(x) = P(x')$$

Hence program P partitions $\mathcal{X}$ in $|\mathrm{Im}(P)|$ equivalence classes induced by $\sim$. We denote with $\mathcal{X}_y$ the equivalence class $P^{-1}(y)$ for any y such that $P^{-1}(y) \neq \emptyset$:

$$\mathcal{X}_y = \{x \in \mathcal{X} | P(y) = x\} = P^{-1}(y).$$

Note that the classes $\mathcal{X}_y$ bound the knowledge of the adversary. Indeed if the attacker sees an output y, he knows only that the corresponding input belongs to the equivalence class $\mathcal{X}_y$. But how much this equivalence class reveals about the secret?

Example 2.1.1 *Consider the situation described in Example 1.1.1 and in Tab. 1. In this case of P_1 there is only an equivalence class that coincides with the entire $\mathcal{X}$. Concerning P_2, the equivalence classes are all singletons and we experiment total leakage of information*[1]*. Finally, in P_3, the equivalence relation $\sim$ has $2^5 = 32$ different classes, each containing 2^{27} elements.*

Probabilistic programs. More generally, P could be a probabilistic program. In this case fixed a given input, we can obtain different outputs each with a certain probability. This means that P does not partition $\mathcal{X}$ as in the deterministic case. Following (CPP08a), we can model a probabilistic program using a conditional probability matrix $p(\cdot|\cdot) \in [0,1]^{\mathcal{X}\times\mathcal{Y}}$, whose rows represent the inputs and whose columns are indexed by the output and each entry specifies the conditional probability of obtaining a certain output given a fixed input, namely $p(Y = y|X = x)$. Note that each row

[1] As argued by Smith in (Smi09), the adversary might not be able to efficiently compute the equivalence classes. But here we are adopting a worst-case, information-theoretic point of view, ignoring the computational aspect.

of the matrix sums up to 1. Deterministic programs are a special case of probabilistic ones, where each row has one entry equal to 1 and all the others equal to 0.

Definition 1 (Information Hiding System) *An* information hiding system *is a quadruple $\mathcal{H} = (\mathcal{X}, \mathcal{Y}, p(\cdot), p(\cdot|\cdot))$, composed by a finite set of inputs $\mathcal{X}$ representing the secret information, a finite set of observable outputs $\mathcal{Y}$, an a priori distribution on $\mathcal{X}$, $p(\cdot)$, and a conditional probability matrix, $p(\cdot|\cdot) \in [0,1]^{\mathcal{X}\times\mathcal{Y}}$, where each row sums up to 1.*

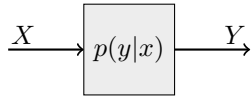


Figure 1: An information hiding system

In some scenarios, it may happen that the prior distribution on $\mathcal{X}$ is unknown. Then we speak about randomization mechanism, instead of information hiding systems.

Definition 2 (randomization mechanism (BP12; BPb)) *A* randomization mechanism *is a triple $\mathcal{R} = (\mathcal{X}, \mathcal{Y}, p(\cdot|\cdot))$, composed by a finite set of* inputs $\mathcal{X}$ *representing the secret information, a finite set of* observables $\mathcal{Y}$ *representing the observable values, and a* conditional probability matrix, $p(\cdot|\cdot) \in [0,1]^{\mathcal{X}\times\mathcal{Y}}$, *where each row sums up to 1.*

For each x, the x-th row of the matrix is identified with the probability distribution $y \mapsto p(y|x)$ on $\mathcal{Y}$, denoted by p_x. We say $\mathcal{R}$ is *non-singular* if $x \neq x'$ implies $p_x \neq p_{x'}$, and *strictly positive* if $p(y|x) > 0$ for each x and y.

Example 2.1.2 *The following example is inspired by (EGS03). The secret information is represented by the set of integers $\mathcal{X} = \{0, ..., 5\}$, and $\mathcal{Y} = \mathcal{X}$. We consider a mechanism that replaces any $x \in \mathcal{X}$ by a number y that retains some information about the original x. More precisely, we let $Y = \lfloor X + \xi \rfloor \mod 6$, where with probability 0.5 ξ is a chosen uniformly at random in $\{-\frac{1}{2}, \frac{1}{2}\}$, and with probability 0.5 it is chosen uniformly at random in $\mathcal{X}$. We can easily compute*

the resulting conditional probability matrix

$$\begin{pmatrix} 0.2500 & 0.2500 & 0.0833 & 0.0833 & 0.0833 & 0.2500 \\ 0.2500 & 0.2500 & 0.2500 & 0.0833 & 0.0833 & 0.0833 \\ 0.0833 & 0.2500 & 0.2500 & 0.2500 & 0.0833 & 0.0833 \\ 0.0833 & 0.0833 & 0.2500 & 0.2500 & 0.2500 & 0.0833 \\ 0.0833 & 0.0833 & 0.0833 & 0.2500 & 0.2500 & 0.2500 \\ 0.2500 & 0.0833 & 0.0833 & 0.0833 & 0.2500 & 0.2500 \end{pmatrix}.$$

Leakage and capacity. We provide a general definition of information leakage and capacity. Consider a randomization mechanism $\mathcal{R}$, as defined in Def. 2, and two (joint) random variables: X, representing the input, and the corresponding output Y. The probability distribution p on $\mathcal{X}$ and the conditional probability $p(y|x)$ together induce a probability distribution q on $\mathcal{X} \times \mathcal{Y}$ defined as $q(x, y) \triangleq p(x) \cdot p(y|x)$, hence a pair of random variables $(X, Y) \sim q$, with X taking values in $\mathcal{X}$ and Y taking values in $\mathcal{Y}$.

Following (1.1), we can measure information leakage as reduction in the adversary's uncertainty, where the *prior uncertainty* is modeled as a function of the secret X, $U(X)$, while the posterior *uncertainty*, $U(X|Y)$, takes into account the observation of Y. In the following section, these uncertainty functions will be instantiated using some forms of entropy.

The maximum leakage we can have is called *capacity*. Formally, we have the following definition.

Definition 3 (Information leakage, Capacity) *The* information leakage due to an input distribution p and a randomization mechanism $\mathcal{R}$ *be defined as*

$$\mathcal{L}(\mathcal{R}; p) \triangleq U(X) - U(X|Y).$$

The entropy capacity *of* $\mathcal{R}$ *is* $C(\mathcal{R}) \stackrel{\triangle}{=} \sup_p \mathcal{L}(\mathcal{R}; p)$.

Note that in the case of an IHS, we simply write $\mathcal{L}(\mathcal{H})$, since the prior probability is already included in the definition of $\mathcal{H}$ (see Def. 1). Finally when we deal with a deterministic program P, we write $\mathcal{L}(P; p)$. We study different forms of uncertainty functions in the following sections.

2.2 Quantifying uncertainty via entropy

Given a (deterministic or probabilistic) program P, we discuss how to measure the amount of information leaked. In literature different measures

exist, each one relying on an underling notion of entropy.

Each entropy notion is related to the type of attacker that we want to model, and to the way we measure its success. Widely used is Shannon entropy (Sha), which counts the average number of binary questions necessary to determine the exact value of the secret with an optimal strategy. Another popular notion of entropy is the min-entropy, proposed by Rényi (Rén61). It corresponds to a one-try attack and it is precisely the negative logarithm of the probability of guessing the true value of the secret, with an optimal strategy. The latter consists, of course, in selecting the element $x \in \mathcal{X}$ with the highest probability. Below we review these measures and another one: the Guessing Entropy discussed by Massey in (Mas94), which will serve as a trait d'union between the first two.

Notation. Let $\mathcal{X}$ be a finite nonempty set. A probability distribution on $\mathcal{X}$ is a function $p : \mathcal{X} \to [0, 1]$ such that $\sum_{x \in \mathcal{X}} p(x) = 1$. The *support* of p is defined as $\mathrm{supp}(p) \stackrel{\triangle}{=} \{x \in \mathcal{X} | p(x) > 0\}$. Given a random variable X taking values in $\mathcal{X}$, we write $X \sim p(x)$ if X is distributed according to $p(x)$, that is for each $x \in \mathcal{X}$, $\Pr(X = x) = p(x)$. We shall only consider discrete random variables. We shall use abbreviations such as $p(y|x)$ for $\Pr(Y = y|X = x)$, whenever no confusion arises about the involved random variables X and Y. Finally, when convenient, we shall denote the conditional probability distribution on $\mathcal{Y}$ given x, $p(\cdot|x)$, $(x \in \mathcal{X})$ by $p_x(\cdot)$. Note that, in what follows, we use $\log$ for the $\log_2$, unless differently specified.

2.2.1 Shannon entropy

One of the more fundamental concepts in Information Theory is entropy. In the context of Security, one's aim is to formalize and quantify the adversary uncertainty, intended as the number of missing bits, in order to completely disclose a result of interest. Let X be a generic random variable, taking values in a finite set $\mathcal{X}$, then the entropy of X, $H(X)$, measures the *prior* uncertainty on the result of X (CT06; Sha). Intuitively, the larger and more uniform is $\mathcal{X}$, the higher will be the uncertainty about X.

Definition 4 (Self-information) *The* self-information $i(x)$*, for* $p(x) > 0$*, is defined as:*

$$i(x) \triangleq -\log p(x) = \log \frac{1}{p(x)}$$

Def. 4 represents the information content of the event x. Notice that the more the event x is unlikely, the larger is $i(x)$[2].

Definition 5 (Shannon-entropy) *Let* X *be a random variable taking values in* $\mathcal{X}$*. The* Shannon entropy *of* X*, denoted as* $H(X)$*, is defined by:*

$$H(X) \triangleq \sum_{x \in \text{supp}(p)} p(x) i(x) = - \sum_{x \in \text{supp}(p)} p(x) \log p(x).$$

We can interpret the entropy as the *average* information conveyed by the observation of outcome of X. Indeed $H(X)$ is defined as the average of the self-information over all possible results of X. Notice that the value of $H(X)$ only depends on the distribution p of X. For this reason, we also use the notation $H(p)$. The entropy is measured conventionally in bits: $H(X) = n$ means that the adversary needs n information bits in order to determine the value of X. In this sense, the entropy provides a lower bound for the number of binary questions needed to find out the X value, that is the questions of the form"is $X \in \mathcal{X}'$?", where $\mathcal{X}' \subseteq \mathcal{X}$. Clearly the strategy of the adversary in choosing the sets $\mathcal{X}'$ will have an influence on the number of queries necessary to determine the value of X. Intuitively the best strategy is to choose $\mathcal{X}'$ so that its mass probability is as close as possible to that of $\mathcal{X}'' \backslash \mathcal{X}'$, where $\mathcal{X}''$ is the set of values that are currently determined as possible for X.

Example 2.2.1 *Let* X *be a six faces dice. We throw the dice, without seeing the result. Our uncertainty can be quantified in the following way. We need* $\log 6 \simeq 2,58$ *bits in order to encode a number between 1 and 6. In other words:*

$$H(X) = -\sum_{i=1}^{6} p(i) \log p(i) = \log 6$$

[2]The use of the logarithm is also motivated by its additive properties. Indeed, we want that, given the conjunction of two independent events, x, y, the equation $i(x, y) = i(x) + i(y)$ holds, true.

since $p(i) = p(X = i) = \frac{1}{6}$, for all $i = 1, \ldots, 6$. Thus in this case, we are totally uncertain, since outcomes are equiprobable.

Recall that $\mathrm{supp}(p) = \{x \in \mathcal{X} | p(x) > 0\}$. We recap some of the properties of the entropy (CT06).

Proposition 1 *Let X be a random variable, with values in $\mathcal{X}$ and $N = |\mathrm{supp}(p)|$ be the total number of possible results. Then:*

$$0 \leq H(X) \leq \log N.$$

In particular:

- *$H(X) = 0 \iff X$ is a constant;*
- *$H(X) = \log N \iff X$ is uniformly distributed on $\mathcal{X}$.*

So far we have focused on the prior uncertainty. However uncertainty might sensibly change after some observations. In this case we speak of *posterior* uncertainty, that represents the residual uncertainty on X, after the observation of a certain event Y (jointly distributed with X). Intuitively the conditional entropy $H(X|Y)$ measures the uncertainty of X given Y, as formalized by the following definition.

Definition 6 (Conditional Entropy) *The* conditional entropy *of X given Y is defined as:*

$$H(X|Y) \triangleq \sum_{y \in \mathcal{Y}} p(y) H(X|Y = y)$$

where $H(X|Y = y) \triangleq -\sum_{x \in \mathcal{X}} p(x|y) \log p(x|y)$ and we only sum over the y such that $p(y) > 0$.

So, the conditional entropy of X given Y is defined as the expected value of $H(X|Y = y)$ over all the possible $y \in \mathcal{Y}$ and, therefore, it is a measure of the average number of residual possibilities for X, once Y has been observed. Note that if X is a function of Y, then $H(X|Y) = 0$.

Example 2.2.2 *Consider the previous Example 2.2.1, but suppose that we know whether x is odd or even. This way, we have eliminated half of the possibilities, gaining one bit of information. Therefore:*

$$H(X|X \textit{ is even}) = H(X) - 1 = \log 6 - 1 = \log 3.$$

We summarize some important properties (CT06).

Proposition 2 *Let X, Y be (joint)random variables, then:*

1. $0 \leq H(X|Y) \leq H(X)$*, with equality on the right iff X, Y are independent;*

2. $H(X,Y) = H(Y,X) = H(X) + H(Y|X) = H(Y) + H(X|Y)$ *(*chain rule*).*

The first statement says that the uncertainty on X decreases, on average, after the observation of a generic event Y or, in the worst case, it remains unchanged ($H(X) = H(X|Y)$) if X and Y are dependent. The second statement shows how we can decompose the uncertainty of the pair (X, Y) as the sum of the Y uncertainty with the X residual uncertainty after the observation of Y.

In order to quantify the X uncertainty reduction after the observation of Y, we introduce the mutual information.

Definition 7 *The* mutual information *between X and Y is the difference between the entropy of X and the conditional entropy of X given Y:*

$$I(X;Y) \triangleq H(X) - H(X|Y).$$

Hence $I(X;Y)$ measures the quantity of information that X and Y share, that is the quantity of information gained by an adversary that can observe Y. Note that mutual information is symmetric.

Remark 2.2.1 *Note that in the case of deterministic programs, Y is determined by X. Then $H(Y|X) = 0$, which implies that $I(X;Y) = I(Y;X) = H(Y) - H(Y|X) = H(Y)$. This means that, with deterministic programs, the mutual information $I(X;Y)$ can be simplified to the entropy $H(Y)$.*

2.2.2 Guessing entropy

Another interesting notion is the *guessing entropy*, that is, the expected number of questions of the form "is X equal to x?" required to determine the value of X.

Definition 8 (Guessing entropy) *Assume that the elements of $\mathcal{X}$ are indexed in order of probability. The* guessing entropy *is defined as:*

$$G(X) \triangleq \sum_{1 \leq i \leq |\mathcal{X}|} i \cdot p(x_i).$$

We can interpret the guessing entropy as the average effort necessary to guess a secret through an exhaustive research in a brute force attack.

Similarly to Shannon entropy, we can define the conditional guessing-entropy that represents the expected number of guesses necessary to determine X, once that Y is known.

Definition 9 (Conditional Guessing-Entropy) *Let X, Y random variables distributed according to p_X and p_Y respectively. Then the* conditional guessing entropy *of X given Y is:*

$$G(X|Y) \triangleq \sum_{y \in \mathcal{Y}} p_Y(y) G(X|Y = y).$$

2.2.3 Min-entropy

In this subsection we introduce an alternative way of measuring information leakage, whose security implications we discuss in the next section. In QIF scenario, min-entropy corresponds to a *one-try adversary* that can ask exactly one question of the form: "is $X = x$?". Before defining the new metric, we introduce some useful concepts related to the success/error probability of a one-try adversary, namely an attacker that is granted with only one attempt to guess the secret. This scenario can be formalized in terms of Bayesian hypothesis testing, as follows. The attacker's strategy is represented by a *guessing function* $g : \mathcal{Y} \to \mathcal{X}$. The *success probability after 1 observation* (relative to g) is defined as by

$$P_{succ}^{(g)} \triangleq \Pr(g(Y) = X). \tag{2.1}$$

Correspondingly, the error probability is

$$P_e^{(g)} \triangleq 1 - P_{succ}^{(g)}. \tag{2.2}$$

It is well-known (see e.g. (CT06)) that optimal strategies, that is strategies maximizing the success probability, are those obeying the following *Maximum A Posteriori* (MAP) criterion.

Definition 10 (Maximum A Posteriori rule, MAP**)** *A function $g : \mathcal{Y} \to \mathcal{X}$ satisfies the* Maximum A Posteriori (MAP) criterion *if for each $y \in \mathcal{Y}$ and $x \in \mathcal{X}$*

$$g(y) = x \text{ implies } p(y|x)p(x) \geq p(y|x')p(x') \ \forall x' \in \mathcal{X} \, .$$

Intuitively, the above definition says that the best strategy for the adversary is to guess the secret that maximizes the posterior probability given the observation y. In what follows, we shall always assume that g is MAP and consequently omit the superscript $^{(g)}$. The quantity P_{succ} admits a number of equivalent formulations. For example, it is straightforward to check that (cf. e.g. (Smi09; BPP11a; BPP11b); the sums below run over y of positive probability)

$$P_{succ} = \sum_y p(y) \max_x p(x|y) \tag{2.3}$$

$$= \sum_y \max_x p(y|x)p(x) \, . \tag{2.4}$$

Equation (2.3) shows clearly that P_{succ} results from an *average* over all observations $y \in \mathcal{Y}$.

We convert the above probability measures to an entropy, using Rényi's *min-entropy* (Ŕ61), defined below.

Definition 11 (Min-entropy) *The* min-entropy *of X is given by*

$$H_\infty(X) \triangleq -\log P_{succ}(X) = -\log \max_{x \in \mathcal{X}} p(x). \tag{2.5}$$

Min-entropy leakage is employed by Smith in (Smi09) as a measure for information flow. It was already adopted to quantify anonymity provided by mix networks (SM03; CPP08a) and discussed in relation to cryptographic guessing attacks by Cachin in (Cac97).

Min-entropy is an instantiation of *Rényi entropy*, introduced in (Ŕ61), as a one-parameter family of entropy measures, intended as a generalization of Shannon entropy:

$$H_\alpha(X) \triangleq \frac{1}{1-\alpha} \log \sum_{x \in \mathcal{D}} p(x)^\alpha, \quad \alpha \geq 0, \alpha \neq 1$$

Min-entropy is obtained when $\alpha \to \infty$, while Shannon entropy when $\alpha \to 1$. Again, we can introduce the conditional version of min-entropy.

Definition 12 (Conditional min-entropy) *The* conditional min-entropy *of X given Y is*

$$H_\infty(X|Y) \triangleq -\log \sum_y p(y) \max_x p(x|y). \tag{2.6}$$

Note that from the above definition it follows that

$$P_{succ}(X|Y) = 2^{-H_\infty(X|Y)} \tag{2.7}$$

that provides a strong security guarantee, since it states that the adversary's success probability, given a certain observation, decreases exponentially fast with the conditional min-entropy.

With min-entropy, we measure the initial uncertainty with $H_\infty(X)$, the remaining uncertainty as $H_\infty(X|Y)$ and the leakage as the difference between the two. This last quantity expresses how much, on the average, one observation increases the success probability of the attacker. The intuition is that a gain of one bit of min-entropy leakage corresponds to doubling the a priori success probability.

2.3 Security guarantees

In this section we clarify how effective the introduced entropy notions are at expressing information leakage. In particular, using Guessing entropy as an intermediate notion, we compare the guarantees offered by Shannon entropy and by min-entropy.

Let us start by considering Shannon entropy first. Following the analysis developed by Smith in (Smi09), the key question is whether the value of $H(X|Y)$ (the remaining uncertainty) accurately reflects the threat to X. A result by Massey (Mas94) seems to justify the use of $H(X|Y)$.

Proposition 3 (Massey) *Let X be a random variable. If $H(X) \geq 2$, then:*

$$G(X) \geq \frac{2^{H(X)} + 1}{e}.$$

Basing on previous proposition, Clark et al. in (CHM01) state that $G(X|Y)$ satisfies a similar bound.

Proposition 4 *If $H(X|Y) \geq 2$, then*

$$G(X|Y) \geq 2^{H(X|Y)-2} + 1. \tag{2.8}$$

Example 2.3.1 *Consider the program introduced in Example 1.1.1*

$$\mathtt{P}(x) = x \;\&\; 037$$

where X is uniformly distributed in $\mathcal{X} = \{0, \ldots, 2^{32} - 1\}$. Then, as already discussed, $H(X|Y) = 27$ since each equivalence class contains 2^{27} elements, uniformly distributed. Using (2.8) *we obtain $G(X|Y) \geq 2^{25} + 1$, which is an acceptable bound of the real expected number of guesses that is $\frac{2^{27}+1}{2}$.*

The problem is that the expected number of guesses is not directly related to the adversary's probability of guessing the secret in just one try.

Fano's inequality provides such a direct relation. In particular it provides a lower bound, in terms of $H(X|Y)$, for the probability of error of the adversary, that is the probability that he/she fails the guess of correctly identify X in one try, given the value of Y.

Proposition 5 (Fano's inequality) *Let X, Y random variables with values in the sets $\mathcal{X}$ and $\mathcal{Y}$ respectively, with $|\mathcal{X}| > 2$. Let $g : \mathcal{X} \to \mathcal{Y}$, then*

$$P_e = \Pr(X \neq g(Y)) \geq \frac{H(X|Y) - 1}{\log(|\mathcal{X}| - 1)}. \tag{2.9}$$

Example 2.3.2 *Consider the program of Example 2.3.1. On this program Fano's inequality gives $P_e \geq \frac{27-1}{\log(2^{32}-1)} \approx 0.8125$. This bound is in fact quite loose, since here the adversary has no knowledge of 27 bits of X, which implies that $P_e \geq \frac{2^{27}-1}{2^{27}} \approx 0.9999999925$.*

Although there exist cases where the Fano's bound and the real value of the error probability are very close, Example 2.3.2 shows that, in general, Fano's inequality does not provide accurate bounds.

As Smith noticed in (Smi09), Shannon and Guessing entropies do not consider the possibility that, even if the needed guessing effort is, on the average, arbitrarily high, the probability to guess the secret in just one try might be significant. The following example, borrowed by Smith (Smi09), clarifies this difference.

Example 2.3.3 *Assume that X is an $8k$-bits integer, with $k \geq 2$, uniformly distributed in $\{0, \ldots, 2^{8k} - 1\}$. This implies that $H(X) = 8k$. Consider the two following programs*

$$P_1(x) : \texttt{if x mod 8 == 0 then y=x else y=1}$$
$$P_2(x) : \texttt{y=(x \& } 0^{7k-1}1^{k+1}\texttt{)}$$

where P_1 outputs the input if it is multiple of 8, otherwise it outputs 1, while P_2 returns the last $k+1$ bits of the input (indeed $0^{7k-1}1^{k+1}$ is a binary constant). Let us now compare the two programs. Let $Y = P(X)$. Concerning P_1, since it is deterministic, its leakage is $H(Y)$ as observed in Remark 2.2.1. Note that the else *branch is executed on $7/8$ of the values of X. This means that $\Pr(Y = 1) = \frac{7}{8}$ and $\Pr(Y = 8n) = 2^{-8k}$ for each $0 \leq n < 2^{8k-3}$. Then*

$$H(Y) = \frac{7}{8} \log \frac{8}{7} + 2^{8k-3} 2^{-8k} \log 2^{8k} \approx k + 0.169.$$

This implies that $H(X|Y) \approx 7k - 0.169$. This suggests that about $7/8$ of the information contained in X remains unleaked. But the problem is that the then *branch is taken $1/8$ of the time, meaning that the adversary correctly guesses the value of X at least $1/8$ of the time, that is a value unacceptably high.*

Concerning P_2, here $H(Y) = k + 1$ and $H(X|Y) = 7k - 1$. Note that, once Y is learned, the adversary's probability of correctly guessing X is just $1/2^{7k-1}$.

So, basing on the Shannon entropy, P_2 is worse that P_1, even if in P_1 X is highly vulnerable while in P_2 it is not.

To stress the differences between Shannon-entropy and min-entropy, we consider again Example 2.3.3.

Example 2.3.4 *Since both P_1 and P_2 are deterministic and X is uniformly distributed, we only care about $|\mathcal{X}|$ and $|\mathcal{Y}|$. Since $|\mathcal{X}| = 2^{8k}$, then $H_\infty(X) = 8k$ as before. While on program P_2 we obtain the same results, on program P_1 we have $|\mathcal{Y}| = 2^{8k-3} + 1$, that implies that the leakage is about $8k - 3$ and the remaining uncertainty is about 3. Therefore the min-entropy approach reflects more faithfully our intuition about leakage.*

Remark 2.3.1 *Note that if X is uniformly distributed on $\mathcal{X}$ then $H_\infty(X) = \log |\mathcal{X}|$. Hence Shannon entropy and min-entropy coincide on uniform distributions. As previous example showed, in general, Shannon entropy can be arbitrarily grater that min-entropy, since $H(X)$ can be arbitrarily high even if X has a value whose probability is close to 1.*

We conclude this section with a standard result, that characterizes maximum information leakage for deterministic programs, for both Shannon and min-entropy leakage. Recall that the image of a program (function) P is the set $\mathrm{Im}(P) \triangleq \{y \in \mathcal{Y} : P^{-1}(y) \neq \emptyset\}$. Let us denote by $[x]$ the inverse image of $P(x)$, that is $[x] \triangleq \{x' \in \mathcal{X} : P(x') = P(x)\}$. The proof can be found in Appendix A.1.

Theorem 2.3.1 *Let $i \in \{\mathrm{Sh}, \infty\}$ and $k = |\mathrm{Im}(P)|$. Then $L_i(P; g) \leq C_i(P) = \log k$. In particular, if $g(x) = \frac{1}{k \times |[x]|}$ for each $x \in \mathcal{X}$, then $L_{\mathrm{i}}(P; g) = C_{\mathrm{i}}(P) = \log k$, for $i \in \{\mathrm{Sh}, \infty\}$.*

In the light of the above result, a crucial security parameter is therefore the image size of P, $k = |\mathrm{Im}(P)|$: with a slight language abuse, we will refer to this value as to *program capacity*. Note that program capacity is especially relevant to min-entropy, because it coincides with actual leakage under the uniform input distribution, which is quite common in practice.

2.4 More Information Theory

In this section we introduce some other useful concepts from Information Theory.

2.4.1 Kullback-Leibler divergence and method of types

In this subsection we develop the relation linking information theory and statistics. We start by first presenting the method of types, introduced by Csiszàr and Körner (Csi98).

Definition 13 (Kullback-Leibler Divergence) *Given two probability distributions p, q, defined on the same set $\mathcal{X}$, the* Kullback-Leibler divergence *between p and q is:*

$$D(p\|q) \triangleq \sum_{x \in \mathcal{X}} p(x) \log \frac{p(x)}{q(x)}$$

with the proviso that $0 \cdot \log \frac{0}{q(x)} = 0$ and that $p(x) \cdot \log \frac{p(x)}{0} = +\infty$ if $p(x) > 0$.

Suppose that the real distribution p is unknown, but that we know an approximation q. Then the Kullback-Leibler divergence quantifies the inaccuracy that we have when we assume that the distribution of X is the approximated one, q, rather than the real one p. Similarly to a real distance, the Kullback-Leibler divergence is always non negative as stated by the following property (CT06).

Lemma 2.4.1 (Gibbs inequality) *The Kullback-Leibler divergence is always non-negative. In particular*

$$D(p\|q) \geq 0 \tag{2.10}$$

and we achieve the equality iff $p = q$.

Let us now introduce the *method of types*. It is based on the classification of the sequences based on their empirical distribution. The goal is, on one hand, to establish an upper bound for the number of possible sequences with such a distribution and, on the other hand, to estimate this probability.

Let $X_1, \ldots, X_n$ be a sequence of n random variables i.i.d. (independent, identically distributed) with distribution p_X. Then we give the following formal definition (CT06).

Definition 14 (Type) *The* type t_x *of a sequence* $x = (x_1, \ldots, x_n) \in \mathcal{X}^n$ *is the relative proportion of occurrences of each symbol* $x \in \mathcal{X}$. *In other worlds, it is a probability distribution on* $\mathcal{X}$ *defined as* $t_x(a) = \frac{N(a|x)}{n}$, *where* $N(a|x)$ *is the number of times that the symbol* a *occurs in the sequence* $x \in \mathcal{X}^n$.

Definition 15 *Let* $\mathcal{T}_n$ *denote the set of types with denominator* n*:*

$$\mathcal{T}_n = \{(\frac{x_1}{n}, \ldots, \frac{x_k}{n}) | x_j \geq 0, \quad \sum_i x_i = n\}.$$

Definition 16 (Type class) *Suppose* $t \in \mathcal{T}_n$, *then the set of* n*-length sequences of type* t *is said* type class *of* t *and is denoted by:*

$$T(t) = \{x \in \mathcal{X}^n | t_x = t\}.$$

We clarify the concepts introduced so far with a concrete example.

Example 2.4.1 *Let* $\mathcal{X} = \{1, 2, 3\}$ *be an alphabet and* $x = 13323$ *a sequence of length* 5. *Then the type* P_x *is:*

$$t_x(1) = \frac{1}{5}, \quad t_x(2) = \frac{1}{5}, \quad t_x(3) = \frac{3}{5}.$$

The type class of t_x is the set of 5-length sequences with a single occurrence of the symbol 1, a single occurrence of the symbol 2 and three occurrences of the symbol 3. In total we have 20 sequences that satisfy this constraint and

$$T(t_x) = \{12333, 21333, 13233, \ldots, 33321\}.$$

The number of elements of $T(t)$ is given by:

$$|T(t)| = \binom{5}{1,1,3} = \frac{5!}{1!1!3!} = 20.$$

The strength of this method is that the number of types is polynomial in n, as shown by the following result (CT06).

Theorem 2.4.1 *The cardinality of the set $\mathcal{T}_n$ over an alphabet $\mathcal{X}$ is bounded in this way:*

$$|\mathcal{T}_n| \leq (n+1)^{|\mathcal{X}|}. \tag{2.11}$$

The crucial point here is that there are only a polynomial number of types of length n. Since the number of sequences is exponential in n, it follows that at least one type has exponentially many sequences in its type class. In fact, the largest type class has essentially the same number of elements as the entire set of sequences, to first order in the exponent.

We define the following sets: let $\mathcal{U}_\varepsilon^{(n)}(p_X)$ be the set of sequences of length n, whose type is far at most ε from p_X in the Kullback-Leibler sense, and let $\mathcal{U}_\varepsilon^{(n)c}(p_X)$ be the complementary set, that is:

$$\mathcal{U}_\varepsilon^{(n)}(p_X) \triangleq \{x \in \mathcal{X}^n | D(t_x \| p_X) \leq \varepsilon\}, \quad \mathcal{U}_\varepsilon^{(n)c}(p_X) \triangleq \mathcal{X}^n \backslash \mathcal{U}_\varepsilon(p_X).$$

The following result (CT06) shows that, under a probability distribution p_X on $\mathcal{X}$:

1. all the sequences with the same type have the same probability;
2. the probability of a sequence of events (or a set of sequences of events, respectively) decrease exponentially fast with the Kullback-Leibler divergence between its type and the distribution p_X

Theorem 2.4.2 (Csizàr-Körner (Csi98)) *Let $X_1, \ldots, X_n$ be i.i.d. random variables distributed according to p_X. Then, setting $p_X^n = \prod_{i=1}^n p_X(x_i)$, we obtain that:*

1. *the probability of a sequence $x = (x_1, \ldots, x_n)$ only depends on its type and is given by:*

$$p_X^n(x) = 2^{-n(H(p_X)+D(t_x\|p_X))}$$

2. *the probability of the set $\mathcal{U}_\varepsilon^{(n)c}(p_X)$ decreases exponentially fast as ε increases:*

$$p_X^n(\mathcal{U}_\varepsilon^{(n)c}(p_X)) \leq (n+1)^{|\mathcal{X}|}2^{-n\varepsilon}.$$

Note that if the sequence x belongs to the type class of p_X, then $p_X^n(x) = 2^{-nH(p_X)}$.

We conclude this subsection introducing some important results, (CT06), regarding this method: the first one provides an estimation of the type class amplitude, while the second one bounds its probability.

Proposition 6 (Type Class Size) *For each type $t \in \mathcal{T}_n$, we obtain that:*

$$\frac{1}{(n+1)^{|\mathcal{X}|}}2^{nH(t)} \leq |T(t)| \leq 2^{nH(t)}.$$

Proposition 7 (Type Class Probability) *For each type $t \in \mathcal{T}_n$ and distribution p_X, the probability if a type class $T(t)$ governed by p_X^n is bounded in the following way:*

$$\frac{1}{(n+1)^{|\mathcal{X}|}}2^{-nD(t\|p_X)} \leq p_X^n(T(t)) \leq 2^{-nD(t\|p_X)}. \tag{2.12}$$

So, to summarize, the previous equations state that there are only a *polynomial* number of types and that there are an *exponential* number of sequences of each type. We also have an exact formula for the probability of any sequence of type t under distribution p_X and an approximate formula for the probability of a type class. These equations allow us to calculate the behavior of long sequences based on the properties of the type of the sequence. For example, for long sequences drawn i.i.d. according to some distribution, the type of the sequence is close to the distribution generating the sequence, and we can use the properties of this distribution to estimate the properties of the sequence.

2.4.2 Rate of convergence

Once we have chosen a security metric, it is interesting to analyze, not only its limit value, but also how fast this value is reached as the number of observation grows. For this reason, we introduce the following concept.

Definition 17 (rate) *Let $f : \mathbb{N} \to \mathbb{R}^+$ be a nonnegative, monotonically non-increasing function. Let $\gamma = \lim_{n\to\infty} f(n)$. The* rate *of f is defined as the nonnegative quantity*

$$\rho(f) \stackrel{\triangle}{=} -\lim_{n\to\infty} \frac{1}{n} \log(f(n) - \gamma)\,. \tag{2.13}$$

We further say that f reaches δ at rate ϵ *if there is a nonnegative, monotonically non-increasing function h s.t. $\lim_{n\to\infty} h(n) \leq \delta$, $\rho(h) \geq \epsilon$ and $f(n) \leq h(n)$ for each n large enough.*

When f reaches γ at rate ρ, we write this as

$$f(n) \doteq \gamma - 2^{-n\rho}\,.$$

Intuitively, for large values on of n, $\doteq$ can be interpreted as $\approx$.

Note that we admit rates of 0, as well as of $+\infty$. The following example shows that slow rate of convergence does not always guarantee security.

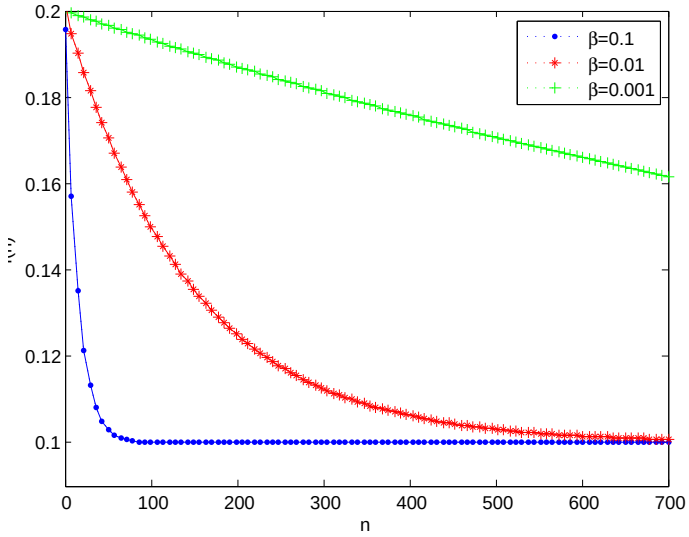
Example 2.4.2 *Consider $f(n) = \alpha + \beta 2^{-n\lambda_1} + \gamma 2^{-n\lambda_2}$, for some nonnegative α, β and γ, and $0 < \lambda_1 < \lambda_2$. Then $f(n) \to \alpha$ and $\rho(f) = \lambda_1$. Since $f(n) \leq h(n) = \alpha + \beta + \gamma 2^{-n\lambda_2}$, one has that f reaches $\alpha + \beta$ at a rate of λ_2. The picture below displays a plot of three functions, characterized by identical values of $\alpha = 0.1$, $\gamma = 0.01$, $\lambda_1 = 0.01$, and $\lambda_2 = 2$, and by three different values of β: $\beta = 0.1$ (top curve), 0.01 (middle curve) and 0.001 (bottom curve). One can see that although the convergence to the limit value, 0.1 for all of them, is extremely slow, convergence to the value 0.11, which is only slightly higher, in the third case is very fast. A system with an error probability function of this shape would not be considered as secure.*

Definition 18 (Chernoff information) *Consider two distributions p and q on $\mathcal{Y}$. The* Chernoff information *between them is the non-negative quantity*

$$C(p,q) \stackrel{\triangle}{=} -\min_{0\leq\lambda\leq 1} \log \sum_{y\in \mathrm{supp}(p)\cap\mathrm{supp}(q)} p^{\lambda}(y) q^{1-\lambda}(y) \tag{2.14}$$

with the convention that $C(p,q) = +\infty$ if $\mathrm{supp}(p) \cap \mathrm{supp}(q) = \emptyset$.

Figure 2: Slow convergence rate is not enough in order to guarantee security



Chernoff Information can be thought of as a sort of distance[3] between p and q: the more p and q are far apart, the the less observations are needed to discriminate between them.

Consider the case of an IHS with $|\mathcal{X}| = 2$, where $\mathcal{X} = \{x_1, x_2\}$, and let $p_{x_1} = p$, $p_{x_2} = q$. A well-known result gives us the rate of convergence for the probabilities of success and error, with the proviso that $p > 0$ and $q > 0$ (cf. (CT06)):

$$P^n_{succ} \doteq 1 - 2^{-nC(p,q)} \tag{2.15}$$

$$P^n_e \doteq 2^{-nC(p,q)} \tag{2.16}$$

where we stipulate $2^{-\infty} = 0$. We stress again that this rate does not depend on the prior distribution, but only on the probability distributions p and q.

[3]Note that $C(p,q) = 0$ iff $p = q$ and that $C(p,q) = C(q,p)$. However $C(\cdot,\cdot)$ fails to satisfy the triangle inequality.

This result extends to the general case $|\mathcal{X}| \geq 2$. Indeed it is enough to replace $C(p,q)$ by $\min_{p_x \neq p_{x'}} C(p_x, p_{x'})$, thus (see (BPP11a; LJ97)):

$$P_{succ}^n \doteq 1 - 2^{-n \min_{p_x \neq p_{x'}} C(p_x, p_{x'})} \tag{2.17}$$

$$P_e^n \doteq 2^{-n \min_{p_x \neq p_{x'}} C(p_x, p_{x'})}. \tag{2.18}$$

We end this section with a relation between KL-divergence and Chernoff Information, that we will exploit in the subsequent chapters. Let p_1 and p_2 be two distributions on $\mathcal{Y}$ and let $\mathcal{I}_{p_1 p_2} = \{q \,|\, D(q||p_1) = D(q||p_2)\}$. If $\mathrm{supp}(p_1) = \mathrm{supp}(p_2)$, then there exits a unique $q^* \in \mathcal{I}_{p_1 p_2}$ that minimizes $D(\cdot||q_1)$ in $\mathcal{I}_{p_1 p_2}$ and such that (see (CT06, Ch.11))

$$C(p_1, p_2) = D(q^*||p_1) = D(q^*||p_2)\,. \tag{2.19}$$

Chapter 3

Information flow under repeated observations

This chapter consists of two main parts, based on two different attack scenarios, reflecting respectively (BPP11a; BPPar) and (BPP11b).

3.1 Motivations

In the first scenario, we consider an adversary that aims to recover the secret from the collected observations. We study the asymptotic behaviour of (a) information leakage and (b) adversary's error probability in information hiding systems modeled as noisy channels. Specifically, we assume the attacker can make a single guess after observing n independent executions of the system, throughout which the secret information is kept fixed. We show that the asymptotic behaviour of quantities (a) and (b) can be determined in a simple way from the channel matrix. Moreover, simple and tight bounds on them as functions of n show that the convergence is exponential. We also discuss feasible methods to evaluate the rate of convergence. Our results cover both the bayesian case, where a prior probability distribution on the secrets is assumed known to the attacker, and the maximum-likelihood case, where the attacker does not know such distribution. In the bayesian case, we identify the distributions that max-

imize the leakage. We consider both the min-entropy setting studied by Smith (Smi09) and the additive form proposed by Braun et al. (BCP09), and show the two forms do agree asymptotically.

Secondly we enrich the proposed model, studying the security of the system, not only quantitatively (*how much* is leaked), but also qualitatively (*what* properties are leaked). To this purpose, we extend information hiding systems (IHS) with *views*: basically, partitions of the state-space. Given a view W and n independent observations of the system, one is interested in the probability that a Bayesian adversary wrongly predicts the class of W the underlying secret belongs to. We offer results that allow one to easily characterize the behaviour of this error probability as a function of the number of observations, in terms of the channel matrices defining the IHS and the view W. In particular, we provide expressions for the limit value as $n \to \infty$, show by tight bounds that convergence is exponential, and also characterize the rate of convergence to predefined error thresholds. We then show a few instances of statistical attacks that can be assessed by a direct application of our model: attacks against anonymity in protocols re-executions and against privacy in sparse datasets.

3.2 Attacker targets the entire secret

Consider an information hiding system $\mathcal{H} = (\mathcal{X}, \mathcal{Y}, p(\cdot), p(\cdot|\cdot))$, as defined in Def. 1, where $\mathcal{X}$ and $\mathcal{Y}$ represent a finite set of, respectively, inputs and outputs, $p(\cdot)$ is a probability distribution on $\mathcal{X}$ and $p(\cdot|\cdot) \in [0,1]^{\mathcal{X}\times\mathcal{Y}}$ is the conditional probability matrix. We start considering the attack scenario considered in (BPP11a; BPPar). Given any $n \geq 0$, we assume the adversary is a passive eavesdropper that gets to know the observations corresponding to n independent executions of the system, $y^n = (y_1, ..., y_n) \in \mathcal{Y}^n$, throughout which the secret state x is kept fixed. Formally, the adversary knows a random vector of observations $Y^n = (Y_1, ..., Y_n)$ such that, for each $i = 1, ..., n$, Y_i is distributed like Y and the individual Y_i are *conditionally independent* given X, that is, the following equality holds true for

each $y^n \in \mathcal{Y}^n$ and $x \in \mathcal{X}$ s.t. $p(x) > 0$

$$\Pr\left(Y^n = (y_1, \ldots, y_n) \,|\, X = x\right) = \Pi_{i=1}^n p(y_i|x)\,.$$

We will often abbreviate the right-hand side of the above equation as $p(y^n|x)$.

Below we extend the definitions of error probability (2.2) and MAP rule (introduced in Def. 10) to the case of repeated observations. For any n, the attacker strategy is modeled by a guessing function $g : \mathcal{Y}^n \to \mathcal{X}$, that represents the single guess the attacker is allowed to make about the secret state x, after observing y^n.

Definition 19 (error probability) *Let $g : \mathcal{Y}^n \to \mathcal{X}$ be a guessing function. The* probability of error after n observations *(relative to g) is given by*

$$P_e^{(g)}(n) \stackrel{\triangle}{=} 1 - P_{succ}(n), \quad \textit{where } P_{succ}^{(g)}(n) \stackrel{\triangle}{=} \Pr(g(Y^n) = X).$$

Even in the case of repeated observations, the optimal strategy for the adversary, that is the one that minimizes the error probability, is the Maximum A Posteriori (MAP) rule, defined in Def. 10 and extended below.

Definition 20 (MAP rule with repeated observations) *A function $g : \mathcal{Y}^n \to \mathcal{X}$ satisfies the* Maximum A Posteriori (MAP) criterion *if for each y^n and x*

$$g(y^n) = x \textit{ implies } p(y^n|x)p(x) \geq p(y^n|x')p(x') \textit{ for each } x'$$

In the above definition, for $n = 0$, it is convenient to stipulate that $p(y^n|x) = 1$: that is, with no observations at all, g selects some x maximizing the prior distribution. With this choice, $P_e^{(g)}(0)$ denotes $1 - \max_x p(x)$. It worth to notice that MAP guessing functions for a given n and $p(x)$ are not in general unique. It is readily checked, though, that $P_e(n)$ does *not* depend on the specific MAP function g that is chosen. Hence, throughout the paper we assume w.l.o.g. a fixed guessing function g for each given n and probability distribution $p(x)$. We shall omit the superscript $^{(g)}$, except where this might cause confusion.

Another widely used criterion is *Maximum Likelihood (*ML*)*, which given y^n selects a state x maximizing the likelihood $p(y^n|x)$ among all the states. ML coincides with MAP if the uniform distribution on the states is assumed.

ML is practically important because it requires no knowledge of the prior distribution, which is often unknown in security applications. Our main results will also apply to the ML rule (see Remark 3.2.2 in the next section).

We now come to information leakage, which measures the information leaked by the system by comparing the posterior (to the observations) and prior success probabilities. Indeed, two flavours of this concept naturally arise, depending on how the comparison between the two probabilities is expressed. If one uses subtraction, one gets the additive form of (BCP09), while if one uses the ratio between them, one gets a multiplicative form. In the latter case, one could equivalently consider the difference of the log's, obtaining the *min-entropy* based definition considered by Smith (Smi09).

Definition 21 (Additive and Multiplicative Leakage (Smi09; BCP09)) *The* additive *and* multiplicative leakage *after n observations are defined respectively as*

$$\mathcal{L}(n) \triangleq P_{succ}(n) - \max_x p(x) \quad \textit{and} \quad \mathcal{L}_\times(n) \triangleq \frac{P_{succ}(n)}{\max_x p(x)}$$

In an information hiding system, it may happen that two secret states induce the same distribution on the observables. An important role in determining the fundamental security parameters of the system is hence played by an indistinguishability equivalence relation over states, which is defined below, following (BK08). Recall that, for each $x \in \mathcal{X}$, we let p_x denote the probability distribution $p(\cdot|x)$ on $\mathcal{Y}$.

Definition 22 (Indistinguishability) *Given $x, x' \in \mathcal{X}$, we let $x \equiv x'$ iff $p_x = p_{x'}$.*

Concretely, two states are indistinguishable iff the corresponding rows in the conditional probability matrix are the same. This intuitively says that there is no way for the adversary to tell them apart, no matter how many observations he performs. We stress that this definition does not depend on the prior distribution on states, nor on the number n of observations.

3.2.1 Bounds and asymptotic behaviour

In (BPP11a; BPPar), it is proven that $P_e(n)$ converges exponentially fast to a quantity that depends on an *indistinguishability* relation on states.

This relation is defined as follows: $x \equiv x'$ if $p(\cdot|x) = p(\cdot|x')$. Assume $\equiv$ partitions $\mathcal{X}$ into K equivalence classes $C_1, ..., C_K$. For each i, let:

$$x_i^* \triangleq \text{argmax}_{x \in C_i} p(x) \quad \text{and} \quad \pi_i \triangleq p(x_i^*) \quad \text{and} \quad p_i^*(\cdot) \stackrel{\triangle}{=} p(\cdot|x_i^*)\,. \tag{3.1}$$

We can assume w.l.o.g. that $p_i^* > 0$ for each i.

For each $i, j \in \{1, \ldots, K\}$, let

$$c_{ij} \triangleq C(p_i, p_j)$$

where $C(\cdot, \cdot)$ is the Chernoff Information (see Def. 18). By adapting the proof for the case $|\mathcal{X}| = 2$ that is given in (CT06) (see also (LJ97)), it is not difficult to prove the following result, which gives the exact rate of convergence for $P_e(n)$, in the case where the distributions $p_1(\cdot), ..., p_K(\cdot)$ *all have the same support*[1]. Recall the definition of rate given in Def. 17.

Proposition 8 *Suppose that* $\text{supp}(p_1) = \cdots = \text{supp}(p_K)$. *Then* $\rho(P_e) = \min_{i \neq j} C(p_i, p_j)$.

More generally, the result below provides a means to tradeoff bounds on error probability with bounds on the rate of convergence. The next theorem has the following interpretation. The attacker focuses on the representative states, $\{x_i^* | i = 1, \ldots, K\}$, and tries to identify one of them as X. This strategy can fail for two reasons: either X is not in the target subset (first term in the error expression), or it is, but the attacker mistakes one state in the subset for another (second term in the error expression). The latter probability decreases exponentially fast with n, at a rate ρ.

Theorem 3.2.1 *Let I a nonempty subset of $\{1, \ldots, K\}$. Let $\rho_I \triangleq \min_{i,j \in I, i \neq j} c_{ij}$. Let $\pi_{\max} = \max_i \pi_i$. Then, for all $n \geq 1$*

$$(1 - \sum_{i \in I} p_i^*) \leq P_e(n) \leq (1 - \sum_{i \in I} p_i^*) + \frac{|I|^2}{2} p_{max}^* 2^{-n\rho_I} \tag{3.2}$$

As a consequence, $P_e(n)$ reaches $(1 - \sum_{i \in I} p_i^*)$ at a rate of ρ_i. In particular, by taking $I = \{1, ..., K\}$, we obtain that $\rho(P_e) \geq \rho_I$.

[1] In the case where the distributions have different supports, the argument of (CT06) does not apply. The ultimate reason is that that $D(p||q)$ is not continuous in the first argument if q has not full support; see also (BV08) for a discussion on this issue.

Proof Fix $n \geq 1$. Let $\mathcal{R} = \{x_i^* | i \in I\}$ and $g : \mathcal{Y}^n \to \mathcal{R}$ be a function satisfying: $g(y^n) = x_i^*$ implies $p(y^n|x_i^*)\pi_i \geq p(y^n|x_j^*)\pi_j$ for each $j \in I$. Note that g need not be MAP for $\mathcal{H}$, and that $g^{-1}(x) = \emptyset$ for $x \notin \mathcal{R}$. For each $i \in I$, let $A_i = g^{-1}(x_i^*)$ be the acceptance region for x_i^*. Then we have (the sums below run over x's s.t. $p_X(x) > 0$)

$$\begin{aligned}
P_e^g(n) &= \sum_{x \in \mathcal{X}} \Pr(g(Y^n) \neq x | X = x) p_X(x) \\
&= \sum_{x \notin \mathcal{R}} \Pr(g(Y^n) \neq x | X = x) p_X(x) + \sum_{i \in I} \Pr(g(Y^n) \neq x_i^* | X = x_i^*) \pi_i \\
&= (1 - \sum_{i \in I} \pi_i) + \sum_{i \in I} p_i(A_i^c) \pi_i \\
&\leq (1 - \sum_{i \in I} \pi_i) + \sum_{i \in I} \sum_{j \in I, j \neq i} p_i(A_j) \pi_i \\
&= (1 - \sum_{i \in I} \pi_i) + \sum_{i \in I} \sum_{j \in I, j > i} p_i(A_j)\pi_i + p_j(A_i)\pi_j \qquad (3.3)
\end{aligned}$$

where the inequality follows from $A_i^c = \cup_{j \in I \setminus \{i\}} A_j$ and a simple union bound, while the last equality is simply a rearrangement of summands. Now, we evaluate $p_i(A_j)\pi_i + p_j(A_i)\pi_j$ for each $i, j \in I$ and $i \neq j$.

Essentially by the same derivation given in (CT06, eqn.(11.239)–(11.251)), one finds that $p_i(A_j)\pi_i + p_j(A_i)\pi_j \leq \pi_i^\lambda \pi_j^{1-\lambda} 2^{-n c_{ij}}$, for a suitable $\lambda \in [0, 1]$. Since $\pi_i^\lambda \pi_j^{1-\lambda} \leq \pi_{\max}^\lambda \pi_{\max}^{1-\lambda} = \pi_{\max}$ and $c_{ij} \geq \rho_I$, we obtain

$$p_i(A_j)\pi_i + p_j(A_i)\pi_j \leq \pi_{\max} 2^{-n\rho_I} \qquad (3.4)$$

Now, if we plug the bound (3.4) in (3.3), and then factor out $\pi_{\max} 2^{-n\rho_I}$ and reorder the summands, we get

$$P_e^g(n) \leq (1 - \sum_{i \in I} \pi_i) + \big(\sum_{i \in I} \sum_{j \in I, j > i} 1\big) \pi_{\max} 2^{-n\rho_I} .$$

Now, use the fact that $\big(\sum_{i \in I} \sum_{j \in I, j > i} 1\big) = \frac{|I|(|I|-1)}{2} \leq \frac{|I|^2}{2}$, which shows that the wanted inequality holds for $P_e^g(n)$. But, from optimality of MAP, $P_e(n) \leq P_e^g(n)$, which completes the proof. □

Remark 3.2.1 *(a) In the practically important case where the prior $p(\cdot)$ on $\mathcal{X}$ is uniform, the term $\frac{|I|^2}{2}\pi_{\max}2^{-n\rho_I}$ is bounded above by $\frac{K}{2}2^{-n\rho_I}$.*

(b) Computation of the Chernoff Information (2.14) is an optimization problem that may be difficult to solve exactly. In practice, setting $\lambda = \frac{1}{2}$ in the argument of the min *often yields a good lower bound of $C(p,q)$, known as* Bhattacharyya distance*. Another lower bound that we will find useful in the case of distributions with sparse support (see Section 3.3.3), is obtained by taking the* min *limited to the cases $\lambda = 0$ and $\lambda = 1$. Letting $\sigma = \mathrm{supp}(p) \cap \mathrm{supp}(q)$, this quantity amounts to $-\min\{\log p(\sigma)\,,\, \log q(\sigma)\}$.*

Corollary 3.2.1 *If the a priori distribution on $\mathcal{X}$ is uniform, then $P_e(n)$ converges exponentially fast to $1 - \frac{K}{|\mathcal{X}|}$.*

Remark 3.2.2 (on the ML rule) *(BCP08) shows that the probability of error under the* ML *rule,* averaged *on all distributions, coincides with the probability of error under the* MAP *rule and the uniform distribution. From Corollary 3.2.1 we therefore deduce that the average* ML *error converges exponentially fast to the value $1 - \frac{K}{|\mathcal{X}|}$ as $n \to \infty$.*

We discuss now some consequences of the above results on information leakage. Recall that for $i = 1, ..., K$, we call x_i^* a representative of the indistinguishability class C_i that maximizes the prior distribution $p(x)$ in the class C_i, and let $p_i^* = p(x_i^*)$. Assume w.l.o.g. that $p_1^* = \max_x p(x)$. In what follows, we denote by p_{max} be the distribution on $\mathcal{X}$ defined by: $p_{max}(x) = \frac{1}{K}$ if $x \in \{x_1^*, ..., x_K^*\}$ and $p_{max}(x) = 0$ otherwise.

Corollary 3.2.2 *Fix any prior distribution $p(x)$.*

1. *$\mathcal{L}(n)$ converges exponentially fast to $\sum_{i=2}^K \pi_i$. This value is maximized by the prior distribution p_{max}, which yields the limit value $1 - \frac{1}{K}$.*

2. *$\mathcal{L}_\times(n)$ converges exponentially fast to $\frac{\sum_{i=1}^K \pi_i}{\pi_1}$. This value is maximized by the prior distribution p_{max}, which yields the limit value K.*

Proof

1. The value of the limit follows directly from the definition of $\mathcal{L}(n)$ and Theorem 3.2.1. Concerning the second part, for any fixed $p(x)$, it is easily checked that $\sum_{i=2}^K \pi_i \leq 1 - \frac{1}{K}$ (this is done by separately considering the cases $\max_x p(x) \geq \frac{1}{K}$ and $\max_x p(x) < \frac{1}{K}$). But the value $1 - \frac{1}{K}$ is obtained asymptotically with the distribution p_{max}.

2. Again, the value of the limit follows directly from the definition of $\mathcal{L}_\times(n)$ and Theorem 3.2.1. Concerning the second part, for any fixed $p(x)$, of course we have $\sum_{i=1}^{K} \frac{\pi_i}{\pi_1} \leq \sum_{i=1}^{K} 1 = K$. But the value K is obtained asymptotically with the distribution p_{max}.

□

Remark 3.2.3 *A consequence of Corollary 3.2.2(2) is that, in the case of uniform distribution on states, the multiplicative leakage, as n goes to infinity, coincides with the number of equivalence classes K. If one considers deterministic systems, that is systems where the channel matrix defines a function $f : \mathcal{X} \to \mathcal{Y}$, the leakage does not depend on the number of observations: $\mathcal{L}_\times(n) = K$ for $n \geq 1$. Moreover k equals the number of distinct counter-images of f; in particular $K \leq |Y|$. In this way we re-obtain a result of Smith in (Smi09) for deterministic systems.*

In (BCP09) additive is contrasted with multiplicative leakage in the case of a single observation ($n = 1$). It turns out that, when comparing two systems, the two forms of leakage are in agreement, in the sense that they individuate the same maximum-leaking system with respect to a fixed prior distribution on inputs. However, (BCP09) also shows that the two forms disagree as to the distribution on inputs that maximizes leakage with respect to a fixed system. This is shown to be the uniform distribution in the case of multiplicative leakage, and a function that uniformly distributes the probability on the set of "corner points" in the case of additive leakage (see (BCP09) for details). Here, we have shown that, despite this difference, additive and multiplicative leakage do agree asymptotically at least on one maximizing distribution.

Remark 3.2.4 *In (KS10), Köpf and Smith observe that, in the case of uniform distribution on $\mathcal{X}$, multiplicative leakage is upper-bounded by the number of types of n-sequences of Y:*

$$\mathcal{L}_\times(n) \leq T_n. \tag{3.5}$$

It is interesting to compare this upper-bound, which depends on n, with our upper-bound, the value K given by Corollary 3.2.2(2). It is clear that, since as $n \to \infty$ one has $T_n \to \infty$ as well, (3.5) *ceases to be useful for large values of n. Recalling that $T_n \leq (n+1)^{|Y|}$ (as stated in Theorem 2.4.1 in Chapt. 2*

and in (CT06)) and using some algebra, one sees that (3.5) *is sharper than our upper-bound K at least as long as*

$$n \leq K^{\frac{1}{|\mathcal{Y}|}} - 1.$$

So it appears that the upper-bound (3.5) *is useful only when the number of rows of the matrix is very large compared to the number of observables.*

3.2.2 Some simple applications

We apply the results stated before in three concrete examples: the anonymity protocol Crowds, the mix-net scenario and the S-box scenario (a fundamental component of the DES coder).

Protocol re-execution in Crowds The Crowds protocol (RR98) is designed for protecting the identity of the senders of messages in a network where some of the nodes may be corrupted, that is, under the control of an attacker. Omitting a few details, the functioning of the protocol can be described quite simply: the sender first forwards the message to a node of the network chosen at random; at any time, any node holding the message can decide whether to (a) forward in turn the message to another node chosen at random, or (b) submit it to the final destination. The choice between (a) and (b) is made randomly, with alternative (a) being assigned probability p_f (forwarding probability) and alternative (b) probability $1 - p_f$. The rationale here is that, even if a corrupted node C receives the message from a node N (in the Crowds terminology, C *detects* N), C, hence the attacker, cannot decide whether N is the original sender or just a forwarder. In fact, given that N is detected, the probability of N being the true sender is only slightly higher than that of any other node being the true sender. So the attacker is left with a good deal of uncertainty as to the sender's identity. Reiter and Rubin have showed that, depending on p_f, on the fraction of corrupted nodes in the network and on a few other conditions, Crowds offers very good guarantees of anonymity (see (RR98)).

Chatzikokolakis et al. have analyzed Crowds from the point of view of information hiding systems and one-try attacks (CPP08a; CPP08b). In

	d_1	d_2	$\cdots$	d_{20}
x_1	0.468	0.028	$\cdots$	0.028
x_2	0.028	0.468	$\cdots$	0.028
$\vdots$			$\vdots$	
x_{20}	0.028	0.028	$\cdots$	0.468

Figure 3: The conditional probability matrix of Crowds for 20 honest nodes, 5 corrupted nodes and $p_f = 0.7$.

their modelling, $\mathcal{X} = \{x_1, ..., x_m\}$ is the set of possible senders (honest nodes), while $\mathcal{Y} = \{d_1, ..., d_m\}$ is the set of observables. Here each d_i has the meaning that node x_i has been detected by some corrupted node. The conditional probability matrix is given by $p(d_j|x_i)$ is given by the probability that x_j is detected, while x_i is the true sender and some honest node has been detected (see (RR98) for details of the actual computation of these quantities). An example of such a system with $m = 20$ users[2], borrowed from (CPP08b), is given in Figure 3.

The interesting case for us is that of re-execution, in which the protocol is executed several times, either forced by the attacker himself (e.g. by having corrupted nodes suppress messages) or by some external factor, and the sender is kept fixed through the various executions. This implies the attacker collects a sequence of observations $y^n = (y_1, ..., y_n) \in \mathcal{Y}^n$, for some n. The repeated executions are assumed to be independent, hence we are precisely in the setting considered in this chapter. This case is also considered in (CPP08b), which gives lower bounds for the error probability holding for any n. Our results in Section 3.2.1 generalize those in (CPP08b) by providing both lower- and upper- bounds converging exponentially fast to the asymptotic error probability. As an example, for the system in table above, we have $P_e(n) \to 0$, independently of the prior distribution on the senders. An achievable convergence rate, estimated with the method proposed in Theorem 3.2.1, is $\rho \approx 0.4482$. This implies that already after observing $n = 30$ re-executions the probability of error

[2] Note that, following (CPP08b) this conditional probability matrix is obtained conditioning on the event that some node has been detected.

is < 0.001.

It is worth to stress that protocol re-execution is normally prevented in Crowds for the very reasons that it decreases anonymity, although it may be necessary in some cases. See the discussion on static vs. dynamic paths in (RR98).

Unlinkability in threshold mix-nets. Statistical attacks against anonymity protocols may take advantage of sender-receiver relationships that remain fixed through repeated rounds of the protocol. In this section, we consider the case of a *mix network*, a concept due to Chaum (Cha81). In a mix-network, messages are relayed through a sequence of trusted intermediary nodes, called *mixes*, in order to hide sender-receiver relationships (*unlinkability*). In the scenario we consider, a single mix is used by a number of senders and receivers. The *threshold* of the mix is $b + 1$: at each round, the mix waits for $b + 1$ messages from the senders and then distributes the messages to the corresponding receivers. We consider the situation where one of the senders is always Alice, with her receiver being always a node Bob, initially unknown to the attacker. The recipients of the remaining b messages are assumed be chosen at random in a set of nodes $R_1, ..., R_N$. A similar scenario is at the basis of the statistical disclosure attack by Danezis (Dan03). We analyse the situation of a local eavesdropper that observes one fixed receiver, say R_j, and after each round is able to tell whether at least one message has reached R_j. More sophisticated forms of eavesdropping could be easily accommodated (e.g. attacker observing all the nodes), but would not change significantly the outcome of the analysis. The task of the attacker is to discover which node is Bob; or at least, to tell if Bob is or not the observed node, R_j.

We can model the scenario described above by an IHS $\mathcal{H}$ where: the set of states is given by all possible nodes (potential receivers of Alice's messages), that is $\mathcal{X} = \{R_1, \ldots, R_N\}$, with $p_X(R_i) = \frac{1}{N}$ for each $i = 1, ..., N$; the set of observations is $\mathcal{Y} = \{0, 1\}$, where $y = 1$ iff R_j has received at least one message at the end of the round.

The conditional probability matrix $p(\cdot|\cdot)$ is then given by the following

equalities:

$$\begin{array}{llll} p(0|R_j) & = 0 & p(1|R_j) & = 1 \\ p(0|R_i) & = (1-\frac{1}{N})^b & p(1|R_i) & = 1-(1-\frac{1}{N})^b \quad \text{for all } i \neq j. \end{array}$$

Here, the first row means that, if Bob=R_j, then the attacker will observe at least one message with certainty. The second row means that, in case Bob is any node different from R_j, then the attacker will observe 0 messages only if all the b messages – other than the one sent to Bob – are not sent to R_j (Alice surely does not send to R_j). In other words, except for a permutation of the rows, we have the matrix below.

$$\begin{bmatrix} (1-\frac{1}{N})^b & 1-(1-\frac{1}{N})^b \\ \vdots & \vdots \\ (1-\frac{1}{N})^b & 1-(1-\frac{1}{N})^b \\ 0 & 1 \end{bmatrix}$$

Here the last row refers to R_j. This means that there are only two classes of indistinguishability: $\mathcal{X}/\equiv$ is $\{C_1, C_2\}$, with $C_1 = \{R_j\}$ and $C_2 = \mathcal{X} \setminus \{R_j\}$.

We apply apply Theorem 3.2.1 to $\mathcal{H}$, which will tell us what is the error probability in case the attacker wishes to know exactly who is Bob. We can set $I = \{i, j\}$, for any $i \neq j$, and get the following bound:

$$P_e(n) \leq \left(1 - \frac{2}{N}\right) + \frac{2}{N}\left(1 - \left(1 - \frac{1}{N}\right)^b\right)^n.$$

As expected, the limit value $1 - \frac{2}{N}$ is > 0, and the security of the system increases as N increases. The corresponding asymptotic information leakage is $\log(N \cdot \frac{2}{N}) = 1$, that is, the attacker gains 1 bit of min-entropy on the limit about the identity of Bob.

Hamming weight attacks against S-boxes. Timing (Koc96) and power analysis (KJJ99) are two flavours of side-channel *correlation attacks* against cryptographic devices (BCO04; SMY09). These attacks presuppose, explicitly or implicitly, that attacker knows the inputs (messages) processed

by the target device[3]. Basically, the attack is carried out by simulating the device's computations under the different candidate keys, each time using as inputs the same messages processed by the device. This way, the attacker obtains different samplings of the leakage from the side-channel, one for each candidate key. He will then choose the key that generates the sampling that is most correlated with the one obtained from the device.

Here, we wonder to what extent knowledge of the messages is necessary to extract significant amount of information from the side-channel. Differently from correlation attacks, will therefore assume that input messages have a nonzero, moderate redundancy, but not that they are known to the attacker. We analyze the case of DES S-boxes. Similar analyzes could be conducted against different types of symmetric keys devices. Our analysis applies to any round, in fact, to any context where an adversary may get to observe the Hamming weight of the S-box output. A DES S-box can be described as a function that takes as an input a pair of a message and a key and yields as an output a block of ciphertext, $SB : \mathcal{K} \times \mathcal{M} \rightarrow \mathcal{C}$, where: $\mathcal{K} = \{0,1\}^6$ is the set of keys, $\mathcal{M} = \{0,1\}^6$ is the set of messages and $C = \{0,1\}^4$ is the set of ciphertexts. The internal details of the device are unimportant for the purpose of this illustration. We assume a uniform prior distribution on $\mathcal{K}$ and some known prior distribution on $\mathcal{M}$, say p_M. Given a block of bits B, let us denote by $B[i_1, ..., i_k]$ the block obtained by concatenating the bits of position $i_1, ..., i_k$ in B. Let use denote by $B \oplus B'$ the bitwise XOR of B and B'.

Concisely, we can describe a DES S-box as a function $SB : \mathcal{K} \times \mathcal{M} \rightarrow \{0,1\}^4$. The input bits are used to index a 4×16 matrix T of integers; each row of T consists of a distinct permutation of the integers from 0 to 15. More precisely, we have

$$SB(m,k) \stackrel{\triangle}{=} T(i,j) \text{ where } i = (m \oplus k)[5,0] \text{ and } j = (m \oplus k)[4,3,2,1]\,.$$

In other words, once the message and the key have been XOR-ed, the most and least significant bits of the resulting block yield a row index i,

[3]In some circumstances, this knowledge is granted by the application. For example, in an attack against the final round of any Feistel cipher, the left half of the output is also the input of the target round function (see (KSWH00)).

while the four central bits yield a column index j, which are used together to access the matrix T. Similarly to (KSWH00), we assume the attacker can create a side-channel delivering him the Hamming weight of a S-box' output. To the S-box thus described there corresponds an information hiding system where: $\mathcal{X} = \mathcal{K}$, $\mathcal{Y} = \{0, 1, 2, 3, 4\}$ is the set of observables, i.e. the Hamming weights, and $p(y|k)$ is defined as

$$p(y|k) \stackrel{\triangle}{=} \sum_{m \in \mathcal{M}: W(SB(m,k))=y} p_M(m)$$

where $W(\cdot)$ is the Hamming weight function.

We report here on our results about the first of the eighth S-boxes of DES. Analysis of other S-boxes leads to similar conclusions. The distribution of the plaintext, p_M, plays a crucial role here: the lower the redundancy, the less information is expected to be extracted from the side-channel. For example, p_M is the uniform distribution (0% redundancy), then it is easy to see that all the rows of the matrix $p(y|k)$ are the same, hence $P_e(n) = 1 - 1/64$ for each n: the adversary cannot do any better than random guessing. For our analysis, we have chosen a plaintext with a redundancy of about 27% ($H(p_M) = 4.39$ bits), obtained by sampling ASCII text from some web pages. In the resulting matrix, $p(y|k)$, all the rows are different, which implies that $P_e(n) \to 0$. Concerning the rate of convergence, Theorem 3.2.1 yields $\rho \approx 1.6 \cdot 10^{-3}$. This means that with $n \geq 7.2 \cdot 10^3$ observations the error probability is < 0.011. Discarding the keys corresponding to the shortest norm-1 distances, one would get $\rho \approx 2 \cdot 10^{-3}$. Applying Theorem 3.2.1, one gets an error probability ≤ 0.011 already with $n = 5.4 \times 10^3$ observations.

In a more realistic scenario, the attacker could not directly measure the Hamming weight of the target S-box, but rather the global weight of the eight S-boxes composing the round function of DES. This scenario can be modeled as a noisy version of the previous one. The Hamming weight of the target S-box, Y, is now disturbed by the noise N, the sum of the Hamming weights of the remaining seven S-boxes, say $W_2, ..., W_8$. Assuming that the variables W_i are independent from each other and from Y and identically distributed – this is not strictly true, but seems

a reasonable approximation – the central limit theorem would tell us that their sum $N = \sum_{i=2}^{8} W_i$ has approximately a gaussian distribution. Here, for simplicity we have modeled N as a random variable having binomial distribution $B(n, p)$ with $n = 28$ and $p = \frac{1}{2}$. What is observed by the attacker now is $Y' \triangleq Y + N$. Hence the new set of observables is $\mathcal{Y}' = \{0, ..., 32\}$. Explicitly, for each $i \in \mathcal{Y}'$ and $k \in \mathcal{K}$, the entries of the new conditional probability matrix $p'(\cdot|\cdot)$ are given by

$$p'(i|k) \triangleq \Pr(Y + N = i \,|\, K = k) \;=\; \sum_{j=0}^{\min\{i,4\}} p(j|k) \cdot \binom{28}{i-j} \cdot 2^{-28} \,.$$

Theorem 3.2.1 applied to the matrix $p'(\cdot|\cdot)$ yields a rate of $\rho \approx 5.963 \cdot 10^{-6}$. Theorem 3.2.1 gives $P_e(n) < 0.011$ for $n \geq 1.9295 \cdot 10^6$. As expected, the convergence rate is lower than in the noiseless case. However, the effort needed to break the system is certainly in the reach of a well determined attacker.

Our simple analysis confirms that unprotected implementations of DES S-boxes are quite vulnerable to attacks based on Hamming weights. Software simulations have reinforced this conclusion, showing that, in practice, a good success probability for the adversary is achieved with a relatively small n. For instance, in the noiseless case, already with $n = 10^3$, we have obtained an experimental success rate of 98%.

3.3 Attacker targets a property of the secret

A drawback of the QIF approach so far is that it focuses exclusively on the quantitative aspect of the analysis (*how much* is leaked), while ignoring the qualitative aspect (*what* is leaked) at all. In (BPP11a) it is shown that, when a uniform distribution on the secrets is assumed, the asymptotic information leakage of a system corresponds to the log of the number of indistinguishability classes in the system – where two states are indistinguishable if they induce the same probability distribution on the observables. To make an extreme example, consider the two small imperative procedures `P1` and `P2` below. Both of them receive as an argument a

confidential variable h that can take on a value in the set $\mathcal{X} = \{0, ..., 15\}$, possibly corresponding to user identifiers or other sensitive information. Part of the information about h is disclosed by the procedures through the public variable l.

```
P1(h): l=-1; if (h==0) then l=0;
P2(h): l=h mod 4;
```

In the case of P1, there are two possible observables, -1 and 0, hence $\mathcal{X}$ is partitioned into two indistinguishability classes: thus, assuming h is uniformly distributed, P1 leaks 1 bit of information about h. In the case of P2 there are four classes, hence P2 leaks two bits. From a global point of view, P1 is therefore more secure than P2. Needless to say, though, *from the point of view of user 0*, P2 is preferable over P1. One would like to conduct the analysis both at a quantitative and at a qualitative level, revealing not only how much is leaked, but also what. This is particularly relevant in relation to the privacy of individuals or groups.

In (BPP11b) we propose a framework to deal with this issue by extending the IHS's considered in (BPP11a; BPPar) and elsewhere with *views*.

3.3.1 A model with *views*

A view is, in short, a partition of the states, representing perhaps a subdivision in "buckets" of a large population (in fact, we are more general and also admit probabilistic partitions). In the example above, the view of interest to user 0 is the partition of $\mathcal{X}$ into $(\{0\}, \mathcal{X} \setminus \{0\})$. Given a view W, one is interested in the adversary's probability of wrongly predicting the class of W the secret belongs to, after observing n independent executions of the system, throughout which the secret state is kept fixed: call this quantity $P_e^W(n)$. In the example above, the involved systems are deterministic, hence a single observation is all the attacker needs. One easily finds that $P_e^W(1)$ equals 0 in the case of P1, and $\frac{1}{16}$ in the case of P2. In the general case of probabilistic systems, computation of the limit value of $P_e^W(n)$ is not as obvious.

Definition 23 (views) *Let $\mathcal{H} = (\mathcal{X}, \mathcal{Y}, p(\cdot), p(\cdot|\cdot))$ be a* IHS. *A* view *of $\mathcal{H}$ is a pair $(\mathcal{W}, q(\cdot|\cdot))$, where $\mathcal{W}$ is a finite alphabet and $q(\cdot|\cdot) \in [0,1]^{\mathcal{X}\times\mathcal{W}}$ is a matrix where all rows sum to 1.*

Informally, $q(w|s)$ is the probability that the property w holds when in state x. The probability distribution p on $\mathcal{X}$ and the conditional probability matrices $p(y|x)$ and $q(w|x)$ induce a probability distribution r on $\mathcal{W}\times\mathcal{X}\times\mathcal{Y}$, defined as $r(w,x,y) \triangleq p(x)\cdot p(y|x)\cdot q(w|x)$. This distribution induce a triple of discrete random variables $(W, X, Y) \sim r$, taking values in $\mathcal{W} \times \mathcal{X} \times \mathcal{Y}$. We shall denote the marginal probability distributions of this triple for X, W and Y by p_X, p_W and p_Y, respectively. Of course, $p_X(\cdot)$ coincides with the prior $p(\cdot)$ given in the IHS, while the marginal distributions p_W and p_Y can be computed from the given data, $p(\cdot)$, $p(\cdot|\cdot)$ and $q(\cdot|\cdot)$.

Let us now discuss the observation scenario. Given any $n \geq 0$, we assume the adversary is a passive eavesdropper that gets to know the observations corresponding to n independent executions of the system, $y^n = (y_1, ..., y_n) \in \mathcal{Y}^n$, throughout which both the secret state x and the corresponding view w are kept fixed. Formally, the adversary knows a random vector of observations $Y^n = (Y_1, ..., Y_n)$ such that, for each $i = 1, ..., n$, Y_i is distributed like Y. Moreover, the individual Y_i and the view W are *conditionally independent* given X. This means that the following equality holds true for each $y^n \in \mathcal{Y}^n$, $w \in \mathcal{W}$ and $x \in \mathcal{X}$ s.t. $p(x) > 0$

$$\begin{aligned}&\Pr\left(Y^n = (y_1, \ldots, y_n),\, W = w \,|\, X = x\right)\\ &= \Pi_{i=1}^n \Pr(Y_i = y_i | X = x)\, \Pr(W = w | X = x)\,.\end{aligned}$$

Note that the right-hand side of the above equality can be equivalently written as $\Pi_{i=1}^n p(y_i|x)q(w|x)$.

In the following we shall drop the subscripts from the above defined (conditional) probability distributions when no ambiguity can arise. We will often abbreviate $\Pi_{i=1}^n p(y_i|x)$ as $p(y^n|x)$. Moreover, by slightly abusing notation, we will freely identify a view $(\mathcal{W}, q(\cdot|\cdot))$ of $\mathcal{H}$ with the induced random variable W.

We discuss now the goal of an attacker that targets a property of states represented by a view W. Similarly to the case considered in (BPP11a;

BPPar), the attacker's strategy corresponds to a guessing function, which this time is of the form $g : \mathcal{Y}^n \to \mathcal{W}$. The corresponding error probability (after n observations, relative to g) is

$$P_e^{g,W}(n) \stackrel{\triangle}{=} \Pr\left(g(Y^n) \neq W \right). \tag{3.7}$$

A function g minimizes this quantity if it is W-MAP, that is if satisfies the following condition. For each $y^n \in \mathcal{Y}^n$ and $w \in \mathcal{W}$

$$g(y^n) = w \text{ implies } p(y^n|w)p(w) \geq p(y^n|w')p(w'), \text{ for each } w' \in \mathcal{W}.$$

Unless otherwise stated, given a view of $\mathcal{H}$, we shall assume an underlying guessing function that is W-map. Consequently, we shall normally omit the indication of g from $P_e^{g,W}(n)$.

In many systems, the practically important views are those that partition the state-space into equivalence classes. A view W is called a *partition* of $\mathcal{H}$ if W is a function of X, that is $W = f(X)$ for some function $f : \mathcal{X} \to \mathcal{W}$. Equivalently, the matrix $q(\cdot|\cdot)$ has a single entry '1' for each row. Let $\mathcal{W} = \{w_1, ..., w_L\}$, and let $E_i \stackrel{\triangle}{=} f^{-1}(w_i)$ for $1 \leq i \leq L$. Of course $E_1, ..., E_L$ forms a partition of $\mathcal{X}$, in the set-theoretic sense.

Accordingly to this scenario, we can extend the definition of information leakage as follows.

Definition 24 (Information leakage relative to a view) *The* information leakage *after n observations* relative to a view W *is defined as*

$$\mathcal{L}_\times^W(n) \stackrel{\triangle}{=} \log\left(\frac{P_{succ}^W(n)}{\max_w p_W(w)}\right).$$

3.3.2 Asymptotic error probability

We analyze now the case of P_e^W, where W is a generic view of an IHS $\mathcal{H}$. We follow the notation and terminology established in the previous section. It would be tempting to proceed as follows: build a new IHS, say $\mathcal{H}^W$, where the states are $\mathcal{W}$ and the channel matrix is $p_{Y|W}$. The error probability function for $\mathcal{H}^W$ would then coincide with $P_e^W(n)$. It would then be enough to apply Theorem 3.2.1 to $\mathcal{H}^W$. This approach however

is doomed to failure. In fact, the assumption that the observations Y_i are conditionally independent given W is in general false:

$$p(y_1 \cdots y_n|w) \neq p(y_1|w) \cdots p(y_n|w)\,.$$

As a consequence, the IHS $\mathcal{H}^W$ is meaningless for what concerns our purposes. However, conditional independence of the Y_i's given W is guaranteed, and the approach outlined above *does* work, in the special case where W is a partition finer than $\equiv$. This intuition leads us to develop to the method illustrated below for P_e^W in the general case.

Some more notation first. For notational simplicity, assume $\mathcal{W}$ is a set of integers $\{1, ..., |\mathcal{W}|\}$. Let $q(\cdot|\cdot)$ be the matrix defining the view W.

Definition 25 (Indistinguishability with views) *We denote by $\sim_W$ the equivalence relation on $\mathcal{X}$ induced by $q(\cdot|\cdot)$, that is*

$$x \sim_W x' \text{ iff for each } y \in \mathcal{Y}:\ q(y|x) = q(y|x')\,. \tag{3.8}$$

In other words, two states are $\sim_W$-equivalent if the corresponding rows of $q(\cdot|\cdot)$ are equal. Let $\mathcal{X}/\sim_W$ be $\{E_1, ..., E_L\}$, the equivalence classes of $\sim_W$. The intersection $\equiv \cap \sim_W$ is still an equivalence relation on $\mathcal{X}$, that is finer than both $\equiv$ and $\sim_W$. Recall that $\mathcal{X}/\equiv$ is $\{C_1, ..., C_K\}$. For $1 \leq i \leq K$ and $1 \leq j \leq L$, we let the equivalence classes of $\equiv \cap \sim_W$ be denoted as

$$F_{ij} \stackrel{\triangle}{=} C_i \cap E_j \tag{3.9}$$

and furthermore

$$F_i^* \stackrel{\triangle}{=} \max_j p_X(F_{ij}) \quad \text{and} \quad q_j^* \stackrel{\triangle}{=} \max_w q(w|x), \text{ for an arbitrary } x \in E_j\,. \tag{3.10}$$

The next theorem has the following interpretation. The attacker focuses on a subset of the representative states, $\{x_i^*|i \in I\}$. He tries to identify first the class C_i of X, then guesses the class F_{ij} – this is given by the j that maximizes $p_X(F_{ij})$. Finally he guesses the view w that is most likely in E_j. This strategy can fail for two reasons: either w is wrong (first term in the expression), or F_{ij} is wrong (second + third term).

Theorem 3.3.1 *Let I and ρ_I be chosen as in Theorem 3.2.1. Let W be a view of $\mathcal{H}$. Let $\Pi_{\max} = \max_{i\in I} F_i^*$. Then*

$$P_e^W(n) \;\le\; \sum_{j=1}^{L}(1-q_j^*) \;+\; (1-\sum_{i\in I} F_i^*) \;+\; \frac{|I|^2}{2}\Pi_{\max}2^{-n\rho_I}\,. \qquad (3.11)$$

Proof Denote a pair of indices $(i,j) \in \{1,...,K\}\times\{1,...,L\}$ as ij. For each $x \in \mathcal{X}$, define $\mathrm{ind}(x) = ij$ iff $x \in F_{ij}$. Fix $n \ge 1$ and any function $g' : \mathcal{Y}^n \to \{1,...,K\}\times\{1,...,L\}$, and let $Succ'$ be the event $(g'(Y^n) = \mathrm{ind}(X))$. That is, $Succ'$ is the event that g' correctly classifies the index (of the equivalence class F_{ij}) of X. Now define a guessing function for $\mathcal{H}$, $g : \mathcal{Y}^n \to \mathcal{W}$, as $g(y^n) \stackrel{\triangle}{=} w$, where $g'(y^n) = ij$ and $w = \mathrm{argmax}_w\, q(w|x)$ for any $x \in E_j$ (note that the information about i provided by g' is ignored by g). Let Err be the event $(g(Y^n) \ne W)$. We have

$$\begin{aligned} P_e^W(n) &= \Pr(Err, Succ') + \Pr(Err|\neg Succ')\Pr(\neg Succ') \qquad (3.12)\\ &\le \Pr(Err, Succ') + \Pr(\neg Succ')\,. \qquad (3.13)\end{aligned}$$

Let us estimate $\Pr(Err, Succ')$ and $\Pr(\neg Succ')$ separately. It is an easy matter to prove that

$$\begin{aligned} \Pr(Err, Succ') &= \sum_{j=1}^{L}(1-q_j^*)\Pr(X\in E_j, Succ')\\ &\le \sum_{j=1}^{L}(1-q_j^*)\,. \qquad (3.14)\end{aligned}$$

We now estimate $\Pr(\neg Succ')$. Consider the new IHS $\mathcal{H}' \stackrel{\triangle}{=} (\{1,...,K\}\times\{1,...,L\}, \mathcal{Y}, p'(\cdot), p'(\cdot|\cdot))$, where $p'(ij) \stackrel{\triangle}{=} p_X(F_{ij})$ and $p'(y|ij) \stackrel{\triangle}{=} p_i(y)$. Note that $ij \equiv i'j'$ iff $i = i'$. Hence we have K distinct classes in this system, whose representatives are elements $x'_1 = 1j_1, ..., x'_K = Kj_K$ such that $j_i = \mathrm{argmax}_j\, p_X(F_{ij})$, hence $p'(x'_i) = F_i^*$, for $i = 1,...,K$. The corresponding representative distributions (rows of the matrix $p'(\cdot|\cdot)$) are $p'_1(\cdot) = p_1(\cdot), ..., p'_K(\cdot) = p_K(\cdot)$.

Now take the function g' above to be a MAP guessing function for $\mathcal{H}'$. Call $P'_e(n)$ the error probability of $\mathcal{H}'$: clearly, $\Pr(\neg Succ') = P'_e(n)$. Take

$I \subseteq \{1, ..., K\}$ and apply Theorem 3.2.1 to $\mathcal{H}'$ and I to get

$$\Pr(\neg Succ') \leq 1 - \sum_{i \in I} F_i^* \ + \ \frac{|I|^2}{2} \Pi_{\max} 2^{-n\rho_I} \, . \tag{3.15}$$

When we plug the bounds (3.14) and (3.15) into (3.13), we get the wanted result. □

Note that the determination of the upper-bound in (3.11) is computationally practical: the partitions induced by $\equiv \cap \sim_W$ can be directly computed by inspection of the matrices $p(\cdot|\cdot)$ and $q(\cdot|\cdot)$. Their intersection (3.9), and the probability mass of the corresponding classes $p_X(F_{ij})$, are then straightforward to compute. Theorem 3.3.1 only provides an (exponential) upper bound to $P_e^W(n)$. The following theorem provides the exact limit of $P_e^W(n)$ in the special, but important case when W is a partition.

We recall a few concepts of the method of types introduced in Section 2.4.1. that will be used in the proof. Fix $n \geq 1$. Given a sequence $y^n \in \mathcal{Y}^n$ and $y \in \mathcal{Y}$, denote by $\mathrm{n}(y, y^n)$ the number of occurrences of y inside y^n. The empirical distribution or type of y^n is the distribution on $\mathcal{Y}$ defined as $t_{y^n}(y) \stackrel{\triangle}{=} \mathrm{n}(y, y^n)/n$, for each $y \in \mathcal{Y}$, as defined in Def. 14. The "balls" of center $p_i(\cdot)$ and radius $\epsilon > 0$ in $\mathcal{Y}^n$ are defined as $U_\epsilon^n(p_i) \stackrel{\triangle}{=} \{y^n : D(t_{y^n}||p_i) \leq \epsilon\}$. From Theorem 2.4.2 of Csizàr and Körner, it follows that, as $n \to +\infty$, $p_i(U_\epsilon^n(p_i)) \to 1$, while, for any $p \neq p_i$ there is $\epsilon > 0$ small enough s.t. $p(U_\epsilon^n(p_i)) \to 0$. Moreover, the convergence is exponential in both cases.

Theorem 3.3.2 *Let W be a partition of $\mathcal{H}$. Then $P_e^W(n)$ converges exponentially fast to $1 - \sum_{i=1}^K F_i^*$. More precisely, with the same notation of Theorem 3.3.1, for each $n \geq 1$,*

$$1 - \sum_{i=1}^K F_i^* \leq P_e^W(n) \leq (1 - \sum_{i=1}^K F_i^*) + \frac{K^2}{2} \Pi_{\max} 2^{-n\rho_I},$$

where $I = \{1, ..., K\}$.

Proof First, note that for W a partition, the first term in (3.11) vanishes, as each q_j^* equals 1. The upper bound is then a consequence of Theorem

3.3.1 with $I = \{1, ..., K\}$. We now seek for a lower bound of $P_e^W(n)$. We equivalently focus on an upper bound of $P_{succ}^W(n)$. Assume without loss of generality that $\mathcal{W} = \{1, ..., L\}$. For any $n \geq 1$, let $g : \mathcal{Y}^n \to \{1, ..., L\}$ be a W-MAP guessing function, and let $A_j = g^{-1}(j)$, for $j \in \{1, ..., L\}$, be the acceptance region in $\mathcal{Y}^n$ for j. It is a routine task to check that

$$P_{succ}^W(n) \;=\; \sum_{i=1}^{K} \sum_{j=1}^{L} p_i(A_j) p_X(F_{ij})\,. \tag{3.16}$$

Now, fix any $i \in \{1, ..., K\}$, and let $j_i = \mathrm{argmax}_{j=1,...,L}\, p_X(F_{ij})$, that is $p_X(F_{ij_i}) = F_i^*$. We claim that $p_i(A_{j_i}) \to 1$ as $n \to +\infty$. In fact, fixed $\epsilon > 0$ small enough, for any n large enough A_{j_i} contains the "ball" $U_\epsilon^n(p_i)$ of center $p_i(\cdot)$ and radius ϵ in $\mathcal{Y}^n$. To see that this is true, note that a sufficient condition for $y^n \in A_{j_i}$ is that for each $j \neq j_i$

$$\begin{aligned} p_{Y^n|W}(y^n|j_i) p_W(j_i) &= \sum_{l=1}^{K} p_l(y^n) p_X(F_{lj_i}) \\ &> \sum_{l=1}^{K} p_l(y^n) p_X(F_{lj}) = p_{Y^n|W}(y^n|j) p_W(j). \end{aligned} \tag{3.17a}$$

Now from results of the method of types it follows that, for $y^n \in U_\epsilon^n(p_i)$, we have that all the $p_l(y^n)$ with $l \neq i$ go exponentially fast to 0 as n grows. Thus the condition (3.17a) reduces, for n large enough, to $F_i^* = p_X(F_{ij_i}) > p_X(F_{ij})$: this is satisfied by definition of j_i[4]. Now $A_{j_i} \supseteq U_\epsilon^n(p_i)$ implies that $p_i(A_{j_i})$ goes to 1 exponentially fast as n grows; for the same reason, $p_i(A_j)$ goes to 0 for each $j \neq j_i$ as n grows (recall that the A_j's form a partition of $\mathcal{Y}^n$). This way, and taking (3.16) into account, we have proved that

$$\lim_{n\to\infty} P_{succ}^W(n) \;=\; \sum_{i=1}^{K} F_i^*\,.$$

[4]If there is more than one index j maximizing $p_i(F_{ij})$, then the choice of j_i gets more involved: among those j's that maximize $p_X(F_{ij})$, one chooses the one that maximizes $p_X(F_{i'j})$, where $p_{i'}(\cdot)$ is the distribution closest to $p_i(\cdot)$ in terms of KL-distance, if this j is unique; otherwise one must look at the second closest distribution $p_{i''}(\cdot)$, and so on. We omit the details here.

Since $P^W_{succ}(n)$ is monotonically non-decreasing, we have proved that $P^W_{succ}(n) \leq \sum_{i=1}^K F_i^*$ holds true for each $n \geq 1$. This implies in turn the wanted statement. □

3.3.3 Some simple applications

Below we consider two applications of the above model. In the first one we apply the result of this section to the mix-net scenario, introduced in Section 3.2.2. Secondly we introduce a statistical attack, related to privacy protection in sparse datasets.

Unlinkability in threshold mix-nets Consider the scenario described in Section 3.2.2. To see qualitatively *what* the single bit gained by the attacker corresponds to, we analyse the error probability with respect to the view $W \in \{0, 1\}$ given by:

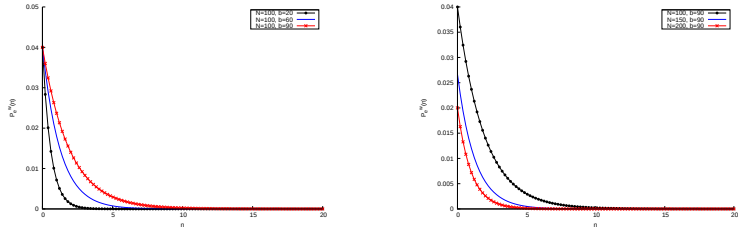
$$W = 1 \text{ iff } X = R_j \,.$$

That is, W yields 1 iff Bob is R_j. The partition induced on $\mathcal{X}$ by W coincides with $\equiv$, hence its classes are C_1, C_2. Concerning the sets F_{ij}, we note that: $F_{11} = \{R_j\},\ F_{12} = F_{21} = \emptyset$ and $F_{22} = \mathcal{X} \setminus \{R_j\}$. Since the distribution on the states is uniform, we have: $F_1^* = \frac{1}{N}$ and $F_2^* = 1 - \frac{1}{N}$. Take $I = \{i, j\}$ as defined as above. According to Theorem 3.3.1, the limit of $P_e^W(n)$ vanishes, moreover

$$P_e^W(n) \leq \frac{2}{N}\left(1 - \left(1 - \frac{1}{N}\right)^b\right)^n .$$

The attacker's success probability of guessing whether R_j =Bob or not approaches very fast 1. It is also interesting to study the behaviour of the rate $\rho_I = -\log\left(1 - (1 - \frac{1}{N})^b\right)$ depending on b and N. It is easy to see that as b increases, ρ_I decreases; on the contrary, as N increases and b is kept fixed, ρ_I increases. The shape of $P_e^W(n)$ is illustrated qualitatively by the plots in the figures 4: very few rounds of the protocols ($n < 10$) are sufficient to achieve $P_e^W \approx 0$.

As mentioned above, it is easy to repeat this kind of analysis with more sophisticated observations on the part of the attacker: we do not do so



(a) $P_e^W(n)$ depending on parameter b (b) $P_e^W(n)$ depending on parameter N

Figure 4: Plots of $P_e^W(n)$ depending on two different parameters.

here for lack of space. On the other hand, note that just repeating this simple attacks for each of the potential Alice's receivers (that is, setting $R_j = R_1, R_2, ..., R_{N-1}$ in turn), would lead the attacker to uncover the identity of Bob after a low number of rounds. This is sufficient to show that the single threshold mix system is totally insecure.

Privacy in sparse datasets. We consider datasets collecting *micro-data* – preferences, recommendations, transaction records, health histories and so on – about a large number of individuals. Datasets of this kind are sometimes published for commercial or research purposes. Making micro-data public poses serious threats to the privacy of individuals, even when the data are released in anonymized form – that is with personal identifiers, such as SSN's, removed. The risk is that an attacker, using a little of background information about a given individual and cross-correlation of attributes, might *re-identify* the individual within the dataset, leading to the disclosure of the whole set of her/his attributes. An example of this technique is the spectacular de-anonymization attack of Narayanan and Shmatikov against the Netflix Prize dataset (NS08a)[5].

[5]The Netflix Prize dataset collects anonymous movie ratings of 500,000 subscribers. Using background information publicly available from the Internet Movie Database, Narayanan and Shmatikov successfully re-identified known users within the Netflix dataset.

In this section, we show that (sparse) datasets naturally arise as instances of IHS, and that assessment of statistical attacks against dataset privacy is easily accomplished using the general results of Section 3.3.2.

We view a dataset as a table $\mathcal{D}$, with rows and columns corresponding to individuals (or more generally, records) and attributes, respectively. Formally, $\mathcal{D} \in \mathcal{V}^{R \times A}$, where $\mathcal{V}$, R and A are finite nonempty sets of values, records and attributes, respectively. One can view any dataset $\mathcal{D}$ as an IHS $\mathcal{H}_{\mathcal{D}}$, as follows. Records are equiprobable states, that is we set $\mathcal{X} = R$ and let $p_X(\cdot)$ be the uniform distribution on R. Concerning observables, there is a variety of sensible choices, depending on the observation power one wishes to grant the attacker with. For instance, a sensible choice is $\mathcal{Y} \triangleq A \times \mathcal{V}$. Another choice, if $\mathcal{V}$ is a totally ordered, is to observe attributes and *ranges* of values. The last choice is more robust than the former in case the dataset is published in a perturbed form. In fact, even setting $\mathcal{Y} \triangleq A$ is sensible, as just knowledge of non-null attributes of a record provides a great deal of information[6]. In any case, the technical development presented below does not depend on the specific choice of $\mathcal{Y}$. Finally, the conditional probability matrix models the process of acquiring background information about the individuals in the dataset. Depending on its exact nature, this information might come from various sources, e.g. personal blogs, Google searches, or even a water-cooler conversation with a colleague (see (NS08a)). For example, if $\mathcal{Y} = A$, then it is sensible to assume that the background knowledge consists of randomly chosen attributes and set, for each record r and attribute a

$$p(a|r) \triangleq \begin{cases} \frac{1}{n_r} & \text{if } a \text{ is a non-null attribute of } r \\ 0 & \text{otherwise} \end{cases}$$

where n_r is the number of non-null attributes in the row of the dataset corresponding to r. Of course, non-uniform distributions can be equally accommodated, e.g. if it is felt that certain attributes are more likely to be publicly released than others.

Having shown how to model a dataset as a IHS, we have to point out that, in the formal development below, there is no need to restrict to IHS's

[6]See (NS08a) for further considerations on the structure of sparse datasets.

of the form $\mathcal{H}_{\mathcal{D}}$. To work in full generality, we will just assume a dataset is simply an IHS.

In a sparse dataset, most of the entries in the table are null. Specifically, we consider a dataset sparse if, except possibly for a small fraction of records, for no record there is another "similar" record in the dataset. To make the notion of sparsity precise, we have first to make precise the notion of similarity between records. We will work with a similarity function $\mathrm{Sim} : \mathcal{X} \times \mathcal{X} \to [0, 1]$. The intuition underlying the following definition, which is different from that proposed in (NS08a), is that the similarity of x' to x is related to the fraction of non-null attributes they share. More precisely, it is the fraction of non-null attributes that can be inferred on any of the two by looking at the other.

Definition 26 (similarity) *Given an* IHS $\mathcal{H}$*, for any* $x, x' \in \mathcal{X}$*, let* $\sigma_{xx'} = \mathrm{supp}(p(\cdot|x)) \cap \mathrm{supp}(p(\cdot|x'))$*. We set*

$$\mathrm{Sim}(x, x') \stackrel{\triangle}{=} \min \left\{ p(\sigma_{xx'} \mid x) \, , \, p(\sigma_{xx'} \mid x') \right\} .$$

Note that $\mathrm{Sim}(x, x') = 1$ iff $\mathrm{supp}(p(\cdot|x)) = \mathrm{supp}(p(\cdot|x'))$.

The following notion of sparsity is not related to - not weaker nor stronger than - the one considered in (NS08a). It seems to be satisfied by typical sparse datasets, like the Neflix Prize (NS08a). In fact, our results extend, although in a different form, to the notion of sparsity of (NS08a), but we shall not give any detail here.

Definition 27 (sparsity) *Let* $\mathcal{H}$ *be a* IHS *with* $p_X(\cdot)$ *the uniform distribution. Let* $\epsilon > 0$ *and* $\delta > 0$*. We say* $\mathcal{H}$ *is* (ϵ, δ)-sparse *if*

$$\Pr \big(\max_{x : x \neq X} \mathrm{Sim}(X, x) \geq \epsilon \big) < \delta \, . \tag{3.18}$$

Our results apply to a situation where the attacker gets to know an entire copy of the dataset.We begin with a result on error probability.

Theorem 3.3.3 *Let* $\mathcal{H}$ *be* (ϵ, δ)*-sparse, with* $|\mathcal{X}| = N$*. Then* $P_e(n)$ *reaches* δ *at a rate* $-\log \epsilon$*. More precisely,*

$$P_e(n) \ \leq \ \delta + \frac{1}{2}[(1-\delta)^2 N + 2(1-\delta) + \frac{1}{N}]\epsilon^n .$$

Proof By definition of sparsity, it is possible to find a subset of the records, say $\mathcal{R} = \{x_i^* | i \in I\}$, s.t. for each $x \in \mathcal{R}$, there is no other record in $\mathcal{X}$ which is ϵ-similar to x, and such that $p_X(\mathcal{R}) \geq 1 - \delta$. Moreover, by uniform distribution of the probability mass on records, we can choose the size of I satisfying $\frac{|I|-1}{N} < (1-\delta) \leq \frac{|I|}{N}$, which means $(1-\delta)N \leq |I| < (1-\delta)N+1$. Next note that, with the notation introduced in Section 3.3.2 and by virtue of Remark 3.2.1(b), for any $i, j \in I$ with $i \neq j$, the Chernoff information c_{ij} satisfies: $c_{ij} \geq -\log \mathrm{Sim}(x_i^*, x_j^*) \geq -\log \epsilon$. Applying Theorem 3.2.1 we get the thesis. □

In some cases, all the adversary needs to determine about a record is its "similarity class". In fact, knowledge of this class already provides him with almost all the information about the record. If this class is disclosed then a privacy breach has occurred. The next definition formalizes this intuition. Recall from (3.8) that $\sim_W$ is the equivalence relation induced on $\mathcal{X}$ by W.

Definition 28 ((ϵ, δ, ρ)-breach) *Let $\mathcal{H}$ be a* IHS. *Consider a partition W of $\mathcal{H}$ such that whenever $x \sim_W x'$ then $\mathrm{Sim}(x, x') \geq \epsilon$. We say W is an* (ϵ, δ, ρ)-breach *if $P_e^W(n)$ reaches δ at rate ρ.*

The following result establishes strong upper bounds on the resistance to privacy breaches in sparse datasets.

Theorem 3.3.4 *Any (ϵ, δ)-sparse* IHS *has an $(\epsilon, \delta, -\log \epsilon)$-breach W. In particular, to $P_e^W(n)$ the same bound applies as given for $P_e(n)$ in Theorem 3.3.3.*

Proof The proof is similar to that of Theorem 3.3.3. Consider the set $\mathcal{R} = \{x_i^* \,|\, i \in I\}$. Build the partition W as follows: take as blocks the singletons $\{x_i^*\}$, for $i \in I$, plus the blocks obtained by breaking $\mathcal{X} \setminus \mathcal{R}$ in such a way that any two records in the same block are ϵ-similar. Then apply Theorem 3.3.1 with W and I, taking into account the bounds for $|I|$ given in the proof of Theorem 3.3.3. □

Example 3.3.1 *Real-world datasets tend to be extremely sparse. For instance, $(0.15, 0.2)$-sparsity in a dataset containing $N = 5 \times 10^5$ records should not*

be considered as exceptional (cf. (NS08a, Fig.1), referring to the Netflix Prize dataset). Applying the bound of Theorem 3.3.3 to these figures, we see that already after coming across $n = 10$ randomly chosen attribute values of a target individual, the probability of uncorrect re-identification in the dataset is < 0.201. This may still seem quite high in absolute terms. Consider, however, that the success probability prior to the observations was $\frac{1}{5\times 10^5}$. In terms of information leakage, this means that the attacker has obtained $\mathcal{L}_\times(10) \approx 18.6$ bits of min-entropy, out of $\log N \approx 18.9$. The privacy breach is therefore absolutely relevant. Note that attacks against real-world datasets can exploit specific features of the target and get more impressive success probabilities (NS08a).

3.4 Further and related work

The last few years have seen a flourishing of research on quantitative models of information leakage. In the context of language-based security, Clark et al. (CHM01) first motivated the use of mutual information to quantify information leakage in a setting of imperative programs. Boreale (Bor09) extended this study to the setting of process calculi, and introduced a notion of rate of leakage. In both cases, the considered systems do not exhibit probabilistic behaviour. Closely related to ours is the work by Chatzikokolakis, Palamidessi and their collaborators. We have already mentioned (CPP08a), that examines information leakage mainly from the point of view of Shannon entropy and capacity. (CPP08a) also contains results on asymptotic error probability, showing that, independently from the input distribution, the ML rule approximates the MAP rule. (CPP08b) studies error probability mainly relative to one observation ($n = 1$), but also offers a lower-bound in the case of repeated observations (CPP08b, Proposition 7.4). This lower-bound is generalized by our results. Compositional methods based on process algebras are discussed in (BCP08); there, the average ML error probability is characterized in terms of MAP error probability under a uniform distribution of inputs. (BCP09) introduces the notion of additive leakage, $\mathcal{L}_+$, and compares it to the min-entropy based leakage, $\mathcal{L}_\times$, considered by Smith (Smi09), but again in the case of a single observation.

A model of "unknown-message" attacks is considered by Backes e

Köpf in (BK08). This model is basically equivalent to the information hiding systems considered in (CPP08a; CPP08b; BCP09) and in the present chapter. Backes e Köpf too consider a scenario of repeated independent observations, but from the point of view of Shannon entropy, rather than of error probability. They too rely on the information-theoretic method of types to determine the asymptotic behaviour of the considered quantities. An application of their setting to the modular exponentiation algorithm is the subject of (KD09), where the effect of bucketing on security of RSA is examined. As mentioned, this study has been extended to the case of one-try attacks by Köpf and Smith in (KS10). Earlier, Köpf and Basin had considered a scenario of adaptive chosen-message attacks (KB07). They offer an algorithm to compute conditional Shannon entropy in this setting, but not a study of its asymptotic behaviour, which seems very difficult to characterize.

In the context of side-channel cryptanalysis, Standaert et al. propose a framework to reason on side-channel correlation attacks (SMY09). Both a Shannon entropy based metric and a security metric are considered. This model does not directly compares to ours, since, as already discussed in Section 3.2.2, correlation attacks are inherently known-message, that is, they presuppose the explicit or implicit knowledge of the plaintext on the part of the attacker.

Our work is also related, at least conceptually, to the notion of probabilistic *opacity* as studied by Brard, Mullins and Sassolas (BMS10). Indeed, although their setting is different – they work with finite-state machines – our partitions could be viewed as a generalization of the binary predicates they consider. Note however that (BMS10) is based on Shannon entropy, and considers observations consisting of a single run of the system, rather than repeated observations, hence not statistical attacks.

Hypothesis testing is at the basis of the analysis considered in the present paper. The classical, binary case is covered in (CT06, Ch.11). Baignres and Vaudenay (BV08) refine these results and characterize the optimal asymptotic rate of convergence in a number of variations of the basic setting, including the case where one of the two hypotheses is "composite" – that is, consisting of several sub-hypotheses chosen according to a prior

probability distribution. They apply these results to study the *advantage* an attacker may have in distinguishing the output of a given cipher from a random output. The optimal asymptotic rate of convergence in the general case of multiple bayesian hypothesis testing is characterized by Leang and Johnson in (LJ97).

In the future, we would like to systematically characterize achievable rates of convergence given an error probability threshold, thus generalizing the present results. It would also be natural to generalize the present one-try attack scenario to the case of k-tries attack, for $k \geq 2$. Experiments and simulations with anonymity protocols may be useful to asses at a practical level the theoretical results of our study. Finally, the application to sparse datasets prompts a connection to databases privacy issues that deserves further attention.

Chapter 4

Worst- and average-case information flow

4.1 Motivations

We view a randomization mechanism as an information-theoretic channel with inputs in $\mathcal{X}$ and outputs in $\mathcal{Y}$. The starting point of our study is a semantical notion of breach. Assume $\mathcal{X}$ is a finite set of items containing the secret information X, about which the adversary has some background knowledge or belief, modeled as a prior probability distribution $p(x)$. Consider a predicate $Q \subseteq \mathcal{X}$ – in a dataset about individuals, one may think of Q as gender, or membership in a given ethnical group etc. The mere fact that X is in Q or not, if ascertained, may convey sensitive information about X. Henceforth, any observation $y \in \mathcal{Y}$ that causes a significant change in the adversary's posterior belief about $X \in Q$ must be regarded as dangerous. In probabilistic terms, Q is a *breach* if, for some prior probability on $\mathcal{X}$, the posterior probability of Q after interaction with the randomization mechanism exhibits a significant change, compared to its prior probability. We decree a randomization mechanism as secure at level ϵ, if it exhibits *no breach* of level $> \epsilon$, independently of the prior distribution on the set of secret data $\mathcal{X}$. The smaller ϵ, the more secure the mechanism. This simple idea, or variations thereof, has been proposed

elsewhere in the Data Mining literature – see e.g. (EGS03). Here, we are chiefly interested in analyzing this notion of breach according to the following dimensions.

1. Worst- vs. average-case security. In the worst-case approach, one is interested in bounding the level of any breach, independently of how likely the breach is. In the average-case, one takes into account the probability of the observations leading to the breach.

2. Single vs. repeated, independent executions of the mechanism.

3. Expected utility of the mechanism and its asymptotic behavior, depending on the number of observations and on a user-defined loss function.

To offer some motivations for the above list, we observe that worst-case is the type of breach considered in DP, while average-case is the type considered in QIF. In the worst-case scenario, another issue we consider is resistance to background information. In the case of DP, this is often stated in the following terms (Dwo06): *Regardless of external knowledge, an adversary with access to the sanitized database draws the same conclusions whether or not my data is included*. A formalization along these lines is in (GKS08). We investigate how this kind of resistance relates to the notion of privacy breach we consider, which also intends to offer protection against arbitrary background knowledge.

Concerning the second point, a scenario of repeated observations seems to arise quite naturally in many applications. In addition to the examples discussed in Section 1.1, consider an online, randomized data-releasing mechanism might offer users the possibility of asking the same query a number of times, thus potentially allowing an adversary to remove enough noise to learn valuable information about the secret. This is an instance of the *composition* attacks well known in the context of DP, where they are thwarted by allotting each user or group of users a *privacy budget* that limits the overall number of queries to the mechanism; see e.g. (McS09; FS10). In general, one would like to assess the security of a mechanism in these situations. In particular, one would like to determine exactly *how fast*

the level of any potential breach grows, as the number n of independent observations grows.

The third point, concerning utility, has been the subject of intensive investigation lately – see the related work paragraph. Here, we are interested in studying the growth of expected utility in the model of Ghosh et al. (GRS09) as the number of independent observations grows, and to understand how this is related to security. In summary, the main results we obtain are the following.

- In the scenario of a single observation, both in the average and in the worst case, we characterize the security level (absence of breach above a certain threshold) of the randomization mechanism in a simple way that only depends on certain row-distance measures of the underlying matrix.

- We prove that our notion of worst-case security is stronger than DP. However, we show the two notions coincide when one confines to background information that factorises as the product of independent measures over all individuals. This, we think, sheds further light on resistance of DP against background knowledge.

- In the scenario of repeated, independent observations, we determine the exact asymptotic growth rate of the (in)security level, both in the worst and in the average case.

- In the scenario of repeated, independent observations, we determine the exact asymptotic growth rate of any reasonable expected utility. We also give bounds relating this rate to ϵ-DP, and exact expressions in the case of the geometric mechanisms. In this respect, we argue that the geometric mechanism is superior to its *truncated* version (GRS09).

Some of the above results are summarized in Table 2, which has the following notation: $\pi_{M,y}$ (resp. $\pi_{m,y}$) is the maximum (resp. minimum) in column y of the matrix; p_x is row x of the matrix; $||\cdot||_1$ is norm-1; $C(\cdot,\cdot)$ is Chernoff Information. In the third column, ϵ is the DP parameter.

Table 2: Summary of results for the different privacy notions.

	Security Level	Security Level Rate	Utility Rate
Worst Case	$\epsilon = \log \max_y \frac{\pi_{M,y}}{\pi_{m,y}}$	$\rho = \epsilon$	$\rho = \min_{x \neq x'} C(p_x, p_{x'}) \ \leq \ \epsilon$
Average Case	$\epsilon = \log(\frac{\max_{x,x'} \|p_x - p_{x'}\|_1}{2} + 1)$	$\rho = \max_{x,x'} C(p_x, p_{x'})$	$\rho = \frac{\epsilon}{2} + \log \frac{1+2^{-\epsilon}}{2}$ (geometric mechanism)

The interpretation of the results is the following. In the *Security Level* column, ϵ is the maximum level of a breach, hence the lower ϵ, the more secure the system. Note that ϵ is a logarithmic measure. In the *Security Level Rate* column, ρ is the positive exponent governing the growth of the insecurity level after n observations, ϵ_n: approximately, this means that $2^{\epsilon_n} \approx \tau - 2^{-n\rho}$, for a certain limit value τ. The third column is about the growth rate of expected utility and its relation with ϵ-DP. The third column is about the growth rate of expected utility and its relation with ϵ-DP.

4.2 Semantic security of randomization mechanisms

We shall consider two scenarios. In the worst-case scenario, one is interested in the seriousness of a breach, independently of how much the breach is likely; this is also the scenario underlying differential privacy, which we will examine in Section 4.6. In the average-case scenario, one considers, so to speak, the seriousness of the breach averaged on the probability of the observed Y. The definitions of semantics security involve a universal quantification over all possible priors. In each scenario, our aim is to characterize when a randomization mechanism is secure in a concrete way, which only depends on the channel matrix at hand. We fix a generic randomization mechanism $\mathcal{R} = (\mathcal{X}, \mathcal{Y}, p(\cdot|\cdot))$ for the rest of the section, as defined in Def. 2.

4.2.1 The worst-case scenario

In the worst-case scenario, we compare the probability of a predicate $Q \subseteq \mathcal{X}$ of the inputs, before and after one observation $y \in \mathcal{Y}$: a large variation in the posterior probability relative to some y implies a breach. Note that even the situation when the posterior probability is small compared to the prior is considered as dangerous, as it tells the adversary that $X \in Q^c$ is likely.

Definition 29 (worst-case breach) *Let $\epsilon \geq 0$. A* ϵ-breach *(*privacy breach of level ϵ*) for $\mathcal{R}$ is a subset $Q \subseteq \mathcal{X}$ such that for some a priori probability distribution $p(x)$ on $\mathcal{X}$, we have $p(Q) > 0$ and*

$$\max_{p(y)>0} |\log \frac{p(Q|y)}{p(Q)}| > \epsilon \,.$$

$\mathcal{R}$ is ϵ-secure *if it has no breach of level ϵ. The* security level *of $\mathcal{R}$ is defined as* $\epsilon_{\mathcal{R}} \stackrel{\triangle}{=} \inf\{\epsilon \geq 0 : \mathcal{R} \text{ is } \epsilon\text{-secure}\}$.

If $|\log \frac{p(Q|y)}{p(Q)}| > \epsilon$, we say *$y$ causes a Q-breach of level ϵ*.

Remark 4.2.1 *The use of $|\log(\cdot)|$ in the definition of worst-case breach has the desired effect that a large variation between the prior probability $p(Q)$ and the posterior $p(Q|y)$ results in a breach. Equivalently, the condition $\max_y |\log \frac{p(Q|y)}{p(Q)}| > \epsilon$ could be reformulated as $\max_y \max\{\frac{p(Q|y)}{p(Q)}, \frac{p(Q)}{p(Q|y)}\} > 2^\epsilon$.*

We give the following concrete characterization of worst-case security: it requires that any two elements of the matrix in the same column be of similar magnitude. The intuition behind this condition is as follows. Assume that, for some column y, an element in row x is much larger than an element in row x'. Suppose there is a prior that assigns almost all the probability mass to x', and the rest to x: the predicate $Q = \{x\}$ has then a very low (prior) probability. Now, assume y is observed: the probability of Q becomes quite high. Intuitively, this is the case because y is much more likely to have been generated by x than by x'. This large variation in probability makes for a breach. A similar property (*amplification*) was considered as a sufficient condition for the absence of breaches in (EGS03).

Let us introduce some notation to state the formal result. For each $y \in \mathcal{Y}$, let $\pi_{M,y}$ and $\pi_{m,y}$ be the maximum and the minimum in the column y of the matrix $p(\cdot|\cdot)$, respectively. In the theorem below, we stipulate that $\frac{\pi_{M,y}}{\pi_{m,y}} = +\infty$ if $\pi_{M,y} > 0$ and $\pi_{m,y} = 0$.

Theorem 4.2.1 *$\mathcal{R}$ is ϵ-secure iff* $\log \max_y \frac{\pi_{M,y}}{\pi_{m,y}} \leq \epsilon$.

Proof Let $y \in \mathcal{Y}$ be the column which achieves the maximum of $\frac{\pi_{M,y}}{\pi_{m,y}}$, and let $\pi_M = \pi_{M,y}$ and $\pi_m = \pi_{m,y}$. First, suppose that $\log \frac{\pi_M}{\pi_m} > \epsilon$: we shall exhibit a breach Q of level $> \epsilon$. To this purpose, let $x_1 = \operatorname{argmax}_x p(y|x)$ and $x_2 = \operatorname{argmin}_x p(y|x)$ be two rows achieving the maximum and minimum values in column y. Consider the subset $Q = \{x_1\}$ and a generic prior distribution $p(x)$ such that $p(x_1), p(x_2) > 0$. Then we have

$$\frac{p(Q|y)}{p(Q)} = \frac{p(x_1|y)}{p(x_1)} = \frac{p(y|x_1)p(x_1)}{p(x_1)p(y)} = \frac{\pi_M}{p(y)}. \tag{4.1}$$

Now observe that: $p(y) = p(y|x_1)p(x_1) + p(y|x_2)p(x_2) = p(x_1)\pi_M + p(x_2)\pi_m$. Note that this term can be made arbitrarily close to π_m, as $p(x_1) \to 0$, hence $\frac{p(Q|y)}{p(Q)} \to \frac{\pi_M}{\pi_m} > 2^\epsilon$. Thus there exists a certain value of $p(x_1)$ for which $\frac{p(Q|y)}{p(Q)} > 2^\epsilon$, showing that the subset Q actually is an ϵ-breach.

Consider now the opposite implication. Suppose there are Q and an a priori distribution $p(x)$ s.t. $p(Q) > 0$. Then we have (the sums below run over those x's s.t. $p(x) > 0$):

$$\frac{p(Q|y)}{p(Q)} = \frac{\sum_{x \in Q} p(y|x)p(x)}{p(y)p(Q)} \leq \frac{\pi_{M,y}p(Q)}{p(y)p(Q)} \leq \frac{\pi_M}{\pi_m} \leq 2^\epsilon$$

where in the last step we have exploited $p(y) = \sum_x p(x)p(y|x) \geq \pi_{m,y}$. Similarly, we have:

$$\frac{p(Q)}{p(Q|y)} = \frac{p(y)p(Q)}{\sum_{x \in Q} p(y|x)p(x)} = \frac{p(Q)\sum_x p(y|x)p(x)}{\sum_{x \in Q} p(y|x)p(x)} \leq \frac{\pi_M}{\pi_m} \leq 2^\epsilon$$

as $\sum_{x \in Q} p(y|x)p(x) \geq \pi_m p(Q)$ and $\sum_x p(y|x)p(x) \leq \pi_M$. □

Example 4.2.1 *Consider Example 2.1.2. The worst-case security level of the mechanism is* $\epsilon = \log \frac{0.25}{0.083} \approx 3.0012$. *It is interesting to note that the utility of this mechanism is very low. In fact, assume a user does not know anything initially about the secret - his prior is the uniform distribution - and observes for example* $y = 0$*: he may conclude with equal confidence that* x *is either 0, 1 or 5.*

4.2.2 The average-case scenario

We want to asses the security of $\mathcal{R}$ by comparing the prior and posterior success probability for an adversary wanting to infer whether the secret is in Q or not after observing Y. This will give us an *average* measure of the seriousness of the breach induced by Q.

Fix a prior probability distribution $p(x)$ on $\mathcal{X}$. For every nonempty $Q \subseteq \mathcal{X}$, we shall denote by $\hat{Q}$ the binary random variable $I_Q(X)$, where $I_Q : \mathcal{X} \to \{0, 1\}$ is the indicator function of Q – in this notation, the dependence from $p(x)$ is left implicit, as $p(x)$ will always be clear from the context. An adversary, after observing Y, wants to determine whether it holds $\hat{Q}$ or $\hat{Q}^c$. This scenario can be formalized in terms of Bayesian hypothesis testing, as explained in Section 2.2.3. We recall that information leakage is defined in terms of min-entropy as follows

$$\mathcal{L}(\mathcal{R}; p) \stackrel{\triangle}{=} H_\infty(X) - H_\infty(X|Y) = \log \frac{P_{succ}}{\max_x p(x)} . \tag{4.2}$$

We can express (4.2) focusing on $\hat{Q}$ in the following way:

$$\mathcal{L}(\mathcal{R}; \hat{Q}; p) = H_\infty(\hat{Q}) - H_\infty(\hat{Q}|Y).$$

Definition 30 (Average-case breach) *Let* $\epsilon \geq 0$. *A* ϵ-A-breach (average case breach of level ϵ) *of* $\mathcal{R}$ *is a* $Q \subseteq \mathcal{X}$ *s.t. for some a priori distribution* $p(x)$ *on* $\mathcal{X}$*, we have that* $p(Q) > 0$ *and* $\mathcal{L}(\mathcal{R}; \hat{Q}; p) = H_\infty(\hat{Q}) - H_\infty(\hat{Q}|Y) > \epsilon$. $\mathcal{R}$ *is* ϵ-A-secure *if it has no average case breach of level* ϵ. *The* A-security level *of* $\mathcal{R}$ *is defined as* $\epsilon^A_{\mathcal{R}} \stackrel{\triangle}{=} \inf\{\epsilon \geq 0 : \mathcal{R}$ *is* ϵ*-A-secure*$\}$.

Of course, since there are only 2 possible secrets (and the vulnerability can at most increase from $\frac{1}{2}$ to 1), Y leaks at most one bit about the truth of Q: $0 \leq \mathcal{L}(\mathcal{R}; \hat{Q}; p) \leq 1$.

Remark 4.2.2 *Note that, by expanding the min-entropy terms in the definition, we can equivalently express the information leakage relative to a subset Q and to a distribution p in the following form, which makes it apparent that we are dealing with an average-case measure:*

$$\mathcal{L}(\mathcal{R}; Q; p) = \log \frac{\sum_y p(y) \max\{p(Q|y), p(Q^c|y)\}}{\max\{p(Q), p(Q^c)\}}.$$

In the binary Bayesian hypothesis testing, it is well-known that the success probability depends on the norm-1 distance between the probability distributions underlying the two hypotheses – or, more precisely, their total variation distance, which is one half the norm-1 distance. The next theorem exploits this fact to show that the security level in the average case is precisely determined by the (maximal) norm-1 distance between the rows of the matrix. Recall that, for each $x \in \mathcal{X}$, we let $p_x(\cdot)$ denote the distribution $p(\cdot|x)$. We denote the norm-1 of a vector $q \in \mathbb{R}^{\mathcal{Y}}$ as $||q||_1 \stackrel{\triangle}{=} \sum_y |q(y)|$.

Theorem 4.2.2 *Let $l \stackrel{\triangle}{=} \max_{x,x'} ||p_x - p_{x'}||_1$ and $\epsilon \geq 0$. Then $\mathcal{R}$ is ϵ-A-secure iff $\log(\frac{l}{2} + 1) \leq \epsilon$.*

Note that $\log(\frac{l}{2} + 1)$ is actually the maximum min-entropy leakage on priors whose support has size 2. In order to prove Theorem 4.2.2, we introduce the following lemma whose proof can be found in Appendix A.2.

Lemma 4.2.1 *In the binary Bayesian hypothesis testing problem with two distributions p_1 and p_2, having prior probabilities α and β respectively ($\alpha + \beta = 1$), the success probability P_{succ} defined in* (2.1) *satisfies*

$$P_{succ} = \frac{||\alpha p_1 - \beta p_2||_1 + 1}{2}.$$

Proof [Theorem 4.2.2] First, assume $\log(\frac{l}{2} + 1) \leq \epsilon$. Consider any $Q \subseteq \mathcal{X}$ and any prior $p(x)$ on $\mathcal{X}$ s.t. $p(Q) > 0$. The adversary has to decide whether $p(\cdot|Q)$ or $p(\cdot|Q^c)$ is the true distribution, so this is a binary Bayesian hypothesis testing problem. Assume without loss of generality that $p(Q) \geq p(Q^c)$, and let $\lambda = \frac{p(Q^c)}{p(Q)} \in [0, 1]$. Taking (4.2) into account and using the expression for P_{succ} provided by Lemma A.2.1 with $\alpha = p(Q)$ and $\beta = p(Q^c)$,

after some algebra we get:

$$\begin{aligned}\mathcal{L}(\mathcal{R};\hat{Q};p) &= \log\frac{||p(\cdot|Q)p(Q)-p(\cdot|Q^c)p(Q^c)||_1+1}{2\max\{p(Q),p(Q^c)\}}\\ &= \log\frac{1}{2}\big(||p(\cdot|Q)-p(\cdot|Q^c)\lambda||_1+1+\lambda\big)\,. \qquad (4.3)\end{aligned}$$

Using the triangle inequality, one has

$$\begin{aligned}||p(\cdot|Q)-p(\cdot|Q^c)\lambda||_1 &= ||p(\cdot|Q)-p(\cdot|Q)\lambda+p(\cdot|Q)\lambda-p(\cdot|Q^c)\lambda||_1\\ &\leq ||p(\cdot|Q)-p(\cdot|Q)\lambda||_1+||p(\cdot|Q)\lambda-p(\cdot|Q^c)\lambda||_1\\ &= 1-\lambda+\lambda||p(\cdot|Q)-p(\cdot|Q^c)||_1\end{aligned}$$

which, when plugged into (4.3), yelds

$$\mathcal{L}(\mathcal{R};\hat{Q};p)\leq\log\frac{1}{2}\big(\lambda||p(\cdot|Q)-p(\cdot|Q^c)||_1+2\big)\,. \qquad (4.4)$$

Now, we note that $p(\cdot|Q)$ is a convex combination of probability distributions on $\mathcal{Y}$: indeed, $p(\cdot|Q)=\sum_{x\in Q}\frac{p(x)}{p(Q)}p(\cdot|x)$. Similarly, $p(\cdot|Q^c)$ is a convex combination of probability distributions on $\mathcal{Y}$: indeed, $p(\cdot|Q^c)=\sum_{x'\in Q^c}\frac{p(x')}{p(Q^c)}p(\cdot|x')$. From the convexity of norm-1 distance in both arguments, or equivalently from the triangle inequality for $||\cdot||_1$, we get

$$\begin{aligned}||p(\cdot|Q)-p(\cdot|Q^c)||_1 &\leq \sum_{x\in Q}\frac{p(x)}{p(Q)}\sum_{x'\in Q^c}\frac{p(x')}{p(Q^c)}||p(\cdot|x)-p(\cdot|x')||_1\\ &\leq \sum_{x\in Q}\frac{p(x)}{p(Q)}\sum_{x'\in Q^c}\frac{p(x')}{p(Q^c)}l=l\end{aligned}$$

which when plugged into (4.4) yields

$$\mathcal{L}(\mathcal{R};\hat{Q};p)\leq\log\big(\frac{\lambda l}{2}+1\big)\leq\log(\frac{l}{2}+1)\leq\epsilon\,.$$

Conversely, assume $\mathcal{R}$ is ϵ-A-secure. Let x_1 and x_2 be the (distinct) rows of the matrix that achieve $l=\max_{x_1,x_2}||p(\cdot|x_1)-p(\cdot|x_2)||_1$. Consider the prior probability distribution on $\mathcal{X}$ defined as $p(x_1)=p(x_2)=\frac{1}{2}$ and let

$Q = \{x_1\}$. With this particular Q, Lemma A.2.1 gives us $P_{succ} = \frac{l/2+1}{2}$, hence

$$\begin{aligned} \epsilon &\geq \mathcal{L}(\mathcal{R}; \hat{Q}; p) \\ &= \log \frac{P_{succ}}{\max\{p(Q), p(Q^c)\}} \\ &= \log(\frac{l}{2} + 1)\,. \end{aligned}$$

□

Example 4.2.2 *Consider again the mechanism of Example 2.1.2. The maximal norm-1 distance between any two rows is $l = 1$; hence the average-case security level of this mechanism is $\epsilon^A = \log(\frac{1}{2} + 1) \approx 0.584$.*

4.3 Worst-case security vs. differential privacy

We first introduce DP, then formally relate it to worst-case security. The exposition in this section is mostly technical; a more general discussion on the significance of the results we will obtain is deferred to the next section.

The definition of differential privacy (DMNS06; Dwo06) relies on a notion of "neighborhood" between inputs of an underlying randomization mechanism. In the original formulation, two neighbors x and x' are two database instances that only differ by one entry. More generally, one can rely upon a notion of *adjacency*. An undirected graph is a pair (V, E) where V is a set of nodes and E is a set of unordered pairs $\{u, v\}$ with $u, v \in V$ and $u \neq v$. We also say that E is an adjacency relation on V and if $v \sim v'$ say v and v' are adjacent.

Definition 31 (differential privacy) *A* differentially private mechanism *$\mathcal{D}$ is a pair $(\mathcal{R}, \sim)$ where $\mathcal{R} = (\mathcal{X}, \mathcal{Y}, p(\cdot|\cdot))$ is a randomization mechanism and $\sim$ is an adjacency relation on $\mathcal{X}$, that is, $(\mathcal{X}, \sim)$ forms an undirected graph.*

Let $\epsilon \geq 0$. We say $\mathcal{D}$ provides ϵ-differential privacy *if for each $x, x' \in \mathcal{X}$ s.t. $x \sim x'$, it holds that for each $y \in \mathcal{Y}$:*

$$\max_y |\log \frac{p(y|x)}{p(y|x')}| \leq \epsilon\,. \tag{4.6}$$

If x and x' are two connected nodes in the graph $(\mathcal{X}, \sim)$ at distance d (that is $x \sim x_1 \sim \cdots \sim x_d = x'$ is the shortest path from x to x'), we note that (4.6) implies the following condition, for each $y \in \mathcal{Y}$:

$$2^{-\epsilon d} \leq \frac{p(y|x)}{p(y|x')} \leq 2^{\epsilon d}\,. \tag{4.7}$$

Note that condition (4.6) is exactly that given in Theorem 4.2.1 to characterize worst-case security, but limited here to pairs of adjacent rows x and x'. This prompts the question of the exact relationship between the two notions of worst-case security and DP. To answer this question, in the rest of the section we will consider the standard input domain $\mathcal{X} = \{0,1\}^n$ of *databases* (for some fixed $n \geq 1$), corresponding to the subsets of a given set of individuals $\{1, ..., n\}$. We deem two databases x, x' adjacent if they differ for the value of exactly one individual, that is if their Hamming distance is 1 (DMNS06). Throughout the section, we let $\mathcal{D} = (\mathcal{R}, \sim)$ be a generic mechanism equipped with this $\mathcal{X}$ and this adjacency relation. Moreover, we will denote by Q_i ($i \in \{1, ..., n\}$) the set of databases $\{x \in \mathcal{X} \mid x_i = 1\}$, that is databases containing individual i.

The following theorem provides a precise characterization of worst-case ϵ-security in terms of privacy of individuals: interaction with the mechanism does not significantly change the belief about the participation of any individual to the database.

Theorem 4.3.1 *$\mathcal{R}$ satisfies ϵ-security iff for each $i \in \{1, ..., n\}$ and prior $p(\cdot)$, Q_i is not an ϵ-breach.*

Proof One direction trivially follows from the definition. Assume now that $\mathcal{R}$ does not satisfy ϵ-security: this means there are x and x' and y such that $p(y|x)/p(y|x') > 2^\epsilon$. We shall exhibit a Q_i and a prior $p(x)$ such that Q_i is an ϵ-breach. Assume that x contains the individual i, while x' does not. Consider a generic prior concentrating all probability mass on x, x'. Then

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

Now, since $p(y) = p(y|x)p(x) + p(y|x')p(x')$, we see that the above quantity can be made arbitrarily close to $p(y|x)/p(y|x')$: as $p(x) \to 0$, one has $\frac{p(Q_i|y)}{p(Q_i)} \to p(y|x)/p(y|x') > 2^\epsilon$. □

Remark 4.3.1 *The above theorem is of course still valid if one strengthens the "only if" part by requiring that* both Q_i and Q_i^c *are not* ϵ*-breach.*

We proceed now by linking (worst-case) ϵ-security to ϵ-DP. The next result sets limits to the "arbitrariness" of background information against which DP offers guarantees: for example, it fails in some cases where an adversary has sufficient background information to rule out all possible databases but two, which are substantially different from each other.

Theorem 4.3.2 *If* $\mathcal{R}$ *satisfies* ϵ*-security then* $\mathcal{D} = (\mathcal{R}, \sim)$ *provides* ϵ-DP*. On the contrary, for each* n *there exist mechanisms providing* ϵ-DP *but not* ϵ*-security; in particular, these mechanisms exhibit* Q_i*-breaches (*$i \in \{1, ..., n\}$*) of level arbitrarily close to* $n\epsilon > \epsilon$*.*

Proof The first assertion is obvious in the light of Theorem 4.2.1 and of the definition of DP. We show the second assertion. Assume $\mathcal{D}$ provides ϵ-DP. Consider a generic prior that concentrate the probability mass on two databases x and x' at Hamming distance n from one another. Equation (4.7) implies that, for every y, $\frac{p(y|x)}{p(y|x')} \leq 2^{n\epsilon}$: but in fact, it is easy to construct ϵ-DP mechanisms where this holds with *equality* (see e.g. (AACP11a)). This already shows that $\mathcal{R}$ is not ϵ-secure. In particular, assuming $x \in Q_i$:

$$\frac{p(Q_i|y)}{p(Q_i)} = \frac{p(y|x)}{p(y)} = \frac{p(y|x)}{p(y|x)p(x) + p(y|x')p(x')}.$$

As $p(x) \to 0$, the above term approaches to $\frac{p(y|x)}{p(y|x')} = 2^{n\epsilon} > 2^{\epsilon}$. □

Example 4.3.1 *Let us consider the mechanism with input domain* $\mathcal{X} = \{0, 1\}^2$ *corresponding to the following matrix:*

$$\begin{array}{c} \texttt{00} \\ \texttt{01} \\ \texttt{11} \\ \texttt{10} \end{array} \begin{pmatrix} \frac{4}{9} & \frac{2}{9} & \frac{1}{9} & \frac{2}{9} \\ \frac{2}{9} & \frac{4}{9} & \frac{2}{9} & \frac{1}{9} \\ \frac{1}{9} & \frac{2}{9} & \frac{4}{9} & \frac{2}{9} \\ \frac{2}{9} & \frac{1}{9} & \frac{2}{9} & \frac{4}{9} \end{pmatrix}.$$

This mechanism provides ϵ-DP *with* $\epsilon = 1$*. However, it is not* ϵ*-secure, as e.g.* $\frac{4/9}{1/9} = 4 > 2^{\epsilon}$.

We recover coincidence between ϵ-security and ϵ-DP if we confine ourselves to background knowledge that can be factorised as the product of independent measures over individuals. This provides another characterization of ϵ-DP. In what follows, for any $x \in \mathcal{X}$, we denote by $x_{\setminus i}$ the element of $\{0,1\}^{n-1}$ obtained by removing the i-th component from x.

Theorem 4.3.3 *The following statements are equivalent:*

1. *$\mathcal{D}$ satisfies ϵ-DP;*
2. *for each $i \in \{1,...,n\}$ and $p(x)$ of the form $p_i(x_i)q(x_{\setminus i})$, Q_i is not an ϵ-breach;*
3. *for each $p(x)$ of the form $\prod_{j=1}^{n} p_j(x_j)$ and for each $i \in \{1,...,n\}$, Q_i is not an ϵ-breach.*

Proof We will show that (1) $\Rightarrow$ (2) $\Rightarrow$ (3) $\Rightarrow$ (1).
(1) $\Rightarrow$ (2). Note that the absence of Q_i-breaches can be written as

$$|\log \frac{p(Q_i|y)}{p(Q_i)}| \leq \epsilon\,. \tag{4.8}$$

Let $X = (X_1,...,X_n)$, with X distributed according to $p(x)$, be the r.v. corresponding to the input of the mechanism, and Y be the corresponding output. Then we can write

$$\frac{p(Q_i|y)}{p(Q_i)} = \frac{p(X_i = 1|Y = y)}{p(X_i = 1)} = \frac{p(y|X_i = 1)}{p(y)}.$$

Abbreviating $p(y|X_i = j)$, for $j = 0, 1$, by $p(y|j)$, we get that, since $p(y) = p(y|0)p(0) + p(y|1)p(1)$, the above term can be bounded thus:

$$\frac{p(y|X_i = 1)}{p(y)} = \begin{cases} 1 \leq 2^{\epsilon} & \text{if } \min\{p(y|0), p(y|1)\} = p(y|1) \\ \frac{p(y|1)}{p(y|0)} & \text{otherwise.} \end{cases}$$

We now prove that $\frac{p(y|1)}{p(y|0)} \leq 2^{\epsilon}$ (the proof that $\frac{p(y|1)}{p(y|0)} \geq 2^{-\epsilon}$ is similar). We can expand both numerator and denominator in the following way,

recalling that $p(x) = p_i(x_i)q(x_{\setminus i})$ (the index x^{n-1} runs over $\{0,1\}^{n-1}$):

$$\begin{aligned} p(y|1) &= \sum_{x^{n-1}} \frac{p(Y=y, X_{\setminus i} = x^{n-1}, X_i = 1)}{p(X_i = 1)} \\ &= \sum_{x^{n-1}} \frac{p(y|X_{\setminus i} = x^{n-1}, X_i = 1)p(X_{\setminus i} = x^{n-1}, X_i = 1)}{p(X_i = 1)} \\ &= \sum_{x^{n-1}} \frac{p(y|X_{\setminus i} = x^{n-1}, X_i = 1)q(x^{n-1})p(X_i = 1)}{p(X_i = 1)} \\ &= \sum_{x^{n-1}} p(y|X_{\setminus i} = x^{n-1}, X_i = 1)q(x^{n-1})\,. \end{aligned}$$

Similarly, $p(y|0) = \sum_{x^{n-1}} p(y|X_{\setminus i} = x^{n-1}, X_i = 0)q(x^{n-1})$. Therefore, abbreviating $p(y|X_{\setminus i} = x^{n-1}, X_i = j)$ as $p(y|x^{n-1}, j)$, we obtain

$$\frac{p(y|1)}{p(y|0)} = \frac{\sum_{x^{n-1}} p(y|x^{n-1}, 1)q(x^{n-1})}{\sum_{x^{n-1}} p(y|x^{n-1}, 0)q(x^{n-1})} \leq \max_{x^{n-1}} \frac{p(y|x^{n-1}, 1)}{p(y|x^{n-1}, 0)} \leq 2^{\epsilon}$$

where: in the last but one step we use the general inequality $\frac{\sum_j m_j}{\sum_j M_j} \leq \max_j \frac{m_j}{M_j}$, which holds for nonnegative reals m_j, M_j; and in the last step we use the ϵ-DP provided by the mechanism and the fact that we are dealing with adjacent databases.

(2) $\Rightarrow$ (3). Any prior of the form in (3) is clearly a prior that factorizes as required by (2), for any i, hence the thesis.

(3) $\Rightarrow$ (1). Assume $\mathcal{D}$ does not provide ϵ-DP, that is there exist $y, x \sim x'$ such that $\frac{p(y|x)}{p(y|x')} > 2^{\epsilon}$. Then, for a prior as required by (3), we can build an ϵ-breach Q_i. Assume w.l.o.g. that x contains i and x' does not. We can build a prior of the wanted form that concentrates all probability mass on x and x', as follows: $p(x) \stackrel{\triangle}{=} \prod_j p_j(x_j)$ where the $p_j(\cdot)$'s are such that, for $j \neq i$, $p_j(x_j) \stackrel{\triangle}{=} 1$ and $p_j(\overline{x}_j) \stackrel{\triangle}{=} 0$. We can then repeat the reasoning in the proof of Theorem 4.3.1 to show that, as $p(x) = p_i(x_i) \to 0$, one has that $\frac{p(Q_i|y)}{p(Q_i)} \to \frac{p(y|x)}{p(y|x')} > 2^{\epsilon}$; hence Q_i is an ϵ-breach for $p(x)$ as required. $\square$

4.4 Discussion

We round off the technical analysis of the previous section with an informal discussion about the significance of the obtained results, and the relationship between DP and worst-case security. The notion of DP is often associated with popular claims of robustness in the face of arbitrary background/external knowledge. The actual meaning of this resistance is best understood by contrasting the following two properties, which are related to DP and worst-case security respectively, with one another.

- Property 1: semantical privacy, SP (Dwo06). Regardless of external knowledge, an adversary with access to the sanitized database draws the same conclusions about the database whether or not my data is included in the original data.
- Property 2: no evidence of participation, NEP. Regardless of external knowledge, adversary's access to the sanitized database does not modify his belief about inclusion of my data.

Property 1, SP, is stated in (DMNS06; Dwo06) and formalized in (GKS08), to which we refer for a formal definition and a proof that it is satisfied by DP. Basically, SP requires that for any individual i and observation y, the posterior distributions over $\mathcal{X}$ given by $p(\cdot|y)$ and $p_{-i}(\cdot|y)$ be indistinguishable, where $p_{-i}(\cdot|y)$ denotes the posterior induced by a version of the mechanism that suppresses individual i from the database before computing the answer. Although SP is an intuitively desirable property, it is not the same as Property 2, NEP, which is what one would ideally expect from a privacy-enforcing mechanism.

As seen in Theorem 4.3.1, NEP precisely characterizes worst-case security. On the other hand, Theorem 4.3.2 makes it clear that NEP is stronger than DP, hence SP: a differentially private mechanism may exhibit privacy breaches of arbitrarily large seriousness, depending on the size of the database n. An inspection of the proof reveals the reason: NEP conveys protection against *correlation*, whereas DP/SP in general does not. To see why, assume there is a group $G \subseteq \{1, ..., n\}$ of highly correlated individuals, whose size is non negligible with respect to n. Inclusion or

exclusion of the whole G from the (sanitized) database can noticeably affect the statistics, hence the reported answer. Since the adversary knows about G's correlation, as encoded by his prior, he can detect inclusion or exclusion of G given the reported answer. This is equivalent to detecting participation/non-participation of any individual in G, hence to a violation of NEP. Yet, DP may not be violated in these cases, because inserting/removing a single individual - as opposed to a whole group - does not significantly change the computed and reported answers. The purpose of the following example is to provide a concrete, if slightly artificial, illustration of such a situation.

Example 4.4.1 *The following background information is available to an adversary. G is a basketball team, say the LA Lakers, that got involved in some accident: nothing serious, maybe they went to a local hospital for a check up, maybe not. The adversary now wants to know whether the player Kobe Bryant has been hospitalized or not at the local hospital, being initially neutral with respect to the two possibilities. The adversary queries the hospital's database for the average height of people hospitalized this week. The answer mechanism enforces ϵ-DP, for a small ϵ: indeed, removing or inserting any specific individual, however tall, does not change noticeably the statistics and the reported answers. However, including or removing the whole G does noticeably affect the statistics and the answer. In particular, if the reported answer is noticeably greater than, say, the U.S. average height, then with high probability the team has been hospitalized, otherwise it has not. Hence the adversary learns with high probability if Kobe Bryant is in the database or not, so* NEP *is breached.*

Examples similar in spirit to the one above are provided also in (KM12; KM11). For instance, a key role is played by correlation between records, or correlation induced by previously released *exact* answers. In all these cases, DP is not strong enough to guarantee that the changes in the adversary's beliefs after interaction with the mechanism are small. It is therefore natural to seek for natural conditions under which DP provides a form of NEP. This we have done in Theorem 4.3.3: DP and worst-case security coincide when removing correlation; that is, when confining to priors that encode independent participation of individuals to the database. We note that a partial coincidence is recovered also in other situations: for instance, assuming 'informed' adversaries who know all of the database but the

entry of a target individual i, cf. (DMNS06). Intuitively, if the adversary knows almost everything about the database, under DP the observation of y cannot significantly change his belief about i's participation. However, such informed adversaries seem quite unrealistic in practice. And, as a matter of fact, an ignorant adversary can potentially gain much more information about i than an informed one – in particular, an adversary knowing everything has nothing to learn.

The other side of the coin concerns the tradeoff between privacy and utility. In fact, results in among others (KM12) - somewhat simplifying those by Dwork and Naor in (Dwo06; DN10) - imply that it is basically impossible to guarantee both NEP-like privacy and utility without making assumptions about the data generation mechanism. This implies that, in general, worst-case security *per se* conveys nearly zero utility: it serves just as a benchmark against which to compare more useful notions of privacy. From our point of view, zero or low correlation among individuals seems a good rule of thumb for obtaining strong privacy guarantees from employing DP.

4.5 Privacy under repeated observations

We assume that the attacker collects a tuple $y^n = (y_1, y_2, \ldots, y_n) \in \mathcal{Y}^n$ of observations generated i.i.d from the mechanism $\mathcal{R}$. We expect that, given Q, as n grows, the breach level approaches a threshold value. In order to the characterize synthetically the security of the randomization mechanism, though, it is important to characterize *how fast* this threshold is approached. We recall the definition of rate given in Def. 17. Let $\{a_n\}_{n\geq 0}$ be a sequence of nonnegative reals. Assume that $\tau = \lim_{n\to+\infty} a_n$ exists and that $a_n \leq \tau$ for each n. Then the rate of $\{a_n\}_{n\geq 0}$ is $\mathrm{rate}(\{a_n\}) \triangleq \lim_{n\to+\infty} -\frac{1}{n}\log(\tau - a_n)$ provided this limit exists. More generally, we define the upper-rate (resp. lower-rate) $\overline{\mathrm{rate}}(\{a_n\})$ (resp. $\underline{\mathrm{rate}}(\{a_n\})$) by replacing the $\lim$ by $\limsup$ (resp. $\liminf$). Again, we distinguish a worst-from an average-case scenario and, for the rest of the section, fix a generic randomization mechanism $\mathcal{R} = (\mathcal{X}, \mathcal{Y}, p(\cdot|\cdot))$ as defined in Def. 2.

4.5.1 Worst-case scenario

We begin with an obvious generalization of the notion of breach.

Definition 32 (n-breach of level ϵ) *A* (n, ϵ)-privacy breach *is a subset* $Q \subseteq \mathcal{X}$ *s.t. for some prior distribution* $p(x)$ *on* $\mathcal{X}$*, we have that* $p(Q) > 0$ *and*

$$\max_{p(y^n)>0} |\log \frac{p(Q|y^n)}{p(Q)}| > \epsilon\,.$$

The next proposition says that a notion of security based on bounding the level of n-breaches is not viable. For the sake of simplicity, we shall discuss some of the following results in the case $\mathcal{R}$ is non-singular (all rows of the matrix are distinct).

Proposition 9 *Assume* $\mathcal{R}$ *is non-singular. For* n *large enough,* $\mathcal{R}$ *has* n*-breaches of arbitrary level. More explicitly, for any nonempty* $Q \subseteq \mathcal{X}$ *and any* $\epsilon \geq 0$ *there is a prior distribution* $p(x)$ *s.t. for any* n *large enough there is* y^n *(*$p(y^n) > 0$*) such that* $\log \frac{p(Q|y^n)}{p(Q)} > \epsilon$*.*

The above proposition suggests that, in the case of a large number of observations, worst-case analysis should focus on how fast $p(Q|y^n)$ can grow, rather than looking at the maximum level of a breach.

Definition 33 (rate of a breach) *Let* $\rho \geq 0$*. A* breach of rate ρ *is a subset* $Q \subseteq \mathcal{X}$ *such that there exist a prior distribution* $p(x)$ *on* $\mathcal{X}$ *with* $p(Q) > 0$ *and a sequence of tuples,* $\{y^n\}_{n\geq 0}$*, with* $p(y^n) > 0$*, such that* $p(Q|y^n) \doteq 1 - 2^{-n\rho'}$ *with* $\rho' > \rho$*. A randomization mechanism is* ρ-rate secure *if it has no privacy breach of rate* ρ*. The* rate security level *is defined as* $\rho_{\mathcal{R}} \stackrel{\triangle}{=} \inf\{\rho \geq 0 :$ $\mathcal{R}$ *is* ρ*-rate secure*$\}$*.*

Theorem 4.5.1 $\mathcal{R}$ *is* ρ*-rate secure iff* $\rho \geq \log \max_y \frac{\pi_{M,y}}{\pi_{m,y}}$*.*

Proof Assume first $\rho \geq \log \frac{\pi_M}{\pi_m}$, we show that the mechanism is ρ-rate secure. Consider any Q and $p(x)$ s.t. $p(Q) > 0$ and $p(Q|y^n) \to 1$. Assume w.l.o.g. that $p(Q^c) > 0$. We show that $\mathrm{rate}(p(Q|y^n)) \leq \rho$. We let $\pi_M = p(y^*|x_1)$ and $\pi_m = p(y^*|x_2)$, where $(y^*, x_1, x_2) = \mathrm{argmax}_{y,x,x'} \frac{p(y|x)}{p(y|x')}$. We

lower-bound $1 - p(Q|y^n) = p(Q^c|y^n)$ as follows

$$p(Q^c|y^n) = \frac{\sum_{x\in Q^c} p(y^n|x)p(x)}{\sum_x p(y^n|x)p(x)} \geq \sum_{x\in Q^c} \frac{p(y^n|x)p(x)}{p(y^n|x^*)}$$
$$= \sum_{x\in Q^c} \Pi_{i=1}^n \frac{p(y_i|x)}{p(y_i|x^*)} p(x) \geq \sum_{x\in Q^c} p(x)(\frac{\pi_m}{\pi_M})^n = p(Q^c)(\frac{\pi_m}{\pi_M})^n,$$

where $x^* = \text{argmax}_{p(x)>0}\, p(y^n|x)$. Taking the $-\frac{1}{n}\log$ of both sides of the inequality thus obtained, we get

$$-\frac{1}{n}\log(1 - p(Q|y^n)) \leq -\frac{\log p(Q^c)}{n} + \log\frac{\pi_M}{\pi_m}\,.$$

As $n \to +\infty$, we get $\text{rate}(p(Q|y^n)) \leq \log\frac{\pi_M}{\pi_m} \leq \rho$.

On the contrary, assume $\rho < \log\frac{\pi_M}{\pi_m}$: then take $Q = \{x_1\}$, $p(x_1) = p(x_2) = \frac{1}{2}$ and, for each $n \geq 0$, $y^n = (y^*, \cdots, y^*)$ (n times). Clearly, $p(Q|y^n)$ reaches 1 at rate $\log\frac{\pi_M}{\pi_m} > \rho$, so the mechanism is not ρ-rate secure. □

The above theorem says that, for large n, the seriousness of the breach, for certain y^n, can be as bad as $\approx \log\dfrac{1-(\pi_m/\pi_M)^n}{p(Q)}$. This result, however, does not tell us *how likely* a serious breach is depending on n. The next result shows that the probability that *some* observable y^n causes a Q-breach grows exponentially fast. We premise some notation.

Fix a prior $p(x)$ over $\mathcal{X}$. Recall that we let $X \sim p(x)$ denote a random variable representing the secret information, and $Y^n = (Y_1, ..., Y_n)$ be the corresponding random vector of n observations, which are i.i.d. given X. Let us fix $Q \subseteq \mathcal{X}$ s.t. $p(Q) > 0$. Then $p(Q|Y^n)$ is a random variable. For any fixed $\epsilon > 0$, let us consider the two events

$$Breach_n^\epsilon \triangleq \{\frac{p(Q|Y^n)}{p(Q)} > 2^\epsilon\} \quad \text{and} \quad \overline{Breach}_n^\epsilon \triangleq \{\frac{p(Q)}{p(Q|Y^n)} > 2^\epsilon\}\,.$$

Clearly, the event $Breach_n^\epsilon \cup \overline{Breach}_n^\epsilon$ is the event that Y^n causes a Q-breach of level ϵ. As n grows, we expect that the probability of this event approaches 1 quite fast. The next theorem tells us exactly how fast.

Theorem 4.5.2 *Assume $\mathcal{R}$ is non-singular and strictly positive. Then, with the notation introduced above*

$$\Pr(Breach_n^\epsilon | X \in Q) \doteq 1 - 2^{-nC} \quad \textit{and} \quad \Pr(\overline{Breach}_n^\epsilon | X \in Q^c) \doteq 1 - 2^{-nC}$$

where $C = \min_{x \in Q, x' \in Q^c} C(p_x, p_{x'})$, with the understanding that x and x' in the min *are taken of positive probability. As a consequence, the probability that Y^n causes a Q-breach reaches* 1 *at rate at least C.*

The proof can be found in Appendix A.2.

Remark 4.5.1 *Note that the rate at which the probability of a breach approaches 1 does not depend on the level ϵ; moreover, it only depends on the support of the prior distribution. See Appendix A.2 for further details.*

4.5.2 Average-case scenario

It is straightforward to extend the definition of average-case breach to the case with multiple observations. For any nonempty subset $Q \subseteq \mathcal{X}$, and random variable $X \sim p(x)$, s.t. $p(Q) > 0$, we consider $\hat{Q} = I_Q(X)$ and define the leakage imputable to Q after n observations as

$$\mathcal{L}^n(\mathcal{R}; \hat{Q}; p) \triangleq H_\infty(\hat{Q}) - H_\infty(\hat{Q}|Y^n).$$

An n-breach of level $\epsilon \geq 0$ is a Q such that $\mathcal{L}^n(\mathcal{R}; \hat{Q}; p) > \epsilon$. Recall from (2.7) that $P_{succ}^n = 2^{-H_\infty(\hat{Q}|Y^n)}$ is the success probability of guessing between $p(\cdot|Q)$ and $p(\cdot|Q^c)$ after observing Y^n. Provided $p(\cdot|Q) \neq p(\cdot|Q^c)$, (2.17) implies that, as $n \to +\infty$ we have $P_{succ}^n \to 1$, hence $\mathcal{L}^n(\mathcal{R}; \hat{Q}; p) \to -\log \max\{p(Q), 1 - p(Q)\}$. If $p(\cdot|Q) = p(\cdot|Q^c)$ then P_{succ}^n is constantly $\max\{p(Q), 1 - p(Q)\}$, so that the observations give no advantage to the attacker. These remarks suggest that, in the case of repeated observations, it is again important to characterize how fast $P_{succ}^n \to 1$.

Definition 34 (rate of a breach - average case) *Let $\rho \geq 0$. An* A-breach of rate *ρ is a subset $Q \subseteq \mathcal{X}$ such that for some prior distribution $p(x)$ on $\mathcal{X}$ with $p(Q) > 0$ one has that $P_{succ}^n \doteq 1 - 2^{-n\rho'}$, for some $\rho' > \rho$. A randomization mechanism is* ρ-rate A-secure *if it has no privacy breach of rate ρ. The* rate A-security level *is defined as $\rho_{\mathcal{R}}^A \triangleq \inf\{\rho \geq 0 : \mathcal{R}$ is ρ-rate A-secure$\}$.*

Now we can proceed with the following theorem.

Theorem 4.5.3 *$\mathcal{R}$ is ρ-rate A-secure iff* $\max_{x,x'} C(p_x, p_{x'}) \leq \rho$.

In the following proof we use the fact that the Chernoff Information $C(p,q)$ is a convex function of both p and q. See Appendix A.2 for a complete proof of this claim.

Proof First assume that $\max_{x,x'} C(p_x, p_{x'}) \leq \rho$, we will show that the mechanism is ρ-rate A-secure. Assume Q is an A-breach of rate ρ'. This implies that $p(\cdot|Q) \neq p(\cdot|Q^c)$, otherwise $P^n_{succ} \not\to 1$. Also assume w.l.o.g. that $p(Q^c) > 0$. From (2.17) we have:

$$\rho' = \mathrm{rate}(P^n_{succ}) = C(p(\cdot|Q), p(\cdot|Q^c))\,.$$

Both $p(\cdot|Q)$ and $p(\cdot|Q^c)$ can be written as convex combinations of the distributions p_x's: $p(\cdot|Q) = \sum_{x\in Q} \lambda_x p_x$ and $p(\cdot|Q^c) = \sum_{x'\in Q^c} \lambda_{x'} p_{x'}$, where $\lambda_x = \frac{p(x)}{p(Q)}$ and $\lambda_{x'} = \frac{p(x')}{p(Q^c)}$. Using Lemma A.2.2 and Jensen's inequality

$$\begin{aligned}
\rho' &= C(p(y|Q), p(y|Q^c)) \leq \sum_{x\in Q, x'\in Q^c} \lambda_x \lambda_{x'} C(p_x, p_{x'}) \\
&\leq \max_{x\in Q, x'\in Q^c} C(p_x, p_{x'}) \\
&\leq \max_{x,x'} C(p_x, p_{x'}) \leq \rho\,.
\end{aligned}$$

On the other hand, assume there are $x, x' \in \mathcal{X}$ such that $C(p_x, p_{x'}) > \rho$, let $p(x) = p(x') = \frac{1}{2}$ and let $Q = \{x\}$: clearly $p(\cdot|Q) = p_x$ and $p(\cdot|Q^c) = p_{x'}$, so that $\mathrm{rate}(P^n_{succ}) = C(p_x, p_{x'}) > \rho$ in this case. That is, Q is a breach of rate ρ. □

We end the section with an example where three different mechanisms are compared with respect to the various scenarios we have analyzed. This example also demonstrates another important point: there is no general implication between the two considered security notions. That is, it may happen that mechanism A is better than B according to worst-case security, but worse than B according to average-case security.

Example 4.5.1 *The mechanisms considered in this example are inspired by examples in (EGS03). The private information is represented by a set of integers*

$\mathcal{X} = \{0, ..., 1000\}$, and $\mathcal{Y} = \mathcal{X}$. We consider three mechanisms that replace any $x \in \mathcal{X}$ by a random number $y = R_i(x)$ $(i = 1, 2, 3)$, respectively, that retains some information about the original x. More specifically, given $x \in \mathcal{X}$, we let:

1. *$R_1(x)$ be x with probability 0.2 and some other number (chosen uniformly at random) with probability 0.8;*
2. *$R_2(x)$ be $x + \xi \mod 1001$, where ξ is chosen uniformly at random in $\{-100, \ldots, 100\}$;*
3. *$R_3(x)$ be $R_2(x)$ with probability 0.5 and a uniformly random number otherwise.*

We can easily compute the conditional probability matrices. In the case of R_1, we have: $p_1(y|x) = 0.2$ if $y = x$, $p_1(y|x) = \frac{0.8}{1000}$ if $x \neq y$. Therefore the diagonal elements are all equals to 0.2, while the extra-diagonal ones are $\frac{0.8}{1000}$. In the case of R_2, if we let $d_{yx} \triangleq (y - x \mod 1001)$ and $d_{xy} \triangleq (x - y \mod 1001)$, then we have $p_2(y|x) = \frac{1}{201}$ if $\{d_{yx}, d_{xy}\} \cap \{0, \ldots, 100\} \neq \emptyset$, $p_2(y|x) = 0$ otherwise. Finally, the matrix form for R_3 is easily computed from the previous one: $p_3(y|x) = \frac{0.5}{1001} + 0.5p_2(y|x)$. The following table summarizes the results obtained for the mechanisms R_1, R_2, R_3.

	R_1		R_2		R_3	
	ϵ	ρ	ϵ	ρ	ϵ	ρ
Worst	*7.965*	*7.965*	∞	∞	*2.58*	*2.58*
Average	*0.262*	*0.278*	*1*	∞	*0.584*	*0.339*

Recalling that from a pure privacy-preserving point of view the smaller the better, it is clear that R_2 is the worst of the three mechanisms. Whether R_1 or R_3 should be preferred depends on the chosen notion of security: indeed, R_1 is less reliable than R_3 in the worst case, but more reliable in the average case.

4.6 Utility under repeated observations

We next turn to the study of utility. In the rest of the section, we fix a mechanism $\mathcal{R}$ and a prior distribution $p(\cdot)$. Without any significant loss of generality, we shall assume that $\mathcal{R}$ is strictly positive and that $\mathrm{supp}(p) = \mathcal{X}$. Moreover, in this section, we shall work under the more general assumption that $\mathcal{Y}$ is *finite or denumerable*.

For any $n \geq 1$, we are now going to define the expected utility of $\mathcal{R}$, depending on user-specific belief, modeled as a prior $p(\cdot)$ on $\mathcal{X}$, and on

function $loss : \mathcal{X} \times \mathcal{X} \to \mathbb{R}^+$. Here, $loss(x, x')$ represents the loss of a user who interprets the result of an observation of $\mathcal{R}$ as x', given that the real answer is x. For the sake of simplicity, we shall assume that $loss$ achieves a proper minimum when $x = x'$: for each $x \neq x'$, $loss(x, x) < loss(x, x')$. We also presuppose a *guessing function* $g : \mathcal{Y}^n \to \mathcal{X}$. The *expected utility* of $\mathcal{D}$ – relative to g – after n observations is in fact defined as an expected loss (the lower, the better), thus

$$U_n \stackrel{\triangle}{=} \sum_x p(x) \sum_{y^n} p(y^n|x) loss(x, g(y^n)) \,. \tag{4.11}$$

Note that this definition coincides with that of Ghosh et al. (GRS09) when one interprets our guessing function g as the *remap* considered in (GRS09). This is also the utility model of Alvim et al. (AACP11a), modulo the fact they only consider the 0/1-loss, or better, the complementary gain.

Example 4.6.1 *When $\mathcal{Y}$ is a subset of the reals, legal $loss$ functions include the absolute value error $loss(x, x') = |x' - x|$ and the squared error $loss(x, x') = (x' - x)^2$. The binary loss function defined as 0 if $x = x'$ and 1 otherwise is another example: the resulting expected loss is just error probability, $U_n = P_e^n$.*

It is not difficult to argue that g is asymptotically optimal if it respects the MAP criterion: $p(g(y^n)|y^n) \geq p(x|y^n)$ for each $x \in \mathcal{X}$; by asymptotically optimal, we mean that g maximizes the growth rate of success probability, see (BC13) for details. Henceforth, we just assume that g is a fixed MAP function. Below, we study the behavior of utility in relation to differential privacy. The crucial quantity is

$$\rho_{\mathcal{R}} \stackrel{\triangle}{=} \min_{x,x' \in \mathcal{X}, p_x \neq p_{x'}} C(p_x, p_{x'}) \,. \tag{4.12}$$

We will show that the the asymptotic rate of utility is determined solely by $\rho_{\mathcal{R}}$. Note that this quantity does not depend on the user-defined $loss$ function, nor on the prior $p(\cdot)$. For the sake of simplicity, below we discuss the result only in the case when $\mathcal{R}$ is non-singular[1].

Remark 4.6.1 *We note that the formula* (2.14) *for Chernoff Information extends to the case when $p(\cdot)$ and $q(\cdot)$ have the same* denumerable *support.*

[1]The result carries over to the general case, at the cost of some notational burden: one has to replace $U_{\mathcal{R}}$ with a more complicated expression.

Theorem 4.6.1 *Assume $\mathcal{R}$ is non-singular. Then $U_n \doteq U_{\mathcal{R}} + 2^{-n\rho_{\mathcal{R}}}$, where $U_{\mathcal{R}} \stackrel{\triangle}{=} \sum_x p(x) loss(x,x)$.*

Proof Fix $n \geq 1$ and let $g : \mathcal{Y}^n \to \mathcal{X}$ be the underlying MAP guessing function. For each x, let $A_n(x) = g^{-1}(x)$ the subset of $\mathcal{Y}^n$ corresponding to choosing x (the acceptance region for x); clearly $A_n^c(x) = \cup_{x' \neq x} A_n(x')$, and the $A_n(x')$'s are pairwise disjoint; note that the probability of error can be written as $P_e^n = \sum_x p(x) p_x(A_n^c)$. Let $G = \max_{x \neq x'} loss(x, x')$. For each x we have

$$\begin{aligned}
\alpha_n(x) &\stackrel{\triangle}{=} \sum_{y^n} p(y^n|x) loss(x, g(y^n)) \\
&= \sum_{x' \neq x} p_x(A_n(x')) loss(x, x') + p_x(A_n(x)) loss(x,x) \\
&\leq \sum_{x' \neq x} p_x(A_n(x')) G + loss(x,x) = p_x(A_n^c(x)) G + loss(x,x)\,.
\end{aligned}$$

This, when averaged over all x's, yields

$$U_n = \sum_x p(x)\alpha_n(x) \leq \sum_x p(x) p_x(A_n^c(x)) G + U_{\mathcal{R}} = P_e^n G + U_{\mathcal{R}}\,. \quad (4.14)$$

On the other hand, proceeding similarly to the previous case and letting $H \triangleq \max_x loss(x,x)$:

$$\begin{aligned}
\alpha_n(x) &\geq p_x(A_n(x)) loss(x,x) \\
&= (1 - p_x(A_n^c(x)) loss(x,x) \geq loss(x,x) - p_x(A_n^c(x)) H
\end{aligned}$$

which, when averaged over all x's, yields

$$U_n = \sum_x p(x)\alpha_n(x) \geq U_{\mathcal{R}} - P_e^n H\,. \quad (4.15)$$

By (2.18) we know that $P_e^n \to 0$; therefore (4.14) and (4.15) together imply that $U_n \to U_{\mathcal{R}}$. Concerning the rate, inequality (4.14) implies that $\underline{\mathrm{rate}}(\{U_n\}) \geq \mathrm{rate}(\{P_e^n\}) = \rho_{\mathcal{R}}$, from (2.18). On the other hand, inequality (4.15) implies that $\overline{\mathrm{rate}}(\{U_n\}) \leq \mathrm{rate}(\{P_e^n\}) = \rho_{\mathcal{R}}$, again from (2.18) (note that the nonnegativity of $U_{\mathcal{R}} - P_e^n H$ is guaranteed for n large enough). Therefore $\mathrm{rate}(\{U_n\}) = \rho_{\mathcal{R}}$. In conclusion, $U_n \doteq U_{\mathcal{R}} + 2^{-n\rho_{\mathcal{R}}}$. □

Having established the centrality of $\rho_{\mathcal{R}}$ in the asymptotic behavior of utility, we now discuss the relationship of this quantity with the worst-case security level ϵ provided by the mechanism. The first result provides us with a simple, general bound relating $\rho_{\mathcal{R}}$ and ϵ.

Theorem 4.6.2 *Assume $\mathcal{R}$ is worst-case ϵ-secure. Then $\rho_{\mathcal{R}} \leq \epsilon$. The same conclusion holds if $\mathcal{D} = (\mathcal{R}, \sim)$ provides ϵ-DP.*

Proof Worst-case security is a special case of DP if one takes as $\sim$ the clique topology over $\mathcal{X}$. So we only consider DP below. Let $x \neq x'$ and let d denote the length of the shortest path from x to x' in $\mathcal{X}$. Using (4.7), we have that for any $\lambda \in [0, 1]$ (below, we stipulate that $0 \cdot \frac{p_x(y)}{0} = 0$)

$$\sum_y p_x^{\lambda}(y) p_{x'}^{1-\lambda}(y) = \sum_y p_{x'}(y) (\frac{p_x(y)}{p_{x'}(y)})^{\lambda} \geq \sum_y p_{x'}(y) 2^{-\epsilon\lambda d} = 2^{-\epsilon\lambda d}\,.$$

From this it follows $-\log(\sum_y p_x^{\lambda}(y) p_{x'}^{1-\lambda}(y)) \leq \epsilon\lambda d \leq \epsilon d$, that achieves the minimum taking $d = 1$ (that is $x \sim x'$), hence the thesis by definition of Chernoff Information (2.14). □

In what follows, we will obtain more precise results relating ϵ to the utility rate $\rho_{\mathcal{R}}$ in the case of a class of mechanisms providing ϵ-DP. Specifically, we will consider mechanisms with a finite input domain $\mathcal{X} = \{0, 1, \ldots, N\}$, a denumerable $\mathcal{Y} = \mathbb{Z}$ and a conditional probability matrix of the form $p_i(j) = Mc^{|i-j|}$, for some positive $c < 1$. This class of mechanisms includes the geometric mechanism (a discrete variant of the Laplacian mechanism, see (GRS09)) and also a version extended to $\mathbb{Z}$ of the optimal mechanism considered by Alvim et al. (AACP11a).

Theorem 4.6.3 *Let $\mathcal{R}$ be a mechanism as described above. Then*

$$\rho_{\mathcal{R}} = \log(1 + c) - \frac{1}{2} \log c - 1\,.$$

Proof The statement is proven by the following main steps. In what follows, we let $h, i \in \mathcal{X}$.

1. Let j run over $\mathcal{Y}$ in the sum below, then

$$\begin{aligned}\sum_j p_h^{\lambda}(j)p_i^{1-\lambda}(j) &= M\sum_j c^{\lambda d_{hj}+(1-\lambda)d_{ij}} \\ &= M\cdot f(\lambda) \qquad (4.16)\end{aligned}$$

   where $f(\lambda) \stackrel{\triangle}{=} \sum_j c^{\lambda d_{hj}+(1-\lambda)d_{ij}}$.

2. It is easy to see that $f(\lambda)$ is a convex and differentiable function of λ in $[0,1]$. Hence, any point $\lambda \in (0,1)$ in which its derivative vanishes corresponds to a global minimum of f in $[0,1]$. We compute the derivative $df/d\lambda$:

$$\frac{df}{d\lambda}(\lambda) = \ln c \sum_j c^{\lambda d_{hj}+(1-\lambda)d_{ij}}(d_{hj}-d_{ij}). \qquad (4.17)$$

   We show that (4.17) vanishes at $\lambda = \frac{1}{2}$. Indeed, for each $j \in \mathcal{Y}$ that is at distance d from h and d' from i, there is a distinct $j' \in \mathcal{Y}$ that is at distance d' from h and d from i: in this way the terms $c^{\frac{1}{2}(d_{hj}+d_{ij})}(d_{hj}-d_{ij})$ and $c^{\frac{1}{2}(d_{hj'}+d_{ij})}(d_{hj'}-d_{ij'})$ in the above sum cancel out with one another. Hence $\frac{1}{2}$ is a global minimum for $f(\lambda)$ in $[0,1]$.

3. We compute $f(\frac{1}{2})$ as follows. We can partition the integers $\mathcal{Y}$ into those in between h and i, and those outside this interval. This gives rise to the following summations:

$$\begin{aligned}\sum_{j\in\mathcal{Y}} c^{\frac{1}{2}(d_{hj}+d_{ij})} &= (|h-i|+1)c^{\frac{1}{2}d_{hi}} + 2\sum_{d\geq 1} c^{\frac{1}{2}(d_{hi}+2d)} \\ &= (|h-i|+1)c^{\frac{1}{2}d_{hi}} + 2c^{\frac{1}{2}d_{hi}}\sum_{d\geq 1} c^{d} \\ &= c^{\frac{1}{2}d_{hi}}[|h-i|+1+2(\frac{1}{1-c}-1)]\,. \qquad (4.18)\end{aligned}$$

   Using this observation and the fact (4.18) is minimized when $|h-i| = 1$ we can compute as follows:

$$f(\frac{1}{2}) = c^{\frac{1}{2}}(\frac{2}{1-c})\,. \qquad (4.19)$$

4. Note that $M = \frac{1-c}{1+c}$. From the definition of Chernoff Information, (4.18) and (4.19)

$$\begin{aligned} C(p_h, p_i) &= \max_{\lambda} -\log(\sum_j p_h^{\lambda}(j) p_i^{1-\lambda}(j)) \\ &= -\log(M \cdot f(\frac{1}{2})) \\ &= \log(1+c) - \frac{1}{2}\log c - 1\,. \end{aligned} \tag{4.20}$$

Recalling that $\rho_{\mathcal{R}} = C(p_h, p_i)$ we get the wanted equality from (4.20).

□

Example 4.6.2 *The geometric mechanism is obtained by equipping the above described mechanism with the the line topology over $\mathcal{X} = \{0, ..., N\}$: $i \sim j$ iff $d_{ij} \stackrel{\triangle}{=} |i - j| = 1$. This is the topology for counting queries in "oblivious" mechanisms, for example. If we set $c = 2^{-\epsilon}$, then this mechanism provides ϵ-DP. The above theorem tells us that in this case $\rho_{\mathcal{R}} = \frac{\epsilon}{2} + \log \frac{1+2^{-\epsilon}}{2}$. By setting e.g. $\epsilon = 1$, one gets $\rho_{\mathcal{R}} \approx 0.085$.*

For any mechanism $\mathcal{R}$ with input $\mathcal{X} = \{0, ..., N\}$ and output $\mathcal{Y} = \mathbb{Z}$, we can consider the corresponding *truncated* mechanism $\mathcal{R}'$: it has $\mathcal{X} = \mathcal{Y} = \{0, 1, \ldots, N\}$ and its matrix is obtained from $\mathcal{R}$'s by summing all the columns $y < 0$ to column $y = 0$, and all the columns $y > N$ to column $y = N$.

Corollary 4.6.1 *Assume $\mathcal{R}'$ is the truncated version of a mechanism $\mathcal{R}$. Then $\rho_{\mathcal{R}'} < \rho_{\mathcal{R}}$.*

Proof We prove that the least Chernoff Information between any two different rows of $\mathcal{R}'$ is strictly less than the least Chernoff Information between any two different rows of $\mathcal{R}$. We first note that the function $x \mapsto x^{\lambda}$ is strictly concave in $[0, 1]$ for $\lambda \in (0, 1)$. Then using Jensen's inequality, we easily obtain:

$$\sum_y p_x^{\lambda}(y) p_{x'}^{1-\lambda}(y) < A^{\lambda} B^{1-\lambda} + \sum_{1<y<N} p_x^{\lambda}(y) p_{x'}^{1-\lambda}(y) + A'^{\lambda} B'^{1-\lambda}$$

where $A \triangleq \sum_{y \leq 0} p_x(y)$, $B \triangleq \sum_{y \leq 0} p_{x'}(y)$, $A' \triangleq \sum_{y \geq N} p_x(y)$ and $B' \triangleq \sum_{y \geq N} p_{x'}(y)$. We get the wanted result applying $-\log$ on both sides of the above inequality. □

In the case of a single observation case, treated by Ghosh et al. (GRS09), there is no substantial difference between the geometric mechanism and the truncated geometric one. Corollary 4.6.1 shows that the situation is different in the case with repeated observations.

4.7 Further and related work

There is a large body of literature on QIF (BCP09; KS10; BPP11a; BPP11b) and DP (Dwo06; DMNS06). The earliest proposal of a worst-case security notion is, to the best of our knowledge, found in (EGS03). As mentioned, the investigation of the relations between quantitative notions of leakage and DP has begun recently. Both (BK11) and (AACP11b; AACP11a) discuss the implication of ϵ-DP on information leakage guarantees, and vice-versa, in the case of a single observation. In the present work, we propose and characterize both worst- and average-case semantic notions of privacy breach, encoding resistance to arbitrary side-information, and clarify their relationships with QIF and DP. We also study the asymptotic behavior of privacy breaches depending on the number of observations.

Concerning the theme of resistance to background information, we note that results very similar to ours have recently been pointed out by Kifer et al. (KM12; KM11); this has been discussed in Section 4.4.

The notion of utility has been the subject of intensive investigation in the field of DP, see e.g. (MT07; GRS09; AAC$^+$11; AACP11b; AACP11a) and references therein. A general goal is that of designing mechanisms achieving optimal expected utility given a certain security level ϵ. Ghosh et al. (GRS09) propose a model of expected utility based on user preferences, and show that both the geometric mechanism and its truncated version achieve universal optimality. Here we provide the growth rate of utility, and we highlight a difference between a mechanism and its truncated version, in the presence of repeated observations. Alvim et al. (AAC$^+$11)

have shown the tight connection between utility and Bayes risk, hence information leakage, in the case of a single observation. A different, somewhat stronger notion of utility, called *accuracy*, is considered by McSherry and Talwar (MT07). They do not presuppose any user-specific prior over the set of possible answers; rather, they show that, in the exponential mechanism they propose, for any database, the expected *score* of the answer comes close to the maximum.

A problem left open by our study is the exact relationship between our average-case security notion and the maximum leakage considered in QIF – see e.g. (KS10). We would also like to apply and possibly extend the results of the present chapter to the setting of de-anonymization attacks on dataset containing micro-data. (NS08b) has shown that the effectiveness of these attacks depends on certain features of sparsity and similarity of the dataset, which roughly quantify how difficult it is to find two rows of the dataset that are similar. The problem can be formalized in terms of randomization mechanisms with repeated observations – see (BPP11b) for some preliminary results on this aspect. Then the row-distance measures considered in the present chapter appear to be strongly related to the notion of similarity, and might play a crucial in the formulation of a robust definition of dataset security.

The definitions of semantic security investigated in this chapter involve a quantification over all possible priors. In certain contexts, however, it may be sensible to put constraints on the form of the prior: one case we have considered is independent participation of individuals to a database, see Sections 4.3 and 4.4. Another possibility is to assume lower bounds on the entropy of the prior. Indeed, certain proofs obtained here - and elsewhere in the literature - exploit in a crucial way the existence of low entropy priors, where most of the probability mass is concentrated on very few elements. In several application scenarios (e.g. secret-key cryptography), such priors are unrealistic and could be ruled out. This would be in the line of work in entropic security, see e.g. (DS05).

Chapter 5

Estimating information flow

In this chapter, basing on (BPa), we address the problem of estimating information flow of a deterministic program. We start explaining the motivations that stimulated our investigation.

5.1 Motivations

The problem of exactly computing the information leakage of programs, or even that of giving nontrivial bounds that hold with certainty, turns out to be computationally intractable, even in the deterministic case (YT11). For this reason, there has been recently much interest towards methods for approximate calculation of information leakage (CCG10; CG11; CKN13; CK13; KR13).

Köpf and Rybalchenko method (KR13; KR10) relies on structural properties of programs: it leverages static analysis techniques to obtain bounds on information leakage. This method has proven quite successful in specific domains, like analysis of cache based side-channels (DFK$^+$13). In general terms, however, its applicability depends on the availability and precision of static analysis techniques for a specific domain of application. In many fields of Computer Science, sometimes a viable alternative to static analysis is represented by simulation. In the case of QIF, this prompts interesting research issues. We recall some of the questions that

we posed in Section 1.1: can one dispense with structural properties and adopt a black box approach, in conjunction with statistical techniques? What kind of formal guarantees can such an approach provide? And under what circumstances, if any, is it effective? This Chapter, based on (BPa), is meant as a systematic examination of such questions. We are not aware of previous work for QIF that systematically deals with such issues. Most related to ours are a few papers by Tom Chothia and collaborators (CCG10; CG11; CKN13; CK13; CKNP), which we will discuss in Section 5.7.

Our main object of study is maximum information leakage, known as *capacity*, for deterministic programs. For both Shannon- and min-entropy based QIF, capacity is easily proven to equal $\log k$, where k is the maximum number of distinct *outputs* the program can produce, taken over all possible input probability distributions. Capacity can also be explained in intuitive terms. An input distribution may be taken to represent the behaviour of a class of users feeding (secret) inputs to the the program. A program could come equipped with a synthetic safety guarantee in the form of a bound about its capacity. Violation of this guarantee implies that, due to unforeseen input distributions (behaviours of users), surprising or unexpected outputs can be triggered. In more detail, we give the following contributions.

1. After introducing a black-box, statistical set-up, we give a formal definition of program capacity estimator (Section 5.2). We then prove a few negative results (Section 5.3). Depending on the available knowledge about input generation, either no estimator exists (case of unknown input distribution); or no estimator exists that is significantly more efficient than exhaustive search on the input domain (case of input known to be uniform). Another result indicates that the source of this difficulty lies essentially in obtaining accurate *upper* bounds on capacity.

2. Motivated by the negative results, we introduce a weak estimator, J_t (Section 5.4): this provides lower bounds with high confidence under all distributions, and accurate upper bounds under output

distributions that are known to be, in a precise sense, close to uniform. The size of the required samples does not depend on that of the input domain.

3. We define sampling methods which, ideally, converges to an *optimal* input distribution, that is an input distribution under which the corresponding output is uniform (Section 5.5). Under such distribution, J_t can in principle provide both accurate lower and upper bounds on capacity. We then discuss how to turn this ideal algorithm into a practical methodology that at least provides good input distributions. This method is demonstrated with a few simulations (Section 5.6) that give encouraging results.

All in all, these results demonstrate some limits of the black-box approach (1), as well as the existence of positive aspects (2,3) that could be used, we believe, in conjunction with static analysis techniques. For ease of reading, the proofs of some technical results have been confined in Appendix A.3.

5.2 Statistical set up

We assume an input set $\mathcal{X} = \{x_1, x_2, ...\}$ and an output set $\mathcal{Y} = \{y_1, y_2, ...\}$, which are both finite and nonempty. As introduced in Section 2.1, we view a deterministic program P simply as a function $P : \mathcal{X} \to \mathcal{Y}$. We restrict our attention to terminating programs, or, equivalently, assume termination is an observable value in $\mathcal{Y}$. We model the program's input as a random variable X taking values in $\mathcal{X}$ according to some probability distribution. Since in the following we mostly focus on the probability distribution of Y, for sake of simplicity we denote by g the probability distribution of X and by p that of Y. For simplicity, we restrict ourselves to input distributions that have full support, that is $\mathrm{supp}(g) \triangleq \{x : g(x) > 0\} = \mathcal{X}$, and denote by $\mathcal{I}$ the set of such input distributions. Once fed to a program P, X induces an output $Y = P(X)$. We denote the set of outputs of nonzero probability as $\mathrm{supp}(Y)$ or $\mathrm{supp}(p)$. We let $|Y| \triangleq |\mathrm{supp}(p)|$ denote the size of this set. We let $g(x|y) \triangleq \Pr(X = x|Y = y)$ denote the a posteriori

input distribution after observation of an output y. Recall the definition of leakage in Def. 3, that, in the case of a deterministic program P, can be written as

$$\mathcal{L}_i(P;p) \triangleq H_i(X) - H_i(X|Y)$$

where $i \in \{\mathrm{Sh}, \infty\}$. Consequently the capacity, defined in Def. 3, can be expressed as $C_i(P) \stackrel{\triangle}{=} \sup_g L_i(P;g)$. Finally, recall that the image of a program (function) P is the set $\mathrm{Im}(P) \stackrel{\triangle}{=} \{y \in \mathcal{Y} : P^{-1}(y) \neq \emptyset\}$. Let us denote by $[x]$ the inverse image of $P(x)$, that is $[x] \stackrel{\triangle}{=} \{x' \in \mathcal{X} : P(x') = P(x)\}$. In this setting Theorem 2.3.1 is a crucial result that highlight the importance of the image size of P, $k = |\mathrm{Im}(P)|$ as security parameter.

Example 5.2.1 (cache side-channels) *Programs can leak sensitive information through timing or power absorption behaviour. Side-channels induced by cache behaviour are considered a particularly serious threat. The basic observation is that the lookup of a variable will take significantly more* CPU *cycles if the variable is not cached (*miss*), than if it is (*hit*). Consider the following program fragment, taken from (AS01).*

```
if ( h>0 )
    z = x;
else
    z = y;
z = x;
```

Assume an adversary can perform accurate time measurements, so that $\mathcal{Y} = \{t_1, ..., t_n\}$, a discrete set of execution times (or buckets thereof). Following the terminology of (DFK+13), we refer to this type of adversary as time*-based. If neither* `x` *nor* `y` *are cached before execution, there are only two possible execution times for the above program, say t_{short} and t_{long}. Observing t_{short} implies that the* `if` *condition is true, while t_{long} implies that the* `if` *condition is false. Assuming the value of the variable* `h` *is uniformly distributed over signed integers, this behaviour will actually reveal one bit about* `h`.

In a different scenario, related to e.g. power absorption measurements, the adversary might be able to detect if each individual memory lookup instruction causes a miss or a hit. In this case, the set of observables is $\mathcal{Y} = \{H, M\}^*$, where actually only a finite subset of these traces will have positive probability. For the program above, the only two possible traces are $tr_{short} = MH$ and $tr_{long} = MM$ and, again, they can reveal up to one bit about `h`. *We refer to this type of adversary as* trace*-based.*

In yet another scenario, the adversary might only be able to observe the final cache state, that is, the addresses of the memory blocks that are cached at the end of the execution - not their content. We refer to this type of adversary as access-*based. See (DFK+13) for additional details on the formalization of cache side-channels.*

We seek for a method to statistically estimate program capacity k, based on a sample of the outputs. This method should come equipped with formal guarantees about the accuracy and confidence in the obtained results, depending on the size of the sample. We will not assume that the code of the program is available for inspection. Of course, for any such method to be really useful, the size of the sample should be significantly smaller than the size of the input domain. That is, the statistical method should be substantially more efficient than exhaustive input enumeration.

We define the following statistical set up. We postulate that the input and output domains, $\mathcal{X}$ and $\mathcal{Y}$, are known to the analyst, and that P can be independently run a certain number of times with inputs generated according to some probability distribution g. However, we do not assume that the code of P is accessible for analysis. Moreover, we will typically assume the analyst has only some very limited or no information about the input distribution g. These assumptions model a situation where P's code cannot be analyzed, perhaps because it is a "live" application that cannot be inspected[1]; or P and its input generation mechanism are just too complex to fruitfully apply analysis tools. In Section 5.5, we will relax the assumption on the input distribution, and grant the analyst with some control over g.

Let $\mathcal{D}$ denote the set of all probability distributions on $\mathcal{Y}$. For any $\epsilon \geq 0$, define $\mathcal{D}_\epsilon \stackrel{\triangle}{=} \{p \in \mathcal{D} : \text{ for each } y \in \text{supp}(p), p(y) \geq \epsilon\}$; clearly $\mathcal{D}_\epsilon \subseteq \mathcal{D} = \mathcal{D}_0$. Another useful subset is $\mathcal{D}_u \stackrel{\triangle}{=} \{p \in \mathcal{D} : P(X_u) \sim p \text{ for some program } P\}$, where X_u is the uniform distribution over $\mathcal{X}$: this is the set of output distributions generated by considering all programs under the uniform input distributions. It is easy to see that $\mathcal{D}_u \subseteq \mathcal{D}_{\frac{1}{|\mathcal{X}|}}$.

[1]Note that we must distinguish the position of the analyst from that of the attacker. The analyst wishes to estimate capacity of a system he has only a limited, black-box access to. The attacker wants to recover (part of) the secret, and the black-box restriction may possibly not apply to him (e.g., he may have access to the source code of P).

Program outputs are generated a certain fixed number $m \geq 1$ of times, independently (this number may also depend on predefined accuracy and confidence parameter values; see below). In other words, we obtain a random sample of m outputs

$$S \stackrel{\triangle}{=} Y_1, ..., Y_m \quad \text{with } Y_i \text{ i.i.d. } \sim p(y) \tag{5.1}$$

for some in general *unknown* distribution $p \in \mathcal{D}$.

Our ideal goal is to define a random variable I that estimates the capacity $k = |Y| = |\text{supp}(p)|$. This estimator should be a function solely of the sequence of observed outputs, S: indeed, while the size of the input and output domains, $\mathcal{X}$ and $\mathcal{Y}$, are assumed to be known, both the program and its input distribution are in general unknown. Formally, we have the following definition, where the parameters γ and δ are used to control, respectively, the accuracy of the estimation and the confidence in it: a good estimator would have γ close to 1 and δ close to 0. The parameter $\mathcal{D}' \subseteq \mathcal{D}$ implicitly encodes any a priori partial information that may be available about the output distribution p (with $\mathcal{D}' = \mathcal{D}$ meaning no information).

Definition 35 (estimators) *Let $0 < \delta < 1/2$, $\gamma > 1$ and $\mathcal{D}' \subseteq \mathcal{D}$ be given. We say a function $I : \mathcal{Y}^m \to \mathbb{R}$ ($m \geq 1$) γ-approximates* program capacity under $\mathcal{D}'$ with confidence $1 - \delta$ *if, for each $p \in \mathcal{D}'$, the random variable $I \stackrel{\triangle}{=} I(S)$, with S like in* (5.1)*, satisfies the following where $k = |\text{supp}(p)|$:*

$$\Pr\left(k/\gamma \leq I \leq \gamma k\right) \quad \geq \quad 1 - \delta. \tag{5.2}$$

Note that in the above definition the size of the sample, m, may depend on the given δ, γ and set $\mathcal{D}'$.

5.3 Limits of program capacity estimation

We will show that, under very mild conditions on γ, the problem of γ-approximation of program capacity cannot be solved, unless nontrivial a priori information about the distribution p is available to the analyst. We first deal with the case where an $\epsilon > 0$ is known such that $p \in \mathcal{D}_\epsilon$.

Theorem 5.3.1 *Assume $\epsilon > 0$ and $\gamma < \min\{\sqrt{|\mathcal{Y}|}, \sqrt{(1-\epsilon)/\epsilon}\}$. Then any I that γ-approximates capacity under $\mathcal{D}_\epsilon$ requires a sample size of at least $m \geq \frac{\ln 2}{\gamma^2 \epsilon} - 1$, independently of any fixed confidence level.*

Proof Consider two distributions $p, q \in \mathcal{D}$ defined as follows. The distribution p concentrates all the probability mass in some $y_0 \in \mathcal{Y}$. The distribution q assigns y_0 probability $1 - h\epsilon$, and ϵ to each of h distinct elements $y_1, ..., y_h \in \mathcal{Y}$, where $h \stackrel{\triangle}{=} \lfloor \gamma^2 \rfloor$. Note that $1 \leq h \leq \gamma^2 < h + 1$, hence $h\epsilon \leq \gamma^2 \epsilon < \frac{1-\epsilon}{\epsilon} \cdot \epsilon = 1 - \epsilon$, and $h \leq \gamma^2 < |\mathcal{Y}|$, so that the distribution q is well-defined and in $\mathcal{D}_\epsilon$. Of course, p is in $\mathcal{D}_\epsilon$ as well.

Fix an arbitrary $0 < \delta < 1/2$, and assume by contradiction there is an estimator I under $\mathcal{D}_\epsilon$ with confidence $1 - \delta$ that uses m i.i.d. extractions, and that $m < \frac{\ln(1-\delta)}{\ln(1-\gamma^2\epsilon)}$. Now consider the sequence y_0^m, in which only y_0 occurs m times. Given the above strict upper bound on m, some easy calculations show that under either p or q, y_0^m has probability $> 1 - \delta > 1/2$. Let $a = I(y_0^m)$. By definition of estimator under $\mathcal{D}_\epsilon$, we must have both $1/\gamma \leq a \leq \gamma$ (since I is supposed to estimate $|\mathrm{supp}(p)| = 1$) and $(h+1)/\gamma \leq a \leq (h+1)\gamma$ (since I is supposed to estimate $|\mathrm{supp}(q)| = h+1$). From these inequalities, and using the fact that by construction $\gamma^2 < h + 1$, we have: $a \leq \gamma < (h + 1)/\gamma \leq a$, which implies $a < a$, a contradiction. Hence it must be $m \geq \frac{\ln(1-\delta)}{\ln(1-\gamma^2\epsilon)}$. Given that δ is arbitrary, by letting $\delta \to \frac{1}{2}$ we obtain that $m \geq \frac{\ln(1/2)}{\ln(1-\gamma^2\epsilon)} = -\frac{\ln 2}{\ln(1-\gamma^2\epsilon)}$.

Now by Taylor expansion, one sees that $-1/\ln(1-x) \geq x^{-1} - 1$ for $x \in (0, 1)$, which implies the wanted result. □

A variation of the above result concerns the case where we know that the output distribution is induced by a a uniform *input* distribution.

Theorem 5.3.2 *Assume $\gamma < \min\{\sqrt{|\mathcal{Y}|}, \sqrt{|\mathcal{X}| - 1}\}$. Then any I that γ-approximates capacity under $\mathcal{D}_u$ requires a sample size of at least $m \geq \frac{\ln 2}{\gamma^2}|\mathcal{X}| - 1$, independently of any fixed confidence level.*

Proof Let $\epsilon = 1/|\mathcal{X}|$, $h = \lfloor \gamma^2 \rfloor$ and consider distinct elements $x_0, x_1, ..., x_h$ in $\mathcal{X}$ and $y_0, y_1, ..., y_h$ in $\mathcal{Y}$. The output distributions p and q over $\mathcal{Y}$ defined in the proof of Theorem 5.3.1 are generated, under the uniform distribution

over $\mathcal{X}$, by the following two programs (functions) $P_i : \mathcal{X} \to \mathcal{Y}$, for $i = 1, 2$, respectively:

- $P_1(x) = y_0$ for each $x \in \mathcal{X}$;
- $P_2(x) = y_i$ if $x = x_i$ $(i = 1, ..., h)$, $= y_0$ otherwise.

Note that by virtue of the condition $\gamma < \min\{\sqrt{|\mathcal{Y}|}, \sqrt{|\mathcal{X}| - 1}\}$, P_2 is well defined and $q \in \mathcal{D}_u$. The rest of the proof is identical to that of Theorem 5.3.1. □

Hence, for γ close to 1, approximately $|\mathcal{X}|/\gamma^2 \approx |\mathcal{X}|$ i.i.d. extractions of Y are necessary for estimating capacity when it is known that $p \in \mathcal{D}_u$: this shows that no such estimator exists that is substantially more efficient than exhaustive search. We next give the main result of the section: if no partial information is available about p, then it is just impossible to estimate capacity.

Theorem 5.3.3 (non existence of estimators) *Assume that $\gamma < \sqrt{|\mathcal{Y}|}$. Then there is no I that γ-approximates capacity under $\mathcal{D}$, independently of the fixed confidence level.*

Proof Assume by contradiction the existence of one such estimator I, and let m be the size of the sample it uses. Take now $\epsilon > 0$ small enough such that $\frac{1-\epsilon}{\epsilon} > |\mathcal{Y}|$, so that $\gamma < \sqrt{\frac{1-\epsilon}{\epsilon}}$, and such that $m < \frac{\ln 2}{\gamma^2 \epsilon} - 1$. Since I is an estimator under $\mathcal{D}$, it is also an estimator under $\mathcal{D}_\epsilon$. Now applying Theorem 5.3.1, we would get $m \geq \frac{\ln 2}{\gamma^2 \epsilon} - 1$, which is a contradiction. □

We end the section with a result indicating that that the difficulty of estimating k is mostly due to obtaining accurate *upper* bounds on it. Let us say that a function $I : \mathcal{Y}^m \to \mathbb{R}$ is *reasonable* if its achieves its minimum for a string where a single output occurs m times. That is, for some $y_0 \in \mathcal{Y}$, $y_0^m \in \operatorname{argmin}_{\mathbf{y} \in \mathcal{Y}^m} I(\mathbf{y})$. The meaning of the following theorem is that any reasonable estimator I can nowhere be significantly smaller than the trivial upper bound $|\mathcal{Y}|$.

Theorem 5.3.4 *Assume there is a reasonable I such that, for some $\gamma > 1$ and $0 < \delta < 1/2$ and for any $p \in \mathcal{D}$ and $k = |\mathrm{supp}(p)|$, it holds that $\Pr(I \geq k/\gamma) \geq 1 - \delta$. Then it must be $I \geq |\mathcal{Y}|/\gamma$.*

Proof Assume w.l.o.g. that $|\mathcal{Y}| > 1$. Consider any I satisfying the hypotheses, and let $k^- \stackrel{\triangle}{=} \min\{I(\mathbf{y}) : \mathbf{y} \in \mathcal{Y}^m\}$, with $I(y_0^m) = k^-$. Choose any $0 < \epsilon < 1$ such that $\ln(1-\delta)/\ln(1-\epsilon) > m$. Consider the distribution p over $\mathcal{Y}$ which assigns y_0 probability $1-\epsilon$ and divides evenly the rest among the remaining elements: $p(y) = \epsilon/(|\mathcal{Y}|-1)$ for each $y \neq y_0$; clearly, $|\mathrm{supp}(p)| = k = |\mathcal{Y}|$. Now, since $m < \ln(1-\delta)/\ln(1-\epsilon)$, it is easy to see that, under p, $\Pr(S = y_0^m) > 1-\delta > \frac{1}{2}$. Hence, according to the hypotheses on I, it must be $I(y_0^m) = k^- \geq |\mathcal{Y}|/\gamma$. □

Note that no similar result can hold for lower bounds of program capacity: the trivial estimator $I \stackrel{\triangle}{=}$ n. of distinct elements occurring in S - which holds even with $\delta = 0$ and $\gamma = 1$ - shows that lower bounds exist that can be arbitrarily larger than the trivial 1. In the next section, we will concentrate our effort on analysing one such lower bound.

5.4 A weak estimator

The negative results on the existence of general estimators in the preceding section suggest we should relax our goals, and concentrate our efforts on lower bounds, and/or assume that some partial information about the program's output distribution is available. For example, if the code of P is accessible for inspection, this information could be derived by employing abstract interpretation techniques as suggested in (KR13). We seek for an estimator J that, with high confidence: (a) gives a lower-bound of k under all output distributions; (b) be close to k under the additional assumption that the underlying output distribution is "close to uniform". The last requirement means that, at least in the ideal situation where all output values are equally likely to be generated, the estimator should approximate k accurately. Finally, we would like the estimator be substantially more efficient than exhaustive search.

In what follows, we define a weak estimator based on the number of distinct element occurring in our sample. This is of course not the only statistics one could consider, but it turns out to be easy to analyse and

quite effective in practice[2]. Assume the distribution of Y is uniform, or close to uniform, on k elements. It is well known that, in this case, the expected number of distinct elements occurring at least once in a i.i.d. sample of Y is $k\big(1-(1-\frac{1}{k})^m\big) \approx k(1-\exp(-m/k))$, where m is the size of the sample (see below). Viewing this expression as a function of k, say $f(k)$, this suggests the estimation $k \approx f^{-1}(D)$, where D is the number of distinct elements that have *actually* been observed in S, rather than its expected value. There are a few points that need to be addressed in order to prove this a weak estimator in the above sense: mainly, the fact that the distribution we are faced with is possibly not uniform, and to what extent one can approximate $E[D]$ via D. We tackle these issues below.

Formally, let S be the sample defined in (5.1), obtained under a generic output distribution $Y \sim p(y)$.Consider the function defined as $f(x) \stackrel{\triangle}{=} x(1-(1-\frac{1}{x})^m)$, for $x \in [1, +\infty)$. It is easy to see that f is an increasing function of x; hence, its inverse f^{-1} exists and is in turn increasing on the same interval. We consider the following random variables, where the second one depends on a parameter $t \in \mathbb{R}$:

$$\begin{aligned} D &\stackrel{\triangle}{=} |\{y \in \mathcal{Y} : y \text{ occurs at least once in } S\}| \\ J_t &\stackrel{\triangle}{=} f^{-1}(D-t) \end{aligned}$$

with the proviso that $f^{-1}(x) \stackrel{\triangle}{=} 1$ for $x < 1$. Let us indicate by $E_u[D]$ the expected value of D under the distribution that is uniform on $\mathrm{supp}(p)$. Moreover, for each $\eta \geq 1$, let us say that p is *η-close to uniform* if for each $y \in \mathrm{supp}(p)$, $p(y) \geq \frac{1}{\eta|\mathrm{supp}(p)|}$. Note that p is uniform on its own support iff $\eta = 1$. The proof of the next lemma is reported in Appendix A.3.

Lemma 5.4.1 *Let $k = |\mathrm{supp}(p)|$. Then: (a) $f(k) = E_u[D] \geq E[D]$; (b) if additionally p is η-close to uniform, then $E[D] \geq f(k)/\eta$.*

The next lemma ensures that the measure of D is quite concentrated around its expected value: roughly, D is unlikely to deviate from $E[D]$ by more than $O(\sqrt{m})$. The proof is a standard application of McDiarmid's theorem, see for example (DP09, Chapter 5).

[2]In particular, we have found that statistics based on the number of collisions, such the index of coincidence, although easy to analyse, perform poorly in terms of accuracy

Lemma 5.4.2 *Let $t > 0$. Then $\Pr(D > E[D] + t) \leq \exp(-2t^2/m)$ and $\Pr(D < E[D] - t) \leq \exp(-2t^2/m)$.*

The next theorem formalizes the fact that J_t is a family of weak estimators.

Theorem 5.4.1 *Let $k = |\mathrm{supp}(p)|$. Let $m \geq 1$, $0 < \delta < 1/2$ and t such that $t \geq \sqrt{m \ln(1/\delta)/2}$. Then*

1. $\Pr(J_t \leq k) \geq 1 - \delta$;
2. *assume additionally that p is η-close to uniform. Let $t' \triangleq \eta t + (\eta - 1)D$. Then $\Pr(J_{-t'} \geq k) \geq 1 - \delta$.*

Proof Concerning part 1, from Lemma 5.4.1(a), we have that $f(k) \geq E[D]$. From Lemma 5.4.2, $E[D] \geq D - t$ with probability at least $1 - \delta$; hence, with probability at least $1 - \delta$, $f(k) \geq D - t$. Applying $f^{-1}(\cdot)$ on both sides of this inequality, the wanted statement follows.

Concerning part 2, from Lemma 5.4.2 we have that, with probability at least $1 - \delta$, $D + t \geq E[D]$. From Lemma 5.4.1(b), $E[D] \geq f(k)/\eta$; hence $D + t' = \eta(D + t) \geq f(k)$ with probability at least $1 - \delta$. Applying $f^{-1}(\cdot)$ on both sides of this inequality, the wanted statement follows. □

Part 1 of the above theorem says that J_t is in fact an underestimation of k. Moreover, under a p η-close to uniform, part 2 allows us to overapproximate k by $J_{-t'}$. Let us stress that the values of t, t' and m *do not depend on the size of the input* domain, $\mathcal{X}$. However, the above result does not give direct indications as to how m and t are related to the error in the estimation. This is is resolved in the following result, which establishes an explicit connection between m, t and the absolute error incurred when estimating k by J_t. This also gives indications as to the number of extractions m that is necessary to take in order to keep this error under control.

Corollary 5.4.1 *Let $k = |\mathrm{supp}(p)|$. Let $m \geq 1$, $0 < \delta < 1/2$ and t such that $t \geq \sqrt{m \ln(2/\delta)/2}$. Assume p is η-close to uniform and let t' as defined above. Then with probability at least $1 - \delta$,*

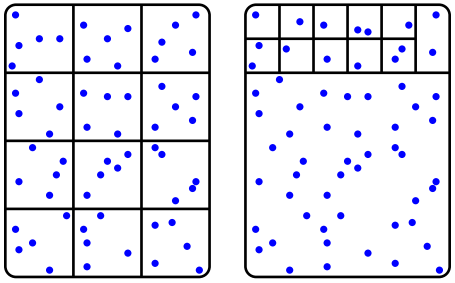
$$|k - J_t| \leq \frac{(\eta - 1)D + (\eta + 1)t}{f'(J_{-t'})} \approx \frac{(\eta - 1)D + (\eta + 1)t}{1 - e^{-m/J_{-t'}}(1 + m/J_{-t'})}\,.$$

Proof From the previous theorem, we know that with probability at least $1-\delta$, it holds that $J_t \leq k \leq J_{-t'}$. Hence it suffices to bound $J_{-t'} - J_t$. Now, $J_{-t'} - J_t = f^{-1}(\eta(D+t)) - f^{-1}(D-t)$; moreover, it is easy to check that that f^{-1} is Lipschitzian in the interval $[D-t, \eta(D+t)]$, with constant $\max_{z\in[D-t,\eta(D+t)]}(f^{-1})'(z) \leq 1/f'(f^{-1}(\eta(D+t))) = 1/f'(J_{-t'})$. It follows that $J_{-t'} - J_t = f^{-1}(\eta(D+t)) - f^{-1}(D-t) \leq \frac{(\eta-1)D+(\eta+1)t}{f'(J_{-t'})}$, hence the thesis. The approximation follows from $f(x) \approx x(1-e^{-m/x})$ and $f'(x) \approx 1 - e^{-m/x}(1+m/x)$. □

The right-hand side expression in the above corollary indicates that the absolute error will be small provided that the ratio $m/J_{-t'}$ is not too small. In practice, one should keep on sampling until $m/J_{-t'}$ is sufficiently high to guarantee the desired accuracy. The relative error $|k-J_t|/k$ can be bounded with high probability by $|k-J_t|/J_t$, hence the above corollary can still be used to this purpose.

Example 5.4.1 *Assume we are faced with a distribution p that is uniform on its own support, that is $\eta = 1$ and $t' = t$. We set our confidence level to $\delta = 0.001$ and decide to sample until we find $m/J_{-t} \geq 0.2$. Suppose we get this ratio when $m = 10^5$ and $D = 9 \times 10^4$. Setting t to the expression in Theorem 5.4.1, this gives $J_{-t} \approx 4.97 \times 10^5$, $J_t \approx 4.38 \times 10^5$ and a relative error bound by $(J_{-t} - J_t)/J_t \leq 0.135$. The theoretical bound we would get from Corollary 5.4.1 is slightly higher, about 0.15.*

Unfortunately, part 2 of Theorem 5.4.1 and Corollary 5.4.1 cannot be directly applied when we do not know η. Nevertheless, they do tell us something useful even in this case. First, Corollary 5.4.1 tells us that we can decrease the absolute error by decreasing $m/J_{-t'}$. Second, both Theorem 5.4.1(2) and Corollary 5.4.1 imply we can improve the quality of the estimation by tweaking the input distribution so as to make the resulting output distribution as close as possible to uniform - that is, η as close as possible to 1. In the next section, we define a methodology to this purpose.



(a) Balanced (b) Unbalanced

Figure 5: Balanced and unbalanced equivalence relations.

5.5 Searching for good input distributions

For a fixed program P, computing k is the same as counting the number of classes of the equivalence relation over $\mathcal{X}$ given by $x \sim x'$ iff $P(x) = P(x')$ – that is, counting the number of nonempty inverse images $P^{-1}(y)$, for $y \in \mathcal{Y}$. The weak estimator J_t implements a Monte-Carlo method to count such classes, based on counting the number of distinct outputs when inputs x are drawn according to a distribution $g(x)$. It should be intuitively clear that the accuracy of J_t depends critically on this g. In particular, if g is the uniform distribution on $\mathcal{X}$, as often found in practice, the resulting estimator J_t implements what we may call a *crude Monte Carlo* (CMC) method. CMC can perform well when the equivalence classes are more or less all of the same size, like in Fig. 5(a). On the other hand, consider a situation like in Fig. 5(b): under g, most of the sampled x's will fall in the big class, thus giving rise to a low value of D, hence of J_t, that will lead to severely underestimate the true value of k. In terms of Theorem 5.4.1(2), the resulting output distribution will have an η much higher than 1. In order to rectify this, one should modify g so as to "squeeze" the probability mass out of the big class and redistribute it on the small classes. Ideally, one should be able to obtain an optimal input distribution g^* such that all outputs (equivalence classes) are equally likely.

These considerations lead us to study a scenario where we grant the

analyst with some control over the input distribution. We will define two methods to search for good input distributions, one based on a Markov Chain Monte Carlo (MCMC) sampling algorithm, and one based on an Accept-Reject (AR) criterion. We will then outline a practical methodology based on these algorithms.

5.5.1 A Metropolis Monte Carlo method

We first note that, once P is fixed, an optimal input distribution g^* always exists. For example, denoting by $[x]$ the $\sim$-equivalence class of each $x \in \mathcal{X}$ according to P, we can set

$$g^*(x) \triangleq \frac{1}{k \times |[x]|} .$$

Of course, this expression requires knowledge of k, hence does not yield *per se* a method for computing g^*. We note, however, that for any two given x and x', the calculation of the ratio $g^*(x')/g^*(x) = |[x]|/|[x']|$ does not require k, which is canceled out. Now, it is reasonable to assume that this ratio can be, at least roughly, approximated. For example, we could count the number of elements so far found in $[x]$ and $[x']$ in a random walk in the state space $\mathcal{X}$ (see below); or we could use abstract interpretation techniques as suggested by (KR13). This puts us in a position to apply the *Metropolis* MCMC, as explained below. For a general treatment of MCMC see e.g. (CR).

The general idea of MCMC is that sampling from $\mathcal{X}$ according to a distribution, say g^*, can be accomplished by defining a Markov Chain whose (unique) stationary distribution is g^*. Metropolis MCMC, instantiated to our setting, can be described as follows. We fix a *proposal distribution* $Q(x'|x)$, according to which candidate next-states will be chosen. In the version we consider, this is required to be symmetrical and positive: $Q(x'|x) = Q(x|x') > 0$. Starting from an arbitrary initial state $x_0 \in \mathcal{X}$, a random walk on $\mathcal{X}$ is performed: at each step, starting from the current state x_t, a candidate next state x' is drawn according to $Q(x'|x_t)$. Now, x' can be either *accepted* ($x_{t+1} = x'$) or *rejected* ($x_{t+1} = x_t$): the first event

occurs with probability $\alpha(x, x') \stackrel{\triangle}{=} \min\{1, |[x]|/|[x']|\}$, the second one with probability $1 - \alpha(x, x')$.

This defines a random walk $\{X_t\}_{t\geq 0}$ on $\mathcal{X}$, which, so to speak, "keeps off the big classes": once a a big class $[x]$ is entered, it is likely to be exited immediately. This will happen indeed provided the proposal distribution will select a small class $[x']$ at the next step , as $|[x]|/|[x']| > 1$. As a result, after an initial "burn in" period, the random walk will tend to spend the same amount of time on each class. The theorem below is an instance of a far more general result (see e.g. (CR, Th.7.2)).

Theorem 5.5.1 *The stochastic process $\{X_t\}_{t\geq 0}$ defined above is a time-homogeneous Markov chain over $\mathcal{X}$ whose stationary distribution is g^*. That is, letting $X_t \sim g_t$, one has $\lim_{t\to+\infty} g_t = g^*$.*

A practical consequence of this theorem is that, after a burn in period of T steps, we can use the realizations x_t ($t \geq T$) of the stochastic processes as extractions drawn according to g^*. If we insist these extractions to be *independent*, we can run independently say m copies of the Markov chain for T steps each, and only keep the last state of each of the resulting random walks, say $x_T^{(i)}$, for $i = 1, ..., m$. Next, we can compute m independent samples of Y as $y_i \stackrel{\triangle}{=} P(x_T^{(i)})$, for $i = 1, ..., m$.

5.5.2 An Accept-Reject method

The Accept-Reject (AR) sampling method is a technique used to generate independent observations from a certain distribution. It is based on the observation that, to sample from a random variable, one can sample uniformly from the region under the graph of its density function.

The AR method generates sampling values from an arbitrary probability distribution function $f(x)$, by using an *instrumental* distribution $\hat{f}(x)$, under the only restriction that $f(x) < M\hat{f}(x)$, where $M > 1$ is an appropriate bound on $\frac{f(x)}{\hat{f}(x)}$. This sampling method is usually used in cases where the form of the distribution $f(x)$ makes sampling difficult. Instead of sampling directly from the distribution , we use an envelope distribution $M\hat{f}(x)$ where sampling is easier. These samples from $M\hat{f}(x)$ are probabilistically accepted or rejected.

We give a sketch of the procedure:

1. sample x according to $\hat{f}(x)$

2. generate u uniformly at random in $[0, 1]$

3. if $u < \frac{f(x)}{M\hat{f}(x)}$ accept x and return it; else reject x and repeat from 1.

Note that when simulating the pair $(x, v \triangleq uM\hat{f}(x))$, one produces a uniform simulation over the subgraph of $M\hat{f}(x)$. Accepting only pairs such that $u < \frac{f(x)}{M\hat{f}(x)}$ then produces pairs (x, v) uniformly distributed over the subgraph of $f(x)$ and thus, marginally, a simulation from $f(x)$. For further details on the correctness of this procedure, see (CR, Ch. 2).

In our case the instrumental distribution is the uniform distribution on $\mathcal{X}$, namely it is $\hat{f}(x) \triangleq \frac{1}{|\mathcal{X}|}$. The constant M must satisfies the relation $g^*(x) = \frac{1}{k|[x]|} \leq \frac{M}{|\mathcal{X}|}$ for any $x \in \mathcal{X}$. This means that we take $M = \frac{|\mathcal{X}|}{k \min_x |[x]|}$.

5.5.3 Methodology

In order to turn the previous methods into a practical methodology, we have to take some issues into account.

1. For both MCMC and AR, the values $|[x]|$ cannot possibly be computed exactly. In practice, we approximate $|[x]|$ during a pre-computation phase, where inputs are sampled uniformly at random and used to compute a histogram of the most frequent outputs. In the actual sampling phase of either methods, the accept rule will make use of this histogram to ensure that such outputs are, in practice, not generated too often.

2. For MCMC, convergence to g^* only holds in the limit: if we fix a maximum number of steps T, the resulting distribution $\hat{g}$ can only approximate g^*. The determination of this "burn in" period T can be nontrivial and often proceeds by trials and errors.

The above approximations imply that, in general, what one can hope for is to find a "good" input distribution $\hat{g}$, that is, one that does significantly better than the uniform input distribution, in terms of output's J_t. This is *not* to say, however, that we are not able to give *formal* bounds on the program's capacity. Indeed, Theorem 5.4.1(1) applies to *any* output distribution, be it optimal or not. Additional details on the methodology are reported below.

Two aspects of practical concern of the sampling methodology are the following.

(a) For MCMC, the choice of the proposal distribution, $Q(x'|x)$.

(b) The 'waste' involved in rejecting samples.

Aspect (a) is related to the specific input domain $\mathcal{X}$ under consideration. In our experiments, we have considered $\mathcal{X} = \{0,1\}^n$, the set of n-bit vectors. With such a domain, we have found that the random function $Q(x'|x)$ that computes x' by flipping independently each individual bit of the current state x, with a fixed probability $0 < \beta < 1$, has given encouraging results; in particular, the value $\beta = 0.165$ has experimentally been found to give good results.

Concerning (b) in the case of AR, from step 3 of the algorithm it is easy to deduce that, for x chosen uniformly at random, the probability of acceptance is $\mu\frac{k}{|\mathcal{X}|}$, where $\mu \stackrel{\triangle}{=} \min_{x'} |[x']|$. Note that, in a perfectly balanced situation ($|[x]| = \frac{|\mathcal{X}|}{k}$ for each x), this probability is 1. In strongly unbalanced situations ($\mu k << |\mathcal{X}|$), however, we will observe a high rejection rate and inefficiency. We can mitigate this problem by formally replacing M with $M\theta$, for a parameter $\theta \in (0,1]$ that can be tuned by the analyst. This in practice means that the comparison at step 3 becomes $u < \frac{\mu}{\theta|[x]|}$. This modification will affect the sampled distribution, which will not be g^* anymore, but not the soundness of the method, that is, the formal guarantees provided on lower bounds.

In the case of MCMC, problem (b) mostly occurs when generating the m i.i.d. extractions, because all but the last state of each of m random walks should be discarded. We have explored the following alternative method:

we first run a single Markov chain for a suitably long burn in time T, thus reaching a state x_T and producing a histogram h_T of output frequencies. We then use x_T (as a starting state) and h_T to run again m independent copies of the Markov chain, but each for a much shorter time T'. The results thus obtained have been found experimentally to be equivalent to those obtained by sequentially sampling m times a single same Markov chain, after the burn in period. One will have often to tune the various parameters involved in the sampling algorithm (T, β, m, θ) in order to get satisfactory input distributions and guarantees.

5.6 Numerical experiments

We have tested the methodology introduced in the previous section on two classes of simple programs. The first class includes programs with artificially highly unbalanced classes; the second class is about cache side channels in sorting algorithms, a problem discussed in (DFK+13; AS01). The Java code used to run these experiments is available from the authors.

5.6.1 Unbalanced classes

We have tested the methodologies introduced in the previous section on a few instances of a parametric class of programs, $P_{n,l,r}(x)$, with $n \geq l \geq r \geq 1$ and $x \in \mathcal{X} = \{0,1\}^n$ and output in $\mathcal{Y} = \{0,1\}^l$, defined as follows.

$P_{n,l,r}(x)$:

```
z=mod(x,2^l);
if mod(z,2^r)==0 then y=z else y=mod(z,2^r);
return y;
```

The capacity of this program is easy to compute by inspection. In fact, $P_{n,l,r}(x)$ divides the input domain into $k = 2^{r-1} - 1 + 2^{l-r}$ distinct classes: of these, $2^{r-1} - 1$ are "fat" classes with 2^{n-r} elements each (outputs corresponding to the `else` branch); and 2^{l-r} are "small" classes, with 2^{n-l} elements each (outputs corresponding to the `then` branch). Once fixed

n, depending on the value of $l - r$, we will have more or less balanced situations. If $l - r$ is small, we will have a balanced situation, with about 2^l classes of similar size. As $l - r$ gets larger, we will have increasingly unbalanced situations, with relatively few (about 2^r) fat classes and a lot (2^{l-r}) of small classes.

We have put the estimator J_t at work on several instances of this program, using input distributions generated according to either Crude Monte Carlo or Metropolis. Table 3 displays the outcomes of these experiments for $P_{n,l,r}(x)$, for $m = 5 \times 10^5$ and $\delta = 0.001$. In each row, the most accurate obtained outcome is highlighted in boldface. Some more detailed tables can be found in Appendix A.5. For various values of n, we have tested J_t on both balanced (l close r) and unbalanced (r small) situations. We have considered two possible values for the confidence $1 - \delta$. For simplicity, we also preferred to just fix an "affordable" sample size of $m = 5 \times 10^5$ throughout all the experiments. Following Theorem 5.4.1, we have set $t = \sqrt{m \ln(1/\delta)/2}$. Finally, the parameters T, of both Metropolis and Accept-Reject, and β, of Metropolis, have been determined experimentally: $T = 50,000$ and $\beta = 0.165$.

n	l	r	k	CMC	MCMC	AR
24	22	22	4.9143×10^6	$\mathbf{4.0320 \times 10^6}$	2.8233×10^6	4.0101×10^6
	22	20	1.0486×10^6	$\mathbf{1.0317 \times 10^6}$	8.4025×10^5	1.0281×10^6
	22	2	2.0972×10^6	1.1853×10^5	4.0384×10^5	$\mathbf{1.0320 \times 10^6}$
	22	1	2.0972×10^6	2.8133×10^5	7.6395×10^5	$\mathbf{2.0450 \times 10^6}$
28	23	23	8.3886×10^6	$\mathbf{7.7328 \times 10^6}$	4.8261×10^6	7.6385×10^6
	23	20	1.0486×10^6	$\mathbf{1.0339 \times 10^6}$	8.3413×10^5	1.0316×10^6
	23	2	2.0972×10^6	1.2232×10^5	4.8578×10^5	$\mathbf{2.0330 \times 10^6}$
	23	1	4.1943×10^6	2.9472×10^5	9.3437×10^5	$\mathbf{3.9806 \times 10^6}$
32	26	26	6.7108×10^7	$\mathbf{3.8772 \times 10^7}$	1.5140×10^7	3.8651×10^7
	26	23	8.3886×10^6	$\mathbf{7.7432 \times 10^6}$	4.7617×10^6	7.7312×10^6
	26	2	1.6777×10^7	1.2593×10^5	5.9906×10^5	$\mathbf{1.3683 \times 10^7}$
	26	1	3.3554×10^7	3.0841×10^5	1.1975×10^6	$\mathbf{2.4993 \times 10^7}$

Table 3: Capacity lower bounds J_t with different sampling methods.

A few considerations on these results are in order. First, Accept-Reject performs better than Crude Monte Carlo in the unbalanced situations ($l - r$

big); in this scenarios, Metropolis too performs better that Crude MC, but not as better as AR. Nevertheless Metropolis appears more efficient in running simulations and it seems reasonable that it could take its revenge with programs having very irregular equivalence relations, although this hypothesis will be the subject of further experiments. Both AR and Crude MC perform very well in the balanced situations (l close to r). Second, save for the AR method, in the unbalanced cases the absolute error is significant; this can partly be imputed to the relatively small sample size m used, but further experimentation in this respect is needed. Third, the quality of the estimation does not depend at all on the size of the input domain (2^n), but rather on the size of the output domain (approximately, 2^l); this fact is extremely encouraging from the point of view of the scalability of the methodology. We also notice that in cases with a small k compared to m, not reported in the tables, using J_t is in a sense overkill, as the the trivial estimator D performs quite well (see Appendix A.5).

Running each experiment (row of a table) required less than two minutes time on a common notebook. The corresponding MATLAB code has been made available online (BP14).

5.6.2 Cache side-channels in sorting algorithms

Sorting algorithms are fundamental building blocks of many systems. The observable behaviour of a sorting algorithm is tightly connected with the (sensitive) data it manipulates. Therefore, designing secure sorting algorithms is challenging and, as discussed in (AS01), should provide indications on how other algorithms can be implemented securely. An additional reason for analysing sorting algorithms is that this is one of the cases where static analysis alone may fail to provide useful bounds (we discuss this at the end of this subsection).

To give a flavour of the leaks that may arise in this context, consider the following code implementing `InsertionSort`.

```
 1.  public static int InsertionSort(int[] v){
 2.      int n = v.length;
 3.      int i, j, index;
 4.      for(i = 1; i<n; i++){
 5.          index = v[i];
 6.          for(j = i; j>0 && v[j-1]>index; j--){
 7.              v[j] = v[j-1];
 8.          }
 9.          v[j]= index;
10.      } }
```

Suppose all elements of v are distinct, and the secret one wants to protect is the initial ordering of the elements in v, that is, the sequence of indices $(i_1, i_2, ...)$ s.t. $v[i_1] < v[i_2] < \cdots$. One wonders how much information a trace-based adversary, capable of detecting the hit/miss sequence relative to lookups of vector v, can learn about the secret. Assuming for simplicity a cache that is initially empty and large enough that v could fit entirely in it, it is clear that the beginning of each innermost `for` cycle (line 6) will be marked by a miss, corresponding to the lookup of $v[i]$ in line 5. The distance between the last miss and the next will reveal the adversary how many iterations are executed in 6-7 for the current value of i; hence the value of j (if any) s.t., for the current version of the vector, $v[j-1] > v[i]$. It appears the overall execution will give the adversary significant information about the secret, although it is far from easy to exactly determine how much, manually.

We have put our estimator J_t at work on both `InsertionSort` and `BubbleSort` (the code of the latter is reported in the Appendix A.4), for several values of the vector's size n and cache configurations. For simplicity, we have confined ourselves to the AR sampling method. We have considered increasing values of the sample size m, until observing J_t getting essentially stable: this indicates that the maximum leakage has been observed. We have analyzed the case of both access- and trace-based cache side-channel adversaries, introduced in Example 5.2.1. In the first case, we have obtained $J_t = 1$ in all configurations, that is a leakage of 0, which is consistent with what known from previous analyses (DFK$^+$13). The outcomes of the experiments for the trace-based adversary is displayed in Tables 4 and 5, where values are obtained with $\delta = 0.001$ and $m = 5 \times 10^3$ (for $n = 8, 16$) or $m = 10^5$ (for $n = 32$). Here, C and B

n		8		16		32	
$\log_2(n!)$		15.3		44.25		117.66	
	Preload	yes	no	yes	no	yes	no
C	B						
64	2	4.75	7.51	4.10	11.44	6.22	16.01
64	4	4.75	4.95	3.90	10.05	6.35	14.81
64	8	4.85	4.80	3.80	7.77	6.18	12.88
128	2	4.75	7.51	4.10	11.39	6.28	15.99
128	4	4.75	4.85	4.10	10.04	6.18	14.82
128	8	4.80	4.80	3.90	8.49	6.12	12.89
256	2	4.80	7.46	3.80	11.46	6.20	15.99
256	4	4.75	4.85	3.80	10.06	6.20	14.81
256	8	4.85	4.80	3.70	8.52	6.12	12.89

Table 4: Capacity lower bounds $\log J_t$ (in bits) for BubbleSort and several cache configurations and vector lengths.

are respectively the cache and block size, in words. Preload=no means the cache is initially empty. Preload=yes means that the entire vector v is cached before execution of the sorting algorithm. The adopted cache replacement policy is *least recently used* (LRU). Lower bounds on capacity are expressed in bits, that is, $\log J_t$ is reported; the size in bits of the secret, $\log(n!)$, is also given for reference. The running time of each experiment ranges from few seconds to several minutes on a common notebook.

As expected, the observed leakage consistently increases as n increases. For fixed n, ignoring small fluctuations, the observed values consistently decrease as B increases, while they appear to be independent from C. As expected, a preload phase drastically reduces the observed leakage. Note however that, even when the vector is completely cached, there is still some variability as to the number of comparison operations, which depends on the initial ordering of v. This explains the positive leakage observed with preload. Finally, we note that `BubbleSort` is consistently observed to leak less than `InsertionSort`.

We also note that this is a scenario where current static analysis methods are not precise enough to give useful results. In particular, the upper bounds obtained in (DFK+13, Fig.9) for the trace-based adversary are vac-

n		8		16		32	
$\log_2(n!)$		15.3		44.25		117.66	
	Preload	yes	no	yes	no	yes	no
C	B						
64	2	4.80	9.76	3.90	16.23	6.32	18.88
64	4	4.80	6.34	4.10	14.91	6.26	18.89
64	8	4.85	4.80	3.80	9.65	6.22	18.70
128	2	4.85	9.68	3.80	16.20	6.06	18.89
128	4	4.80	6.27	3.90	14.88	6.35	18.92
128	8	4.80	4.80	3.80	9.67	6.18	18.69
256	2	4.80	9.79	3.80	16.26	6.14	18.90
256	4	4.85	6.39	4.10	14.81	6.12	18.89
256	8	4.85	4.85	3.80	9.67	6.33	18.67

Table 5: Capacity lower bounds $\log J_t$ (in bits) for InsertionSort and several cache configurations and vector lengths.

uous, since larger than the size in bits of the secret[3]. A direct comparison of our results with those of (DFK+13) is not possible, because of the different experimental settings - in particular, they analyze compiled C programs, while we simulate the cache behaviour by instrumented Java code. Nevertheless, our results indicate that a black-box analysis of lower bounds is a useful complement to a static analysis of upper bounds, revealing potential confidentiality weaknesses in a program. In the future, we plan to test systematically the examples given in (DFK+13), in a comparable experimental setting.

5.7 Further and related work

Recent works by Terauchi and collaborators proved serious computational limits related to both the exact computation and nontrivial bounding of information leakage of programs even in the deterministic case (YT11). For this reason, a growing research interest towards methods for approximate calculation of information leakage (BDKR05; CCG10; CG11; CKN13;

[3]The authors there also analyse a third sorting algorithm, which gives results identical to `BubbleSort`.

CK13; KR13) developed.

Besides the already discussed Köpf and Rybalchenko's (KR13; KR10), our work is mostly related to some recent papers by Tom Chothia and collaborators. In (CCG10; CK13), methods are proposed to estimate, given an input-output dataset, confidence intervals for information leakage, and therefore test whether the apparent leakage indicates a statistically significant information leak in the system, or whether it is in fact consistent with zero leakage. The LEAKIEST and LEAKWATCH tools (CKN13; CKNP13; CKNP) are based on these results. Issues related to the existence of estimators in our sense are not tackled, though. Moreover, while handling also probabilistic systems, the resulting analysis methods and tools are essentially based on exhaustive input enumeration. In practice, they can be put into use only when the input space is quite small (Cho14).

Batu et al. (BDKR05) consider the problem of estimating Shannon entropy. Their negative results are similar in spirit to ours in Section 5.3. Technically, they mainly rely on birthday paradox arguments to prove lower bounds of (approximately) $\Omega(\sqrt{|\mathcal{Y}|})$ on the number of required samples, for a class of "high entropy" distributions; these arguments could be easily adapted to our case. On the positive side, they provide, among others, an estimator based on the observed number of collisions for the special case of distributions that are uniform on their own support, where Shannon and min entropy coincide. In our experience, collision-based estimators perform poorly in unbalanced situations.

Estimation of program capacity is related to the problem, considered in applied Statistics, of estimating the number of classes in a population, by some form or another of sampling. There exists a large body of literature on this subject, and a variety of estimators have been proposed, see e.g. (CL92) and references therein. To the best of our knowledge and understanding, though, the settings considered in these works are quite different from ours. Indeed, either a parametric approach is adopted, which requires the class probabilities to fit into certain families of distributions, which cannot be assumed for programs; or the focus is rather on obtaining practical estimations of the number of classes, which at best become exact when the

sample is infinite. Issues concerning formal guarantees achievable using finite samples, and the circumstances under which these guarantees can or cannot actually be obtained, are scarcely considered. Another crucial difference is that, in our setting, there is an input distribution the analyst can be assumed to exercise some control on, which of course is not the case for population classes.

The present chapter has mostly focused on theoretical issues. In the future, it would be interesting to investigate to what extent the proposed methodology scales to sizeable, real-world applications. Our results indicates that accurate estimators could be obtained if taking advantage of prior or subjective knowledge about the program: it would be interesting to investigate Bayesian versions of our methodology to this purpose.

Chapter 6

Conclusion

In this thesis we have analyzed the problem of quantifying information flow in terms of both confidentiality and privacy. QIF is mainly concerned with quantifying the degree of protection offered against an adversary trying to guess the whole secret. For this reason, QIF analysis of information leakage mainly follow an average approach. DP is rather concerned with protection of individual bits of the secret, possibly in the presence of background information, like knowledge of the remaining bits. In this case, an average notion of leakage may not be adequate, because there are systems that, even if appear secure on average, they reveal easily exposed to eavesdropping. Hence it is necessary to consider a worst-case notion. In both scenarios we develop a model intended for assessment of system security against passive eavesdropper.

In Chapter 3 we have characterized the asymptotic behaviour of error probability, information leakage and indistinguishability in a scenario of one-try attacks, in the case of repeated independent observations. We refine the proposed model, studying the security of the system not only quantitatively (how much is leaked), but also qualitatively (what properties are leaked). To this purpose, we extend information hiding systems with views. This model allows to analyze a variety of statistical attacks in a uniform fashion. This permits the assessment of systems security against passive eavesdroppers both at the global level and at the level of specific

partitions of the secrets. In particular, we give precise bounds for the probability of misclassification on the part of the attacker, characterizing both the limit value and the rate of convergence of the error probability as a function of the number independent observations. Particular interesting are statistical attacks against privacy in sparse datasets.

In Chapter 4 we propose a semantic notion of security, that expresses absence of any privacy breach above a given level of seriousness, irrespective of any background information We have analyzed security of randomization mechanisms (viewed as information theoretic channels) against privacy breaches with respect to various dimensions (worst vs. average case, single vs. repeated observations, utility). Whenever appropriate, we have characterized the resulting security measures in terms of simple row-distance properties of the underlying channel matrix. We have clarified the relation our worst-case measure with DP.

Unfortunately, whether we speak of privacy or confidentiality, recent studies show that exactly computing or bounding the information leakage turns out to be computationally prohibitive, even in the deterministic case. For this reason, we finally focus on the problem of formally bounding information leakage by statistical estimation. In Chapter 5 we have studied this problem in a set up where little is known about the program under examination and the way its input is generated. We have provided both negative results on the existence of accurate estimators and positive results, the latter in the terms of a methodology to obtain good input sampling distributions. This methodology has been demonstrated with a few simulations that have given encouraging results.

References

[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. In *Formal Aspects in Security and Trust*, pages 39–54, 2011. 87

[AACP11a] Mário S. Alvim, Miguel E. Andrés, Konstantinos Chatzikokolakis, and Catuscia Palamidessi. On the relation between differential privacy and quantitative information flow. In *ICALP (2)*, pages 60–76, 2011. 71, 82, 84, 87

[AACP11b] Mário S. Alvim, Miguel E. Andrés, Konstantinos Chatzikokolakis, and Catuscia Palamidessi. Quantitative information flow and applications to differential privacy. In *FOSAD*, pages 211–230, 2011. 5, 87

[AS01] Johan Agat and David Sands. On confidentiality and algorithms. In *IEEE Symposium on Security and Privacy*, pages 64–77, 2001. 92, 106, 108

[BC13] Michele Boreale and Alessandro Celestini. Asymptotic risk analysis for trust and reputation systems. In *SOFSEM*, pages 169–181, 2013. 82

[BCO04] Eric Brier, Christophe Clavier, and Francis Olivier. Correlation power analysis with a leakage model. In *CHES*, pages 16–29, 2004. 41

[BCP08] Christelle Braun, Konstantinos Chatzikokolakis, and Catuscia Palamidessi. Compositional methods for information-hiding. In *FoSSaCS*, pages 443–457, 2008. 36, 57

[BCP09] Christelle Braun, Konstantinos Chatzikokolakis, and Catuscia Palamidessi. Quantitative notions of leakage for one-try attacks. *Electr. Notes Theor. Comput. Sci. 2009*, vol. 249:pp. 75–91, 2009. 2, 31, 33, 37, 57, 58, 87

[BDKR05] Tugkan Batu, Sanjoy Dasgupta, Ravi Kumar, and Ronitt Rubinfeld. The complexity of approximating the entropy. *SIAM J. Comput.*, 35(1):132–150, 2005. 111, 112

[BK08] Michael Backes and Boris Köpf. Formally bounding the side-channel leakage in unknown-message attacks. In *ESORICS*, pages 517–532, 2008. 33, 58

[BK11] Gilles Barthe and Boris Köpf. Information-theoretic bounds for differentially private mechanisms. In *CSF*, pages 191–204, 2011. 5, 87

[BMS10] Béatrice Bérard, John Mullins, and Mathieu Sassolas. Quantifying opacity. In *QEST*, pages 263–272, 2010. 58

[Bor09] Michele Boreale. Quantifying information leakage in process calculi. *Inf. Comput.*, 207(6):699–725, 2009. 2, 57

[BPa] Michele Boreale and Michela Paolini. On formally bounding information leakageby statistical estimation. submitted. x, 9, 89, 90

[BPb] Michele Boreale and Michela Paolini. Worst- and average-case privacy breaches in randomization mechanisms (full version). submitted. x, 8, 12

[BP12] Michele Boreale and Michela Paolini. Worst- and average-case privacy breaches in randomization mechanisms. In *IFIP TCS*, pages 72–86, 2012. x, 8, 12

[BP14] Michele Boreale and Michela Paolini. Matlab code available at `http://rap.dsi.unifi.it/~boreale/metropolis.m`. 2014. 108

[BPP11a] Michele Boreale, Francesca Pampaloni, and Michela Paolini. Asymptotic information leakage under one-try attacks. In *FOSSACS 2011*, pages pp. 396–410, 2011. x, 2, 8, 19, 29, 30, 31, 33, 44, 45, 46, 87

[BPP11b] Michele Boreale, Francesca Pampaloni, and Michela Paolini. Quantitative information flow, with a view. In *ESORICS 2011*, pages pp. 588–606, 2011. x, 2, 8, 19, 30, 45, 87, 88

[BPPar] Michele Boreale, Francesca Pampaloni, and Michela Paolini. Asymptotic information leakage under one-try attacks (full version). In *Math. Struct. in Computer Science*, to appear. x, 8, 30, 31, 33, 45, 47

[BV08] Thomas Baignères and Serge Vaudenay. The complexity of distinguishing distributions (invited talk). In *ICITS*, pages 210–222, 2008. 34, 58

[Cac97] Christian Cachin. Entropy measures and unconditional security in cryptography. *PhD thesis, Swiss Federal Institute of Technology*, 1997. 19

[CCG10] Konstantinos Chatzikokolakis, Tom Chothia, and Apratim Guha. Statistical measurement of information leakage. In *TACAS 2010*, pages pp. 390–404, 2010. 6, 89, 90, 111, 112

[CG11] Tom Chothia and Apratim Guha. A statistical test for information leaks using continuous mutual information. In *CSF 2011*, pages pp. 177–190, 2011. 6, 89, 90, 111

[Cha81] David Chaum. Untraceable electronic mail, return addresses, and digital pseudonyms. *Commun. ACM*, 24(2):84–88, 1981. 40

[Cha88] David Chaum. The dining cryptographers problem: Unconditional sender and recipient untraceability. *J. Cryptology*, 1(1):65–75, 1988. 3

[CHM01] David Clark, Sebastian Hunt, and Pasquale Malacaria. Quantitative analysis of the leakage of confidential data. *Electr. Notes Theor. Comput. Sci.*, 59(3):238–251, 2001. 2, 21, 57

[Cho14] Tom Chothia. Personal communication to Michele Boreale. 2014. 112

[CK13] Tom Chothia and Yusuke Kawamoto. Statistical estimation of min-entropy leakage. manuscript available at `http://www.cs.bham.ac.uk/research/projects/infotools/leakiest/`. 2013. 6, 89, 90, 112

[CKN13] Tom Chothia, Yusuke Kawamoto, and Chris Novakovic. A tool for estimating information leakage. In *CAV 2013*, pages pp. 690–695, 2013. 6, 89, 90, 111, 112

[CKNP] Tom Chothia, Yusuke Kawamoto, Chris Novakovic, and David Parker. Leakwatch: Estimating information leakage from java programs. 90, 112

[CKNP13] Tom Chothia, Yusuke Kawamoto, Chris Novakovic, and David Parker. Probabilistic point-to-point information leakage. In *CSF*, pages 193–205, 2013. 112

[CL92] Anne Chao and Shen-Ming Lee. Estimating the number of classes via sample coverage. In *Journal of the American statistical Association 1992*, volume vol. 87, pages pp. 210–217, 1992. 112

[CPP08a] Konstantinos Chatzikokolakis, Catuscia Palamidessi, and Prakash Panangaden. Anonymity protocols as noisy channels. *Inf. Comput. 2008*, vol. 206(2-4):pp. 378–401, 2008. 2, 11, 19, 38, 57, 58

[CPP08b] Konstantinos Chatzikokolakis, Catuscia Palamidessi, and Prakash Panangaden. On the bayes risk in information-hiding protocols. *Journal of Computer Security 2008*, vol. 16(5):pp. 531–571, 2008. 2, 38, 39, 57, 58

[CR] George Casella and Christian Robert. Monte carlo statistical methods. *Springer Verlag, Second Edition 2004*. 102, 103, 104

[Csi98] Imre Csiszár. The method of types. *IEEE Transactions on Information Theory*, 44(6):2505–2523, 1998. 23, 25

[CT06] Thomas M. Cover and Joy A. Thomas. Elements of information theory, 2/e. 2006. 14, 16, 17, 19, 24, 25, 26, 28, 29, 34, 35, 38, 58

[Dan03] George Danezis. Statistical disclosure attacks. In *SEC*, pages 421–426, 2003. 40

[DFK+13] Goran Doychev, Dominik Feld, Boris Köpf, Laurent Mauborgne, and Jan Reineke. Cacheaudit: A tool for the static analysis of cache side channels. In *USENIX Security*, pages 431–446, 2013. 89, 92, 93, 106, 109, 110, 111

[DMNS06] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. In *TCC*, pages 265–284, 2006. 5, 69, 70, 74, 76, 87

[DN10] Cynthia Dwork and Moni Naor. On the difficulties of disclosure prevention in statistical databases or the case for differential privacy. In *JPC*, volume 2(1), 2010. 76

[DP09] Devdatt P. Dubhashi and Alessandro Panconesi. *Concentration of Measure for the Analysis of Randomized Algorithms*. Cambridge University Press 2009, 2009. 98

[DS05] Yevgeniy Dodis and Adam Smith. Entropic security and the encryption of high entropy messages. In *TCC*, pages 556–577, 2005. 88

[Dwo06] Cynthia Dwork. Differential privacy. In *ICALP (2)*, pages 1–12, 2006. 5, 61, 69, 74, 76, 87

[EGS03] Alexandre V. Evfimievski, Johannes Gehrke, and Ramakrishnan Srikant. Limiting privacy breaches in privacy preserving data mining. In *PODS*, pages 211–222, 2003. 5, 12, 61, 64, 80, 87

[FS10] Arik Friedman and Assaf Schuster. Data mining with differential privacy. In *KDD*, pages 493–502, 2010. 61

[GKS08] Srivatsava Ranjit Ganta, Shiva Prasad Kasiviswanathan, and Adam Smith. Composition attacks and auxiliary information in data privacy. In *KDD*, pages 265–273, 2008. 61, 74

[GM82] Joseph A. Goguen and José Meseguer. Security policies and security models. In *IEEE Symposium on Security and Privacy 1982*, pages 11–20, 1982. 1

[GRS09] Arpita Ghosh, Tim Roughgarden, and Mukund Sundararajan. Universally utility-maximizing privacy mechanisms. In *STOC*, pages 351–360, 2009. 62, 82, 84, 87

[KB07] Boris Köpf and David A. Basin. An information-theoretic model for adaptive side-channel attacks. In *ACM Conference on Computer and Communications Security*, pages 286–296, 2007. 58

[KD09] Boris Köpf and Markus Dürmuth. A provably secure and efficient countermeasure against timing attacks. In *CSF*, pages 324–335, 2009. 58

[KJJ99] Paul C. Kocher, Joshua Jaffe, and Benjamin Jun. Differential power analysis. In *CRYPTO*, pages 388–397, 1999. 2, 41

[KM11] Daniel Kifer and Ashwin Machanavajjhala. No free lunch in data privacy. In *SIGMOD Conference*, pages 193–204, 2011. 75, 87

[KM12] Daniel Kifer and Ashwin Machanavajjhala. A rigorous and customizable framework for privacy. In *PODS*, pages 77–88, 2012. 75, 76, 87

[Koc96] Paul C. Kocher. Timing attacks on implementations of diffie-hellman, rsa, dss, and other systems. In *CRYPTO*, pages 104–113, 1996. 2, 3, 41

[KR10] Boris Köpf and Andrey Rybalchenko. Approximation and randomization for quantitative information-flow analysis. In *CSF*, pages 3–14, 2010. 89, 112

[KR13] Boris Köpf and Andrey Rybalchenko. Automation of quantitative information-flow analysis. In *SFM*, pages 1–28, 2013. 6, 89, 97, 102, 112, 122

[KS10] Boris Köpf and Geoffrey Smith. Vulnerability bounds and leakage resilience of blinded cryptography under timing attacks. In *CSF 2010*, pages pp. 44–56, 2010. 2, 37, 58, 87, 88

[KSWH00] John Kelsey, Bruce Schneier, David Wagner, and Chris Hall. Side channel cryptanalysis of product ciphers. *Journal of Computer Security*, 8(2/3):141–158, 2000. 42, 43

[LJ97] Charles C. Leang and Don H. Johnson. On the asymptotics of m-hypothesis bayesian detection. *IEEE Transactions on Information Theory*, 43(1):280–282, 1997. 29, 34, 59

[Mal10] Pasquale Malacaria. Risk assessment of security threats for looping constructs. *Journal of Computer Security*, 18(2):191–228, 2010. 2

[Mas94] James Lee Massey. Guessing and entropy. In *IEEE Symposium on Information Theory (ISIT) 1994*, volume vol. 204, 1994. 14, 20

[McS09] Frank McSherry. Privacy integrated queries: an extensible platform for privacy-preserving data analysis. In *SIGMOD Conference*, pages 19–30, 2009. 61

[MT07] Frank McSherry and Kunal Talwar. Mechanism design via differential privacy. In *FOCS*, pages 94–103, 2007. 87, 88

[NS08a] Arvind Narayanan and Vitaly Shmatikov. Robust de-anonymization of large sparse datasets. In *IEEE Symposium on Security and Privacy*, pages 111–125, 2008. 3, 53, 54, 55, 57

[NS08b] Arvind Narayanan and Vitaly Shmatikov. Robust de-anonymization of large sparse datasets. In *IEEE Symposium on Security and Privacy*, pages 111–125, 2008. 88

[Ré61] Alfréd Rényi. On measures of entropy and information. In *Proceedings of 4th Berkeley Symposium on Mathematics, Statistics and Probability*, volume 1, page 547561, 1961. 14, 19

[RR98] Michael K. Reiter and Aviel D. Rubin. Crowds: Anonymity for web transactions. *ACM Trans. Inf. Syst. Secur.*, 1(1):66–92, 1998. 3, 38, 39, 40

[Sha] Claude E. Shannon. A mathematical theory of communication. *Bell System Technical Journal*. 14

[SM03] Andrei Sabelfeld and Andrew C. Myers. Language-based information-flow security. *IEEE Journal on Selected Areas in Communications*, 21(1):5–19, 2003. 19

[Smi09] Geoffrey Smith. On the foundations of quantitative information flow. In *FOSSACS*, pages 288–302, 2009. 2, 11, 19, 20, 21, 31, 33, 37, 57, 122

[SMY09] François-Xavier Standaert, Tal Malkin, and Moti Yung. A unified framework for the analysis of side-channel key recovery attacks. In *EUROCRYPT*, pages 443–461, 2009. 41, 58

[YT11] Hirotoshi Yasuoka and Tachio Terauchi. On bounding problems of quantitative information flow. *Journal of Computer Security*, vol. 19(6):pp. 1029–1082, 2011. 6, 89, 111

Appendix A

Additional proofs and tables

A.1 Proofs of Chapter 2

A.1.1 Proof of Theorem 2.3.1

Theorem A.1.1 (Theorem 2.3.1) *Let $i \in \{\mathrm{Sh}, \infty\}$ and $k = |\mathrm{Im}(P)|$. Then $L_i(P;g) \leq C_i(P) = \log k$. In particular, if $g(x) = \frac{1}{k \times |[x]|}$ for each $x \in \mathcal{X}$, then $L_{\mathrm{i}}(P;g) = C_{\mathrm{i}}(P) = \log k$, for $i \in \{\mathrm{Sh}, \infty\}$.*

Proof The result for $i = \infty$ (min entropy) is in see e.g. (KR13; Smi09). For the case $i = \mathrm{Sh}$ (Shannon entropy) first note that, denoting by $I(\cdot;\cdot)$ mutual information and exploiting its symmetry, we have, for each g: $L_{\mathrm{Sh}}(P;g) = I(X;Y) = H_{\mathrm{Sh}}(Y) - H_{\mathrm{Sh}}(Y|X) = H_{\mathrm{Sh}}(Y) \leq \log k$, where in the last but one step we use $H(Y|X) = 0$ (output is entirely determined by input for deterministic programs), and the last step follows from a well known inequality for Shannon entropy. Now, taking the input distribution g specified in the first item of the theorem, we see that all inverse images of P have the same probability $1/k$, hence Y is uniformly distributed on k outputs. This implies $H_{\mathrm{Sh}}(Y) = \log k$. □

A.2 Proofs of Chapter 4

A.2.1 Proof of Lemma A.2.1

Lemma A.2.1 *In the binary Bayesian hypothesis testing problem with two distributions p_1 and p_2, having prior probabilities α and β respectively ($\alpha + \beta = 1$), the success probability P_{succ} defined in* (2.1) *satisfies*

$$P_{succ} = \frac{||\alpha p_1 - \beta p_2||_1 + 1}{2}.$$

Proof Taking (2.4) into account, and considering that we are in the binary case $|\mathcal{X}| = 2$, we have

$$P_{succ} \quad = \quad \sum_y \max\{p_1(y)\alpha, p_2(y)\beta)\}. \tag{A.1}$$

The wanted result follows by noting that:

1. $\sum_y |\alpha p_1(y) - \beta p_2(y)|$ = $\sum_y \max\{\alpha p_1(y), \beta p_2(y)\}$ - $\min\{\alpha p_1(y), \beta p_2(y)\}$;

2. $\sum_y \max\{\alpha p_1(y), \beta p_2(y)\}$ + $\min\{\alpha p_1(y), \beta p_2(y)\}$=$\sum_y \alpha p_1(y)+\beta p_2(y)$=$\alpha+\beta = 1$.

After some algebra, we obtain that

$$2\sum_y \max\{p_1(y)\alpha, p_2(y)\beta\} = \sum_y |\alpha p_1(y) - \beta p_2(y)| + 1$$

that is $\sum_y \max\{p_1(y)\alpha, p_2(y)\beta\} = \frac{||\alpha p_1 - \beta p_2||_1 + 1}{2}$. □

A.2.2 Proof of Theorem 4.5.2

Theorem A.2.1 (Theorem 4.5.2) *Assume $\mathcal{R}$ is non-singular and strictly positive. Then, with the notation introduced Section 4.5*

$$\Pr(Breach_n^\epsilon | X \in Q) \doteq 1 - 2^{-nC} \quad \textit{and} \quad \Pr(\overline{Breach}_n^\epsilon | X \in Q^c) \doteq 1 - 2^{-nC}$$

where $C = \min_{x \in Q, x' \in Q^c} C(p_x, p_{x'})$, with the understanding that x and x' in the min *are taken of positive probability. As a consequence, the probability that Y^n causes a Q-breach reaches* 1 *at rate at least C.*

Proof Let $n \geq 1$ and let us abbreviate $Breach_n^\epsilon$ as $Breach_n$ in the rest of the proof. We will show that both $p(Breach_n|Q) \doteq 1 - 2^{-nC}$ and that $p(\overline{Breach}_n|Q^c) \doteq 1 - 2^{-nC}$, where $C = \min_{x \in Q, x' \in Q^c} C(p_x, p_{x'})$ is the rate: from this the result will follow, indeed $\alpha_n \stackrel{\triangle}{=} \Pr(Breach_n \cup \overline{Breach}_n) \geq p(Breach_n|Q)p(Q) + p(\overline{Breach}_n|Q^c)$, which implies $\underline{\text{rate}}(\{\alpha_n\}) \geq C$. We focus on $p(Breach_n|Q)$, as the case of $p(Breach_n|Q^c)$ follows by symmetry. Consider the set of sequences in $\mathcal{Y}^n$

$$W_n = \{y^n \in \mathcal{Y}^n \, : \, p(Q|y^n) > 2^\epsilon p(Q)\} .$$

Clearly, $\Pr(Breach_n|Q) = \Pr(Y^n \in W_n | X \in Q) \stackrel{\triangle}{=} p(W_n|Q)$. Observe that $p(W_n|Q) = \sum_{x \in Q} p_x(W_n) \frac{p(x)}{p(Q)}$. We show that, for each $x \in Q$ of positive probability, $p_x(W_n) \to 1$ at rate $\min_{x' \in Q^c} C(p_x, p_{x'})$, from which the result will follow: indeed, the rate of a summation – in our case $\sum_{x \in Q} p_x(W_n) \frac{p(x)}{p(Q)}$ – is the least of the summands' rates. Fix any $x \in Q$ of positive probability. It is easy to show that, for certain constants $\kappa_{x'}$ that depend on $p(x')$ ($x' \in Q^c$), $p(Q)$ and ϵ, the set

$$B_n(x) \stackrel{\triangle}{=} \{y^n : D(t_{y^n}||p_x) \leq D(t_{y^n}||p_{x'}) + \frac{\kappa_{x'}}{n} \; \forall x' \in Q^c\}$$

is included in W_n, for n large enough:

$$B_n(x) \subseteq W_n$$

(informally, $B_n(x)$ contains the set of sequences y^n whose type is closer to p_x than to any $p_{x'}$ with $x' \in Q^c$). Also note that, defining $\Pi_n(x, x') \stackrel{\triangle}{=} \{y^n : D(t_{y^n}||p_x) > D(t_{y^n}||p_x') + \frac{\kappa_{x'}}{n}$ for each $x' \in Q^c\}$, the complement of $B_n(x)$ can be written as

$$B_n^c(x) = \cup_{x' \in Q^c} \Pi_n(x, x') .$$

We examine the convergence of $1 - p_x(W_n) = p_x(W_n^c)$. We have

$$p_x(W_n^c) \leq p_x(B_n^c(x)) \leq \sum_{x' \in Q^c} p_x(\Pi_n(x, x')) \tag{A.2}$$

Now, fix $x' \in Q^c$. From (2.19), we know that there is a *unique* distribution $q^* \in \mathcal{I}_{p_x p_{x'}}$ s.t. $C(p_x, p_{x'}) = D(q^*||p_x) = D(q^*||p_{x'})$. We choose now a

sequence of n-types $\{\xi_n\}$ s.t. $\xi_n = \operatorname{argmin}_{\zeta \in \Pi_n(x,x') \cap \mathcal{T}_n} D(\zeta||p_{x'})$. For any n large enough, (2.11) and (2.12) imply

$$p_x(\Pi_n(x,x')) \leq (n+1)^{|\mathcal{X}|} 2^{-nD(\xi_n||p_x)} . \tag{A.3}$$

Moreover, as $n \to +\infty$ we have $\xi_n \to q^*$; this follows because: (i) q^* is the unique minimizer of $D(\cdot||p_x)$ and $D(\cdot||p_{x'})$ in $\mathcal{I}_{p_x p_{x'}}$; (ii) the set of types is dense among the set of distributions; and (iii) $D(\cdot||p_x)$ and $D(\cdot||p_{x'})$ are continuous. Now, again by continuity, we get that $D(\xi_n||p_x) \to D(q^*||p_x) > 0$, which implies, from (A.3), that $p_x(\Pi_n(x,x')) \to 0$ for each x', hence $p_x(W_n^c) \to 1$. Concerning the rate, take the $-\frac{1}{n}\log$ of both sides of (A.3). For any n large enough, we get

$$-\frac{1}{n}\log p_x(\Pi_n(x,x')) \geq -|\mathcal{X}|\frac{\log(n+1)}{n} + D(\xi_n||p_x) .$$

This, on the limit, yields

$$\underline{\text{rate}}(p_x(\Pi_n(x,x'))) \geq D(q^*||p_x) = C(p_x, p_{x'}) .$$

Since this holds for each $x' \in Q^c$ and the rate of a sum is the minimum rate of the summands, from (A.2) we get

$$\underline{\text{rate}}(p_x(W_n)) \geq \min_{x' \in Q^c} C(p_x, p_{x'}) .$$

Concerning the reverse inequality, note that for each n large enough, from (2.12) we get

$$p_x(\Pi_n(x,x')) \;\geq\; p_x(\mathcal{T}_n(\xi_n)) \;\geq\; (n+1)^{-|\mathcal{X}|} 2^{-nD(\xi_n||p_x)} . \tag{A.4}$$

Again, taking the $-\frac{1}{n}\log$ of both sides and then taking the limit we obtain

$$\overline{\text{rate}}(p_x(W_n)) \leq \min_{x' \in Q^c} C(p_x, p_{x'})$$

from which the thesis follows. □

Remark A.2.1 *Consider Remark 4.5.1. We have seen that the rate at which the probability of a breach approaches 1 does not depend on the level ϵ and it only depends on the support of the prior distribution. Moreover, in principle, it should be possible to give a more precise lower-bound for* $\Pr(Breach_n|Q)$ *than the one implied by* (A.2) *and* (A.3)*, by explicitly taking $p(x)$ and ϵ into account. We shall not investigate this issue here, though.*

A.2.3 Proof of Lemma A.2.2

Lemma A.2.2 *The Chernoff Information $C(p,q)$ is a convex function of both p and q.*

Proof We only look at the first argument, p, as the function is symmetric. Suppose that $p = \mu p_1 + (1-\mu)p_2$, with $\mu \in [0,1]$. Note that for any $\lambda \in [0,1]$, the function $c \mapsto c^\lambda$ is concave in $[0,1]$, therefore:

$$\begin{aligned}\sum_x p^\lambda(x)q^{1-\lambda}(x) &\geq \sum_x \big(\mu p_1^\lambda(x) + (1-\mu)p_2^\lambda(x)\big)q^{1-\lambda}(x)\\ &= \sum_x \mu p_1^\lambda(x)q^{1-\lambda}(x) + (1-\mu)p_2^\lambda(x)q^{1-\lambda}(x)\\ &= \mu\sum_x p_1^\lambda(x)q^{1-\lambda}(x) + (1-\mu)\sum_x p_2^\lambda(x)q^{1-\lambda}(x)\,.\end{aligned}$$

Now exploiting the monotonicity of $\log$ and the above inequality, and then concavity of $\log$, we have:

$$\begin{aligned}\log\sum_x p^\lambda(x)q^{1-\lambda}(x) &\geq \log(\mu\sum_x p_1^\lambda(x)q^{1-\lambda}(x) + (1-\mu)\sum_x p_2^\lambda(x)q^{1-\lambda}(x))\\ &\geq \mu[\log(\sum_x p_1^\lambda(x)q^{1-\lambda}(x))] +\\ &\quad +(1-\mu)[\log(\sum_x (1-\mu)p_2^\lambda(x)q^{1-\lambda}(x))].\end{aligned}$$

Now, we take the $\min_\lambda$ of both sides of the above inequality and then exploit the fact that, for any two functions of λ, say $f_1(\lambda)$ and $f_2(\lambda)$, one has

$$\min_\lambda(f_1(\lambda) + f_2(\lambda)) \geq \min_\lambda f_1(\lambda) + \min_\lambda f_2(\lambda).$$

From this, it is immediate to conclude. □

A.3 Proof of Chapter 5

A.3.1 Proof of Lemma 5.4.1

Lemma A.3.1 (Lemma 5.4.1) *Let $k = |\mathrm{supp}(p)|$. Then: (a) $f(k) = E_u[D] \geq E[D]$; (b) if additionally p is η-close to uniform, then $E[D] \geq \frac{f(k)}{\eta}$.*

Proof Concerning (a), note that:

$$E_u[D] = E_u[\sum_y I_{\{y\in S\}}] = \sum_y E_u[I_{\{y\in S\}}] = \sum_y (1-(1-\frac{1}{k})^m) \quad \text{(A.5)}$$
$$= k(1-(1-\frac{1}{k})^m) = f(k)$$

where in (A.5) $(1-\frac{1}{k})^m$ is the probability that y does not occur in the sample S under a uniform on k outputs distribution. Furthermore $E_u[D] \geq E[D]$, indeed

$$E[D] = \sum_y 1-(1-p(y))^m = k - \sum_y (1-p(y))^m$$
$$= k - k\sum \frac{1}{k}(1-p(y))^m \leq k - k(1-\frac{1}{k})^m = E_u[D] \quad \text{(A.6)}$$

where in (A.6) we have applied the Jensen's inequality to the function $x \mapsto (1-x)^m$ which is convex in $[0,1]$.

Concerning (b), we have

$$E[D] = \sum_y 1-(1-p(y))^m = k - \sum_y (1-p(y))^m$$
$$\geq k(1-(1-\frac{1}{k\eta})^m) = \frac{1}{\eta}f(k\eta) \geq \frac{1}{\eta}f(k) \quad \text{(A.7)}$$

where in (A.7) we use the fact that p is η-close to uniform and that the function $x \mapsto (1-x)^m$ is decreasing, while $f(x)$ is increasing, in $[0,1]$. □

A.4 Additional code for Section 5.6

```
public static int BubbleSort(int[] v){
  int n = v.length; int swap;
  for(int i=0; i<n-1; i++){
    for(int j=0; j<n-i-1; j++){
       if(v[j]>v[j+1]){
          swap = v[j];
          v[j] = v[j+1];
          v[j+1] = swap;
              }
            }
          }
          }
```

A.5 Additional tables for Chapter 5

We give below some more detailed tables, containing the outcomes of some experiments with the three considered sampling methods in Chapter 5 (Crude Monte Carlo, Metropolis Monte Carlo, Accept-Reject). These tables also contain information on D, the number of distinct outputs actually observed during simulation.

Table 6: Lower bounds on program capacity, with CMC and $m = 5 \times 10^5$, $\delta = 0.001$.

n	l	r	k	Crude Monte Carlo	
				D	J_t
28	23	23	8.3886×10^6	4.8592×10^5	7.7328×10^6
	23	20	1.0486×10^6	3.9776×10^5	1.0339×10^6
	23	2	2.0972×10^6	1.2158×10^5	1.2232×10^5
	23	1	4.1943×10^6	2.4201×10^5	2.9472×10^5
32	26	26	6.71088×10^7	4.9810×10^5	3.8772×10^7
	26	23	8.3886×10^6	4.8551×10^5	7.7432×10^6
	26	2	1.6777×10^7	1.2487×10^5	1.2593×10^5
	26	1	3.3554×10^7	2.4912×10^5	3.0916×10^2

Table 7: Lower bounds on program capacity, with MCMC and $m = 5 \times 10^5$, $\delta = 0.001$.

n	l	r	k	Metropolis	
				D	J_t
28	23	23	8.3886×10^6	4.7629×10^5	4.8261×10^6
	23	20	1.0486×10^6	3.7740×10^5	8.3413×10^5
	23	2	2.0972×10^6	3.1354×10^5	4.8578×10^5
	23	1	4.1943×10^6	3.8851×10^5	9.3437×10^5
32	26	26	6.71088×10^7	4.9315×10^5	1.5140×10^7
	26	23	8.3886×10^6	4.7617×10^5	4.8040×10^6
	26	2	1.6777×10^7	3.4036×10^5	5.9906×10^5
	26	1	3.3554×10^7	4.1006×10^5	1.1975×10^6

Table 8: Lower bounds on program capacity, with AR and $m = 5 \times 10^5$, $\delta = 0.001$.

n	l	r	k	Accept-Reject	
				D	J_t
28	23	23	8.3886×10^6	4.8530×10^5	7.6385×10^6
	23	20	1.0486×10^6	3.9756×10^5	1.0316×10^6
	23	2	2.0972×10^6	4.4457×10^5	2.0330×10^6
	23	1	4.1943×10^6	4.7118×10^5	3.9806×10^6
32	26	26	6.71088×10^7	4.9809×10^5	3.8651×10^7
	26	23	8.3886×10^6	4.8548×10^5	7.7312×10^6
	26	2	1.6777×10^7	4.9228×10^5	1.3683×10^7
	26	1	3.3554×10^7	4.9634×10^5	2.4993×10^7