IMT Institute for Advanced Studies, Lucca

Lucca, Italy

Quantitative Models of Information Flow: Tuning the Power of the Adversary

PhD Program in Computer Science and Engineering

XXVI Cycle

By

Francesca Pampaloni

2014

The dissertation of Francesca Pampaloni 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 Francesca Pampaloni has been reviewed by:

Prof. Catuscia Palamidessi, INRIA and LIX, École Polytechnique

Prof. David Clark, University College London

IMT Institute for Advanced Studies, Lucca

2014

Contents

List of Figures

Acknowledgements

Foremost, I would like to express my sincere gratitude to my advisor Prof. Michele Boreale for the continuous support of my Ph.D study and research. His guidance helped me in all the time of research and writing of this thesis.

My sincere thanks also goes to Prof. Matteo Maffei and to his group, for offering me the internship opportunity and leading me working on a very interesting project.

I thank all my friends: for the stimulating discussions, for the support and for all the fun we have had in the last three years. In particular, I am grateful to Michela, that shared with me a long time, among joys, fears and deadlines, and Valeria, that, although studying a completely different topic, read part of my thesis and gave me very useful suggestions.

Last but not the least, I would like to thank my family: my parents and my brother, for supporting me spiritually throughout my life.

The results presented in this thesis are based on joint works with Michele Boreale, University of Florence, and Michela Paolini, IMT Institute for Advanced Studies, Lucca. In particular, Section 3.1 and Chapter 4 are based on (BPP11a; BPPar), joint works with Michele Boreale and Michela Paolini. Section 3.2 is based on (BPP11b), a joint work with Michele Boreale and Michela Paolini. Chapter 5 is based on (BP12a), a joint work with Michele Boreale. Finally, Chapter 6 is based on the results presented in (BPar), a joint work with Michele Boreale.

Vita

September 16, 1986	Born, Bagno a Ripoli (FI), Italy
October 2005 - October 2008	Bachelor Degree in Mathematics University of Florence, Italy, Final mark: 110/110.
October 2008 - October 2010	Master Degree in Mathematics for Applications University of Florence, Italy, Final mark: 110/110 cum laude.
March 2011 - Now	PhD Student, IMT Lucca, Italy.
January - April 2013	Visiting Student, Saarland University, Saarbrücken, Germany.

Publications

Journal papers:

1. M. Boreale, F. Pampaloni, M. Paolini. Asymptotic information leakage under one-try attacks (full version). *MSCS*, to appear.

Conference papers:

2. M. Boreale, F. Pampaloni, M. Paolini. Asymptotic information leakage under one-try attacks. *Proc. of FoSSaCS 2011*, *LNCS* 6604: pp. 396-410, 2011.
3. M. Boreale, F. Pampaloni, M. Paolini. Quantitative information flow, with a view. *Proc. of ESORICS 2011*, *LNCS* 6879: pp. 588-606, 2011.
4. M. Boreale, F. Pampaloni. Quantitative Multirun Security under Active Adversaries. *Proc. of QEST 2012*, *IEEE Computer Society*: pp. 158-167, 2012.
5. M. Boreale, F. Pampaloni. Quantitative information flow under generic leakage functions and adaptive adversaries. *Proc. of FORTE 2014*, *LNCS*, to appear.

Submitted papers:

6. F. Eigner, A. Kate, M. Maffei, F. Pampaloni. PrivaDA: A Generic Framework for Privacy-preserving Data Aggregation.

Extended abstracts:

7. F. Eigner, A. Kate, M. Maffei, F. Pampaloni. PrivaDA: A Generic Framework for Privacy-preserving Data Aggregation. *GRSRD*, to appear.

Presentations

Conference talks:

1. Quantitative Multirun Security under Active Adversaries, 9th International Conference on Quantitative Evaluation of SysTems (QEST), Imperial College, London, United Kingdom, September 2012.
2. Quantitative information flow, with a view, 16th European Symposium on Research in Computer Security (ESORICS), Katholieke Universiteit, Leuven, Belgium, September 2011.

Invited talks:

3. Quantitative Models on Confidentiality and Integrity: Tuning the Power of the Adversary, Saarland University, Saarbrücken, Germany, December 2012.

Abstract

Despite the variety of tools and techniques deployed in order to protect sensitive data, ranging from security types in programming languages to anonymity protocols, data sanitisation, cryptographic algorithms, ..., real-world systems tend to disclose part of the information they are meant to protect. This happens either by design - when the output of the system is public (e.g. a password checker) - or for reasons depending on their actual deployment and implementation (e.g. side-channel attacks against cryptographic devices).

Our work aims to study methods for analysing from a *quantitative* point of view the behaviour of *information flow* in computing systems, that is, the leakage of sensible information via public outputs. In general, we are interested in studying systems with a probabilistic behaviour, in situations where the attacker is allowed to run these systems several times, while the secret is kept fixed.

We analyse quantitative information flow in various scenarios characterised by an increasing power of the adversary. We start from the case of a single, passive attacker, attempting to break the system solely based upon observed data. Then we examine more complex settings, where we are faced with adversaries that collect sequential observations, up to considering active adversaries, capable of directly interacting with the system. In all cases, we consider one-try re-execution attacks, where the adversary can make a single guess after observing a certain number of independent executions of the system. In particular, we define suitable security metrics and study their asymptotic behaviour as the number of observations increases, as well as their rate of convergence. We also consider a number of applications of our analysis techniques.

Chapter 1

Introduction

In this introductory chapter, we outline the motivations at the basis of our research. We then survey the relevant literature in Quantitative Information Flow and related fields. We finally discuss the contributions provided by our study.

1.1 Quantitative Information Flow

Protecting the confidentiality of information manipulated by computer programs is a long-standing problem. The standard way to protect confidential data is (discretionary) *access control*, in which some privileges are required in order to access files or objects containing confidential information (SM03). Only some legitimate users will be able to access to the *high*, secret variables, while the *low*, public ones will be accessible to all. The question at stake is whether, due to a program execution, some information contained in high variables could *flow* into low outputs.

The ideal case is achieving *non-interference*, a security property stating that high information does not interfere with low data output from the system. In language-based security, *Information Flow* is a well-established area with roots in this concept of non-interference (GM82; SM03). However, avoiding all possible information flows from secret inputs to public outputs is often not achievable. Sometimes this loss is inevitable, since it is the program itself that is forced to release a certain amount of information, in order to achieve its goals. This is, for instance, the case of a password checker, which is forced to reveal that the password is not

correct, if a wrong password is entered:

```
h=password;
i=input;
if i=! h then
print("Password Incorrect");
```

In other cases, information leakage is related to the physical implementation of a system, e.g. to execution times or termination behaviour, or even to the emission of physical signals from a device (electromagnetic radiations, power consumption, ...). This flow of information is a crucial concern for security, since such signals are fairly easy to detect for an eavesdropper and contain an amount of information about the secret that can be valuable for an adversary.

Recent years have seen the emergence of *Quantitative* Information Flow (QIF), aiming to measure information flowing from confidential to public data, by comparing the difference between the prior and the posterior attacker's uncertainty, after collecting some observations related to the secret:

$$\text{information leaked} \;=\; \text{prior uncertainty} \;-\; \text{posterior uncertainty}.$$

QIF relies on tools from Information Theory and is related to several fields, such as information flow in programming languages, side-channels analysis in cryptographic hardware and software, anonymity protocols and so on. The basic idea is tolerating small leaks of information, below an accepted predefined threshold, and, at the same time, ensuring that safety guarantees and security requirements are met. This may result in more flexible analysis than the rigid safe/unsafe classification provided by traditional Information Flow techniques.

This thesis aims to study methods for analysing from a quantitative point of view information flow in various attack scenarios, characterised by an increasing power of the adversary. To have an overview of the different scenarios one can be faced with, let us see some examples. Consider a program, or protocol, P, which receives a high input H and produces a low output L. An adversary, observing L, might be able to learn information about H. The point is quantifying the amount of information that can be inferred by the adversary. The high input H and the low output L can be represented as two random variables, taking values respectively in $\mathcal{H}$ and $\mathcal{L}$. Assume that the probability distribution of H is known to the attacker. If P is a *deterministic* program, then the output L

can be seen as a function of H: $L = P(H)$, with $P : \mathcal{H} \to \mathcal{L}$. For example, suppose $\mathcal{H} = \{0,1\}^{32}$ and consider the program

```
P1:     l = h mod 4
```

that, taken h as input, a 32-bit unsigned integer, outputs its two least significant bits. The set of observations is $\mathcal{L} = \{00, 01, 10, 11\}$. This way, the attacker, through the observation of the output l, can learn the two least significant bits of the high value h.

For deterministic programs, once fixed the input, the output is determined. Hence, there is no reason to consider the case of *re-execution*, where we allow the attacker to re-run a program several times, maintaining fixed the secret input. However, in Security, programs are often *probabilistic*, either because of the presence of noise or by design. In this case the analysis of the multi-run case is fundamental. In real-world situations, indeed, re-execution may happen either when forced by the attacker (as in the case of an adversary querying several times a smartcard), or by design (as in the case of routing paths established, repeatedly, between a sender and a receiver in anonymity protocols like Crowds (RR98)). A probabilistic program can be modeled by a conditional probability matrix, containing all probabilities of the form $\Pr(L = l|H = h)$. Consider, for instance, the following program:

```
P2:     l = h +rnd(-1,1) mod 4
```

where `rnd(1,-1)` denotes a random number taking values in the set $\{-1, 0, 1\}$. Intuitively, if the program can be ran just once, the information leaked is not much. However, if we consider the case of re-execution, the amount of information leaked increases. In the program above, for instance, after one execution, an attacker can only eliminate, from the set of possible values, those that present the last two bits differing by 2 from the output. Re-running the program several times, instead, with high confidence he can determine the last two bits of the input.

So far, we have considered attack scenarios where the adversary is a passive eavesdropper, attempting to break the system solely based upon observed data. The situation changes if we consider an *active* adversary, for example able of controlling part of the input (*untrusted* input). In this scenario, the re-execution case concerns also deterministic programs, since at each run the attacker can alter the untrusted input and observe the effect on the output. Consider the following program ($\&$ denotes the bitwise AND operator):

```
P3:     l = h & u
```

that takes as inputs two 32-bit integers h and u, where h is the high input, while u is the untrusted one. Let $\mathcal{U} = \{2^i | i = 0, \ldots, 31\}$ be the set of untrusted inputs. Then, if $u = 2^j$ for a certain $j \in \{0, \ldots, 31\}$, this program outputs the jth bit of h. After one run, an attacker can only learn this value, discovering just a bit of h. Trying all possible values $u \in \mathcal{U}$, he can *completely* determine h. In this case, we try to prevent the attacker from learning sensitive information by re-running the program with modified untrusted inputs and observing the public output.

1.2 Literature Review

In this section, we survey the literature in QIF and in related fields.

1.2.1 Quantitative Information Flow

The last few years have seen a flourishing of research on quantitative models of information flow. 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 (albeit with a different technical meaning than that we have studied in this thesis). In both these works, the considered systems do not exhibit probabilistic behaviour.

The work by Chatzikokolakis, Palamidessi and their collaborators (CPP08a; CPP08b; BCP09) is closely related to our approach. (CPP08a) examines information leakage mainly from the point of view of Shannon entropy and capacity, but 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 a single observation, but also offers a lower-bound in the case of repeated observations.

Smith shows in (Smi09) that the concept of mutual information is not very suitable for modeling the information leakage in scenarios where the adversary attempts to guess the value of the secret in one single try. As an alternative, he proposes to use another metric, based on Renyi min-entropy.

Compositional methods based on process algebras are discussed in (BCP09). In this case, the average ML error probability is characterised

in terms of MAP error probability under a uniform distribution of inputs. (BCP09) introduces the notion of additive leakage, comparing it to the min-entropy based leakage considered by Smith (Smi09), but again in the case of a single observation.

A model of "unknown-message" attacks is considered by Backes and Köpf in (BK08). This model is basically equivalent to the information hiding systems considered in (CPP08a; CPP08b; BCP09). Backes and Köpf too consider a scenario of repeated independent observations, but from the point of view of Shannon entropy, rather than of error probability. Similarly to our studies, they rely on the information-theoretic method of types in order 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. This study is extended to the case of one-try attacks by Köpf and Smith in (KS10), that also offer a general lower bound on the attacker's error probability after a certain number of observations.

A drawback of most of the approaches so far considered is that they focus exclusively on the quantitative aspect of the analysis (*how much* is leaked), while ignoring the qualitative aspect (*what* is leaked) at all. Bérard, Mullins and Sassolas in (BMS10) study the notion of probabilistic *opacity*. Nevertheless, (BMS10) is based on Shannon entropy and considers observations consisting on a single run of the system, rather than repeated observations, hence it does not tackle statistical attacks.

Focusing on more powerful attacks, Köpf and Basin in (KB07) consider a scenario of adaptive chosen-message attacks. They offer an algorithm to compute conditional Shannon entropy in this setting. However, they do not study its asymptotic behaviour, which seems, indeed, very difficult to characterise. Their analysis is conducted essentially on the case of uniform prior distributions and is limited to deterministic mechanisms.

Information flow in interactive systems is studied by Alvim et al. in (AAP12). They describe these systems as probabilistic automata where secrets and observables alternate in the execution. Information theoretically, they characterise them as channels with feedback, giving a Shannon-entropy based definition of leakage.

Birgisson and Sabelfeld in (BS11) tackle the concept of *multi-run security* of a system. Through the use of *policies*, they set a lower bound on the attacker's uncertainty, by partitioning the set of states into classes. They then verify the robustness of programs, including in the case of several

runs, varying all possible secret inputs, according to their policy compliance. They consider imperative programs, ignoring the probabilistic case.

A quantitative model of integrity is discussed by Clarkson and Schneider in (CS10). They study what happens when the adversary can control a part of the input. In particular, they focus on the concepts of *contamination* and *suppression*, respectively related to the amount of untrusted information that contaminates the output, flowing from the untrusted input to the output, and to the amount of trusted information that is totally suppressed. Even here, they study the problem mainly for non probabilistic scenarios.

1.2.2 Side-channel analysis

Side-channel attacks against cryptographic algorithms aim at breaking cryptographic systems by exploiting information that is revealed by the physical execution of the algorithm. As proved by several works, by repeatedly measuring either the time needed by the algorithm to execute a certain computation (Koc96), or the power consumption (JO05; KJJ99), or electromagnetic radiation emitted by the device (GMO01; AARR02), it is possible to get information about the secret key, thus narrowing the search space, or even discover the key itself.

Timing (Koc96) and power analysis (KJJ99) are two flavours of side-channel correlation attacks against cryptographic devices (BCO04; SMY09). Another example of such attacks is given by Hamming weight attacks, where the adversary can determine the Hamming weight of the secret input. An application of this attack to DES is discussed in (KSWH00). These attacks presuppose, explicitly or implicitly, that the adversary knows the inputs processed by the target device [1].

The effectiveness of side-channel attacks represents a serious threat to the security of devices like smart-cards, that are subject to various kinds of measurements. This threat is not addressed by traditional notions of cryptographic security. Consider, for example, attacks based on time measurements, the so-called *Timing attacks*, where the attacker tries to reconstruct the secret by sampling the duration of several independent executions of the system. According to Kocher (Koc96), by this kind of attack an eavesdropper is able to determine the secret exponent used in

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

the modular exponentiation algorithm, consequently identifying Diffie-Hellman exponents, or the factorization of RSA keys, and can break several other encryption systems depending on such algorithm. The reason is that the time needed to perform various computations is not fixed, but depends on the input.

Models to quantify the resistance of a system against this kind of attacks are currently being studied (BK08; KD09; KB07; SMY09). Backes and Köpf in (BK08) analyse the case of unknown message attacks, where the attacker does not know the input that is encrypted/decrypted by the system. As already mentioned, they propose a model to quantify the information leakage, depending on the number of observations collected by the attacker. Köpf and Dürmuth in (KD09) consider timing attacks and propose a countermeasure to apply to them. As a case study, they analyse an implementation of an algorithm for 1024 bit RSA decryption. Köpf and Basin in (KB07) consider a scenario of adaptive chosen-message attacks.

In the context of side-channel cryptanalysis, Standaert et al. propose a framework to analyse side-channel correlation attacks (SMY09). Both a Shannon entropy based metric and a security metric are considered. Since correlation attacks are inherently known-message, their model presupposes the explicit or implicit knowledge of the plaintext on the part of the attacker.

1.2.3 Anonymity Protocols

Anonymous Routing is a collection of techniques that are meant to allow users to communicate, over public networks, without revealing their identity. Their main goal is obfuscating relations between the sender and detectable observations, by routing the message in a random fashion through the nodes of the network. A protocol well known of this kind is *Crowds*, firstly proposed by Reiter and Rubin in (RR98) and then studied in (CPP08a; CPP08b). Crowds 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 the functioning of the protocol that will be described later on, as proved by (RR98), Crowds offers very good guarantees of anonymity, if it is executed for only one time. The strength of the protocol decays in presence of re-executions, when the protocol is executed several times, either forced by the attacker himself (e.g. if corrupted nodes suppress messages) or by some external factor, and the sender is kept

fixed through the various executions. As showed by Reiter and Rubin (RR98), in the case of re-execution it is recommended to use *static paths*, that is, when sending a message several times to a same receiver, it is recommended to follow the same path instead of randomly choosing it. The increasing dynamism of the paths, indeed, tends to decrease the protections of anonymity offered by the system against the corrupted set of collaborating users.

Concerning the level of anonymity offered by Crowds, both (RR98) and (CPP08a) analyse it, showing that, with respect to a corrupted user the so-called *Strong Anonymity* is no longer guaranteed, that is, it is no longer true that observations give no information about the secret, i.e. the identity of the sender. However, if the number of corrupted nodes is not too big, the protocol can still satisfy the *Probable Innocence*, meaning that the probability that the detected user coincides with the real sender is not greater than the sum of the individual probabilities corresponding to each other user.

In order to guarantee anonymity to both sender and receiver and at the same time hide the content of the message, Goldschlag, Reed and Syverson in (GRS96) propose *Onion Routing*. The idea is to protect the transmitted information, including sender and receiver's IP address, through a series of cryptographic layers, that can be removed only by certain users and in a precise order.

1.3 Structure of the thesis and contributions

This thesis aims to analyse, mainly from a quantitative point of view, the information leaked during a system implementation, addressing several attack scenarios, through which the adversary's power is refined. We start from the case of a single *passive* attacker, that can only passively collect observations related to system executions and attempt statistical attacks, to arrive to *active* adversaries, able to directly interact with the system. In particular, we study what happens when the system presents a *probabilistic* behaviour, in the case of *one-try* attacks and system re-execution, where the adversary can make a single[2] guess after observing *several* independent executions of the system, throughout which the secret is kept fixed.

[2]As proved by Smith in (Smi09), this is not a limitation and we can easily adjust the result of a one-try case to another one where the adversary has more possibilities to guess.

We outline the structure of the rest of the thesis and the contributions we provide as follows.

- In Chapter 2 we recall some basic notions of Information Theory, required to introduce the security metrics we will use in the following chapters.
- In Chapter 3 we analyse two attack scenarios, where we are faced with a passive attacker. In the first scenario, the attacker directly targets the secret, while in the second one he targets some predicates of the secrets. We define a security metric, we study its asymptotic behaviour as the number of collected observations (system re-executions) increases, and provide simple and tight bounds, showing that the convergence is exponential. The second scenario allows us to analyse the previous scenario from a qualitative point of view, aiming to discover not only *how much* information is leaked, but also *what* is leaked. This chapter is based on (BPP11a; BPPar; BPP11b).
- In Chapter 4 we analyse a more sophisticated eavesdropping scenario, where the attacker is still passive, but this time is able to collect sequential observations, one for each computation step. We extend the model described in the previous chapter, introducing Hidden Markov Models, and propose an algorithm to characterise the asymptotic behaviour of the security metric. This chapter is based on (BPP11a; BPPar).
- Turning to scenarios with more powerful adversaries, in Chapter 5, we propose a model to analyse threats posed to confidentiality and integrity of probabilistic systems by a class of *active* adversaries. Akin to (BS11), we assume an attacker that can control part of the input, which we deem as *untrusted*. We focus on the non-adaptive case, studying the asymptotic behaviour of the attacker's uncertainty. Given the computational difficulty of directly computing bounds and rate, we propose a sub-optimal, but reasonably efficient, attack strategy, that allows one to compute a lower bound of the rate. Moreover, we analyse the problem of declassification policies (used to state what information can be released) and quantify how serious is a violation of the policy, if any. Concerning the issue of integrity, then, we assess the inherent risk of any decision strategy to determine whether a system is under attack or not. This chapter is based on (BP12a).

- The case of *adaptive* active adversaries, capable to adaptively query the system according to the passed observations, is addressed in Chapter 6. We present a general information-theoretic model to study the limits of adaptive adversaries. A central theme of our study is the comparison of adaptive and non-adaptive attack strategies. Relative to a *generic* uncertainty function, we: (1) characterise the maximum information leakage achievable by adaptive adversaries and show that it can be also achieved with non-adaptive strategies; (2) in terms of strategies length, contrast the efficiency of adaptive against non-adaptive adversaries; (3) point out that maximum information leakage over a given finite horizon can be expressed in terms of a Bellman equation, which can be used to compute optimal finite strategies recursively. All in all, our results indicate that, for reasonably powerful adversaries, there is no dramatic difference between adaptive and non-adaptive strategies, even from the point of view of analysis. These results are based on (BPar).

- Chapter 7, finally, contains some concluding remarks and discussion of future works.

Chapter 2

Preliminaries on Information Theory

In this chapter we recall some preliminary notions of Information Theory, such as Shannon and Rényi entropy, mutual information and Kullback-Leibler distance, necessary to understand the content of this thesis.

In the following, let X be a discrete random variable taking value in a finite set $\mathcal{X}$ and let $p(\cdot)$ be its probability density function (denoted as $X \sim p(\cdot)$), that is:

$$\text{for all } x \in \mathcal{X} \qquad p(x) \triangleq \Pr(X = x).$$

From now, on all the logarithms are taken with base 2.

2.1 Shannon Entropy

Definition 2.1.1 (Shannon Entropy) *Let X be defined as above. Then,* Shannon Entropy *of X, denoted as $H(X)$, is defined by:*

$$H(X) \triangleq -\sum_{x \in \mathcal{X}} p(x) \log p(x).$$

By convention, $0 \cdot \log 0 = 0$. [1]

[1] Such convention is consistent with the fact that $\lim_{x \to 0} x \log x = 0$.

Measured in bits, $H(X)$ quantifies the a priori uncertainty on the value taken by X. That is, it expresses the number of bits that are missing to completely determine this value.

Remark 2.1.2 *By definition, Shannon entropy only depends on the distribution $p(\cdot)$ of X. So, we can rewrite it in the following way:*

$$H(X) = H(p) = -\sum_{i \in \{1,\dots,|\mathcal{X}|\}} p_i \log p_i,$$

where $p_i \triangleq p(x_i)$, with $x_i \in \mathcal{X}$.

Example 1 *Consider a coin flipping. Suppose that we know the probability of obtaining head (H) or tail (T). The result of the coin flipped can be represented as a random variable X taking values in $\{H, T\}$. If the coin is fair, then our uncertainty is maximum, since we can obtain either outcome with the same probability, $\frac{1}{2}$. In this case, therefore, we have the maximum entropy, given by $H(X) = 2(\frac{1}{2} \cdot \log 2) = 1$. This means that we need 1 bit of information to identify the result. Suppose now that the coin is unfair. For example, assume that it gives value H with a probability $q > \frac{1}{2}$. In this case, it is more likely that the result is head and so the entropy will be smaller: $H(X) = -q \log q - (1-q)\log(1-q) < 1$. This means that, on average, we need less than one bit of information to determine X. The extreme case happens when the coin is totally biased, e.g. it yields H with probability 1. In this situation there is not uncertainty on the value of X, because there is only one possible result and so the entropy is 0.*

Remark 2.1.3

- *The uncertainty increases as the number of possible values that X can take on increases and as its distribution approaches the uniform one over $\mathcal{X}$;*
- *clearly we can define the entropy also for a vector of variables, $(X_1, \dots, X_N)$, defined respectively over $\mathcal{X}_1, \dots, \mathcal{X}_N$. In this case, we consider the joint distribution $p(\cdot)$, such that $p(x_1, \dots, x_n) \triangleq \Pr(X_1 = x_1, \dots, X_n = x_n)$:*

$$H(X_1, \dots, X_N) \triangleq -\sum_{(x_1,\dots,x_n) \in \mathcal{X}_1 \times \dots \times \mathcal{X}_n} p(x_1, \dots, x_N) \log p(x_1, \dots, x_N).$$

Proposition 2.1.4 *For each random variable X taking values in $\mathcal{X}$, we have:*

$$0 \leq H(X) \leq \log |\mathcal{X}|$$

where on the left the equality holds if and only if X is a constant, while on the right if and only if X is uniformly distributed.

Example 2 *Let X be the random variable corresponding to the result obtained throwing a die. Suppose we cannot see the outcome. Let us compute our uncertainty, that is, the number of bits we need to exactly determine the result. If the die is fair, there are 6 equiprobable outcomes, so the uncertainty is maximum:*

$$H(X) = 6\big(\frac{1}{6} \cdot \log 6\big) = \log 6 \approx 2.58 \textit{ bits}.$$

If the die is heavily loaded, in such a way to yield always the same outcome, for example 5, in this case the uncertainty is 0, because

$$H(X) = 1 \cdot \log 1 = 0 \textit{ bits}.$$

Finally, consider an intermediate case, where the outcome 5 has probability $\frac{1}{2}$, while all the others have probability $\frac{1}{10}$. Here, the uncertainty is lower than in the fair case ($\log 6 = \log |\mathcal{X}|$), but greater than in the previous one. Indeed we have:

$$H(X) = \frac{1}{2} \cdot \log 2 + 5\big(\frac{1}{10} \cdot \log 10\big) \approx 2.16 \textit{ bits} < \log 6.$$

As mentioned before, entropy quantifies the concept of *a priori* uncertainty. Let us now consider the *a posteriori* uncertainty, that is, the uncertainty left after the observation of a certain event. Let Y be the random variable that represents this event and let $\mathcal{Y}$ be its set of values. In the following, we will denote by $p(x|y)$ the conditional probability that the event $X = x$ occurs, given the event $Y = y$, that is:

$$p(x|y) \triangleq \Pr(X = x|Y = y).$$

Definition 2.1.5 (Conditional Shannon entropy) *Given two random variables X and Y defined on the same probability space, the* conditional Shannon entropy *of X, given Y, denoted by $H(X|Y)$, is given by:*

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

where:

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

$H(X|Y)$ expresses the average uncertainty that remains on the value of X, once Y is known.

Proposition 2.1.6 *Let X and Y be two random variables. Then:*

- $0 \leq H(X|Y) \leq H(X)$*, with equality on the right if and only if X and Y are independent;*
- $H(X,Y) = H(Y) + H(X|Y)$ (chain rule)*;*
- $H(Y,X) = H(X,Y)$*.*

The first item states that the uncertainty about X, on average, decreases after the observation of Y. Moreover, the uncertainty stays the same if and only if the two variables are independent each other. The second item, called *chain rule*, allows us to decompose the uncertainty about a pair of variables (X,Y) into two parts: the a priori uncertainty about Y and the uncertainty that remains about X, given the value of Y. The chain rule is very useful, mainly because it often allows one to determine the entropy relative to a pair of variables (X,Y) without having to directly compute their joint distribution. The third item, finally, allows one to exchange the roles of the two variables in the chain rule.

Remark 2.1.7 *In general, the relation $H(X|Y=y) \leq H(X)$ does not hold. For example, if X represents a loaded die that with probability 0.99 returns 1, otherwise a value different from 1, then $H(X) \approx 0$. Assume we throw the die, without being allow to see the outcome, and assume to be informed that the number obtained is different from 1. Then, $H(X|X \neq 1) = \log 5 > H(X)$. However, since the event that increases the uncertainty is very unlikely, when we compute the average it has a very little weight.*

Example 3 *Let us come back to Example 2, where X is the variable representing the outcome of a fair coin toss. Let Y be another random variable, taking values in $\{0,1\}$, representing the parity of the outcome, such that $Y=0$ if the outcome is even, otherwise $Y=1$. Suppose we get to know the value of Y. Then, it is easy to see that the average uncertainty that remains about X, after observing the value of Y, is less than the a priori one. Intuitively, indeed, when*

we get to know, for example, that the outcome is even, that is $Y = 0$, the possible values of X are reduced to 3:

$$H(X|Y = 0) = 3(\frac{1}{3} \cdot \log 3) = \log 3.$$

The same result is obtained when $Y = 1$. Let us compute now the conditional entropy:

$$\begin{aligned} H(X|Y) &= \frac{1}{2}\big(H(X|Y = 0) + H(X|Y = 1)\big) \\ &= \frac{1}{2}(2 \cdot \log 3) \\ &= \log 3 \approx 1.58 \textit{ bit} < H(X) = \log 6. \end{aligned}$$

We can now define the *mutual information*, linking entropy and conditional entropy.

Definition 2.1.8 (Mutual Information) *Given two variables X and Y, the* mutual information *of X and Y, denoted by $I(X;Y)$, is:*

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

$I(X;Y)$ expresses the reduction of uncertainty that we have about X, once Y is known.

Example 4 *Let us come back to Example 3 and compute the reduction of uncertainty we have about X, the outcome of the die, after observing the value of Y, representing its parity.*

$$I(X;Y) = H(X) - H(X|Y) = \log 6 - \log 3 \approx 1 \textit{ bit}.$$

This means that the observation of the event Y has reduced uncertainty by 1 bit.

Lemma 2.1.9 *Mutual information is symmetric, that is:*

$$I(X;Y) = I(Y;X).$$

This shows that $I(X;Y)$ is also a measure of the information that is *shared* by the two variables. In other words, the reduction of uncertainty that we have about any of them, once we know the other one.

The worst case, that is the the maximum mutual information we can have, is called *capacity* and defined as follows.

Definition 2.1.10 (Capacity) *Given the variables X and Y, let $\mathcal{P}$ be the set of possible distributions $p(\cdot)$ on $\mathcal{X}$. Then the* capacity *is given by:*

$$C \triangleq \max_{p \in \mathcal{P}} I(X, Y).$$

Capacity measures the maximum rate at which the information can be transmitted.

Let us consider now the case when we have more than two variables. The following theorem is an important result, that allows one to express the joint entropy as the sum of the corresponding conditional entropies. It also gives an upper bound of it, as the sum of the marginal entropies.

Theorem 2.1.11 *Let $X_1, \ldots, X_n$ be random variables, taking values respectively in $\mathcal{X}_1, \ldots, \mathcal{X}_n$. Then:*

$$H(X_1, \ldots, X_n) = \sum_{i=1}^{n} H(X_i | X_{i-1}, \ldots, X_1), \tag{2.1}$$

$$H(X_1, \ldots, X_n) \leq \sum_{i=1}^{n} H(X_i), \tag{2.2}$$

with equality if and only if the variables X_i's are independent.

Shannon entropy is not the only measure to express uncertainty. Moreover, it measures the average number of *binary* questions we should ask to completely determine the value of the considered variable. In the following sections, we will introduce two other metrics, related to different notions of uncertainty.

2.2 Guessing Entropy

A metric that is directly connected to the difficulty of correctly guessing the value of a random variable, through an exhaustive search ("brute force" attacks) is given by the *guessing entropy*, discussed by Massey in (Mas94), albeit not under this name.

Definition 2.2.1 (Guessing Entropy) *Let X be a random variable taking values in $\mathcal{X}$ and with distribution $p(\cdot)$. Without loss of generality, assume that the elements of $\mathcal{X}$ are ordered by decreasing probability, such that for each*

$x_i, x_j \in \mathcal{X}$, *we have* $p(x_i) \geq p(x_j)$ *if* $i \leq j$. *Then, the* guessing entropy *of* X, *denoted by* $G(X)$, *is defined by:*

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

$G(X)$ measures the average number of questions of the form "Is $X = x$?" we have to ask to completely determine the value of X, following an optimal strategy (Mas94), that is, starting with the most likely value and proceeding in this way, up to the one with the least probability. Differently from Shannon entropy, it is not measured in bits.

An important result that links the two kinds of entropy we have just seen is the following, due to Massey (Mas94).

Theorem 2.2.2 *Let* X *be a random variable. If* $H(X) \geq 2$, *then*

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

If we think of X as a secret information to be protected, then $G(X)$ says us how many optimal questions, like "Is $X = x$?", the attacker needs on average to ask in order to break the system, starting from the most likely values. The above theorem shows us that such number increases exponentially with $H(X)$, the a priori Shannon entropy of the secret.

Example 5 *Reconsider the case of the loaded die of Example* 3. *Recall that, with probability* $\frac{1}{2}$, *the outcome is* 5, *while, with probability* $\frac{1}{10}$, *any of the other values. Following an optimal strategy, we will start asking if the outcome coincides with* 5, *then we will try the other values. The guessing entropy is then:*

$$G(X) = 1 \cdot \frac{1}{2} + (2 + 3 + 4 + 5 + 6) \cdot \frac{1}{10} = \frac{1}{2} + 20 \cdot \frac{1}{10} = 2.5 \text{ questions}.$$

Let us check Theorem 2.2.2. *Since* $H(X) = 2.16 > 2$, *we obtain*

$$2^{(H(X)-2)} + 1 = 2^{0.16} + 1 \approx 2.12 \leq 2.5 = G(X).$$

Definition 2.2.3 (Conditional Guessing Entropy) *Let* X *and* Y *be two random variables, taking values respectively in* $\mathcal{X}$ *and* $\mathcal{Y}$. *Then, the* conditional guessing entropy *of* X, *given* Y, *denoted by* $G(X|Y)$, *is:*

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

where

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

$G(X|Y)$ represents the average number of optimal questions we need to ask in order to correctly determine the value of X, once known the value of Y. Theorem 2.2.2 can be extended to the conditional entropies via the following proposition.

Proposition 2.2.4 *Given two random variables X and Y, assuming values respectively in $\mathcal{X}$ and $\mathcal{Y}$, with $Y = F(X)$ for some function $F : \mathcal{X} \to \mathcal{Y}$, then, if $H(X|Y) \geq 2$, we have:*

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

2.3 Min-entropy

Shannon and guessing entropies provide strong bounds about the average effort needed by a potential attacker to completely determine the secret information. But as Smith points out in (Smi09), they do not take into account the fact that, in some cases, such effort can be arbitrarily high, even if the probability itself of guessing the secret with only one question (one-try attacks) is quite high. This fact is demonstrated in the following example.

Example 6 ((Smi09)) *Let h be a positive integer of $8k$ bits, for some integer $k \geq 2$, that is $0 \leq h < 2^{8k}$, chosen uniformly at random. Therefore $H(h) = 8k$. Let us consider the following program:*

```
if h mod 8 = 0 then
   l:= h;
else
   l:= 1;
```

It is possible to prove that the average resistance of such program, that is, the residual uncertainty, given the value of the output l, is:

$$H(h|l) = 8k - \left(\frac{7}{8}\log\frac{8}{7} + 2^{8k-3} \cdot 2^{-8k}\log 2^{8k}\right) \approx 7k - 0.169. \qquad (2.3)$$

Let us consider now another program

```
l:= h mod (k+1);
```

Such program leaks $k+1$ bits compared to the total $8k$ bits of h. Therefore the residual uncertainty is:

$$H(h|l) = 8k - (k+1) = 7k - 1. \tag{2.4}$$

Comparing (2.3) *and* (2.4)*, the two programs seem to oppose comparable levels of resistance against brute force attacks. Taking a closer look, however, we note that, while in the second program the probability of guessing the value of h with only one try is $\frac{1}{2^{7k-1}}$, in the first one, instead, with probability $\frac{1}{8}$ the secret h will be completely leaked: such probability is unacceptably high. The problem is due to the fact that Shannon entropy only gives the average number of binary questions required to break the system. Apparently, this is not tightly related to the probability of guessing the secret with only one question.*

This motivates yet another form of uncertainty, this time related to the attacker's *error probability* in the case of one-try attacks.

Definition 2.3.1 (Error probability) *Let X and Y be two random variables, taking values respectively in $\mathcal{X}$ and $\mathcal{Y}$. The* a priori error probability *of X is:*

$$P_e(X) \triangleq 1 - \max_{x \in \mathcal{X}} p(x),$$

while the a posteriori error probability*, after observing the event Y, is:*

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

where:

$$P_e(X|Y = y) \triangleq 1 - \max_{x \in \mathcal{X}} p(x|y).$$

Concretely, let X represent the secret information to be protected and Y be an observation related to X, that can be collected by an attacker. Consider then the complement of P_e, that is the *success probability* for the attacker, given by

$$P_{succ} = 1 - P_e.$$

The success probability measures the maximum probability that the adversary correctly guesses the value of X in one try.

It is sometimes convenient to use a logarithmic measure.

Definition 2.3.2 (Min-entropy) *Given a random variable X taking values in $\mathcal{X}$,* min-entropy *of X, denoted by $H_\infty(X)$, is given by:*

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

Given another variable Y, taking values in $\mathcal{Y}$, the conditional min-entropy of X given Y is

$$H_\infty(X|Y) \triangleq -\log\left(P_{succ}(X|Y)\right) = -\log \sum_{y\in\mathcal{Y}} p(y) \max_{x\in\mathcal{X}} p(x|y).$$

$H_\infty(X)$ measures, in bits, the difficulty of a potential attacker to correctly guess the value of X in one try. The reason why it is denoted with the letter H, like Shannon entropy, with subscript ∞, is due to the fact that it is a particular case of the *Rényi entropy* (Rén61), defined by:

$$H_\alpha(X) \triangleq \frac{1}{1-\alpha} \log\Big(\sum_{x\in\mathcal{X}} \big(p(x)\big)^\alpha\Big).$$

Min-entropy corresponds to the limit case that we have when $\alpha \to +\infty$, while Shannon entropy corresponds to the other limit case, when $\alpha \to 1$. By definition, min-entropy satisfies the equality

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

Min-entropy related measures provide a strong security guarantee, showing us that the attacker's probability of guessing the secret, given a certain observation, decreases exponentially with the conditional min-entropy $H_\infty(X|Y)$.

Also in the case of min-entropy we can compute the reduction of uncertainty that we have after observing some events.

Definition 2.3.3 (Min-entropy leakage) *Given two random variables X and Y, we define* min-entropy leakage *as the difference between the a priori uncertainty about X and the a posteriori one, once Y is known, in min-entropy terms. Denoting such value with $I_\infty(X;Y)$, we have:*

$$I_\infty(X;Y) \triangleq H_\infty(X) - H_\infty(X|Y) = \log \frac{P_{succ}(X|Y)}{P_{succ}(X)}.$$

As Smith shows in (Smi09), the fact that we are confining ourselves to a single try may seem unreasonable, because often the attacker has more

than one possibility. However, if we consider vulnerability against attacks where the adversary can make a certain number n of attempts and denote with P^n_{succ} the correspondent success probability, the following inequalities hold, for all $n \geq 1$:

$$P^n_{succ}(X) \leq n \cdot P_{succ}(X)$$
$$P^n_{succ}(X|Y) \leq n \cdot P_{succ}(X|Y).$$

Therefore, we can study what happens in the case of one-try attacks (a single attempt) and then use the obtained results as an upper bound for the case of n-tries attacks.

2.4 Kullback-Leibler distance and the Method of Types

Another very useful concept is the *Kullback-Leibler distance*.

Definition 2.4.1 (Kullback-Leibler distance) *Given a random variable X, taking values in $\mathcal{X}$, and two probability distributions $p(\cdot)$ and $q(\cdot)$, both defined on $\mathcal{X}$, then the* Kullback-Leibler distance (or divergence) *between $p(\cdot)$ and $q(\cdot)$, denoted by $D(p\|q)$, is defined by:*

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

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

Note that Kullback-Leibler distance is not a real distance: for example, it is not symmetric, nor satisfies the triangle inequality. It is also true that, like real distances, it satisfies the property that it is always non negative and equals 0 if and only if its two arguments coincide, as expressed by the following theorem (CT06, Theorem 2.6.3).

Theorem 2.4.2 (Gibbs Inequality) *Let $p(\cdot)$ and $q(\cdot)$ be two probability distributions defined over the set $\mathcal{X}$. Then*

$$D(p\|q) \geq 0$$

with equality if and only if $p(x) = q(x)$ for all $x \in \mathcal{X}$.

$D(p\|q)$ measures the inaccuracy, or the information divergence, that we would have, assuming that the distribution of X is $q(\cdot)$, when the true distribution is in fact $p(\cdot)$.

Let us now illustrate the Method of Types, a very powerful technique, introduced by Csiszàr and Körner (Csi98), in order to study the behaviour of a set of sequences with the same empirical distribution.

Definition 2.4.3 (Type) *Let $\mathbf{x} = (x_1, \ldots, x_n) \in \mathcal{X}^n$ one of their realisations. The* type $t_{\mathbf{x}}(\cdot)$ *(or empirical distribution) of $\mathbf{x}$ is the relative proportion of occurrences of each symbol of X in $\mathbf{x}$. That is, for each symbol $x \in \mathcal{X}$, we have:*

$$t_{\mathbf{x}}(x) \triangleq \frac{n(x, \mathbf{x})}{n},$$

where $n(x, \mathbf{x})$ denotes the number of times the symbol x occurs in $\mathbf{x}$.

From now on, given the number n of considered random variables, we will denote with $\mathcal{P}_n$ the set of types with denominator n.

Definition 2.4.4 (Type class) *Let $\mathcal{P}_n$ be defined as above and let $p(\cdot)$ be one of its element. Then, the set of sequences of length n and whose type is $p(\cdot)$ is called* type class *of $p(\cdot)$ and denoted with $\mathcal{T}_p^n$:*

$$\mathcal{T}_p^n \triangleq \{\mathbf{x} \in \mathcal{X}^n | t_{\mathbf{x}}(\cdot) = p(\cdot)\}.$$

Example 7 *Given $\mathcal{X} = \{1, 2, 3\}$, a ternary alphabet, let $\mathbf{x} = (2, 1, 3, 3, 2)$. Then, the type of $\mathbf{x}$ is:*

$$t_{\mathbf{x}}(1) = \frac{1}{5}, \qquad t_{\mathbf{x}}(2) = \frac{2}{5}, \qquad t_{\mathbf{x}}(3) = \frac{2}{5}.$$

The type class of $t_{\mathbf{x}}$ is the set of sequences of length 5 with one occurrence of 1, two occurrences of 2 and two occurrences of 3, that is:

$$\mathcal{T}_p^n = \{(1, 2, 2, 3, 3), (1, 2, 3, 2, 3), \ldots, (3, 3, 2, 2, 1)\}.$$

The cardinality of $T(p)$ is $5 \cdot \binom{4}{2} = 30$: we have 5 possibilities of choosing the position of 1, each of which has to be multiplied by $\binom{4}{2}$, that are the possibilities of choosing the positions of 2. The remaining positions will be occupied by 3.

One of the results which makes the Method of Types so powerful is given by the following theorem, that shows that the number of types with a certain fixed length n is at most *polynomial* in n.

Theorem 2.4.5 (Theorem 11.1.1, (CT06)) *Let $\mathcal{P}_n$ and $\mathcal{X}$ be defined as above. Then:*

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

This theorem says that there exists only a polynomial number of types of length n. Since the number of sequences of such length is *exponential* in n, it means that there exists at least one type class containing an exponential number of sequences.

Let us finally see other three important results that will be useful in the next chapters:

Theorem 2.4.6 (Theorem 11.1.2, (CT06)) *Let $\mathbf{x} = (x_1, x_2, \ldots, x_n)$. Then, the probability of $\mathbf{x}$ under $q(\cdot)$ is given by*

$$q(\mathbf{x}) \triangleq \prod_{i=1}^{n} q(x_i) = 2^{-n\left(H(t_{\mathbf{x}}) + D(t_{\mathbf{x}} \| q)\right)}.$$

The probability $q(\mathbf{x})$ depends only on the type of $\mathbf{x}$, $t_{\mathbf{x}}(\cdot)$, and decreases exponentially with the Kullback-Leibler distance between this type and the real distribution $q(\cdot)$.

Theorem 2.4.7 (Theorem 11.1.4, (CT06)) *For any $p(\cdot) \in \mathcal{P}_n$ and any distribution $q(\cdot)$, the probability of the type class $\mathcal{T}_p^n$ under $q(\cdot)$ is*

$$\frac{1}{(n+1)^{|\mathcal{X}|}} 2^{-nD(p\|q)} \leq q(\mathcal{T}_p^n) \leq 2^{-nD(p\|q)}.$$

Theorem 2.4.8 (Equation (11.67), (CT06)) *Let ε be a positive real and $U_\varepsilon^n(q)$ be the set containing the sequences of length n whose type is distant from $q(\cdot)$ less than the value ε, that is:*

$$U_\varepsilon^n(q) \triangleq \{\mathbf{x} \in \mathcal{X}^n : D(t_{\mathbf{x}} \| q) \leq \varepsilon\}.$$

Then, the probability of obtaining sequences whose type is distant from $q(\cdot)$ more than ε decreases exponentially respect to the considered length n:

$$q((U_\varepsilon^n(q))^c) \leq (n+1)^{|\mathcal{X}|} 2^{-n\varepsilon}, \tag{2.5}$$

where by $(U_\varepsilon^n(q))^c$ we denote the complement set of $U_\varepsilon^n(q)$ in $\mathcal{X}^n$.

2.5 Chernoff Information and rate of convergence

Another important concept that will be useful later on is the *rate of convergence*, defined as follows.

Definition 2.5.1 (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*

$$\rho(f) \triangleq -\lim_{n\to\infty} \frac{1}{n} \log(f(n) - \gamma) \tag{2.6}$$

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

Note that we admit rates of 0, as well as of $+\infty$. In our case, the study of the rate of convergence is fundamental. Indeed, once chosen a resistance metric to measure the security of a system, the knowledge of its limit value is important to have an upper bound of the amount of information that can be leaked, but we are also interested in the rate at which this value is reached. The following example will clarify the reason.

Example 8 *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 $\rho(f) = \lambda_1$. On the other hand, since $f(n) \leq h(n) = \alpha + \beta + \gamma 2^{-n\lambda_2}$, one has that f reaches $\alpha + \beta$ at a higher rate of λ_2. Figure 1 displays a plot of three functions, characterised 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 should not be considered secure. It is for this reason that, when we analyse the asymptotic behaviour of a system, we cannot simply compute the limit values of the uncertainty function and, according to it, decide if the system is secure or not.

When we study the rate of convergence an important role is played by the Chernoff information.

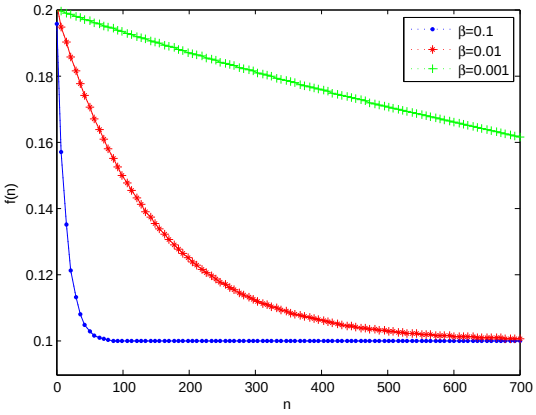

Figure 1: Slow rates of convergence do not guarantee security.

Definition 2.5.2 (Chernoff information) *Given two distributions $p(\cdot)$ and $q(\cdot)$, their* Chernoff Information *is defined as:*

$$C(p,q) \triangleq -\min_{0\leq\lambda\leq 1} \log \sum_{o\in\mathcal{O}} p^{\lambda}(o)q^{1-\lambda}(o), \tag{2.7}$$

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

As we will see later on, in Section 3.2.2 (Proposition 3.1.6, Theorem 3.1.7), if we consider the attacker's error probability and we analyse its asymptotic behaviour, a lower bound of it is given by the least Chernoff information between any two conditional probabilities over the set of observations.

Chapter 3

Passive attackers

In this chapter, we propose two models aiming to analyse different attack scenarios. In both of them we consider one-try attacks and system re-execution and we deal with passive attackers, not able to interact with the system, collecting a single observation for each execution of the system. The differences between these scenarios are the following ones: in the first case the attacker directly targets the states of the system; in the second one, instead, he targets properties related to the states. For both these scenarios, we describe the asymptotic behaviour of error probability and information leakage as the number n of collected observations goes to infinity. We show that the asymptotic values of these quantities can be determined in a simple way from the channel matrix. Moreover, we provide simple and tight bounds on error probability and on the leakage as functions of n, showing that the convergence is exponential. We also discuss feasible methods to compute the rate of convergence. More generally, we give bounds on the rate at which a chosen probability threshold can be reached.[1].

3.1 Passive attackers targeting states

We start with the simple scenario of a single passive eavesdropper, aiming to recover the secret from the collected observations.

[1]It may well be the case that, even if the asymptotic rate to the limit value is extremely slow, convergence to the chosen threshold is very fast, leading to consider the system insecure.

3.1.1 A basic model: Information Hiding Systems

We will consider a generic information hiding system[2] (IHS), that is 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, whose aim is obfuscating the possible relations between the secret and the observations detected by a potential attacker. As shown in Figure 2, we can view an IHS as a channel, that takes as input a secret and gives as outputs an observation. The use of this model in the field of QIF was promoted by Chatzikokolakis et al. in (CPP08a). Usually, we talk about *noisy* channels, since for each input there are (several) different outputs, each of which can be obtained with a certain probability, as shown in Figure 3.

Definition 3.1.1 (Information Hiding System) *An* information hiding system *is a quadruple*

$$\mathcal{H} \triangleq \langle \mathcal{S}, \mathcal{O}, p(\cdot), p(\cdot|\cdot) \rangle \,,$$

where: $\mathcal{S} = \{s_1, \ldots, s_m\}$ *is the finite set of* states*, representing the secret information,* $\mathcal{O} = \{o_1, \ldots, o_l\}$ *is a finite set of* observables*, containing all possible observations that can be collected by the attacker,* $p(\cdot)$ *is an a priori probability distribution on* $\mathcal{S}$ *and* $p(\cdot|\cdot) \in [0,1]^{\mathcal{S}\times\mathcal{O}}$ *is a* conditional probability matrix*, where each row sums up to 1.*

The matrix $p(\cdot|\cdot)$ is also called *channel matrix*. The entry of row s and column o will be written as $p(o|s)$, and represents the probability of observing o, given that s is the (secret) input of the system. For each s, the row of the matrix corresponding to s is identified with the probability distribution $o \mapsto p(o|s)$ on $\mathcal{O}$, denoted by $p_s(\cdot)$.

The probability distribution $p(\cdot)$ on $\mathcal{S}$ and the conditional probability matrix $p(o|s)$ together induce a probability distribution $q(\cdot)$ on $\mathcal{S} \times \mathcal{O}$ defined as $q(s,o) \triangleq p(s) \cdot p(o|s)$, hence a pair of random variables $(S, O) \sim q(\cdot)$, with S taking values in $\mathcal{S}$ and O taking values in $\mathcal{O}$. Note that $S \sim p(\cdot)$ and, for each s and o such that $p(s) > 0$, $\Pr(O = o|S = s) = p(o|s)$.

Before proceeding with the description of the attack model, let us consider four systems and explain how we can view them as IHS's. We will use them as simple running examples throughout this chapter.

[2]The term *information hiding system* here has no connection with the literature on watermarking. It simply denotes a randomisation mechanism.

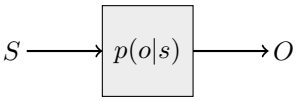

Figure 2: An information theoretic-channel.

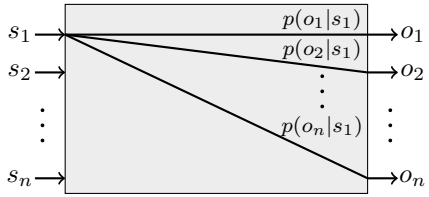

Figure 3: An information hiding system as a *noisy* channel.

Example 9 (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 $1 - p_f$. An instance of Crowds execution is shown in Figure 4. 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 senders identity.*

We can view Crowds (RR98) as an IHS *where the secret is the sender's identity, while observations are the identities of users that forward the message to a corrupted node, thus being* detected. *We have: $\mathcal{S} = \mathcal{O}$ is the set of honest users (i.e. possible senders), $p(s)$ is the prior probability of s being the real sender and $p(o|s)$ is the conditional probability of detecting o when the real sender is s.*

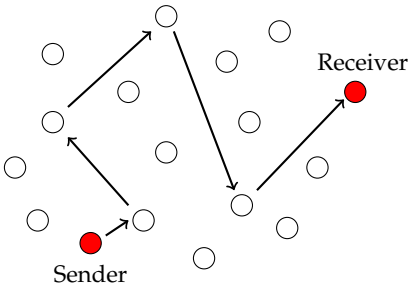

Figure 4: An instance of the Crowds protocol.

Example 10 (Modular exponentiation algorithm) *A typical implementation of modular exponentiation works as follows. The bits of the secret exponent are scanned from right to left, or vice-versa. When the ith bit is considered ($0 \leq i < N$), either only a squaring or a squaring and a multiplication are performed, depending on whether the i-th bit is 0 or 1. Hence, the lower the time needed to execute the exponentiation, the higher will be the amount of bits equal to 0 contained into the exponent (the secret key).*

We can view it as an IHS *where: $\mathcal{S} = \mathcal{K} = \{0,1\}^N$ is the set of private keys, i.e. the possible secret exponents, over which we assume a uniform distribution; $\mathcal{O} = \{t_1, t_2, \ldots\}$ is the set of possible execution times; $p(t|k)$ is the probability that, depending on the deciphered message, the execution of the algorithm takes time t, given that the private key is k. To be more specific concerning the last point, we assume an underlying set of messages $\mathcal{M}$, with a known prior distribution $p_M(m)$, and a function*[3] *$time : \mathcal{M} \times \mathcal{S} \to \mathcal{O}$ that yields the duration of the execution of the algorithm when its argument is a given pair (m, k). Then the entries of the probability matrix $p(t|k)$ can be defined thus*

$$p(t|k) \triangleq \sum_{m \in \mathcal{M}: time(m,k)=t} p_M(m).$$

Example 11 (Hamming weight attacks against S-boxes) S-boxes, *or substitution boxes, are a fundamental component in most symmetric key block ciphers (e.g.* DES*), aiming to obscure the connection between plaintext and ciphertext. In the case of* DES*, an S-box, as illustrated in Figure 5, can be*

[3] A more realistic modeling would make $time(m, k)$ a joint probability distribution. This modification would not substantially affect the final result.

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, $SB : \mathcal{K} \times \mathcal{M} \to \mathcal{C}$, 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. Let us assume a uniform prior distribution on $\mathcal{K}$ and some known prior distribution on $\mathcal{M}$, say $p_M(\cdot)$.

Similarly to (KSWH00), we assume the attacker can create a side-channel delivering him the Hamming weight of the target S-box output. This is a plausible scenario, since, in an unprotected implementation, either the execution time or the power consumption relative to a S-box computation might be more or less closely related to the Hamming weight of the result. Based on this assumption, (KSWH00) carries out a statistical attack against the last round [4] sub-key of DES. *Similar assumptions are also at the basis of the chosen-message model, more recently proposed in (SMY09).*

To the S-box thus described there corresponds an IHS *where: $\mathcal{S} = \mathcal{K}$, $\mathcal{O} = \{0,1,2,3,4\}$ is the set of observables, i.e. the set of possible Hamming weights, and $p(o|k)$ is defined as*

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

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

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, O, is now disturbed by the noise N, given by the sum of the Hamming weights of the remaining seven S-boxes, say $W_2, \ldots, W_8$, as shown in Figure 5. Assuming that the variables W_i are independent from each other and from O 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 normal distribution. Here, for simplicity we model N as a discrete random variable having binomial distribution $B(n,p)$ with $n = 28$ and $p = \frac{1}{2}$. What is observed by the attacker now is $O' \triangleq O + N$. Hence the new set of observables is $\mathcal{O}' = \{0, \ldots, 32\}$. Explicitly, for each $i \in \mathcal{O}'$ and $k \in \mathcal{K}$, the entries of the new conditional probability matrix $p'(\cdot|\cdot)$ are given by*

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

[4] The last round is crucial since the message part of the S-box input is known to the attacker.

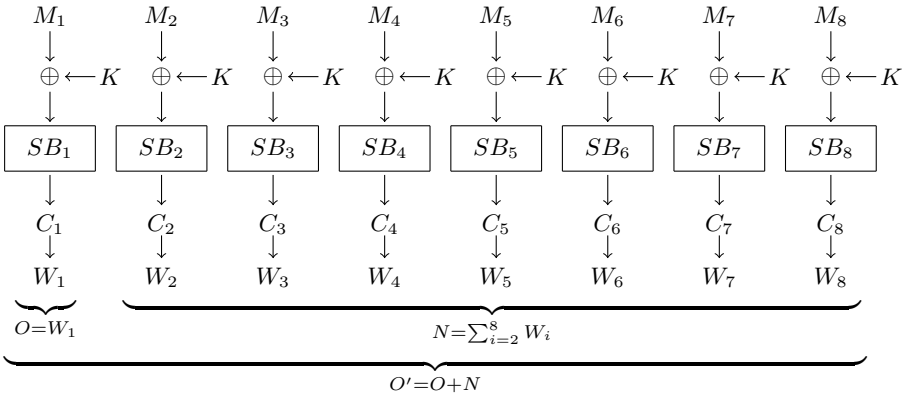

Figure 5: Functioning of DES S-boxes and noisy version of the Hamming weight attack.

Example 12 (Unlinkability in Threshold Mixnets) *Statistical attacks against anonymity protocols may take advantage of sender-receiver relationships that remain fixed through repeated rounds of the protocol. Let us 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 to be chosen at random in a set of nodes $R_1, \ldots, R_N$. 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.*

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 $\mathcal{S} = \{R_1, \ldots, R_N\}$, with $p_S(R_i) = \frac{1}{N}$ for each $i = 1, \ldots, N$; the set of*

observations is $\mathcal{O} = \{0, 1\}$, where $O = 1$ if and only if R_j has received at least one message at the end of the round. The conditional probability matrix $p(\cdot|\cdot)$ is given by the following equalities:

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

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 – that is, all messages of the batch, 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, where the last row refers to the observed node R_j.

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

3.1.2 An attack model with repeated observations

Let us discuss now the attack scenario we will analyse. 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, $o^n = (o_1, \ldots, o_n) \in \mathcal{O}^n$, throughout which the secret state s is kept fixed. Formally, the adversary knows a random vector of observations $O^n = (O_1, \ldots, O_n)$ such that, for each $i = 1, \ldots, n$, O_i is distributed like O and the individual O_i are *conditionally independent* given S, that is, the following equality holds true for each $o^n \in \mathcal{O}^n$ and $s \in \mathcal{S}$ such that $p(s) > 0$

$$\Pr\left(O^n = (o_1, \ldots, o_n) \,|\, S = s\right) = \Pi_{i=1}^n \Pr(O_i = o_i | S = s). \tag{3.1}$$

Note that the right-hand side of the above equality can be equivalently written as $\Pi_{i=1}^n p(o_i|s)$, often abbreviated as $p(o^n|s)$.

Concerning the security metrics, we will use here the min-entropy definition, discussed in Section 2.3, with $X = S$ and $Y = O^n$. We make the definition explicit below for the reader's convenience. For any fixed length n of observations, the attacker's strategy is modeled by a so

called *guessing* (or *decision*) *function*, chosen by the adversary to deduce the value of s by the observations. Let $g : \mathcal{O}^n \to \mathcal{S}$ be this function, representing the single guess the attacker is allowed to make about the secret s, after observing o^n.

Definition 3.1.2 (Error Probability) *Let $g : \mathcal{O}^n \to \mathcal{S}$ be a guessing function. The* error probability after n observations *(relative to g) is given by*

$$P_e^{(g)}(n) \triangleq 1 - P_{succ}^{(g)}(n) \qquad \textit{where} \qquad P_{succ}^{(g)}(n) = \Pr(g(O^n) = S).$$

It is well-known (see e.g. (CT06)) that the optimal strategy for the adversary, that is the one that minimises the error probability, is the Maximum A Posteriori (MAP) rule, defined below.

Definition 3.1.3 (Maximum A Posteriori rule, MAP) *A function $g : \mathcal{O}^n \to \mathcal{S}$ satisfies the* Maximum A Posteriori (MAP) criterion *if for each o^n and s*

$$g(o^n) = s \quad \textit{implies} \quad p(o^n|s)p(s) \geq p(o^n|s')p(s') \textit{ for each } s'.$$

In the above definition, for $n = 0$ one has $o^n = \epsilon$, and it is convenient to stipulate that $p(\epsilon|s) = 1$: that is, with no observations at all, g selects some s maximising the prior distribution. With this choice, $P_e^{(g)}(0)$ denotes $1 - \max_s p(s)$. The error probability associated with this function, also called *Bayes Risk*, can be explicitly computed as follows:

$$P_e^{(g)}(n) = 1 - \sum_{o^n \in \mathcal{O}^n} p(o^n) \max_{s \in \mathcal{S}} \Pr(S = s | (O_1, \ldots, O_n) = o^n) \tag{3.2}$$

$$= 1 - \sum_{o^n \in \mathcal{O}^n} \max_{s \in \mathcal{S}} \Pr((O_1, \ldots, O_n) = o^n | S = s) p(s), \tag{3.3}$$

where $p(o^n) \triangleq \Pr((O_1, \ldots, O_n) = o^n)$. It can be proved that if g is MAP, for each guessing function $g' : \mathcal{O}^n \to \mathcal{S}$, $P_e^{(g')} \leq P_e^{(g)}$ (see (CPP08a, Proposition 10) for details).

It worthwhile to note that, once n and $p(\cdot)$ are fixed, the MAP guessing function is 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 this section we assume without loss of generality a fixed MAP guessing function g for each given n and probability distribution $p(\cdot)$. We shall omit the superscript $^{(g)}$, except where this might cause confusion.

Another widely used criterion for the choice of g is *Maximum Likelihood (ML)*, which given o^n selects a state s maximising the likelihood

$p(o^n|s)$ 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 3.1.10 in the next subsection).

We now come to information leakage: this is a measure of the information leaked by the system, obtained by comparing the prior and the posterior (to the observations) 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 and then takes the logarithm (or, equivalently, consider the difference of the log's), one gets the multiplicative form, obtaining the *min-entropy* based definition considered by Smith (Smi09)[5]. In the latter case, one could equivalently consider the simple ratio, without logarithm, obtaining the multiplicative leakage of (BCP09). The logarithm allows to directly measure the result in bits.

Definition 3.1.4 (Additive and min-entropy leakage (BCP09; Smi09)) *The additive and min-entropy leakage after n observations are defined respectively as*

$$\mathcal{L}_+(n) \triangleq P_{succ}(n) - \max_s p(s) \qquad \textit{and} \qquad \mathcal{L}_\times(n) \triangleq \log \frac{P_{succ}(n)}{\max_s p(s)}.$$

It allows us to compute the leaked information as a function of the number n of observations collected by the attacker. In the next sections, we will mainly use the min-entropy form. Therefore, for the sake of notation, we shall omit the subscript $_\times$, denoting $\mathcal{L}_\times$ as $\mathcal{L}$, except where this might cause confusion.

What we are interested in is the asymptotic behaviour of the leakage, that is, studying what happens if n goes to infinity, in order to have an upper bound of the maximum amount of information that can be leaked by the system. As we can already see from Definition 3.1.4, the asymptotic behaviour of information leakage completely depends on the suc-

[5]Smith (Smi09) defines the leakage as $\log \frac{V_{post}}{V_{pr}}$, where, using our notation, $V_{pr} \triangleq \max_s p(s)$ is the *prior vulnerability* and $V_{post} \triangleq \sum_{o^n} \Pr(O^n = o^n) \cdot \max_s \Pr(S = s|O^n = o^n)$ is the *posterior vulnerability* (after n observations; Smith only defines the case $n = 1$). To see that $V_{post} = P_{succ}(n)$, just note that $P_{succ}(n) = \sum_{o^n} \Pr(O^n = o^n) \cdot \Pr(g(o^n) = S|O^n = o^n) = \sum_{o^n} \Pr(O^n = o^n) \cdot \max_s \Pr(S = s|O^n = o^n)$, where the last equality follows because g is MAP.

cess probability $P_{succ}(n)$, or on its complement $P_e(n)$. It is for this reason that from now on the results we will present mainly concern the error probability. In particular, as we will see later on, the asymptotic behaviour of the leakage strictly depends on an equivalence relation defined over the set of states $\mathcal{S}$, that associates in the same class states that induce the same probability distribution over the observables, thus being indistinguishable from the attacker's point of view. This equivalence relation, called *indistinguishability* relation, plays an important role in determining the fundamental security parameters of the system. The formal definition is as follows.

Definition 3.1.5 (Indistinguishability) *Given* $s, s' \in \mathcal{S}$,

$$s \equiv s' \qquad \textit{if and only if} \qquad p_s = p_{s'}.$$

Concretely, two states are indistinguishable if and only if 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. Note that, in the case when the channel matrix actually defines a deterministic function f, the equivalence classes of $\equiv$ are precisely the counter-images of f in $\mathcal{S}$, that is, the sets $f^{-1}(o)$ for $o \in \mathcal{O}$.

Example 13 *Let us consider again the modular exponentiation algorithm and the corresponding* IHS*, seen in Example 10. It would appear reasonable to assume that, for each message m, the execution time only depends on the number of '1' digits in k. In other words, we assume that whenever k and k' have the same Hamming weight, $time(m, k) = time(m, k')$, for any m. From this assumption and the definition of $p(t|k)$, it follows that whenever k and k' have the same Hamming weight then $p(\cdot|k) = p(\cdot|k')$. Thus, in the system there are at most as many $\equiv$-classes as Hamming weights, that is $N + 1$.*

Example 14 *Consider the small imperative procedure* `c()` *described below. There,* `h` *and* `l` *are two-bits integer global variables, while* `rnd()` *is a procedure returning a random real value in the interval* $[0, 1]$*. Boolean values* `true` *and* `false` *are identified with integers 1 and 0, respectively.*

```
proc c();
{
```

```
 l=rnd();
 if not(h mod 2) then l=(l >= .5)
 else l=1+(l >= (.5 + (h div 2)*10^-5) );
 return l
}
```

Now, assume `h` *is a sensitive variable, whose initial value is chosen in the range 0..3 and then never modified. We assume that* `c()` *can be invoked several times. One is interested in analysing the asymptotic information leakage relative to* `h` *caused by* `c()`*. We can model the procedure* `c()` *as an information hiding system, as follows.*

Let $\mathcal{S} = \{0, 1, 2, 3\}$ *be the set of possible values of* `h`*, and* $\mathcal{O} = \{0, 1, 2\}$ *the set of possible values returned by* `c()`*. The prior probability distribution on* $\mathcal{S}$ *is non-uniform and given by:* $p(0) = p(1) = \frac{1}{2} - 10^{-9}$ *and* $p(2) = p(3) = 10^{-9}$*. The behaviour of* `p()` *can be described by the conditional probability matrix displayed below.*

	o_1	o_2	o_3
s_1	$\frac{1}{2}$	0	$\frac{1}{2}$
s_2	$\frac{1}{2}$	0	$\frac{1}{2}$
s_3	0	$\frac{1}{2}$	$\frac{1}{2}$
s_4	0	$\frac{1}{2} - 10^{-5}$	$\frac{1}{2} + 10^{-5}$

In this case, $s_1 \equiv s_2$ *since they generate with the same probability all possible observations (the corresponding rows in the channel matrix coincide) and* $\mathcal{S}/\equiv = \{\{s_1, s_2\}, \{s_3\}, \{s_4\}\}$.

Example 15 *In Example 12, from the channel matrix we can immediately see that here there are only two classes of indistinguishability:* $\mathcal{S}/\equiv$ *is* $\{C_1, C_2\}$*, with* $C_1 = \{R_j\}$ *and* $C_2 = \mathcal{S} \setminus \{R_j\}$.

3.1.3 Bounds and asymptotic behaviour

Let us introduce some notation that will be used throughout the section. Let $\mathcal{S}/\equiv$ be $\{C_1, \ldots, C_K\}$, the set of equivalence classes of $\equiv$. For each $i = 1, \ldots, K$, let

$$s_i^* \triangleq \text{argmax}_{s \in C_i} p(s), \quad p_i^* \triangleq p(s_i^*) \quad \text{and} \quad p_i(\cdot) \triangleq p(\cdot | s_i^*). \tag{3.4}$$

We assume without loss of generality that $p_i^* > 0$ for each $i = 1, \ldots, K$ (otherwise all the states in class C_i can be just discarded from the system). For each $i, j \in \{1, \ldots, K\}$ let

$$c_{ij} \triangleq C(p_i, p_j), \tag{3.5}$$

where $C(\cdot, \cdot)$ is the Chernoff Information (see Definition 2.5.2). As mentioned in Section 2.5, Chernoff Information is very important when we study the rate of convergence. Indeed, by adapting the proof for the case $|\mathcal{S}| = 2$, given in (CT06; LJ97), it is not difficult to prove the following result, that gives the exact rate of convergence of $P_e(n)$, $\rho(P_e)$, in the case where the distributions $p_1(\cdot), \ldots, p_K(\cdot)$ all have the same support.[6].

Proposition 3.1.6 *Suppose that* $\mathrm{supp}(p_1) = \cdots = \mathrm{supp}(p_K)$. *Then* $\rho(P_e) = \min_{i \neq j} c_{ij}$.

The next theorem provides tight bounds on the error probability and its rate in the general case, although in general not the *exact* rate. The interpretation is the following: the attacker focuses on the set of representative states, $\{s_i^* | i = 1, \ldots, K\}$, and tries to identify one of them as the value of S. This strategy can fail for two reasons: either S is not in the target set (first term in the error upper bound), or it is, but the attacker, mistakes one state in the set for another (second term in the error upper bound). The latter probability decreases exponentially fast with n, at a rate that is at least as big as the minimum "distance" ρ between the distributions $p_i(\cdot)$, for $i = 1, \ldots, K$. We stipulate that $2^{-\infty} = 0$.

Theorem 3.1.7 *Let* $\rho \triangleq \min_{\substack{i,j=1,\ldots,K, \\ i \neq j}} c_{ij}$. *Let* $p_{\max}^* = \max_i p_i^*$. *Then, for all* $n \geq 1$

$$(1 - \sum_{i=1}^{K} p_i^*) \;\leq\; P_e(n) \;\leq\; (1 - \sum_{i=1}^{K} p_i^*) \;+\; \frac{K^2}{2} p_{\max}^* 2^{-n\rho}. \tag{3.6}$$

As a consequence, $P_e(n)$ *reaches* $1 - \sum_{i=1}^{K} p_i^*$ *at a rate of* $\rho(P_e) \geq \rho$.

The theorem has a simple interpretation in terms of the attacker's strategy: after infinitely many observations, he can determine the indistinguishability class of the secret, say C_i, and then guess the most likely state in that class, s_i^*.

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

PROOF Fix $n \geq 1$. Let $\mathcal{R} = \{s_i^* | i = 1, \ldots, K\}$ and $g : \mathcal{O}^n \to \mathcal{R}$ be a MAP function satisfying:

$$g(o^n) = s_i^* \quad \text{implies} \quad p(o^n|s_i^*)p_i^* \geq p(o^n|s_j^*)p_j^* \quad \text{for each } j = 1, \ldots, K.$$

For each $i = 1, \ldots, K$, let $A_i = g^{-1}(s_i^*)$ be the acceptance region for s_i^*. Concerning the lower bound, we have (the sums below run over s's such that $p_S(s) > 0$):

$$\begin{aligned} P_e(n) &= \sum_{s \in \mathcal{S}} \Pr(g(O^n) \neq s | S = s) p_S(s) \\ &= \sum_{s \notin \mathcal{R}} \Pr(g(O^n) \neq s | S = s) p_S(s) + \sum_{i=1}^{K} \Pr(g(O^n) \neq s_i^* | S = s_i^*) p_i^* \\ &= (1 - \sum_{i=1}^{K} p_i^*) + \sum_{i=1}^{K} p_i(A_i^c) p_i^* \end{aligned} \tag{3.7}$$

which, since the second term is never negative, implies the lower bound in the statement. Concerning the upper bound, instead, we can apply the following inequality to Equation (3.7), obtaining:

$$\begin{aligned} P_e(n) &\leq (1 - \sum_{i=1}^{K} p_i^*) + \sum_{i=1}^{K} \sum_{\substack{j=1, \\ j \neq i}}^{K} p_i(A_j) p_i^* \\ &= (1 - \sum_{i=1}^{K} p_i^*) + \sum_{i=1}^{K} \sum_{\substack{j=1, \\ j > i}}^{K} (p_i(A_j) p_i^* + p_j(A_i) p_j^*) \end{aligned} \tag{3.8}$$

where the inequality follows from $A_i^c = \cup_{j \in \{1,\ldots,K\} \setminus \{i\}} A_j$ and a simple union bound, while the last equality is simply a rearrangement of summands. Now, we evaluate $p_i(A_j)p_i^* + p_j(A_i)p_j^*$ for each $i, j = 1, \ldots, K$ and $i \neq j$.

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

$$p_i(A_j)p_i^* + p_j(A_i)p_j^* \leq p_{\max}^* 2^{-n\rho} \tag{3.9}$$

Now, if we plug the bound (3.9) into (3.8), and then factor out $p^*_{\max} 2^{-n\rho}$ and reorder the summands, we get

$$P_e(n) \;\leq\; (1 - \sum_{i=1}^{K} p_i^*) \;+\; \big(\sum_{i=1}^{K} \sum_{\substack{j=1,\\ j>i}}^{K} 1\big)\, p^*_{\max} 2^{-n\rho}\,.$$

Now, use the fact that $\big(\sum_{i=1}^{K} \sum_{\substack{j=1,\\ j>i}}^{K} 1\big) = \frac{K(K-1)}{2} \leq \frac{K^2}{2}$, which completes the proof. □

Remark 3.1.8

- *In the practically important case where the prior p_S on $\mathcal{S}$ is uniform, the term $\frac{K^2}{2} p^*_{\max} 2^{-n\rho}$ is bounded above by $\frac{K}{2} 2^{-n\rho}$.*
- *Computation of the Chernoff Information* (2.7) *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 can be found useful in the case of distributions with sparse support 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 $-\min\{\log p(\sigma)\,,\, \log q(\sigma)\}$.*

The following result shows that, asymptotically, the security of a system is tightly connected to the number of indistinguishability classes - and in the case of uniform distribution *only* depends on this number.

Corollary 3.1.9 *If the a priori distribution on $\mathcal{S}$ is uniform, then $P_e(n)$ converges exponentially fast to $1 - \frac{K}{|\mathcal{S}|}$.*

If all the states are distinguishable, that is the partition induced by this relation is formed by all singleton classes, then $K = |\mathcal{S}|$. So, the limit value of the error probability in this case is 0, that is, asymptotically the whole information will be disclosed. The lower the number K of classes, the lower the limit value obtained, and so the more resistant the system. The minimum value is reached when the prior is the uniform distribution and the partition coincides with $\mathcal{S}$ ($K = 1$), that is all states are indistinguishable for the attacker.

Remark 3.1.10 (on the ML rule) *Braun et al. in (BCP09) show 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 3.1.9 we therefore deduce that the average* ML *error converges exponentially fast to the value $1 - \frac{K}{|\mathcal{S}|}$ as $n \to \infty$.*

We discuss now some consequences of the above results on information leakage. Assume without loss of generality that $p_1^* = \max_s p(s)$. In what follows, we denote by $p_{max}(\cdot)$ the probability distribution on $\mathcal{S}$ defined by: $p_{max}(s) = \frac{1}{K}$ if $s \in \{s_1^*, \ldots, s_K^*\}$ and $p_{max}(s) = 0$ otherwise.

Corollary 3.1.11 *1. $\mathcal{L}_+(n)$ converges exponentially fast to $\sum_{i=2}^K p_i^*$. This value is maximised by the prior distribution $p_{max}(\cdot)$, which yields the limit value $1 - \frac{1}{K}$.*

2. $\mathcal{L}_\times(n)$ converges exponentially fast to $\log \frac{\sum_{i=1}^K p_i^*}{p_1^*}$. This value is maximised when the prior distribution is either uniform or $p_{max}(\cdot)$, both of which yield the limit value $\log K$.

PROOF

1. The value of the limit follows directly from the definition of $\mathcal{L}_+(n)$ and Theorem 3.1.7. Concerning the second part, for any fixed $p(\cdot)$, it is easily checked that $\sum_{i=2}^K p_i^* \leq 1 - \frac{1}{K}$ (this is done by separately considering the cases $\max_s p(s) \geq \frac{1}{K}$ and $\max_s p(s) < \frac{1}{K}$). But the value $1 - \frac{1}{K}$ is obtained asymptotically with the distribution $p_{max}(\cdot)$.

2. Again, the value of the limit follows directly from the definition of $\mathcal{L}_\times(n)$ and Theorem 3.1.7. Concerning the second part, for any fixed $p(\cdot)$, of course we have $\sum_{i=1}^K \frac{p_i^*}{p_1^*} \leq \sum_{i=1}^K 1 = K$. But the value K is obtained asymptotically when the prior is either uniform or the distribution $p_{max}(\cdot)$. Applying now the logarithm to the inequality above and taking into account the fact that it is an increasing function, we obtain the thesis.

□

Remark 3.1.12 *A consequence of Corollary 3.1.11(2) is that, in the case of uniform distribution on states, the min-entropy leakage as n goes to infinity coincides with the logarithm of the number of equivalence classes K. If one considers deterministic systems, that is systems where the channel matrix defines a function $f : \mathcal{S} \to \mathcal{O}$, then the leakage does not depend on the number of observations: $\mathcal{L}_\times(n) = \log K$ for $n \geq 1$. Moreover K equals the number of distinct counter-images of f, that is the number of elements in the range of f; in particular $K \leq |\mathcal{O}|$. This way we re-obtain a result of (Smi09) for deterministic systems.*

In (BCP09) additive leakage is contrasted with multiplicative (or min-entropy) 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 maximises 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 maximising distribution, $p_{max}(\cdot)$.

Remark 3.1.13 *In (KS10), Köpf and Smith observe that, in the case of uniform distribution on $\mathcal{S}$, min-entropy leakage is upper-bounded by the logarithm of the number of types of n-sequences of $\mathcal{O}$:*

$$\mathcal{L}_{\times}(n) \;\leq\; \log |\mathcal{P}_n| \,. \tag{3.10}$$

It is interesting to compare this upper-bound, which depends on n, with our upper-bound, the value $\log K$ given by Corollary 3.1.11(2). It is clear that, since as $n \to \infty$ one has $|\mathcal{P}_n| \to \infty$ as well, (3.10) *ceases to be useful for large values of n. According to Theorem 2.4.5, $|\mathcal{P}_n| \leq (n+1)^{|\mathcal{O}|}$. Using some algebra, one sees that* (3.10) *is sharper than our upper-bound $\log K$ at least as long as*

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

So it appears that the upper-bound (3.10) *is useful only when the number of rows of the matrix is very large compared to the number of observables.*

The above results prompt the following question. Suppose the attacker somehow ignores the rows of the channel matrix that are close together with each other, and only considers those that are far from each other. That is, suppose that he focuses on a subset of the representative states $\{s_i^* | i \in I\}$, for a certain $I \subseteq \{1, \ldots, K\}$. The next result shows that it is possible to achieve an higher rate of convergence $\rho_I > \rho$, although ignoring some rows might lead to a possibly higher asymptotic error probability. In this way, we cover the case where there exist some other values, slightly higher than the limit value obtained above, but that can be reached much faster.

Theorem 3.1.14 *Let I be a nonempty subset of $\{1,\ldots,K\}$. Let $\rho_I \triangleq \min_{\substack{i,j\in I,\\ i\neq j}} c_{ij}$. Let $p^*_{\max} = \max_{i\in I} p^*_i$. Then, for all $n \geq 1$*

$$P_e(n) \;\leq\; (1 - \sum_{i\in I} p^*_i) \;+\; \frac{|I|^2}{2} p^*_{\max} 2^{-n\rho_I}. \tag{3.11}$$

As a consequence, $P_e(n)$ reaches $1 - \sum_{i\in I} p^*_i$ at a rate of ρ_I.

PROOF For any $n \geq 0$ and $s \in \mathcal{S}$, let $A^{(n)}_s$ be the acceptance regions determined by any MAP guessing function. Choose any $i^* \in I$. For any $i = 1,\ldots,K$, define the new acceptance regions $B^{(n)}_i$ as follows: $B^{(n)}_i = \emptyset$ if $i \notin I$, otherwise $B^{(n)}_i = A^{(n)}_{s^*_i}$. For each n, the regions $B^{(n)}_i$ determine a new guessing function, say g', which will in general *not* be MAP. Now, repeating the computation in the proof of Theorem 3.1.7, varying i, j in I instead of $\{1,\ldots,K\}$ and with the regions $B^{(n)}_i$ instead of $A^{(n)}_s$, one finds

$$P^{(g')}_e(n) \;\leq\; (1 - \sum_{i\in I} p^*_i) \;+\; \frac{|I|^2}{2} p^*_{\max} 2^{-n\rho_I}.$$

The wanted result follows from the optimality of the MAP rule, which implies $P_e(n) \leq P^{(g')}_e(n)$. □

Consider again Example 14 and apply to it this result.

Example 16 *Applying Theorem 3.1.7, we find that*

$$1 - E \leq P_e(n) \leq 1 - E + \frac{|I|^2}{2}(1 - 10^{-9}) 2^{-n\rho_I},$$

where $E = 1 - 10^{-9}$. Now, the number of indistinguishable classes here is 3. If we consider $I = \{0,1,3\}$, we obtain that $P_e(n) \longrightarrow_{n\to\infty} 1 - E = 10^{-9}$ at a rate $\rho = 1 - \log(1 - 10^{-5}) \simeq 1.443 \times 10^{-5}$, very slow. One wonders if there is some value $1 - E'$ that is only slightly higher than $1 - E$, but that can be reached much faster. This is indeed the case. Observe that rows 1 and 3 are very close with each other in norm-1 distance: $\|p_1 - p_3\|_1 = 2 \times 10^{-5}$. We can discard row 3, which has a very small probability, and then apply Theorem 3.1.14 with $I = \{0,1\}$ to get

$$P_e(n) \leq 1 - E' + \frac{|I|^2}{2}(1 - 10^{-9}) 2^{-n\rho_I}$$

	d_1	d_2	$\cdots$	d_{20}
s_1	0.468	0.028	$\cdots$	0.028
s_2	0.028	0.468	$\cdots$	0.028
$\vdots$			$\vdots$	
s_{20}	0.028	0.028	$\cdots$	0.468

Figure 6: The conditional probability matrix of Crowds for 20 honest nodes, 5 corrupted nodes and $p_f = 0.7$.

where $E' = \frac{1}{2} - 10^{-9} + \frac{1}{2} - 10^{-9} = 1 - 2 \times 10^{-9}$ *and* $\rho_I = 1$. *This implies that the value* $1 - E'$ *is approached much faster as* n *grows. For instance, already after* $n = 37$ *invocations we get that* $(1 - E')/P_e(n) > 0.99$.

3.1.4 Some applications

Let us consider now more in detail the four applications mentioned in Section 3.1.

Protocol re-execution in Crowds (BPP11a; BPPar) Consider a system with $m = 20$ users. An example of such a system, borrowed from (CPP08b), is given in the table in Figure 6. 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. if 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 $o^n = (o_1, \ldots, o_n) \in \mathcal{O}^n$, for some n. The repeated executions are assumed to be independent, hence we are precisely in the setting considered before. This case is also considered in (CPP08b), which gives lower bounds for the error probability holding for any n. Our results in Section 3.2.2 generalise 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 the 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 of Theorem 3.1.7, is $\rho \approx 0.4482$. This implies that already after observing $n = 30$ re-executions the probability of error is < 0.001.

It is worth to stress that protocol re-execution is normally prevented in Crowds for the very reason that it decreases anonymity, although it

may be necessary in some cases. See the discussion on static vs. dynamic paths in (RR98).

Modular exponentiation algorithm Consider again Example 10. From each of the $N+1$ indistinguishability classes, let us choose a representative s_i^* of probability $p_i^* = \frac{1}{2^N}$. Applying Theorem 3.1.7, we find that $P_e(n) \to 1 - \frac{N+1}{2^N}$, which for realistic values of N, e.g. $N = 1024$, is very close to 1. Accordingly, applying Corollary 3.1.11, we get that additive and min-entropy leakage satisfy, asymptotically,

$$\mathcal{L}_+ \leq \frac{N}{2^N} \qquad \text{and} \qquad \mathcal{L}_\times \leq \log(N+1).$$

For any practical size of the key, say $N = 1024$, these upper bounds yield negligible values: $\mathcal{L}_+ \approx 0$ and $\mathcal{L}_\times \leq 1025$. The latter case tells us that just $\log(1025) = 10.001$ bits of min-entropy are leaked, out of 1024. In conclusion, the modular exponentiation algorithm appears to provide satisfactory guarantees of security against one-try attacks.

Hamming weight attacks against S-boxes (BPP11a; BPPar) We report here on our results concerning the first of the eight S-boxes of DES. Analysis of other S-boxes leads to similar conclusions. The distribution of the plaintext, $p_M(\cdot)$, plays a crucial role here: the lower the redundancy, the less information is expected to be extracted from the side-channel. For example, if $p_M(\cdot)$ is the uniform distribution (0% redundancy), then it is easy to see that all the rows of the matrix $p(o|k)$ 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, $p(o|k)$, all the rows are different, which implies that $P_e(n) \to 0$. Concerning the rate of convergence, Theorem 3.1.7 yields $\rho \approx 1.6 \times 10^{-3}$. This means that with $n \geq 7.2 \times 10^3$ observations the error probability is < 0.011. Discarding the keys corresponding to the 23 shortest norm-1 distances, one would get $\rho \approx 2 \times 10^{-3}$. Applying Theorem 3.1.14, one gets an error probability ≤ 0.011 already with $n \geq 5.4 \times 10^3$ observations.

Concerning the more realistic scenario, instead, Theorem 3.1.7 applied to the matrix $p'(\cdot|\cdot)$ yields a rate of $\rho \approx 5.963 \times 10^{-6}$. It implies that this time $P_e(n) < 0.011$ for $n \geq 1.9295 \times 10^6$. 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%.

Unlinkability in Threshold Mix-Nets (BPP11b) Let $\mathcal{H}$ be the IHS corresponding to this application described in Example 12. Applying Theorem 3.1.14 to $\mathcal{H}$, we can compute 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:

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

where the last inequality follows from the double application of the inequality $1 - x \leq e^{-x}$ for $x < 1$. As expected, the limit value $1 - \frac{2}{N}$ is > 0, and the security of the system increases as N increases. The corresponding asymptotic min-entropy 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.

3.2 Passive attackers targeting state predicates

In the previous section we have taken into account passive attackers that directly target the secret, trying to recover it from the collected observations. Here, instead, we analyse a scenario where the adversary is still passive, but is only interested in understanding if a certain predicate of the secret does hold, or not.

This approach allows us to analyse not only the quantitative aspect of the analysis (*how much* information is leaked), but also the qualitative aspect (*what* is leaked). In the previous section we have shown that, when a uniform distribution on the secrets is assumed, the asymptotic min-entropy leakage of a system corresponds to the logarithm of the number of indistinguishability classes in the system. For instance, an anonymity protocol in which users are grouped into a small number of classes is considered as globally secure. 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 totally exposed to eavesdropping. To make another, extreme example, consider the two small imperative procedures `P1` and `P2` below. Each of them receives as an argument a confidential variable `h` that can take on a value in the set $\mathcal{S} = \{0, \ldots, 15\}$, perhaps corresponding to user identifiers or other sensitive information. Part of the information about `h` is then disclosed 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 $\mathcal{S}$ 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`. However, suppose the attacker is only interested in understanding if `h` is equal to 0 or not. In this case, `P2` is preferable over `P1`, because relatively to the single case `h=0`, it leaks less information: indeed, while `P1` in this particular case completely reveals the secret to the attacker, `P2` leaks only its least two significant bits. In order to cope with such problems, 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 the next sections, we propose a framework to deal with this issue by extending the IHS's considered in the Section 3.1 and elsewhere with *views*.

3.2.1 An extended model: 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 for the case `h=0` (coinciding with the attacker's target) is the partition of $\mathcal{S}$ into $(\{0\}, \mathcal{S} \setminus \{0\})$. 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.

More formally, a view can be defined as follows.

Definition 3.2.1 (views) *Let $\mathcal{H} = \langle \mathcal{S}, \mathcal{O}, p(\cdot), p(\cdot|\cdot) \rangle$ be a* IHS. *A* view *of $\mathcal{H}$ is a pair $(\mathcal{W}, q(\cdot|\cdot))$, where $\mathcal{W}$ is a finite alphabet and $q(\cdot|\cdot) \in [0,1]^{\mathcal{S}\times\mathcal{W}}$ is a matrix where all rows sum to one.*

Informally, $q(w|s)$ is the probability that the predicate w holds when in state s. The probability distribution $p(\cdot)$ on $\mathcal{S}$ and the conditional probability matrices $p(\cdot|\cdot) \in [0,1]^{\mathcal{S}\times\mathcal{O}}$ and $q(\cdot|\cdot) \in [0,1]^{\mathcal{S}\times\mathcal{W}}$ induce a probability distribution $r(\cdot)$ on $\mathcal{W} \times \mathcal{S} \times \mathcal{O}$, defined as $r(w,s,o) \triangleq p(s) \cdot p(o|s) \cdot q(w|s)$. This distribution induce a triple of discrete random variables $(W, S, O) \sim r(\cdot)$, taking values in $\mathcal{W} \times \mathcal{S} \times \mathcal{O}$. We shall denote the marginal probability distributions of this triple for S, W and O by $p_S(\cdot)$, $p_W(\cdot)$ and $p_O(\cdot)$, respectively. Of course, $p_S(\cdot)$ coincides with the prior $p(\cdot)$ given in the IHS, while the marginal distributions $p_W(\cdot)$ and $p_O(\cdot)$ can be computed from the given data, $p(\cdot)$, $p(\cdot|\cdot)$ and $q(\cdot|\cdot)$. Indeed, $p_O(o) = \sum_{s\in\mathcal{S}} p(o|s)p(s)$ and $p_W(w) = \sum_{s\in\mathcal{S}} q(w|s)p(s)$. Concerning the conditional distributions, we have $p_{O|S}(o|s) = p(o|s)$ and $p_{W|S}(w|s) = q(w|s)$, whenever $p(s) > 0$. Concerning $p_{O|W}(\cdot|\cdot)$, this can be computed as $p_{O|W}(o|w) = \sum_{s\in\mathcal{S}} r(w,s,o) \cdot p_W^{-1}(w)$ (with the convention that this denotes an arbitrary value, e.g. 0, if $p_W(w) = 0$). It is worthwhile to stress that these marginalisation operations may be costly if the state-space $\mathcal{S}$ is very large, but fortunately it will not be necessary to carry out them explicitly to apply our results.

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, $o^n = (o_1, \ldots, o_n) \in \mathcal{O}^n$, throughout which both the secret state s and the corresponding view w are kept fixed. Formally, the adversary knows a random vector of observations $O^n = (O_1, \ldots, O_n)$ such that, for each $i = 1, \ldots, n$, O_i is distributed like O. Moreover, the individual O_i and the view W are *conditionally independent* given S. This means that the following equality holds true for each $o^n \in \mathcal{O}^n$, $w \in \mathcal{W}$ and $s \in \mathcal{S}$ such that $p(s) > 0$

$$\begin{aligned} &\Pr\big(O^n = (o_1, \ldots, o_n),\, W = w \,|\, S = s\big) \\ &= \Pi_{i=1}^n \Pr(O_i = o_i | S = s) \Pr(W = w | S = s). \end{aligned} \tag{3.12}$$

Note that the right-hand side of the above equality can be equivalently written as $\Pi_{i=1}^n p(o_i|s)q(w|s)$. Concerning the notation, we shall drop the subscripts from the above defined (conditional) probability distributions when no ambiguity can arise. We will often abbreviate $\Pi_{i=1}^n p(o_i|s)$ as

$p(o^n|s)$. 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.

The attacker's strategy this time corresponds to a *W-guessing function*, $g : \mathcal{O}^n \to \mathcal{W}$. The corresponding error probability (after n observations, relative to g) is

$$P_e^{g,W}(n) \triangleq \Pr\left(\, g(O^n) \neq W \,\right). \tag{3.13}$$

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

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

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 S, that is $W = f(S)$ for some function $f : \mathcal{S} \to \mathcal{W}$. Equivalently, the matrix $q(\cdot|\cdot)$ has a single entry '1' for each row. Let $\mathcal{W} = \{w_1, \ldots, w_L\}$, and let $E_i \triangleq f^{-1}(w_i)$ for $1 \leq i \leq L$. Of course $E_1, \ldots, E_L$ form a partition of $\mathcal{S}$, in the set-theoretic sense.

Concerning the information leakage, we extend the definition as follows (here we have considered only the min-entropy leakage).

Definition 3.2.2 (Information leakage with views) *The information leakage after n observations* relative to a view W *is defined as*

$$\mathcal{L}^W(n) \triangleq \log\big(\frac{P_{succ}^W(n)}{\max_w p_W(w)}\big).$$

Example 17 *Consider again the two programs* `P1` *and* `P2` *seen in the introductive part of this section. Since they are deterministic, a single observation is all the attacker needs. One easily finds that the error probability equals 0 in the case of* `P1`*, and $\frac{1}{16}$ in the case of* `P2`*, while the information leakage, related to the partition $(\{0\}, \mathcal{S} \setminus \{0\})$, equals 4 in the case of* `P1`*, and 2 in the case of* `P2`*. Thus, for any $n \geq 1$, $\mathcal{L}^W(n) = 4$ for* `P1` *and $\mathcal{L}^W(n) = 2$ for* `P2`*.*

In the general case of probabilistic systems, computation of these limit values is not as obvious. Nevertheless, in the next section we will offer results that allow one to easily characterise its behaviour from the channel matrices. In particular, we will show how to determine the limit value and its rate.

3.2.2 Bounds and asymptotic behaviour

Concerning the analysis of the asymptotic behaviour of P_e^W, it would be tempting to proceed as follows: build a new IHS, say $\mathcal{H}^W$, where the set of states is $\mathcal{W}$ and the channel matrix is $p_{O|W}(\cdot|\cdot)$. The error probability function for $\mathcal{H}^W$ would then coincide with $P_e^W(n)$. It would then be enough to apply Theorem 3.1.14 to $\mathcal{H}^W$. This approach however is doomed to failure. In fact, the assumption that the observations O_i's are conditionally independent given W is in general false:

$$p(o_1 \cdots o_n|w) \neq p(o_1|w) \cdots p(o_n|w)\,.$$

As a consequence, the IHS $\mathcal{H}^W$ is meaningless for what concerns our purposes. However, conditional independence of the O_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 the method illustrated below for P_e^W in the general case.

Before describing the model, let us introduce some more notation. For sake of simplicity, we assume $\mathcal{W}$ is a set of integers $\{1, \ldots, |\mathcal{W}|\}$. Let $q(\cdot|\cdot)$ be the matrix defining the view W. We denote by $\sim_W$ the equivalence relation on $\mathcal{S}$ induced by $q(\cdot|\cdot)$.

Definition 3.2.3 (Indistinguishability relation with views) *Given* $s, s' \in \mathcal{S}$

$$s \sim_W s' \qquad \textit{if and only if} \qquad \textit{for each } o \in \mathcal{O}:\ q(o|s) = q(o|s')\,. \tag{3.14}$$

In other words, two states are $\sim_W$-equivalent if the corresponding rows of $q(\cdot|\cdot)$ are equal. Let $\mathcal{S}/\sim_W$ be $\{E_1, \ldots, E_L\}$, the set of equivalence classes of $\sim_W$. The intersection $\equiv \cap \sim_W$ is still an equivalence relation on $\mathcal{S}$, that is finer than both $\equiv$ and $\sim_W$. Recall that $\mathcal{S}/\equiv$ is $\{C_1, \ldots, C_K\}$ and that $s_1^*, \ldots, s_K^*$ denote the representative elements of these equivalence classes. 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

$$F_{ij} \triangleq C_i \cap E_j \tag{3.15}$$

and furthermore

$$F_i^* \triangleq \max_j p_S(F_{ij}) \quad \text{and} \quad q_j^* \triangleq \max_w q(w|s), \text{ for an arbitrary } s \in E_j\,. \tag{3.16}$$

The next theorem has the following interpretation. The attacker focuses on a subset of the representative states, $\{s_i^*|i \in I\}$. He tries to identify

first the class C_i of S, then guesses the class F_{ij} – this is given by the j that maximises $p_S(F_{ij})$. 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 3.2.4 *Let I and ρ_I be chosen as in Theorem 3.1.14. Let W be a view of $\mathcal{H}$. Let $F_{\max} = \max_{i\in I} F_i^*$. Then*

$$P_e^W(n) \;\leq\; \sum_{j=1}^{L}(1-q_j^*) \;+\; (1-\sum_{i\in I} F_i^*) \;+\; \frac{|I|^2}{2}F_{\max}2^{-n\rho_I}\,. \qquad (3.17)$$

PROOF Denote a pair of indices $(i,j) \in \{1,\ldots,K\}\times\{1,\ldots,L\}$ as ij. For each $s \in \mathcal{S}$, define

$$\mathrm{ind}(s) = ij \quad \text{if and only if} \quad s \in F_{ij}.$$

Fix $n \geq 1$ and any function $g' : \mathcal{O}^n \to \{1,\ldots,K\}\times\{1,\ldots,L\}$, and let $Succ'$ be the event $(g'(O^n) = \mathrm{ind}(S))$. That is, $Succ'$ is the event that g' correctly classifies the index (of the equivalence class F_{ij}) of S. Now define a W-guessing function for $\mathcal{H}$, $g : \mathcal{O}^n \to \mathcal{W}$, as $g(o^n) \triangleq w$, where $g'(o^n) = ij$ and $w = \mathrm{argmax}_w q(w|s)$ for any $s \in E_j$ (note that the information about i provided by g' is ignored by g). Let Err be the event $(g(O^n) \neq W)$. We have

$$\begin{aligned} P_e^W(n) &= \Pr(Err, Succ') + \Pr(Err|\neg Succ')\Pr(\neg Succ') \\ &\leq \Pr(Err, Succ') + \Pr(\neg Succ')\,. \end{aligned} \qquad (3.18)$$

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

$$\begin{aligned} \Pr(Err, Succ') &= \sum_{j=1}^{L}(1-q_j^*)\Pr(S\in E_j, Succ') \\ &\leq \sum_{j=1}^{L}(1-q_j^*)\,. \end{aligned} \qquad (3.19)$$

We now estimate $\Pr(\neg Succ')$. Consider the new IHS $\mathcal{H}' \triangleq \langle\{1,\ldots,K\}\times\{1,\ldots,L\}, \mathcal{O}, p'(\cdot), p'(\cdot|\cdot)\rangle$, where $p'(ij) \triangleq p_S(F_{ij})$ and $p'(o|ij) \triangleq p_i(o)$. Note that $ij \equiv i'j'$ if and only if $i = i'$. Hence we have K distinct classes in this system, whose representatives are elements $s'_1 =$

$1j_1, \ldots, s'_K = Kj_K$ such that $j_i = \text{argmax}_j p_S(F_{ij})$, hence $p'(s'_i) = F^*_i$, for $i = 1, \ldots, K$. The corresponding representative distributions (rows of the matrix $p'(\cdot|\cdot)$) are $p'_1(\cdot) = p_1(\cdot), \ldots, p'_K(\cdot) = p_K(\cdot)$.

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 $I \subseteq \{1, \ldots, K\}$ and apply Theorem 3.1.14 to $\mathcal{H}'$ and I to get

$$\Pr(\neg Succ') \leq 1 - \sum_{i \in I} F^*_i \; + \; \frac{|I|^2}{2} F_{\max} 2^{-n\rho_I} \, . \tag{3.20}$$

When we plug the bounds (3.19) and (3.20) into (3.18), we get the wanted result. □

Note that the determination of the upper-bound in (3.17) 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 (3.15), and the probability mass of the corresponding classes $p_S(F_{ij})$, are then straightforward to compute. Theorem 3.2.4 only provides an (exponential) upper bound to $P^W_e(n)$. The following theorem provides the exact limit of $P^W_e(n)$ in the special, but important case when W is a partition.

We will make use of some concepts of the Method of Types from Information Theory (see Section 2.4). Recall that, given a sequence $o^n \in \mathcal{O}^n$ and $o \in \mathcal{O}$, $n(o, o^n)$ denotes the number of occurrences of o inside o^n and $t_{o^n}(o) \triangleq n(o, o^n)/n$, for each $o \in \mathcal{O}$, is the *type* of o^n. The "balls" of center $p_i(\cdot)$ and radius $\varepsilon > 0$ in $\mathcal{O}^n$ are defined as $U^n_i(\varepsilon) \triangleq \{o^n \, : \, D(t_{o^n} \| p_i) \leq \epsilon\}$. It is a result from the Method of Types that, as $n \to +\infty$, $p_i(U^n_i(\varepsilon)) \to 1$, while, for any $p \neq p_i$ there is $\varepsilon > 0$ small enough such that $p(U^n_i(\varepsilon)) \to 0$ (Theorem 2.4.8). Moreover, the convergence is exponential in both cases.

Theorem 3.2.5 *Let W be a partition of $\mathcal{H}$. Then $P^W_e(n)$ converges exponentially fast to $1 - \sum_{i=1}^K F^*_i$. More precisely, with the same notation of Theorem 3.2.4, for each $n \geq 1$,*

$$(1 - \sum_{i=1}^K F^*_i) \; \leq \; P^W_e(n) \; \leq \; (1 - \sum_{i=1}^K F^*_i) + \frac{K^2}{2} F_{\max} 2^{-n\rho_I},$$

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

PROOF First, note that for W a partition, the first term in (3.17) vanishes, as each q^*_j equals 1. The upper bound is then a consequence of

Theorem 3.2.4 with $I = \{1, \ldots, K\}$. 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 $\mathcal{W} = \{1, \ldots, L\}$. For any $n \geq 1$, let $g : \mathcal{O}^n \to \{1, \ldots, L\}$ be a W-MAP guessing function, and let $A_j = g^{-1}(j)$, for $j \in \{1, \ldots, L\}$, be the acceptance region in $\mathcal{O}^n$ for j. It is a routine task to check that

$$P_{succ}^W(n) \;=\; \sum_{i=1}^{K} \sum_{j=1}^{L} p_i(A_j) p_S(F_{ij})\,. \tag{3.21}$$

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

$$\begin{aligned} p_{O^n|W}(o^n|j_i) p_W(j_i) \;&=\; \sum_{l=1}^{K} p_l(o^n) p_S(F_{lj_i}) \;> \\ > \;\sum_{l=1}^{K} p_l(o^n) p_S(F_{lj}) \;&=\; p_{O^n|W}(o^n|j) p_W(j)\,. \end{aligned} \tag{3.22}$$

Now from results of the method of types it follows that, for $o^n \in U_i^n(\varepsilon)$, we have that all the $p_l(o^n)$ with $l \neq i$ go exponentially fast to 0 as n grows. Thus the condition (3.22) reduces, for n large enough, to $F_i^* = p_S(F_{ij_i}) > p_S(F_{ij})$: this is satisfied by definition of j_i[7]. Now $A_{j_i} \supseteq U_i^n(\varepsilon)$ 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 $\mathcal{O}^n$). This way, and taking (3.21) into account, we have proved that

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

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

[7] If there is more than one index j maximising $p_i(F_{ij})$, then the choice of j_i gets more involved: among those j's that maximise $p_S(F_{ij})$, one chooses the one that maximises $p_S(F_{i'j})$, 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.

3.2.3 Some applications

Modular exponentiation algorithm Concerning the modular exponentiation algorithm, it can be interesting to prove that its small leakage is not concentrated in few individual bits of the exponent, which would make them potentially vulnerable. For instance, assume that the attacker now no longer targets the whole exponent, but just its least two significant bits. Let us examine the error probability in this case.

Let W be the partition of $\mathcal{S}$ such that $s \sim_W s'$ if and only if $s \bmod 4 = s' \bmod 4$. We apply Theorem 3.2.5 to P_e^W. We have four $\sim_W$-classes $E_0, \ldots, E_3$, that intersect with the $N+1$ classes C_i (computed in Example 13) to form $4(N+1)$ classes F_{ij}. Assume N even. For all $i = 0, \ldots, \frac{N-2}{2}$, the class F_{ij} that has more elements, hence determines the probability F_i^*, is F_{i0}; by symmetry, for $i = \frac{N-2}{2} + 1, \ldots, N$ the class with more elements is F_{i3}. For $i = \frac{N}{2}$, instead, we can choose between F_{i1} and F_{i2}. According to Theorem 3.2.5 then

$$P_{succ}^W(n) \;\to\; \sum_{i=0}^{N} F_i^* \;\approx\; \frac{1}{2^N}\Big(\sum_{i=0}^{N-2} \binom{N-2}{i}\Big) \;=\; \frac{1}{4}\,.$$

Thus, asymptotically the observations do not increase the prior probability of success, which is already $\frac{1}{4}$. In terms of information leakage, one gets $\mathcal{L}^W(n) \to\approx\; 0$. One can generalise this reasoning to the case where W represents the least m significant bits, and arrive at similar conclusions.

Unlinkability in Threshold Mixnets Let us consider again the application of Threshold Mixnets, discussed in Section 3.1, Examples 12,15. 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:

$$W = 1 \text{ if and only if } \; S = R_j\,.$$

That is, W yields 1 if and only if Bob is R_j. The partition induced on $\mathcal{S}$ by W coincides with $\equiv$, hence its classes are $C_1 = \{R_j\}$ and $C_2 = \mathcal{S} \setminus \{R_j\}$. (see Example 15). Concerning the sets F_{ij}, we note that: $F_{11} = \{R_j\},\ F_{12} = F_{21} = \emptyset$ and $F_{22} = \mathcal{S} \setminus \{R_j\}$. 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 3.2.4, the limit of $P_e^W(n)$

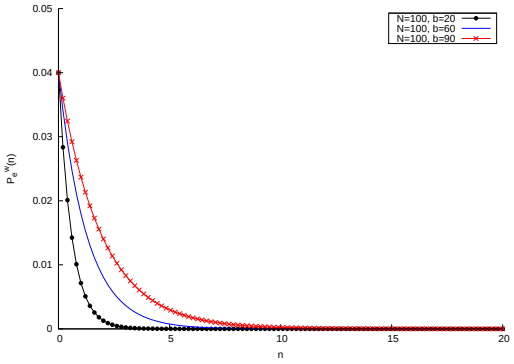

Figure 7: Plots of $P_e^W(n)$ depending on parameter b.

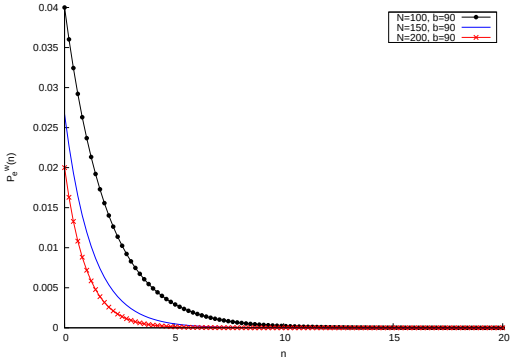

Figure 8: Plots of $P_e^W(n)$ depending on parameter N.

vanishes, moreover

$$P_e^W(n) \;\leq\; \frac{2}{N}\left(1-\left(1-\frac{1}{N}\right)^b\right)^n \;\leq\; \frac{2}{N}e^{-ne^{-\frac{b}{N}}}\,,$$

where the last inequality follows from the double application of the inequality $1-x \leq e^{-x}$ for $x < 1$. 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 Figures 7,8: very few rounds of the protocols ($n < 10$) are sufficient to achieve $P_e^W \approx 0$.

As mentioned in Example 12, it is easy to repeat this kind of analysis with more sophisticated observations on the part of the attacker. On the other hand, note that just repeating this simple attack for each of the potential Alice's receivers (that is, setting $R_j = R_1, R_2, \ldots, R_{N-1}$ in turn), would lead the attacker to discover the identity of Bob after a low number of rounds. This is sufficient to show that the single threshold mix system is totally insecure.

3.3 Concluding remarks

We have characterised the asymptotic behaviour of error probability and information leakage in terms of indistinguishability in a scenario of one-try attacks after repeated independent, noisy observations. Assuming that each execution gives rise to a single observation and that we are faced with a passive attacker, we have first examined the case where the adversary directly targets the secret and then extended our results to the case where he targets some state predicates. We have analysed in a uniform fashion a variety of statistical attacks, allowing for the assessment of systems both at the global level and at the level of specific partitions of the secrets. Our results generalise the lower bound presented in (CPP08b, Proposition 7.4). In particular, we give precise bounds for the probability of misclassification on the part of the attacker, characterising both the limit value and the rate of convergence of the error probability as a function of the number of independent observations.

Our study on views relates, at least conceptually, to the notion of probabilistic *opacity*, as studied by Bérard, Mullins and Sassolas in (BMS10). Although they work with a different setting, finite-state machines, our partition can be seen as a generalisation of the binary predicates they consider.

Chapter 4

Passive attackers and sequential observations

The attack models discussed in the preceding chapter presupposes that the computation involving the secret information takes place in a single step. Or, more accurately, that the intermediate states of the computation are not accessible to the adversary. In this chapter, we consider a more refined scenario, where computations may take several steps to terminate, or even not terminate at all. In any case, to each state of the computation there corresponds one observation on the part of the attacker. Hence, to each computation there corresponds a sequential *trace* of observations. The attacker may collect multiple such traces, corresponding to multiple independent executions of the system. Throughout these executions, the secret information is kept fixed. This set up is well suited to describe situations where the attacker collects information from different sources at different times, like a coalition of different local eavesdroppers. Discrete-time Hidden Markov Models (Rab89) provide a convenient setting to formally model such systems, which we may designate as *sequential* information hiding systems.

4.1 Hidden Markov Models

Let $\mathcal{S}$ and $\mathcal{O}$ be finite sets of states and observations, respectively.

Definition 4.1.1 (Hidden Markov Models) *A (discrete-time, homoge-*

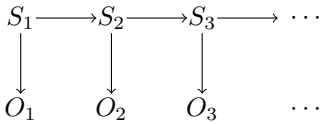

Figure 9: A graphical representation of a sequential IHS.

neous) Hidden Markov Model *(*HMM*) with states in $\mathcal{S}$ and observations in $\mathcal{O}$ is a a pair of random processes $\langle (S_i)_{i\geq 1}\,,\,(O_i)_{i\geq 1}\rangle$, such that, for each $t \geq 1$*

- *S_t and O_t are random variables taking values in $\mathcal{S}$ and $\mathcal{O}$, respectively;*
- *the following equalities hold true (whenever the involved conditional probabilities are defined):*

$$\begin{aligned}&\Pr(S_{t+1} = s_{t+1}|S_t = s_t, O_t = o_t, \ldots, S_1 = s_1, O_1 = o_1) = \\ &= \Pr(S_{t+1} = s_{t+1}|S_t = s_t) \qquad (4.1)\\ &\Pr(O_t = o_t|S_t = s_t, S_{t-1} = s_{t-1}, O_{t-1} = o_{t-1}, \ldots, S_1 = s_1, O_1 = o_1) = \\ &= \Pr(O_t = o_t|S_t = s_t). \qquad (4.2)\end{aligned}$$

Moreover, the value of the above probabilities does not depend on the index t, but only on s_t, s_{t+1} and o_t.

Equation (4.1) says that the state at time $t+1$ only depends on the state at time t, that is $(S_i)_{i\geq 1}$ forms a Markov chain. Equation (4.2) says that the observation at time t only depends on the state at time t. A consequence of this equation is that the state at time $t+1$ is independent from the observation at time t, given the state at time t, that is

$$\Pr(O_t = o_t, S_{t+1} = s_{t+1}|S_t = s_t) \;=\; \Pr(O_t = o_t|S_t = s_t)\cdot\Pr(S_{t+1} = s_{t+1}|S_t = s_t)\,. \qquad (4.3)$$

Graphically, a HMM can be represented by a diagram like the one in Figure 9, where the nodes are random variables and the presence of a pair of arrows $X \leftarrow Y \rightarrow Z$ or $X \rightarrow Y \rightarrow Z$ means conditional independence of X and Z given Y.

Assume now $\mathcal{S} = \{s_1, \ldots, s_m\}$ and $\mathcal{O} = \{o_1, \ldots, o_l\}$. A finite-state HMM on $\mathcal{S}$ and $\mathcal{O}$ is completely specified by, hence can be identified with, a triple (π, F, G) such that:

- $\pi \in \mathbb{R}^{1\times m}$ is a row-vector representing the prior distribution on $\mathcal{S}$, that is $\pi(i) = \Pr(S_1 = s_i)$ for each $1 \leq i \leq m$;
- $F \in \mathbb{R}^{m\times m}$ is a matrix such that $F(i,j)$ is the probability of transition from s_i to s_j, for $1 \leq i,j \leq m$;
- $G \in \mathbb{R}^{m\times l}$ is a matrix such that $G(i,j)$ is the probability of observing o_j at state s_i, for $1 \leq i \leq m$ and $1 \leq j \leq l$.

4.2 An extended model: sequential observations

Given an HMM $\langle (S_i)_{i\geq 1}\,,\,(O_i)_{i\geq 1}\rangle$, with states in $\mathcal{S}$ and observations in $\mathcal{O}$, where $(S_i)_{i\geq 1}$ represents the sequence of (hidden) states crossed by the system, while $(O_i)_{i\geq 1}$ is the corresponding observation trace, assume that the attacker targets the first state of the computation, that is the value of S_1 (lightly modifying the model we can also represent a situation where the adversary targets the whole sequence of states the system passed through before terminating). We are interested in analysing the attacker's error probability after observing n traces of length t, corresponding to n conditionally independent executions of the system up to and including time t, as both n and t go to $+\infty$. Let σ denotes a sequence of observation (or trace) corresponding to an execution of the system, ranging over the set of observation traces $\mathcal{O}^*$. For any $\sigma = o_1 \cdots o_t$ ($t \geq 0$) and $s \in S$, define[1]

$$p(\sigma \mid s) \triangleq \Pr(O_1 = o_1, O_2 = o_2, \ldots, O_t = o_t \mid S_1 = s)$$

with the proviso that $p(\varepsilon \mid s) \triangleq 1$. We note that for any fixed $t \geq 0$ and $s \in \mathcal{S}$, $p(\sigma|s)$ defines a probability distribution as σ ranges over $\mathcal{O}^t$, the set of traces of length t, or *t-traces*. In other words, for *any fixed* t, we have an information hiding system in the sense of Section 3.1.1, with $\mathcal{S}$ as a set of states, $\mathcal{O}^t$ as a set of observables, a conditional probability matrix $p(\sigma|s)$ ($s \in \mathcal{S}, \sigma \in \mathcal{O}^t$) and p^* as a prior distribution on states. Call $\mathcal{H}^{(t)}$ this system. We have the following error probabilities of interest ($t \geq 0$):

[1]Or, more formally, $p(\sigma \mid s) \triangleq \Pr(O_h = o_1, O_{h+1} = o_2, \ldots, O_{h+t-1} = o_t \mid S_h = s)$, for any index h s.t. $\Pr(S_h = s) > 0$. Note that this definition does not depend on the chosen index h, given that the chain is homogeneous. Also, we are assuming w.l.o.g. here that for each s there is an index h s.t. $\Pr(S_h = s) > 0$.

Definition 4.2.1 (Error probabilities and leakage for HMM)

$$\begin{aligned} P_e^{(t)}(n) &\triangleq \textit{error probability after } n \textit{ observations (of } t\textit{-traces) in } \mathcal{H}^{(t)} \\ P_e^{(t)} &\triangleq \lim_{n\to\infty} P_e^{(t)}(n) \qquad (4.4) \\ P_e &\triangleq \lim_{t\to\infty} P_e^{(t)} \qquad (4.5) \end{aligned}$$

The corresponding min-entropy leakage quantities ($P_{succ} = 1 - P_e$) are:

$$\mathcal{L}^{(t)}(n) \triangleq \log \frac{P_{succ}^{(t)}(n)}{\max_s p^*(s)} \qquad \mathcal{L}^{(t)} \triangleq \log \frac{P_{succ}^{(t)}}{\max_s p^*(s)} \qquad \mathcal{L} \triangleq \log \frac{P_{succ}}{\max_s p^*(s)}.$$

Additive leakages are defined similarly.

We will show in the next section that the limits (4.4) and (4.5) exist and are easy to compute.

4.3 Bounds and asymptotic behaviour

The existence of limit (4.4) is an immediate consequence of Theorem 3.1.7 applied to $\mathcal{H}^{(t)}$. Indeed, let us denote by $\equiv^{(t)}$ the indistinguishability relation on states for $\mathcal{H}^{(t)}$, that is, explicitly

$$s \equiv^{(t)} s' \quad \text{if and only if for each } \sigma \in \mathcal{O}^t : p(\sigma|s) = p(\sigma|s') .$$

Let $C_1^{(t)}, \ldots, C_{K_t}^{(t)}$ be the equivalence classes of $\equiv^{(t)}$ and let $p_i^{*(t)} \triangleq \max_{s\in C_i^{(t)}} p^*(s)$. Then we have by Theorem 3.1.7 that

$$P_e^{(t)} \;=\; 1 - \sum_{i=1}^{K_t} p_i^{*(t)} \qquad (4.6)$$

Note that, for any fixed t, Corollary 3.1.11 carries over to $\mathcal{H}^{(t)}$. We now consider the case $t \to \infty$. We introduce the following fundamental relation.

Definition 4.3.1 (Indistinguishability for HMM) *The* indistinguishability relation *on a* HMM *is defined as*

$$\equiv \;\triangleq\; \bigcap_{t\geq 0} \equiv^{(t)} .$$

Equivalently, $s \equiv s'$ if and only if for every $\sigma \in \mathcal{O}^*$, $p(\sigma|s) = p(\sigma|s')$.

It is immediate to check that $\equiv$ is an equivalence relation on $\mathcal{S}$. Let $C_1, \ldots, C_K$ be its equivalence classes and let $p_i^* \triangleq \max_{s \in C_i} p^*(s)$, for $i = 1, \ldots, K$. Concerning the limit (4.5), we have the following result.

Proposition 4.3.2

$$P_e \;=\; 1 - \sum_{i=1}^{K} p_i^* .$$

PROOF First, we note that $\{\equiv^{(t)}\}_{t \geq 0}$ forms a monotonically non-increasing chain of relations: $\equiv^{(0)} \supseteq \equiv^{(1)} \supseteq \equiv^{(2)} \;\cdots$. To prove this fact, note that, for each t, $\sigma \in \mathcal{O}^t$ and $s \in \mathcal{S}$, $p(\sigma|s) = \sum_{o \in \mathcal{O}} p(\sigma \cdot o|s)$. Therefore, $s \equiv^{(t+1)} s'$ implies $s \equiv^{(t)} s'$.

The above fact implies that the sequence $\{P_e^{(t)}\}_{t \geq 0}$ is monotonically non-increasing: indeed, the finer the equivalence classes of $\equiv^{(t)}$, the greater the value of the sum in (4.6). Therefore, the limit (4.5) exists. In order to determine the value of this limit, we reason as follows. Since $\mathcal{S}$ is finite and the chain of sets $\{\equiv^{(t)}\}_{t \geq 0}$ is monotonically non-increasing, there must exist t_0 such that

$$\equiv^{(t_0)} \;=\; \equiv^{(t_0+1)} \;=\; \cdots \;=\; \equiv \;.$$

According to (4.6) then, from t_0 onward the sequence $\{P_e^{(t)}\}_{t \geq 0}$ stabilises to the value $P_e \;=\; 1 - \sum_{i=1}^{K} p_i^*$. □

The actual computation of P_e, and of the corresponding information leakage quantities, is therefore reduced to the computation of $\equiv$. Below, we show that this computation can be performed quite efficiently. We do so by using some elementary linear algebra. Let us introduce some additional notation. We define the transition matrices $M_{o_k} \in \mathbb{R}^{m \times m}$, for any $o_k \in \mathcal{O}$, as follows[2]:

$$\begin{array}{rcl} M_{o_k}(i,j) & \triangleq & \Pr(S_{t+1} = s_j, O_t = o_k | S_t = s_i) \\ & = & F(i,j) \cdot G(i,k) \end{array}$$

where the last equality is justified by equation (4.3). Note that a row of M_{o_k} does not necessarily sum to 1. For any $\sigma = o_1 \cdots o_t$, let

$$M_\sigma \triangleq M_{o_1} \times \cdots \times M_{o_t} .$$

[2]Again, due to homogeneity, in the definition we can choose any index t such that $\Pr(S_t = s_i) > 0$.

Finally, we let $e_i \in \mathbb{R}^{1\times m}$ denote the row vector with 1 in the i-th position and 0 elsewhere and let $e \triangleq \sum_{i=1}^{m} e_i$ denote the everywhere 1 vector. The following lemma provides an alternative characterization of $\equiv$; the lemma is easily proven by induction on the length of σ.

Lemma 4.3.3 *For each σ and $s_i \in \mathcal{S}$, $p(\sigma \mid s_i) = e_i M_\sigma e^T$. Hence*

$$s_i \equiv s_j \quad \textit{if and only if} \quad \textit{for each } \sigma \in \mathcal{O}^* \;\; e_i M_\sigma e^T = e_j M_\sigma e^T .$$

We say a row vector v is *orthogonal* to a set of column vectors U, written $v \perp U$, if $vu = 0$ for each $u \in U$. Also, for any set of vectors U, $U^\perp$ denotes the orthogonal complement of U given by $U^\perp = \{v \mid v \perp U\}$. It is easily seen that $U^\perp$ is a sub-space of the space of column vectors. Moreover, $U \subseteq V$ implies $V^\perp \subseteq U^\perp$. Of course, the above definitions extend as expected when exchanging the roles of "row" and "column". We finally note that if U is a vector space, then $(\cdot)^\perp$ is an involution, that is $(U^\perp)^\perp = U$.

Theorem 4.3.4 *Let B be a basis of the (finite-dimensional) sub-space of $\mathbb{R}^{m\times 1}$ spanned by $\bigcup_{\sigma\in\mathcal{O}^*}\{M_\sigma e^T\}$. For $s_i, s_j \in \mathcal{S}$,*

$$s_i \equiv s_j \qquad \textit{if and only if} \qquad (e_i - e_j) \perp B.$$

PROOF The condition of Lemma 4.3.3 can be expressed as

$$\begin{gathered}\text{for each } \sigma \in \mathcal{O}^* : \; (e_i - e_j) M_\sigma e^T = 0 \\ \text{if and only if} \\ (e_i - e_j) \in \cap_\sigma \{M_\sigma e^T\}^\perp = (\cup_\sigma \{M_\sigma e^T\})^\perp \\ \text{if and only if} \\ (e_i - e_j) \perp B\,.\end{gathered}$$

□

A basis B of span$\big(\bigcup_\sigma \{M_\sigma e^T\}\big)$ can be expressed as

$$B = \{M_\sigma e^T \mid \sigma \in \mathcal{F}\} \tag{4.7}$$

for a suitable finite, prefix-closed $\mathcal{F} \subseteq \mathcal{O}^*$. More precisely, B can be computed by a procedure that starts with the set $B := \{e^T\}$ and iteratively updates B by joining in the vectors $M_{o\cdot\sigma} e^T = M_o \cdot (M_\sigma e^T)$, with

$M_\sigma e^T \in B$ and $o \in \mathcal{O}$, that are linearly independent from the vectors already present in B, until no other vector can be joined in. This procedure must terminate in a number of steps $\leq m$. The set of strings $\mathcal{F}$ can be computed alongside with B.

We now briefly discuss the rate of convergence to P_e. We have already seen that $P_e^{(t_0)} = P_e$. Therefore, there is no advantage, for an attacker wanting to determine $\equiv$, in considering traces of length greater than t_0. The convergence rate for the attacker is hence determined by the matrix of the system $\mathcal{H}^{(t_0)}$. For this reason, it is of practical importance to be able to compute t_0. This is in fact quite an easy task, as stated by the following proposition.

Proposition 4.3.5 *Let B be a basis of the space spanned by $\cup_\sigma \{M_\sigma e^T\}$ and $\mathcal{F}$ the corresponding set of strings, as specified by (4.7). Assume B and $\mathcal{F}$ have been obtained by the algorithm described above. Then $t_0 = \max\{|\sigma| \,:\, \sigma \in \mathcal{F}\}$.*

PROOF For any equivalence relation R over $\mathcal{S}$, let the *kernel* of R be the subspace of $\mathbb{R}^{1\times m}$ defined thus

$$\ker(R) \triangleq \mathrm{span}(\{e_i - e_j \,|\, s_i \, R \, s_j\}) \,.$$

Now, by a reasoning similar to that in the proof of Theorem 4.3.4, for any t we have

$$\ker(\equiv^{(t)}) \;=\; (\mathrm{span}(\cup_{\sigma\in\mathcal{O}^t} M_\sigma e^T))^\perp \tag{4.8}$$

while, by definition of B and $\mathcal{F}$

$$\ker(\equiv) \;=\; (\mathrm{span}(\cup_{\sigma\in\mathcal{F}} M_\sigma e^T))^\perp \,. \tag{4.9}$$

Let R, R' be two equivalence relations of the form $\equiv$ or $\equiv^{(t)}$. The above equations imply that $s_i R s_j$ if and only if $e_i - e_j \in \ker(R)$. Moreover, $R \subseteq R'$ if and only if $\ker(R) \subseteq \ker(R')$. Thus, the equivalence relations of interest are completely characterised by their kernels. By Lemma 4.3.3, we deduce that for each t, $\ker(\equiv^{(t)}) \supseteq \ker(\equiv^{(t+1)})$. From this fact, and using the fact that $U \subseteq V$ implies $V^\perp \subseteq U^\perp$, and that $(U^\perp)^\perp = U$, we obtain that for each t, $\mathrm{span}(\cup_{\sigma\in\mathcal{O}^t} M_\sigma e^T) \subseteq \mathrm{span}(\cup_{\sigma\in\mathcal{O}^{t+1}} M_\sigma e^T)$, hence

$$\ker(\equiv^{(t)})^\perp \;=\; \mathrm{span}(\cup_{\sigma\in\mathcal{O}^t} M_\sigma e^T) \;=\; \mathrm{span}(\cup_{0\leq i\leq t} \cup_{\sigma\in\mathcal{O}^i} M_\sigma e^T) \,.$$

Take now $t = \max\{|\sigma| \,:\, \sigma \in \mathcal{F}\}$ in the equation above: we obtain

$$\ker(\equiv^{(t)})^\perp \;=\; \mathrm{span}(\cup_{0\leq i\leq t}\cup_{\sigma\in\mathcal{O}^i} M_\sigma e^T) \;\supseteq\; \mathrm{span}(\cup_{\sigma\in\mathcal{F}} M_\sigma e^T) \;=\; \ker(\equiv)^\perp$$

hence $\ker(\equiv^{(t)}) \subseteq \ker(\equiv)$, which implies $\ker(\equiv^{(t)}) = \ker(\equiv)$, that is $\equiv^{(t)} = \equiv$.

On the other hand, take any $t < \max\{|\sigma| : \sigma \in \mathcal{F}\}$. Assume by contradiction that $\equiv^{(t)} = \equiv$, that is $\ker(\equiv^{(t)}) = \ker(\equiv)$. By (4.8) and (4.9), and using $(U^{\perp})^{\perp} = U$, we obtain that $\mathrm{span}(\cup_{\sigma\in\mathcal{O}^t} M_\sigma e^T) = \mathrm{span}(\cup_{\sigma\in\mathcal{F}} M_\sigma e^T)$. This implies that there is a string of maximal length in $\mathcal{F}$, say σ_0, s.t. $M_{\sigma_0} e^T$ can be obtained as a linear combination of vectors $M_\sigma e^T$, for σ of length $t < |\sigma_0|$. But, by construction of B and $\mathcal{F}$, this cannot be the case. □

The practical computation of the rate relative to P_e can be carried out applying Theorem 3.1.7 to the system $\mathcal{H}^{(t_0)}$, which requires one has at hand the conditional probability matrix of the system. The entries of this matrix are of the form $p(\sigma|s)$ with $\sigma \in \mathcal{O}^{t_0}$. The computation of individual entries $p(\sigma|s)$ can be performed quite efficiently, running the so-called *Forward-Backward algorithm* on the underlying HMM (see (Rab89)). Unfortunately, the number of columns in the matrix, i.e. of traces of length t_0, is exponential in t_0. Most likely, this makes the exact computation of the rate impractical for significant systems (say, systems with thousands of states). Forms of approximations are conceivable to tackle this problem, such as "lumping" the matrix by aggregating sets of columns: this leads to tractable dimensions, but also to underestimating the rate. We will not discuss this issue further.

Remark 4.3.6 *Model checking of Markov chains is based on viewing properties to analyse as sets of* infinite *sequences of states. One could adopt a similar perspective when analysing* HMM*'s from the point of view of information leakage, and stipulate that an* observable *is a set of infinite sequences $\mathcal{P} \subseteq \mathcal{O}^\omega$, taken from a cylinder-generated sigma-algebra (see e.g. (BK08)). However, this approach would not lead to substantially different results. Indeed, the probability measures defined on the sigma-algebra entirely depend on the probability assigned to cylinders, which is in turn determined by the probability of the* finite *prefixes $\sigma \in \mathcal{O}^*$ that define the cylinders. Therefore, even in this seemingly richer setting of observations, one would end up having that $\equiv$ coincides with $\equiv^{(t_0)}$.*

4.4 An application: analysing routing information

We discuss now a scenario where messages are routed from a sender to a receiver in a network with a fixed topology, as can be found for instance in a structured peer-to-peer overlay. Anonymity protocols such as *onion routing* (GRS96) are designed to protect the identity of the sender and/or of the receiver in the presence of corrupted nodes. Initially, the routing path from the sender to the receiver is established randomly. In each exchanged message, nested layers of encryptions ensure that any intermediate node on the path node only gets to know the preceding and the next node in the path, but not the identity of the original sender and of the final receiver.

We present and analyse a model of this protocol below. We should warn the reader that, for the sake of presentation, we have chosen to analyse an over-simplified version of the protocol. For example, we assume that, upon receiving a message, a corrupted node can tell whether the message pertains to the target sender-receiver conversation, but cannot identify the predecessor node in the path followed by the message. More powerful forms of eavesdropping can be easily accommodated. Again, we are interested in the case of re-execution, where, for some reason, the initiator is forced to establish new paths with the responder several times. We will concentrate on the asymptotic error probability and leakage, ignoring issues related to the rate of convergence.

We assume the topology of the network is specified by a nonempty graph $\mathcal{G} = (V, E)$. For each node $v \in V$, we let $N(v)$ denote the set of neighbours of v, that is the set of nodes u for which an arc $\{v, u\}$ in E exists; $N(v)$ is always assumed nonempty. Let $C \subseteq V$ represent the subset of corrupted nodes. We let $\mathcal{S} \triangleq V \times V$ be the set of states of the system; in $(v, r) \in \mathcal{S}$, v represents the node currently holding the message, while r represent the final receiver. We let $\mathcal{O} \triangleq C \cup \{*\}$ be the set of observables; here $c \in C$ means that the message is presently held by the corrupted node c, while $*$ means no observation other than the elapse of a discrete time interval. What the attacker can observe are therefore traces like σ in the picture in Figure 10.

We assume the sender and the receiver are chosen at random independently from each other, and that the sender is always a honest node, as there is no point for the attacker in eavesdropping on corrupted nodes. This formally means that the first state of the Markov chain is a random

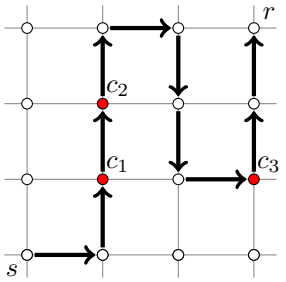

Figure 10: A random route from s to r in a network with three corrupted nodes, and the corresponding observation σ.

vector $S_1 = (S, R)$, where S and R are independent random variables taking values uniformly in $V \setminus C$ and V, respectively. The transitions and the observations of the hidden Markov model are defined by the following equations, where $u, v, r \in V$, $c \in C$ and $s \in V \setminus C$. The first two lines define the entries of matrix F, while the others are the entries of G:

$$
\begin{aligned}
p\big((u,r)\,|\,(v,r)\big) &\triangleq \begin{cases} \frac{1}{|N(v)|} & \text{if } u \in N(v) \text{ and } v \neq r \\ 0 & \text{if } u \notin N(v) \text{ and } v \neq r \end{cases} \\
p\big((r,r)\,|\,(r,r)\big) &\triangleq 1 \\
p\big(c\,|\,(c,r)\big) &\triangleq 1 \\
p\big(*\,|\,(s,r)\big) &\triangleq 1\,.
\end{aligned}
$$

The above equations define a hidden Markov model, say $\mathcal{M}$. For any specific topology $\mathcal{G}$, it is easy to compute the corresponding probability P_e defined by (4.5), as indicated by Proposition 4.3.2. Recall that P_e is the probability that, after observing n independent executions of the system up to time t, for $n, t \to \infty$, the attacker fails to correctly guess the *pair* (s, r) of the true sender and receiver.

In fact, in order to assess the degree of anonymity provided by the system, it is more convenient to have at hand the error probabilities for the sender and for the receiver separately. To see how these probabilities are defined and computed, we examine in detail the case of the sender; the receiver case is basically the same. Formally, for each $\sigma \in \mathcal{O}^t$ and

sender $s \in V \setminus C$, let

$$p_{send}(\sigma|s) \triangleq \Pr(O^t = \sigma | S = s)\,. \tag{4.10}$$

Note that $p_{send}(\sigma|s)$ can be actually computed as an average $\sum_{r\in V} p(\sigma|(s,r))p_R(r)$. The quantities $p(\sigma|(s,r))$ can be computed as described by Lemma 4.3.3. For any fixed $t \geq 1$, (4.10) defines a conditional probability matrix; using this matrix, we can form an information hiding system where the states are the senders and the observables are t-traces: $\langle V \setminus C, \mathcal{O}^t, p_S(\cdot), p_{send}(\cdot|\cdot)\rangle$. Let us denote by $P^{(t)}_{e,send}$ the corresponding asymptotic error probability. The probability we are after is obtained by letting t go to ∞:

$$P_{e,send} \triangleq \lim_{t\to\infty} P^{(t)}_{e,send}\,.$$

Reasoning as we did for Proposition 4.3.2, one checks that $P_{e,send}$ can be computed from the limit indistinguishability relation as $t \to \infty$, say $\equiv_{send}$. Explicitly, this relation can be defined as

$$s \equiv_{send} s' \quad \text{if and only if} \quad \text{for each } \sigma \in \mathcal{O}^* \quad p_{send}(\sigma|s) = p_{send}(\sigma|s').$$

The next lemma says how $\equiv_{send}$ can be computed starting from the hidden Markov model $\mathcal{M}$ defined above, by a suitable aggregation of the rows of the basis matrix B. The proof consists of easy manipulations of the transition matrices M_σ and is omitted. Recall that the states of $\mathcal{M}$ are pairs (u, v), thus $e_{(u,v)}$ denotes the row vector in $\mathbb{R}^{1\times|V|^2}$ whose entry corresponding to the element (u, v) is 1, while the others are 0. For each s, we let f_s denote the row vector $\sum_{(s,v)\in\mathcal{S}} e_{(s,v)}$.

Lemma 4.4.1 *Let B be a basis like in the hypotheses of Theorem 4.3.4 for the hidden Markov model $\mathcal{M}$ defined above. For any two senders s and s',*

$$s \equiv_{send} s' \qquad \textit{if and only if} \qquad (f_s - f_{s'}) \perp B.$$

We have applied this setting to a few instances of a grid network, like the one in Figure 10, relative to different sizes d of the grid and different sets C of corrupted nodes. Table 1 summarises the outcomes of these experiments. The nodes in the grid are numbered from 1 to d^2, starting from the top left corner and proceeding row-wise from left to right. To avoid end effects, we make the grid wrap up, i.e. the top and bottom rows are connected together, as well as the rightmost and leftmost columns. The sets C are chosen so as to give rise to configurations where

d	C	K_{send} $= L_{\times,send}$	K_{rec} $= L_{\times,rec}$	$P_{e,send}$	$P_{e,rec}$	$L_{+,send}$	$L_{+,rec}$
3	$\{1\}$	2	4	0.75	0.56	0.12	0.33
3	$\{1,5\}$	4	9	0.43	0	0.43	0.89
3	$\{2,4,6,8\}$	5	9	0	0	0.8	0.89
4	$\{1\}$	4	9	0.73	0.44	0.2	0.5
4	$\{1,6\}$	7	12	0.5	0.25	0.43	0.69
4	$\{2,5,7,10\}$	12	16	0	0	0.92	0.94
5	$\{1\}$	5	15	0.79	0.4	0.17	0.56
5	$\{1,7\}$	13	25	0.43	0	0.52	0.96
5	$\{2,6,8,12\}$	21	25	0	0	0.95	0.96
6	$\{1\}$	10	10	0.71	0.72	0.26	0.25
6	$\{1,8\}$	19	36	0.44	0	0.53	0.97
6	$\{2,7,9,14\}$	32	36	0	0	0.97	0.97

Table 1: Sender and receiver anonymity for several instances of a grid network.

no two corrupted nodes are directly connected: we have checked experimentally that these are the most advantageous for the attacker; otherwise, the relative distance of the corrupted nodes seems unimportant. K_{send} and K_{rec} denote the number of classes of $\equiv_{send}$ and of $\equiv_{rec}$, respectively. Moreover, from Corollary 3.1.11(2) in Section 3.2.2, we know that the asymptotic min-entropy leakage coincides with the logarithm of the number of classes in the case of uniform distribution. The probability $P_{e,send}$ is computed as $1-\frac{K_{send}}{|V|-|C|}$, while $P_{e,rec}$ is computed as $1-\frac{K_{rec}}{|V|}$. Finally, additive leakages are computed as indicated by Corollary 3.1.11(2).

Although a systematic study of anonymous routing protocols is outside the scope of this thesis, some qualitative considerations can be drawn from these data. If one keeps d fixed and lets $|C|$ grow, the data are simple to interpret: the error probability goes to 0 and the leakage gets larger. On the other hand, if one keeps $|C|$ fixed and compares configurations of different size d, the interpretation becomes less obvious. The leakage tends to increase when moving from smaller to larger values of d, which is particularly evident from the columns of min-entropy leakage. This increase occurs barely because, as the number of nodes grows, the number of indistinguishability classes tends to grow as well: all this means is that a large system tends to leak more information than a small one. Concerning error probability, which is supposed to measure the "absolute" resistance of a system against passive eavesdropping, the

data seem to partially contradict the intuition that the more nodes in a network, the stronger the guarantee of anonymity. Indeed, it may happen that the error probability *decreases* when moving from smaller to larger values of d. Also, the receiver seems more vulnerable than the sender from the point of view of anonymity.

At the moment we have no exact explanation to offer for these phenomena. Heuristically, the first phenomenon (decrease of error probability) seems to be connected with the fact that, as d grows, the number of indistinguishability classes may grow faster than the number of nodes, because a great deal of new observables (traces) becomes available. The second phenomenon (receiver's vulnerability) is connected with the fact that, given enough time, the message will reach its destination and, if this is a corrupted node, the adversary will know that for sure. A more systematic study of anonymous routing protocols is called for to quantitatively assess their security .

4.5 Concluding remarks

We have extended the previous results to a more sophisticated attack scenario, where the computation of the system may take several steps to terminate, or even not terminate at all, and where each state crossed during an execution induces one observation. We formalise this scenario in terms of discrete-time *Hidden Markov Models* (Rab89). This set up allowed us to describe situations where the attacker collects information from different sources at different times, like in a coalition of different local eavesdroppers. The HMM model we consider is similar in spirit to the fully probabilistic automata considered by Andrés et al. in (APvRS10). Their purpose is different, though, as they aim at feasible methods for computing the channel matrix associated with the automaton, whereas we focus on the asymptotic behaviour of leakage and error probability.

Chapter 5

Active attackers: a non adaptive scenario

The models described in the previous chapter, as most of the models of QIF so far proposed, concern with the case where the attacker is a passive eavesdropper. Only very recently, quantification of threats posed by active attackers, able to directly interact with the system, has been considered. In particular, (CS10) provides a definition of quantitative integrity, by measuring information flowing from untrusted inputs to public outputs, (BS11) tracks the knowledge an attacker can obtain in several runs of the program, each one with a new attacker-controlled input, while (KB07) considers a scenario of adaptive chosen-message attacks. However, all of them are limited to deterministic programs. In this and the next chapter we extend previous results to the probabilistic case.

In this chapter we focus on the non-adaptive case, studying the threats posed to confidentiality and integrity of probabilistic systems by a class of active adversaries. As in (BS11), we assume that part of the input is under the control of an active attacker, hence *untrusted*, and consider re-execution attacks. In this scenario, the attacker can take advantage of multiple runs in two ways: (1) like in (BS11), at each run the attacker can alter the untrusted input and observe the result of this modification; (2) since the system is probabilistic, the attacker can accumulate statistical evidence about the behaviour of the program and hence the secret, assuming the runs are independent and the secret remains fixed throughout the runs. As we will show, the interplay between these two

capabilities is non-trivial.

We define a notion of quantitative multi-run leakage, based on min-entropy (Smi09), and give a simple characterization of its asymptotic behaviour, depending on the number n of independent runs of the program. In particular, we determine the exponential growth rate of the leakage, which allows us to give tight numerical estimates depending on n.

Focussing on the qualitative setting, given a declassification policy we provide a multi-run quantitative measure of policy violation, in terms of *conditional* min-entropy. Again, the asymptotic behaviour of this measure is characterized in simple terms. This allows to combine analysis of *what* and *how much* information is leaked. In a multi-run setting, we formulate, in terms of statistical hypothesis testing, the problem of deciding whether an attack against a system is occurring. Given an integrity policy, specified as a class of "suspect" behaviours, we characterise an optimal decision strategy and quantify the inherent risks (error probabilities) involved in taking a decision.

5.1 An extended model: trusted and untrusted inputs

Let $\mathcal{S}$, $\mathcal{U}$ and $\mathcal{O}$ be nonempty sets. We consider probabilistic programs $c(s, u)$ depending on two inputs: a *trusted* and *confidential* input $s \in \mathcal{S}$ and an *untrusted* one $u \in \mathcal{U}$; the latter is under the control of the attacker, who can read and/or modify it at will. The program c outputs a result, or more generally an observation[1], $o \in \mathcal{O}$. If we view untrusted inputs as actions chosen by the adversary, we can formally represent the program c as an *action-based* randomisation mechanism, defined as follows.

Definition 5.1.1 (Action-based randomisation mechanism) *An* action-based *randomisation mechanism is a 4-tuple*

$$\mathcal{S} = \langle \mathcal{S}, \mathcal{O}, \mathcal{U}, \{p_u : u \in \mathcal{U}\} \rangle$$

where (all sets finite and nonempty): $\mathcal{S}, \mathcal{O}$ and $\mathcal{U}$ are respectively the sets of secrets, observations and actions *(or untrusted inputs) and for each $u \in \mathcal{U}$, p_u is a stochastic matrix of dimensions $|\mathcal{S}| \times |\mathcal{O}|$.*

[1]For simplicity, we assume the program always terminates, or that non-termination can be detected by e.g. timing considerations, thus becoming an observable value. See e.g. (BS11).

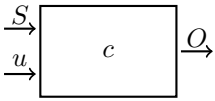

Figure 11: Noisy channel with an untrusted input.

For each fixed untrusted input $u \in \mathcal{U}$, the behaviour of c is defined by the matrix $p_u(\cdot|\cdot) \in [0,1]^{\mathcal{S}\times\mathcal{O}}$. For any s, the row of this matrix is a probability distribution on $\mathcal{O}$, denoted by $p_u(\cdot|s)$: here, $p_u(o|s)$ is the probability of observing o, given that the secret input to the program is s and the untrusted input is u. A system is *deterministic* if each entry of each $p_u(\cdot|\cdot)$ is either 0 or 1. Note that to any deterministic system there corresponds a function $f : \mathcal{S} \times \mathcal{U} \to \mathcal{O}$ defined by $f(s,u) = o$, where $p_u(o|s) = 1$.

Assume a prior distribution $p(\cdot)$ on $\mathcal{S}$ is given and known to the attacker, and let $S \sim p(\cdot)$ be the random variable representing the confidential input. From the attacker's perspective, c is a noisy channel with two inputs: the random variable S and a value $u \in \mathcal{U}$ chosen by the attacker himself. The channel output is a random variable O taking values in $\mathcal{O}$ and such that S and O are jointly distributed according to the distribution $q_u(s,o) \triangleq p(s) \cdot p_u(o|s)$, depending on the chosen value u. This is pictorially represented in Figure 11; we stress that u is *not* a random variable, but an actual value chosen by the attacker.

We are interested in a scenario of multi-run security, where the attacker is granted the possibility of observing several independent executions of the program c, with a fixed, unknown input S and different values of u, chosen by himself. The ability of tweaking the system by trying different values of $\mathcal{U}$, thus influencing the output, clearly gives the adversary extra strength. Suppose that the attacker can perform enough experiments, with the input s remaining fixed, in order to try all possible values of u. If $\mathcal{U} = \{u_1, \ldots, u_l\}$, we shall assume that the attacker can try, in some order, all l values for u. Let $\mathbf{u} = [u_1, \ldots, u_l]$ be the sequence containing all values for u. We can see $\mathbf{u}$ as a *strategy* where all possible actions are played in a fixed order. Note that this order is fixed in advance: the attacker is not allowed to play actions adaptively, that is, by taking into consideration intermediate observations. The adaptive scenario will be examined in Chapter 6 (see Definition 6.1.2, which in fact generalises the present setting).

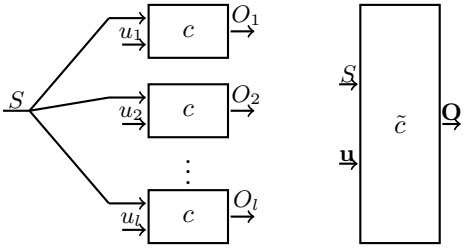

Figure 12: Two different ways to view a multi-run IHS

For each $i = 1, \ldots, l$, the attacker observes O_i, the random variable corresponding to the output of the channel with input (S, u_i). Once $\mathbf{u}$ has been defined, we will find convenient to view the set of all l observations as a output of a 'big' channel, as explained below.

Definition 5.1.2 (multi-run IHS) *Let c be a probabilistic program defined, for each $u \in \mathcal{U}$, by a stochastic matrix $p_u(\cdot|\cdot) \in [0,1]^{(\mathcal{S}\times\mathcal{O})}$ as detailed above. A* multi-run IHS *for c is*

$$\mathcal{H}^{\mathbf{u}} \triangleq \langle \mathcal{S}, \mathcal{O}^l, p(\cdot), p_{\mathbf{u}}(\cdot|\cdot) \rangle \,.$$

where $p(\cdot)$ is a prior distribution on $\mathcal{S}$ and

$$p_{\mathbf{u}}(\mathbf{o}|s) \triangleq p_{u_1}(o_1|s) \cdot \cdots \cdot p_{o_l}(o_l|s) \tag{5.1}$$

for any $\mathbf{o} = [o_1, \ldots, o_l] \in \mathcal{O}^l$.

Since for each of these l experiments the value of the secret s and the channel c are the same, it is as if we were executing l parallel programs (identical and equal to c), each of which takes S and one of the values u_i as input, giving as result the corresponding $O_i \sim p_{u_i}(o|s)$. This is the same as considering a 'big' channel $\tilde{c}$ with two inputs, the random variable S and the vector $\mathbf{u} \triangleq [u_1, \ldots, u_l]$, and as an output the random vector $\mathbf{O} \triangleq [O_1, \ldots, O_l]$. The situation is pictorially represented in Figure 12. Note that the O_i's are independent given S.

The main difference from the previous model is that here a part of the input, u, is known to the attacker and so he can condition the distributions on the secrets with the knowledge of it. Secondly, the observations here correspond to l-tuples of elements of $\mathcal{O}$.

Let us fix a specific notation for multi-run IHS's. A boldface letter $\mathbf{o}$ (respectively $\mathbf{O}$) will denote a generic element of (respectively random variable taking values in) $\mathcal{O}^l$. The notations, definitions and results introduced in Chapter 3 for general IHS's do of course specialise to multi-run ones, taking this notation into account. For example, $\mathbf{O^n} = (\mathbf{O_1}, \ldots, \mathbf{O_n})$ denotes a generic sequence of random variable i.i.d. given S, thus according to (3.1):

$$\Pr(\mathbf{O^n} = (\mathbf{o_1}, \ldots, \mathbf{o_n}) \mid S = s) \;=\; \prod_{i=1}^{n} p_{\mathbf{u}}(\mathbf{o_i}|s) \;=\; \prod_{i=1}^{n}\prod_{j=1}^{l} p_{u_j}(o_{ij}|s)$$

where we let $o_{ij} \in \mathcal{O}$ denote the jth component of the tuple $\mathbf{o_i} \in \mathcal{O}^l$. Note that $\mathbf{O^n}$ corresponds to observing the system $n \times l$ times, n times for each u_j, $1 \leq j \leq l$. We will use index $^{\mathbf{u}}$ to remind ourselves that we are dealing with a multi-run IHS, and the corresponding set of untrusted inputs. Thus for example, the error probability and the information leakage can be expressed as follows.

Definition 5.1.3 (Error probability and leakage for multi-run IHS's) *Given a multi-run* IHS $\mathcal{H}_{\mathbf{u}}$ *and a guessing function* g*, the error probability is*

$$P_e^{(g),\mathbf{u}}(n) \triangleq \Pr(g(\mathbf{O^n}) \neq S) \qquad \textit{where} \qquad P_{succ}^{(g),\mathbf{u}} = 1 - P_e^{(g),\mathbf{u}}(n).$$

As consequence, the information leakage becomes

$$\mathcal{L}^{\mathbf{u}}(n) \triangleq \log \frac{P_{succ}^{(g),\mathbf{u}}(n)}{\max_s p(s)}\,.$$

Note that the guessing function g, which will be assumed MAP unless otherwise stated, here has type $g : \mathcal{O}^{l\times n} \to \mathcal{S}$.

Example 18 *Let* $\mathcal{S} = \{0, \ldots, 10^{16} - 1\}$*, possibly representing credit card numbers,* $\mathcal{U} = \{1, \ldots, 16\}$*, representing positions of digits in (the decimal expansion to sixteen digits of) a credit card number, and* $\mathcal{O} = \{*, 0, \ldots, 9\}$*, representing the set of decimal digits plus a default value. Consider a program* $c(s, u)$ *that flips a fair coin to decide whether output the* u*-th digit of the decimal expansion of* s *or output* $*$*. The conditional probability matrix defining the program is* $p_u(o|s) = \frac{1}{2}\delta_{o,s[u]}$*, where* $s[u]$ *denotes the* u*-th decimal digit of* s*, and* $\delta_{i,j}$ *is Kronecker's symbol with the convention that* $\delta_{*,j} = 1$ *for each* j*. Take the uniform prior distribution* $p(\cdot)$ *on* $\mathcal{S}$*. According to Definition 5.1.2, a multi-run* IHS *can be formed with this program and* $p_{\mathbf{u}}(\mathbf{o}|s) = \prod_{i=1}^{16} p(o_i|s, u_i) = \prod_{i=1}^{16} \frac{1}{2}\delta_{o_i,s[u_i]}$.

5.2 Bounds and asymptotic behaviour

Let us now study the asymptotic behaviour of $\mathcal{L}^{\mathbf{u}}(n)$, depending on the number n of independent observations. Throughout the section, $\mathcal{H}^{\mathbf{u}}$ will denote a generic multi-run IHS for a program $p_u(o|s)$, as in Definition 5.1.2.

The indistinguishability relation instantiated to $\mathcal{H}_{\mathbf{u}}$ is defined as follows.

Definition 5.2.1 (Indistinguishability for multi-run IHS) *For each $s, s' \in \mathcal{S}$*

$$s \equiv_{\mathbf{u}} s' \quad \textit{if and only if} \quad \textit{for each } \mathbf{o} \in \mathcal{O}^l \;\; p_{\mathbf{u}}(\mathbf{o}|s) = p_{\mathbf{u}}(\mathbf{o}|s').$$

Let $\mathcal{S}/\equiv_{\mathbf{u}} = \{C_1, \ldots, C_k\}$ be the set of equivalence classes of $\equiv_{\mathbf{u}}$. For each $i = 1, \ldots, k$ let $s_i^* \triangleq \operatorname{argmax}_{s \in C_i} p(s)$ and $p_i^* \triangleq p(s_i^*)$. We assume without loss of generality that $p_i^* > 0$ for each $i = 1, \ldots, k$ (otherwise all the secrets in class C_i can be just discarded from the system) and that $p_1^* = \max_{s \in \mathcal{S}} p(s)$. Applying to $\mathcal{H}^{\mathbf{u}}$ Corollary 3.1.11(2), Remark 3.1.12 and Theorem 3.1.7, we immediately get the following result.

Proposition 5.2.2 *The information leaked by the multi-run IHS $\mathcal{H}^{\mathbf{u}}$ after an infinite number of experiments is:*

$$\lim_{n \to \infty} \mathcal{L}^{\mathbf{u}}(n) = \log \frac{\sum_{i=1}^{k} p_i^*}{p_1^*}.$$

If the distribution on secrets is uniform:

$$\lim_{n \to \infty} \mathcal{L}^{\mathbf{u}}(n) = \log |\mathcal{S}/\equiv_{\mathbf{u}}|.$$

In any case, the rate of convergence of $P^{\mathbf{u}}_{succ}(n)$ is given by:

$$\rho_{\mathbf{u}} \triangleq \min_{\substack{i,j \in \{1,\ldots,k\} \\ i \neq j}} C\left(p_{\mathbf{u}}(\cdot|s_i^*), p_{\mathbf{u}}(\cdot|s_j^*)\right).$$

Applying Proposition 5.2.2 may be difficult. Indeed, the matrix $p_{\mathbf{u}}(\cdot|\cdot)$ has a size exponential in l, $|\mathcal{S}| \times |\mathcal{O}|^l$, which can be very large in practice. This can make direct calculations of both the indistinguishability relation $\equiv_{\mathbf{u}}$ and the rate $\rho_{\mathbf{u}}$ very costly. In the rest of the section, we offer results to cope with both problems.

For each $i = 1, \ldots, l$, consider the equivalence relation $\equiv_i$ on $\mathcal{S}$, defined by setting $u = u_i$ in the system, that is:

$$s \equiv_i s' \quad \text{if and only if} \quad \text{for each } o \in \mathcal{O} \;\; p_{u_i}(o|s) = p_{u_i}(o|s').$$

Each of these relations is much easier to compute than $\equiv_{\mathbf{u}}$, as it only involves a 'small' matrix of size $|\mathcal{S}|\times|\mathcal{O}|$. On the other hand, computation of $\equiv_{\mathbf{u}}$ is reduced to computing the set-theoretic intersection of the $\equiv_i$'s, as stated by the following lemma.

Lemma 5.2.3

$$\equiv_{\mathbf{u}} = \bigcap_{i=1,\ldots,l} \equiv_i.$$

PROOF We have to prove that for each $s, s' \in \mathcal{S}$:

$$s \equiv_{\mathbf{u}} s' \quad \text{if and only if} \quad s \equiv_i s' \text{ for each } i = 1, \ldots, l.$$

The $\Leftarrow$-direction is obvious. Concerning the $\Rightarrow$-direction, we consider for notational simplicity just the case $i = 1$. Note that for each $o \in \mathcal{O}$

$$\begin{aligned} p_{u_1}(o|s) &= \sum_{o^{l-1} \in \mathcal{O}^{l-1}} p_{u_1}(o|s) p_{u_2}(o_1|s) \cdots p_{u_l}(o_{l-1}|s) \\ &= \sum_{o^{l-1} \in \mathcal{O}^{l-1}} p_{\mathbf{u}}((o, o^{l-1})|s) \\ &= \sum_{o^{l-1} \in \mathcal{O}^{l-1}} p_{\mathbf{u}}((o, o^{l-1})|s') = p_{u_1}(o|s') \end{aligned} \tag{5.2}$$

where the third equality stems from $p_{\mathbf{u}}((o, o^{l-1})|s) = p_{\mathbf{u}}((o, o^{l-1})|s')$ for each o^{l-1}, by definition of $\equiv_{\mathbf{u}}$. This shows that $s \equiv_1 s'$. □

We now come to the computation of the rate $\rho_{\mathbf{u}}$. In fact, it is not obvious if, and how, this quantity can be computed without first building the 'big' matrix $p_{\mathbf{u}}(\cdot|\cdot)$. What we shall do is to propose a sub-optimal (non-MAP) but reasonably efficient guessing strategy g for $\mathcal{H}^{\mathbf{u}}$; the rate $\rho^\star$ of the corresponding success probability, $P_{succ}^{(g),\mathbf{u}}(n)$, can be easily computed. This way, we will get the lower-bound $\rho_{\mathbf{u}} \geq \rho^\star$.

The idea behind the strategy g is as follows. Upon collecting a n-sequence of observations generated from $\mathcal{H}^{\mathbf{u}}$, say $\mathbf{o^n} = (\mathbf{o_1}, \ldots, \mathbf{o_n})$, the attacker analyses separately the l subsequences obtained by setting $u = u_i$ for $i = 1, \ldots, l$: call them $o_1^n, \ldots, o_l^n$. Explicitly, $o_i^n \in \mathcal{O}^n$ denotes

the sequence $(o_{1i}, \ldots, o_{ni})$, corresponding to the untrusted input u_i. For each $i = 1, \ldots, l$, the attacker chooses the equivalence class of $\equiv_i$ that is most likely, given the sequence o_i^n. Next he builds the intersection of all these classes and then chooses the most likely secret s that lies in this intersection.

To state concisely the next result, we introduce a few abbreviations. For each $i = 1, \ldots, l$, let

$$\mathcal{S}/\equiv_i = \{D_{i1}, \ldots, D_{ik_i}\}$$

be the equivalence classes of $\equiv_i$; let

$$p_i(o|D_{ij}) \triangleq \sum_{s \in D_{ij}} \frac{p_{u_i}(o|s)p(s)}{p(D_{ij})}$$

be the probability of observing o given that the untrusted input is set to u_i and the secret belongs to class D_{ij}; note that $p_i(\cdot|D_{ij}) = p_{u_i}(\cdot|s)$, for any $s \in D_{ij}$. We define

$$\rho_i \triangleq \min_{j \neq h} C(p_i(\cdot|D_{ij}), p_i(\cdot|D_{ih})) \qquad \text{and} \qquad q_i^* \triangleq \max_{1 \leq j \leq k_i} p(D_{ij}).$$

Note that the computation of each ρ_i can be carried out starting from the 'small' $|\mathcal{S}| \times |\mathcal{O}|$ matrix $p_{u_i}(\cdot|\cdot)$, obtained by setting $u = u_i$. Then, it is possible to prove the following result.

Theorem 5.2.4 *The rate of convergence of $P_e^{\mathbf{u}}$ is at least*

$$\rho_{\mathbf{u}} \geq \rho^* \triangleq \min_{i=1,\ldots,l} \rho_i.$$

More precisely, for $\gamma^* \triangleq \sum_{i=1}^{l} \frac{k_i^2}{2} q_i^*$:

$$(1 - \sum_{i=1}^{k} p_i^*) \leq P_e^{\mathbf{u}}(n) \leq (1 - \sum_{i=1}^{k} p_i^*) + \gamma^* 2^{-n\rho^*}. \tag{5.3}$$

PROOF The LHS inequality comes from Theorem 3.1.7 applied to $\mathcal{H}^{\mathbf{u}}$. Consider now the RHS. For any $n \geq 1$, we define a guessing function $g : \mathcal{O}^{l \times n} \to \mathcal{S}$ for $\mathcal{H}^{\mathbf{u}}$ as follows. For each $i = 1, \ldots, l$, consider the 'small' IHS having as secrets $\mathcal{S}/\equiv_i$ and defined thus

$$\mathcal{H}_i \triangleq \langle \mathcal{S}/\equiv_i, \mathcal{O}, p_i(\cdot), p_i(\cdot|\cdot) \rangle \tag{5.4}$$

where $p_i(\cdot)$ is the prior probability on $\mathcal{S}/\equiv_i$ defined as $p_i(D_{ij}) \triangleq p(D_{ij})$ for $j = 1, \ldots, k_i$, and $p_i(\cdot|\cdot)$ is as defined immediately above the statement of Theorem 5.2.4. Let $g_i : \mathcal{O}^n \to \mathcal{S}/\equiv_i$ be a MAP guessing function for $\mathcal{H}_i$. We set, for any $\mathbf{o^n} = (\mathbf{o_1}, \ldots, \mathbf{o_n}) \in \mathcal{O}^{l \times n}$,

$$g(\mathbf{o_1}, \ldots, \mathbf{o_n}) \triangleq \operatorname{argmax}_{s \in \bigcap_{i=1}^{l} g_i(o_i^n)} p(s)$$

with the convention that the RHS denotes the probability of a default secret, e.g. s_1, if the intersection $\bigcap_{i=1}^{l} g_i(o_i^n)$ is empty. We shall now give an upper-bound for $P_e^{(g),\mathbf{u}}(n)$.

Informally, the function g can be wrong in two ways: either the wrong $\equiv_\mathbf{u}$-class of the secret is selected, or the correct class is guessed, but the selected secret within this class is wrong. Formally, let $\mathbf{O^n} = (\mathbf{O_1}, \ldots, \mathbf{O_n})$ be a sequence of observations i.i.d. given S. Let Err be the event $(g(\mathbf{O^n}) \neq S)$ and let $Succ'$ be the event that the class is correctly chosen by g, that is $\left([g(\mathbf{O^n})]_{\equiv_\mathbf{u}} = [S]_{\equiv_\mathbf{u}}\right)$. By Lemma 5.2.3, the latter event can be equivalently expressed as $(g_1(O_1^n) = [S]_{\equiv_1} \wedge \ldots \wedge g_l(O_l^n) = [S]_{\equiv_l})$, where $O_i^n = (O_{i1}, \ldots, O_{in})$ for $i = 1, \ldots, l$. The probability of error $P_e^{(g),\mathbf{u}}(n) = \Pr(Err)$ can be decomposed as

$$P_e^{(g),\mathbf{u}}(n) = \Pr(Err, Succ') + \Pr(\neg Succ') \tag{5.5}$$

where in the second summand we have taken into account the fact that $\Pr(Err|\neg Succ') = 1$. Now, we bound separately the two summands in (5.5). $P(Err, Succ')$ is the probability that the secret belongs to the chosen class, but does not coincide with the element chosen by g. Below, we denote the event $\left([S]_{\equiv_\mathbf{u}} = C_i\right)$ by just C_i, and recall that $k = |\mathcal{S}/\equiv_\mathbf{u}|$. We have:

$$\begin{aligned}
&\Pr(Err, Succ') \\
&= \textstyle\sum_{i=1}^{k} \Pr(Err, Succ', C_i) \\
&= \textstyle\sum_{i=1}^{k} \Pr(Err|Succ', C_i) \Pr(Succ', C_i) \\
&\leq \textstyle\sum_{i=1}^{k} \Pr(Err|Succ', C_i) p(C_i) \\
&= \textstyle\sum_{i=1}^{k} \left(1 - \frac{p_i^*}{p(C_i)}\right) p(C_i) \\
&= 1 - \textstyle\sum_{i=1}^{k} p_i^*.
\end{aligned} \tag{5.6}$$

Concerning the second summand in (5.5), a simple union bound yields

$$P(\neg Succ') \leq \sum_{i=1}^{l} \Pr(g_i(O_i^n) \neq [S]_{\equiv_i}).$$

Each summand $\Pr(g_i(O_i^n) \neq [S]_{\equiv_i})$ is just the n-error probability of the IHS $\mathcal{H}_i$ defined in (5.4). Now, we note that the indistinguishability relation on $\mathcal{H}_i$ is trivial: by definition of $p_i(\cdot|\cdot)$, if $D_{ij} \neq D_{ij'}$ then $p_i(\cdot|D_{ij}) \neq p_i(\cdot|D_{ij'})$. Hence, applying Theorem 3.1.7 to each $\mathcal{H}_i$, we get

$$\Pr(\neg Succ') \leq \sum_{i=1}^{l} \frac{k_i^2}{2} q_i^* 2^{-n\rho_i} \leq \big(\sum_{i=1}^{l} \frac{k_i^2}{2} q_i^*\big) 2^{-n\rho^*} = \gamma^* 2^{-n\rho^*}. \qquad (5.7)$$

Combining (5.5), (5.6) and (5.7) and recalling that $P_e^{\mathbf{u}}(n) \leq P_e^{(g),\mathbf{u}}(n)$ by the optimality of MAP, we get the wanted result. □

Example 19 *Reconsider Example* 18 *and estimate how many observations are needed by an attacker to half the initial min-entropy of the secret. Let us first determine the parameters $\sum_i p_i^*$, ρ^* and γ^* necessary to apply Theorem 5.2.4.*

Fix $i = 1, \ldots, 16$. Then $\equiv_i$ partitions $\mathcal{S}$ into $k_i = 10$ classes, $D_{i0}, \ldots, D_{i9}$. Here D_{ij} contains all the numbers having the digit j at position i. Now $p_i(\cdot|D_{ij})$ is a vector that is everywhere 0, except in columns $*$ and j, where it contains $1/2$. It is easy to compute the least Chernoff Information between any two rows $p_i(\cdot|D_{ij}) \neq p_i(\cdot|D_{ij'})$: $\rho_i = -\log \frac{1}{2} = 1$, which in turn implies $\rho^* = \min_i \rho_i = 1$; moreover, $q_i^* = \frac{1}{10}$. Finally, applying Lemma 5.2.3, we see that $\equiv_{\mathbf{u}}$ is the identity, that is it has singleton classes, hence $\sum_{i=1}^{k} p_i^* = 1$. With these data, we can apply Theorem 5.2.4 and obtain:

$$0 \leq P_e^{\mathbf{u}}(n) \leq \sum_{i=1}^{16} \frac{10^2}{2} \frac{1}{10} 2^{-n} = 10 \cdot 2^{3-n}$$

and

$$\log\left(10^{16}(1 - 10 \cdot 2^{3-n})\right) \leq \mathcal{L}^{\mathbf{u}}(n) \to 16 \log 10.$$

This implies that with $n = 7$ observations of $\mathcal{H}^{\mathbf{u}}$, the initial min-entropy gets halved: $\mathcal{L}^{\mathbf{u}}(n) \geq 8 \log 10$. Note that each observation in $\mathcal{H}^{\mathbf{u}}$ is a tuple $\mathbf{o} \in \mathcal{O}^{16}$: this means that the attacker will in fact collect $7 \times 16 = 112$ individual $o \in \mathcal{O}$.

Example 20 (Hamming weight attacks against DES **S-boxes)** *Reconsider the noisy scenario of Hamming weight attacks against S-boxes, described in Example 11. Suppose now that the attacker is able to control the message inserted in one of the S-boxes, e.g. S_1, besides measuring the Hamming weight of the output of all eight S-boxes. Assuming a uniform prior distribution on $\mathcal{K}$, a multi-run* IHS *$\mathcal{H}^{\mathbf{u}}$ for this system can be defined as prescribed by Definition 5.1.2. Note that observations in this multi-run* IHS *are tuples of the form*

$\mathbf{o} = [o_0, \ldots, o_{63}]$, *with* o_i *produced by setting the message* m *to the binary expansion of* i *on six bits.*

Let us estimate the asymptotic behaviour of information leakage for this IHS. *In what follows, we target the S-box* S_1 *of* DES. *We compute the parameters* $\sum_i \pi_i$, q_i^* *and* ρ^* *necessary to apply Theorem 5.2.4. First, by inspection of the S-box, it is readily computed that* $\equiv_{\mathbf{u}} = \cap_{i=0}^{63} \equiv_i$ *is just the identity: hence* $\sum_i \pi_i = 1$. *For each* $i = 0, \ldots, 63$, *it is easy to see that* $k \equiv_i k' \Leftrightarrow W(SB(k, m_i)) = W(SB(k', m_i))$, *therefore each* $\equiv_i$ *partitions* $\mathcal{S}$ *into* $k_i = 5$ *classes. Moreover, for each* i *and for each* $j = 0, \ldots, 4$, *the distribution* $p_i(\cdot|D_{ij})$ *coincides with the distribution of the random variable* $N + j$. *It is easy to see that* $\rho_i = \min_{j \neq j'} C(N + j, N + j') = C(N, N + 1) \approx 0.027$. *Therefore,* $\rho^* \approx 0.027$. *Concerning the computation of the* q_i^**'s, we reason as follows. Fix a message* m_i. *We have to compute how many keys are in class* D_{ij}, *for* $j = 0, \ldots, 4$. *By construction of the* DES *S-boxes, for each* $z \in \{0, \ldots, 15\}$ *(seen as a* 4*-bits block) there exist exactly* 4 *keys such that* $SB(k, m_i) = z$. *Thus, for each* $j = 0, \ldots, 4$, *there are exactly* $4 \times \binom{4}{j}$ *keys that give an output with Hamming weight* j, *hence are in class* D_{ij}. *The largest such class is then obtained for* $j = 2$, *and* $|D_{i2}| = 6$. *This yields* $q_i^* = \frac{24}{64} = \frac{3}{8}$, *independently of* i. *Applying now Theorem 5.2.4 to compute* $P_{succ}^{\mathbf{u}}(n)$*:*

$$\log \frac{1 - 300 \cdot 2^{-n \cdot 0.027}}{64} \leq \mathcal{L}^{\mathbf{u}}(n) \to 6\,.$$

That means that already $n = 310$ *observations* $\mathbf{o} \in \mathcal{O}^{64}$ *of* $\mathcal{H}^{\mathbf{u}}$ *are sufficient to determine* 5.9 *bits out of the total* 6 *of the key; this means that the attacker will collect a total of* 64×310 *individual observables* $o \in \mathcal{O}$.

5.3 Declassification policies

Now that we have seen how much information can be leaked by the system, we wish to express a policy which sets limits on this leakage. In many contexts, the release of partial information concerning a secret is unavoidable or even desirable. A *declassification policy* – see (BS11; MSZ04; SM03) and references therein – states in a precise way *what* information can be safely disclosed. For example, one may state that the four least significant digits of a credit card number can be safely disclosed. A policy can be expressed as a partition of the input domain into equivalence classes: each class contains values that should remain indistinguishable for the attacker. A program is compliant with a policy

if all information it may disclose is already implied by the knowledge of the equivalence class of the secret.

In this section we will study declassification policies in a probabilistic, multi-run setting. Throughout the section we fix a generic IHS $\mathcal{H} = \langle \mathcal{S}, \mathcal{O}, p(\cdot), p(\cdot|\cdot) \rangle$. The result we will obtain do of course apply also to the special case of a multi-run IHS $\mathcal{H}^{\mathbf{u}}$ – we will say more on this at the end of the section.

Definition 5.3.1 (Policy) *A* policy $\sim_\pi$ *for the input domain* $\mathcal{S}$ *is an equivalence relation on* $\mathcal{S}$*.*

The knowledge of the partition induced by the policy is the information that we grant to the adversary. In other words, the attacker may identify the class of the secret according to the policy, but nothing more.

A policy $\sim_\pi$ induces a partition, $\mathcal{S}/\sim_\pi = \{\pi_1, \ldots, \pi_t\}$, in addition to the one induced by the indistinguishability relation, $\mathcal{S}/\equiv$. Knowing both of them, the attacker can study the finer relation, given by their intersection, and possibly learn new information. A policy expresses a form of *qualitative* information disclosure, e.g. *which* data can be safely disclosed. Here, we want to enrich policies with a quantitative analysis, by computing *how much* extra information is leaked, given the policy. That is, we want to quantify how serious is a policy violation, if there is any. The following definition is a natural extension to a probabilistic setting of the notion of declassification policy known for deterministic programs (BS11; MSZ04; SM03).

Definition 5.3.2 (Policy compliance) *Let* c *be a (probabilistic) program that takes as input a secret* s *and a policy* $\sim_\pi$ *for* c*. Then,* c *is* policy compliant *with respect to* $\sim_\pi$ *if and only if*

$$\textit{for each } \; s \in \mathcal{S} \qquad [s]_{\sim_\pi} \subseteq [s]_{\equiv}.$$

In other words, if and only if the relation $\sim_\pi$ is finer, or at most equal, than $\equiv$. Note that in the special case where the matrix $p(\cdot|\cdot)$ of $\mathcal{H}$ is deterministic[2], that is when $p(\cdot|\cdot)$ defines an input/output function $f : \mathcal{S} \to \mathcal{O}$, the above definition specialises to the standard definition of policy compliance for deterministic programs: $s \sim_\pi s'$ implies $f(s) = f(s')$.

[2]For each s there is exactly one o s.t. $p(o|s) = 1$.

Example 21 *Let $\mathcal{S} = \{0, \ldots, 9\}^{16}$ represent credit card numbers. Consider the declassification policy stating that it is safe to disclose the four least significant digits of any number: for any $s, s' \in \mathcal{S}$*

$$s \sim_\pi s' \quad \textit{if and only if} \quad s = s' \mod 10^4.$$

Consider now two probabilistic programs, $c_1(s)$ and $c_2(s)$, taking as confidential input a value $s \in \mathcal{S}$. As shown below, c_1 tosses a fair coin and, depending on the outcome, outputs either the least three significant digits of s or a default value -1. c_2 tosses a fair coin and, depending on the outcome, outputs either the fourth or the fifth least significant digit of s.

```
c1: r=rnd(0..1);
    if r then l=-1 else l=h mod 10^3;

c2: r=rnd(0..1);
    if r then l=(h div 10^4) mod 10
    else l=(h div 10^5) mod 10;
```

Assuming a uniform prior on $\mathcal{S}$, we can view c_1 and c_2 as two IHS*'s $\mathcal{H}_1$ and $\mathcal{H}_2$, respectively. Denoting by $s\{4,5\}$ the* set *of the fourth and fifth least decimal digit of s, the indistinguishability relations of these* IHS*'s can be characterised as follows:*

- $s \equiv_1 s'$ *if and only if* $s = s' \mod 10^3$*;*
- $s \equiv_2 s'$ *if and only if* $s\{4,5\} = s'\{4,5\}$.

For c_1 two states generate all possible observations with the same probability if they have in common the three least significant digits. For c_2, after several observations it reveals both the fourth and the fifth digit of the card, although without revealing which is which. In order that two states are indistinguishable according to c_2, either their fourth and fifth digit coincide or they are swapped. Considering the intersection relation, we note that, in the first case, it coincides with $\sim_\pi$, while in the second one it is a finer relation that associates in the same class numbers formed by the same least five significant digits. It means that $\mathcal{H}_1$ is policy compliant with respect to $\sim_\pi$, while $\mathcal{H}_2$ is not. Indeed, all numbers equivalent according to $\sim_\pi$ are equivalent also according to $\equiv_1$. Hence, in this case no more information is revealed then that obtainable by knowing the policy. In the second case, instead, all pairs of numbers that have the least significant four digits in common, but where the fifth digits differ, are equivalent according to $\sim_\pi$ but not according to $\equiv_2$. Therefore, it is not true that for each $s \in \mathcal{S}$ $[s]_{\sim_\pi} \subseteq [s]_{\equiv_2}$. Intuitively, indeed, the knowledge of $\equiv_2$ reveals new information about the secret, i.e. the fifth digit.

The above example shows that, in the presence of declassification policies, beside *how much* it is important to understand *what* information is released by a program. More generally, one could be interested in quantifying how serious is a violation of the policy, if any.

In the rest of the section, we let $\sim_\pi$ be a generic policy on $\mathcal{S}$. We denote by Π the random variable representing the class of the secret according to the policy $\sim_\pi$, that is

$$\Pi \triangleq [S]_{\sim_\pi}.$$

Thus, Π takes values in $\mathcal{S}/\sim_\pi = \{\pi_1, \ldots, \pi_t\}$. The attacker strategy is modeled by a guessing function that takes as arguments both a sequence of observables o^n and an equivalence class π_i, since the latter is assumed to be known to the attacker. Formally, for each $n \geq 0$, we consider guessing functions of type $g : \mathcal{O}^n \times (\mathcal{S}/\sim_\pi) \to \mathcal{S}$. Moreover, we assume that g follows the MAP rule. That is, for each $n \geq 0$, for all $o^n \in \mathcal{O}^n$ and $i = 1, \ldots, t$:

$$g(o^n, \pi_i) = s \text{ implies } p(s|o^n, \pi_i) \geq p(s'|o^n, \pi_i) \quad \text{for each } s' \in \mathcal{S}. \quad (5.8)$$

Let us stress that when $n = 0$, g takes just a class π_i as an argument and chooses the secret that maximises $p(s|\pi_i)$, that is the secret that is most likely in class π_i.

Definition 5.3.3 (Error probability given a policy) *The error probability after $n \geq 0$ observations, conditioned on the knowledge of the policy class Π of the secret, is defined thus*

$$P_e(n|\Pi) \triangleq \Pr(g(O^n, \Pi) \neq S).$$

Definition 5.3.4 (Leakage given a policy) *Given an* IHS *$\mathcal{H}$ and a policy $\sim_\pi$, the* information leakage of $\mathcal{H}$ after n observations conditioned by the policy $\sim_\pi$ *is:*

$$\mathcal{L}^\Pi(n) \triangleq \log \frac{P_{succ}(n|\Pi)}{P_{succ}(0|\Pi)}.$$

Remark 5.3.5 *In the language of min-entropy, the above definition can be equivalently expressed in terms of conditional mutual information:*

$$\mathcal{L}^\Pi(n) = I_\infty(S; O^n|\Pi).$$

$\mathcal{L}^\Pi(n)$ expresses how much information about the secret a potential attacker would infer, after collecting the observations O^n, assuming that

he already knows the information disclosed according to the policy $\sim_\pi$. Of course, if the program is policy compliant respect to $\sim_\pi$, the leakage must be equal to 0, that is the attacker does not infer anything more than what he has already obtained by the knowledge of the policy. Intuitively, the greater the number n of observations the attacker can collect, the more the information about the secret the system leaks.

We introduce some abbreviations to state the results in the rest of the section. Fix a policy $\sim_\pi$. We have two partitions of the set $\mathcal{S}$: $\mathcal{S}/\equiv = \{C_1, \ldots, C_k\}$ and $\mathcal{S}/\sim_\pi = \{\pi_1, \ldots, \pi_t\}$. We will also consider a third partition, induced by the intersection equivalence relation $\equiv \cap \sim_\pi$, and let $m \triangleq |\mathcal{S}/(\equiv \cap \sim_\pi)|$ denote the number of classes of the intersection relation. For each $1 \leq i \leq k$ and $1 \leq j \leq t$, let $B_{ij} \triangleq C_i \cap \pi_j$ denote a generic (possibly empty) intersection class. For each $j = 1, \ldots, t$, let $I_j \triangleq \{i | B_{ij} \neq \emptyset\}$. In the rest of the section, unless otherwise stated, we will let j range over $1, \ldots, t$ and i over I_j. Note that $\mathcal{S}/(\equiv \cap \sim_\pi) = \{B_{ij} \mid j = 1, \ldots, t, i \in I_j\}$. For $i \in I_j$, we let

$$s^\star_{ij} \triangleq \mathrm{argmax}_{s \in B_{ij}} p(s), \qquad p^\star_{ij} \triangleq p(s^\star_{ij}) \qquad \text{and} \qquad p^\star_j \triangleq \max_{i \in I_j} p^\star_{ij},$$

where the latter is the maximum probability of a secret in class π_j. Finally, we let

$$\rho_j \triangleq \min_{r,h \in I_j, r \neq h} C(p(\cdot|s^\star_{rj}), p(\cdot|s^\star_{hj}))$$

be the least Chernoff Information between any two distinct rows in class π_j and

$$\rho_\Pi \triangleq \min_j \rho_j.$$

Proposition 5.3.6 *As $n \to \infty$,*

$$P_e(n|\Pi) \to 1 - \sum_{j,i} p^\star_{ij}.$$

Moreover this limit is reached at rate at least ρ_Π. More precisely, for $p_{max} \triangleq \max_j (p(\pi_j) \cdot p^\star_j)$ and $\gamma_\Pi \triangleq \frac{m^2}{2} p_{max}$, we have

$$(1 - \sum_{j,i} p^\star_{ij}) \leq P_e(n|\Pi) \leq (1 - \sum_{j,i} p^\star_{ij}) + \gamma_\Pi \cdot 2^{-n\rho_\Pi}. \tag{5.9}$$

PROOF For each j, consider the IHS obtained from $\mathcal{H}$ by conditioning the probability on the class π_j:

$$\mathcal{H}_j \triangleq \langle \pi_j, \mathcal{O}, p_j(\cdot), p_j(\cdot|\cdot) \rangle,$$

where for each $s \in \pi_j$,

$$p_j(s) \triangleq \frac{p(s)}{p(\pi_j)}$$

and for each o,

$$p_j(o|s) \triangleq p(o|s).$$

Fix $n \geq 0$. We equivalently concentrate ourselves on the success probability $P_{succ} = 1 - P_e$. Let $g_j : \mathcal{O}^n \to \pi_j$ be a MAP function for $\mathcal{H}_j$ and $P_{j,succ}(n) \triangleq \Pr(g_j(O^n) = S)$ be the corresponding success probability. It is an easy matter to check that, for each j

$$\Pr(g(O^n, \pi_j) = S|\Pi = \pi_j) = P_{j,succ}(n). \tag{5.10}$$

Using (5.10), we can decompose the success probability given Π thus:

$$P_{succ}(n|\Pi) = \sum_j P_{j,succ}(n)p(\pi_j)\,. \tag{5.11}$$

We consider now separately the probabilities $P_{j,succ}(n)$. Applying Theorem 3.1.7 to the IHS $\mathcal{H}_j$, we obtain

$$\sum_i \frac{p^\star_{ij}}{p(\pi_j)} - \frac{|I_j|^2}{2} p^\star_j \cdot 2^{-n\rho_j} \leq P_{j,succ}(n) \leq \sum_i \frac{p^\star_{ij}}{p(\pi_j)}. \tag{5.12}$$

Plugging inequality (5.12) into (5.11), we obtain:

$$\sum_{j,i} p^\star_{ij} - \sum_j \frac{|I_j|^2}{2} p^\star_j \cdot 2^{-n\rho_j} p(\pi_j) \leq P_{succ}(n|\Pi) \leq \sum_{j,i} p^\star_{ij}. \tag{5.13}$$

Now, since $\sum_j |I_j| = m$, we have

$$\sum_j \frac{|I_j|^2}{2} p^\star_j \cdot 2^{-n\rho_j} p(\pi_j) \leq \frac{m^2}{2} p_{max} \cdot 2^{-n\rho_\Pi},$$

which concludes the proof. □

Concerning leakage, we have the following result, which also identifies the situation of 0-leakage as policy compliance.

Theorem 5.3.7

$$\lim_{n\to\infty} \mathcal{L}^\Pi(n) = \log \frac{\sum_{j,i} p^\star_{ij}}{\sum_j p^\star_j}\,. \tag{5.14}$$

Moreover, $\mathcal{L}^\Pi(n) = 0$ for each $n \geq 0$ if and only if $\mathcal{H}$ is policy compliant.

PROOF It is immediate to check that $P_{succ}(0|\Pi) = \sum_j p^\star_j$; then the first part follows from Proposition 5.3.6. Concerning the second part, assume first that $\mathcal{L}^\Pi(n) = 0$ for each n. Then, $\lim_{n\to\infty} \mathcal{L}^\Pi(n) = \log \frac{\sum_{j,i} p^\star_{ij}}{\sum_j p^\star_j} = 0$. This implies that $\sum_{j,i} p^\star_{ij} = \sum_j p^\star_j$: the only possibility for this to be the case is that for each j, $|I_j| = 1$ (recall that p_j is the maximum probability of a secret in policy class π_j). That is, each class π_j is contained in exactly one class C_i of $\equiv$: $\sim_\pi \subseteq \equiv$. On the other hand, if $\sim_\pi \subseteq \equiv$, then for each j we have $|I_j| = 1$, and again $\sum_{j,i} p^\star_{ij} = \sum_j p^\star_j$. Hence $\log \frac{\sum_{j,i} p^\star_{ij}}{\sum_j p^\star_j} = 0$. Since $\mathcal{L}^\Pi(n) \geq 0$ and $\mathcal{L}^\Pi(n)$ is a monotonically non-decreasing function of n, it follows $\mathcal{L}^\Pi(n) = 0$ for each n. □

When specialised to the uniform prior distribution, the previous result takes a particularly simple form. Recall that $t = |\mathcal{S}/\sim_\pi|$ and $m = |\mathcal{S}/(\equiv \cap \sim_\pi)|$.

Corollary 5.3.8 *If the a priori distribution on $\mathcal{S}$ is uniform then*

$$\lim_{n\to\infty} \mathcal{L}^\Pi(n) = \log \frac{m}{t}.$$

PROOF Under the uniform distribution, for each $i = 1, \ldots, m \quad p^\star_{ij} = \frac{1}{|\mathcal{S}|}$ and so the numerator of (5.14) becomes $\frac{m}{|\mathcal{S}|}$, while the denominator $\frac{t}{|\mathcal{S}|}$. □

Example 22 *Let us consider again Example 21 and estimate the leakage of programs c_1 and c_2, both in the absence and in the presence of the policy $\sim_\pi$. Recall that $\mathcal{S} = \{0, \ldots, 9\}^{16}$, $\mathcal{O}_1 = \{0, \ldots, 9\}^3 \cup \{-1\}$ and $\mathcal{O}_2 = \{0, \ldots, 9\}$. Moreover, a uniform distribution on the secrets is assumed. Let us first compute leakage when the policy is ignored. Then clearly program c_1 is less secure than program c_2. Indeed, applying Corollary 3.1.11(2), on the limit we have*

$$\lim_{n\to\infty} \mathcal{L}_1(n) = \log 10^3 > \log\left(\binom{10}{2} + 10\right) = \log 55 = \lim_{n\to\infty} \mathcal{L}_2(n) \quad (5.15)$$

where $\mathcal{L}_i$ denotes leakage caused by program c_i, for $i = 1, 2$.

We now compute leakage given the policy $\sim_\pi$. Recall that the number of policy classes is $t = 10^4$. It is easy to compute the number of intersection classes, $m_i \triangleq \mathcal{S}/(\equiv_i \cap \sim_\pi)$, for $i = 1, 2$. We have: $m_1 = 10^4$ for c_1 (indeed, $(\equiv_1 \cap \sim_\pi) = \sim_\pi$) and $m_2 = 10^5$ for c_2 (indeed, $s(\equiv_2 \cap \sim_\pi)s'$ requires the five least significant digits of s, s' be the same). Applying Corollary 5.3.8, we

therefore obtain:

$$\lim_{n\to\infty} \mathcal{L}_1^{\Pi}(n) = \log \frac{10^4}{10^4} = 0 < 3.321 \approx \log 10 = \log \frac{10^5}{10^4} = \lim_{n\to\infty} \mathcal{L}_2^{\Pi}(n)\,. \tag{5.16}$$

That is, c_1 leaks nothing more than that implied by the policy (in fact, $\mathcal{L}_1(n)$ is constantly 0, and c_1 policy-compliant), while c_2 leaks, on the limit, one decimal digit, the fifth. To get precise estimates for c_2, we can apply Proposition 5.3.6, where, as easily seen, $\rho_\Pi = 1$ and $\Gamma_\Pi = \frac{1}{2}\cdot 10^{-10}$. E.g., we see that $n = 5$ observations are sufficient to gain the attacker 3 bits out of the ≈ 3.321 that are available on the limit: $\mathcal{L}_2^{\Pi}(5) \geq 3$.

To conclude the section, we note that the previous results can of course be specialised to the case when $\mathcal{H}$ is a multi-run IHS $\mathcal{H}^{\mathbf{u}}$. In particular, $\equiv_{\mathbf{u}} \cap \sim_\pi$ is, more explicitly, $\equiv_1 \cap \ldots \cap \equiv_l \cap \sim_\pi$.

5.4 Risk level of integrity policies

So far we have studied the confidentiality guaranteed by a system, by analysing and quantifying information that flows from secret inputs to (public) outputs, viewing the security of a system as the level of secrecy it can guarantee. When we analyse scenarios where the adversary can directly interfere with the system, however, we cannot confine ourselves to study the level of confidentiality. Another important issue is integrity, that concerns the accuracy and completeness of the results returned by systems against intentional, unauthorised or accidental changes. The aim of this section is to study risks connected to integrity. We analyse here the situation from the perspective of a user who wants to detect if an active adversary is exercising any undue influence on a deployed system, e.g. by suppressing or substituting part of the legitimate input.

Our scenario is characterised by the following, informally stated assumptions. (a) The user can observe/experiment on the system several times; (b) the outcomes of these observations are i.i.d.; (c) a 'black-box' specification of the correct system behaviour is available to the user; (d) an attack will result in a behaviour that is different from the specified one. Assumption (b) is somewhat strong, but is practically appropriate in many situations[3]. Without assumptions (c,d), there would be no way

[3]E.g., when the user may re-start the system; or when the system exhibits a cyclic behaviour and the observations are performed at random time intervals.

to detect an attack. The task of the user is to devise an *integrity policy* that would allow him to conclude, depending on the outcomes of his experiments, if the system is under attack or not. The general strategy will be to first define a set of "uncorrect" or "suspect" (probabilistic) behaviours. These are the behaviours that exhibit an appreciable deviation from the specification. For example, one may deem as suspect a behaviour where, upon repeated inspections, more than 10% of the times the system is found probing a remote host for open TCP ports, thus pointing to an infection. We stress that it is not possible to assess the risk of false negatives without postulating the existence of such a set of behaviours.

The user's task reduces then to decide whether the observed behaviour of a system is suspect or not. Due to the probabilistic nature of both the specification and the observed system, there are inherent risks in taking such a decision. A natural requirement is that, once the policy – the set of suspect behaviours – is given, the decision rule should be designed so as to minimise these risks, in the following sense. While a small risk of false positives (report a non-existent attack) can be tolerated, the decision rule should make the risk of false negatives (miss an attack) as close to zero as possible. It should come as no surprise that this task, and the resulting risk levels, can be analysed in terms of hypothesis testing (CT06; CS04), as we argue below.

Formally, we let $\mathcal{O}$ be a nonempty, finite set of observables. We pose no restrictions to the nature of this set: it might contain values, security-relevant event traces, etc. Let $\mathcal{D}$ be the set of all probability distributions on $\mathcal{O}$. We let the specification of the correct behaviour of the system be a $p(\cdot) \in \mathcal{D}$, and an integrity policy be a subset $\emptyset \neq \Gamma \subseteq \mathcal{D}$. Intuitively, Γ represents the set of "suspect" behaviours.

A user's decision strategy is defined by a family $\{A_n\}$, with $A_n \subseteq \mathcal{O}^n$ ($n \geq 1$), of acceptance regions for $p(\cdot)$. The meaning of this is that, upon observing a n-sequence o^n, produced i.i.d. by the system under consideration, if $o^n \in A_n$ the user accepts $p(\cdot)$ as the real behaviour of the system (no attack); if $o^n \in A_n^c$, he accepts that some $q(\cdot) \in \Gamma$ is the real behaviour of the system (attack). This decision can be affected by one of two types of errors:

- *false positive*: the real distribution is $p(\cdot)$, but o^n falls outside A_n, so the user believes that the system is under attack while instead it is secure. This happens with probability $\alpha_n \triangleq p(A_n^c)$;
- *false negative*: the real distribution is $q(\cdot) \in \Gamma$, but o^n falls in A_n, so the user believes that the system is secure, while instead

it has been attacked. This happens with worst-case probability $\beta_n \triangleq \sup_{q(\cdot)\in\Gamma} q(A_n)$.

In order to measure the risk level, we can consider their sum. More precisely:

Definition 5.4.1 (Integrity policy and risk level) *An* integrity policy *is a pair $(p(\cdot), \Gamma)$ with $\Gamma \neq \emptyset$. Given a family $\{A_n\}$ of acceptance regions, the* risk level after n observations *of the policy under $\{A_n\}$ is, for α_n and β_n as defined above:*

$$r_n \triangleq \alpha_n + \beta_n.$$

Let $0 \leq \varepsilon < 1$. The family $\{A_n\}$ is

- ε-sound *if $\beta_n \to 0$ and for any n large enough $\alpha_n \leq \varepsilon$;*
- sound *if it is ε-sound for some ε;*
- *(asymptotically)* optimal *if it is sound and for any sound family $\{B_n\}$ with error probabilities (α'_n, β'_n), one has:* $\liminf_{n\to\infty} -\frac{1}{n}\log\beta_n \geq \limsup_{n\to\infty} -\frac{1}{n}\log\beta'_n$.

In essence, an optimal acceptance strategy achieves small probability of false positive, and probability of false negative approaching 0 faster than any other sound strategy. The resulting risk r_n is thus the "inherent risk" connected to the policy $(p(\cdot), \Gamma)$. It remains to show that optimal policies exist: we do so below. We also show that $r_n \to 0$ and characterise the optimal rate at which β_n vanishes, which again only depends on $(p(\cdot), \Gamma)$.

For $\Gamma \subseteq \mathcal{D}$, we let

$$D(p\|\Gamma) \triangleq \inf_{q(\cdot)\in\Gamma} D(p\|q),$$

where $D(\cdot\|\cdot)$ is the Kullback-Leibler distance (see Definition 2.4.1). Let $n \geq 1$. Given $o^n \in \mathcal{O}^n$, recall that $t_{o^n}(\cdot)$ denotes the type of o^n (see Definition 2.4.3). We now define an optimal family of acceptance regions for $(p(\cdot), \Gamma)$. For each $n \geq 1$, let

$$\delta_n \triangleq \frac{|\mathcal{O}|\log n}{n} \qquad \text{and} \qquad A_n^* \triangleq \{o^n \in \mathcal{O}^n \mid D(t_{o^n}\|p) < \delta_n\}.$$

Intuitively, we put in A_n^* only those sequences whose empirical distribution is "very close" to $p(\cdot)$, so as to guarantee that β_n vanishes quite fast.

At the same time, A_n^* is large enough to guarantee that α_n too approaches 0, although possibly not as fast as β_n.

Below, we use the fact that $\mathcal{D}$ inherits its topology from $\mathbb{R}^{|\mathcal{O}|}$. The assumption $p \notin \overline{\Gamma}$ means that p is not "too close" to the class of suspect behaviours. The proof of the following result goes along the lines of similar results in (CS04, Sec.2).

Theorem 5.4.2 (optimal strategy) *Assume $p \notin \overline{\Gamma}$. Then $\{A_n^*\}$ is an optimal family of acceptance regions for the policy $(p(\cdot), \Gamma)$. Moreover, $r_n \to 0$ and $\lim_{n\to\infty}(-\frac{1}{n}\log\beta_n) = D(p\|\Gamma)$.*

Before discussing this result, recall two (variations of) Theorems 2.4.5 and 2.4.7 that will be used in the proof. Remember that $\mathcal{P}_n$ is the set of n-types and for each $q(\cdot) \in \mathcal{P}_n$, $\mathcal{T}_q^n \triangleq \{o^n \in \mathcal{O}^n | t_{o^n}(\cdot) = q(\cdot)\}$ is the type class of $q(\cdot)$.

Lemma 5.4.3 *For each $n \geq 1$, $|\mathcal{P}_n| \leq (n+1)^{|\mathcal{O}|-1}$.*

Lemma 5.4.4 *For any distribution $p(\cdot)$ on $\mathcal{O}$ and any $q(\cdot) \in \mathcal{P}_n$, for any n large enough:*

$$\frac{2^{-nD(q\|p)}}{(n+1)^{|\mathcal{O}|-1}} \leq p(\mathcal{T}_q^n) \leq 2^{-nD(q\|p)}.$$

Theorem 5.4.5 *Assume $p \notin \overline{\Gamma}$. Let $\{A_n\}$ be a sound family of acceptance regions and α_n, β_n be the corresponding error probabilities. Then*

$$\limsup_{n\to\infty}(-\frac{1}{n}\log\beta_n) \leq D(p\|\Gamma).$$

PROOF According to Theorem 2.2 of (CS04), for any two distinct distributions on $\mathcal{O}$, say $p_1(\cdot)$ and $p_2(\cdot)$, such that there exists $\varepsilon < 1 : p_1(A_n^c) \leq \varepsilon$ for any n large enough, we have that:

$$\limsup_{n\to\infty}(-\frac{1}{n}\log(p_2(A_n))) \leq D(p_1\|p_2). \tag{5.17}$$

Now, take $p_1(\cdot) = p(\cdot)$ and take any distribution $p_2(\cdot) \in \Gamma$. By definition of β_n, we have $-\frac{1}{n}\log\beta_n \leq -\frac{1}{n}\log(p_2(A_n))$, and this inequality is preserved by lim sup. Now, by (5.17) we have that $\limsup_{n\to\infty}(-\frac{1}{n}\log\beta_n) \leq D(p\|p_2)$. This inequality holds for *any* distribution $p_2(\cdot) \in \Gamma$: this implies that it still holds when we replace the RHS with $\inf_{p_2(\cdot)\in\Gamma} D(p\|p_2)$. □

Theorem 5.4.6 *Assume $p(\cdot) \notin \overline{\Gamma}$. Let α_n, β_n be the error probabilities of $\{A_n^*\}$. Then:*

1. $\lim_{n\to\infty} \alpha_n = 0$;
2. $\lim_{n\to\infty} \beta_n = 0$; *and*
3. $\liminf_{n\to\infty} (-\frac{1}{n} \log \beta_n) \geq D(p\|\Gamma)$.

PROOF The proof of part 1) can be found in (CS04, Theorem 2.3). Concerning parts 2) and 3), we can write $\beta_n = q_n^*(A^n)$, where[4]

$$q_n^*(\cdot) \triangleq \text{argmax}_{q(\cdot)\in\overline{\Gamma}} q(A_n^*).$$

Now

$$\begin{aligned}
\beta_n = q_n^*(A_n^*) &= \sum_{q(\cdot)\in\mathcal{P}_n : D(q\|p)<\delta_n} q_n^*(\mathcal{T}_q^n) \\
&\leq \sum_{q(\cdot)\in\mathcal{P}_n : D(q\|p)<\delta_n} 2^{-n\xi_n} \qquad \text{(Lemma 5.4.4)} \\
&\leq (n+1)^{|\mathcal{O}|-1} 2^{-n\xi_n} \qquad \text{(Lemma 5.4.3)} \qquad (5.18)
\end{aligned}$$

with

$$\xi_n \triangleq \min_{\substack{q(\cdot)\in\mathcal{P}_n: \\ D(q\|p)<\delta_n}} D(q\|q_n^*) = D(s_n\|q_n^*),$$

for some $s_n(\cdot)$ such that $D(s_n\|p) < \delta_n \to 0$. Therefore $s_n(\cdot) \to p(\cdot)$, since convergence in $D(\cdot\|\cdot)$ implies convergence in $\|\cdot\|_1$. Let us take now a convergent subsequence[5] of $\{q_n^*(\cdot)\}$, say $\{q_{i_n}^*(\cdot)\}$ and let $q^{**}(\cdot) \in \overline{\Gamma}$ be its limit. Due to the lower semi-continuity of $D(\cdot\|\cdot)$ (CS04), we have

$$\liminf_{n\to\infty} \xi_n \geq D(p\|q^{**}) \geq D(p\|\Gamma). \qquad (5.19)$$

Now, using the inequalities (5.18) and (5.19) and the fact that $p(\cdot) \notin \overline{\Gamma}$ implies $D(p\|\Gamma) > 0$, we obtain both the thesis 2) and 3). □

The proof of Theorem 5.4.2 simply reduces as follows.

PROOF [Theorem 5.4.2]
The thesis follows applying both Theorem 5.4.5 and Theorem 5.4.6. □

[4]This $q_n^*(\cdot)$ exists since the function $q(\cdot) \mapsto q(A_n^*)$ is continuous over the set $\overline{\Gamma}$, closed and bounded.

[5]The existence of this subsequence is guaranteed by Bolzano-Weierstrass.

Example 23 *Assume $\mathcal{O} = \{0, 1\}$, with $0/1$ representing absence/presence of a given security-relevant event, e.g. port-probing. Through simulations on a non-infected system, it has been established that the correct behaviour is $p(1) = 0.01$. Consider now the integrity policy given by $\Gamma = \{q(\cdot) \in \mathcal{D} | q(1) > 0.1\}$. Now $D(p\|\Gamma) \approx 0.103$: according to Theorem 5.4.2, $\beta_n \approx 2^{-0.103 \cdot n}$. In order to guarantee $\beta_n \approx 10^{-9}$, it is sufficient to collect about $n \approx 290$ observations. Moreover, it can be seen that $\alpha_n \approx 1/n$. Finally, note the more restrictive policy (larger Γ), the lower the rate $D(p\|\Gamma)$.*

5.5 Concluding remarks

We have proposed models to analyse security threats against probabilistic systems posed by a class of active adversaries that control part of the system input, in a multi-run scenario. We have quantified the degree of violation of a declassification policy, thus combining quantitative and qualitative facets in the study of Information Flow. We have finally cast the problem of deciding whether a given system is under attack in terms of hypothesis testing, characterising the asymptotically optimal decision strategy.

Chapter 6

Active attackers: adaptive scenario

In this chapter, we tackle the adaptive case, presenting an information-theoretic model and deriving several general results on the limits of adaptive adversaries. We assume that, based on a secret piece of information, the mechanism responds to a sequence of queries/actions, adaptively submitted by an adversary, thus producing a sequence of answers/observations. Responses to individual queries are in general probabilistic, either because of the presence of noise or by design. Moreover, the mechanism is stateless, thus answers are independent from one another. The adversary is assumed to know the distribution according to which the secret has been generated (the prior) and the input-output behaviour of the mechanism. An adaptive adversary can choose the next query based on past observations, according to a predefined strategy. Once a strategy and a prior are fixed, they together induce a probability space over sequences of observations. Observing a particular sequence gives the adversary some information that modifies his belief about the secret, possibly reducing his uncertainty. We measure information leakage as the *average reduction in uncertainty*. An important aspect of our approach is that we work with a *generic* measure of uncertainty, $U(\cdot)$. Formally, $U(\cdot)$ is a a real-valued function over the set of probability distributions on the secret, which represents possible beliefs of the adversary. Just two properties are assumed of $U(\cdot)$: concavity and continuity. Note that leakage functions commonly employed in QIF, such as Shan-

non entropy, guessing entropy and error probability (the additive version of min-entropy), do fall in this category.

A central theme of our study is the comparison of adaptive with the simpler non-adaptive strategies. All in all, our results indicate that, for reasonably powerful adversaries, there is no dramatic difference between the two, in terms of difficulty of analysis. A more precise account of our contributions follows.

1. We put forward a general model of adaptive QIF; we identify mild general conditions on the leakage function under which it is possible to derive general and significant results on adaptive QIF in this model.

2. We compare the difficulty of analysing mechanisms under adaptive and non-adaptive adversaries. We first note that, for the class of mechanisms admitting a "concise" syntactic description - e.g. devices specified by a boolean circuit - the analysis problem is intractable (NP-hard), even if limited to very simple instances of the non-adaptive case. This essentially depends on the fact that such mechanisms can feature exponentially many actions in the syntactic size. In the general case, we show that non-adaptive finite strategies are as efficient as adaptive ones, up to an *expansion factor* in their length bounded by the number of distinct actions available. Practically, this indicates that, for mechanisms described in explicit form (e.g. by tables, like a DB) hence featuring an "affordable" number of actions available to the adversary, it may be sufficient to assess resistance of the mechanism against non-adaptive strategies. This is important, because simple analytical results are available for such strategies (BP12a).

3. We show that the maximum leakage is the same for both adaptive and non-adaptive adversaries, and only depends on an indistinguishability equivalence relation over the set of secrets.

4. We show that maximum information leakage over a finite horizon can be expressed in terms of a Bellman equation. This equation can be used to compute optimal finite strategies recursively. As an example, we show how to do that using Markov Decision Processes (MDP's) and backward induction.

6.1 An extended model: attack trees

So far we have tackled attack scenarios where the adversary can re-execute multiple times the system, collecting several observations, all related to the same secret, but, once given it, independent from each other. In this section we analyse the adaptive case, where instead, after each collected observation, the adversary can update his belief about the secret and, depending on it, choose the next value for the untrusted input, aiming to maximise the gained information.

Let us assume that, based on a secret piece of information $X \in \mathcal{X}$, the system responds to a sequence of queries/actions $a_1, a_2, \ldots$ ($a_i \in Act$), adaptively submitted by an adversary, thus producing a sequence of answers/observations $Y \in \mathcal{Y}^*$. Responses to individual queries are probabilistic, either because of the presence of noise or by design. Suppose that the system is stateless; thus answers are independent from one another. As before, the adversary is assumed to know the distribution $p(\cdot)$ according to which X has been generated (the prior) and the input-output behaviour of the system, given by the matrix $p(\cdot|\cdot,\cdot) \in [0,1]^{(\mathcal{S}\times Act)\times\mathcal{O}}$. Differently from previous scenarios, an adaptive adversary can choose the next query based on *past* observations, according to a predefined strategy. Once a strategy and a prior are fixed, they together induce a probability space over sequences of observations. Observing a particular sequence gives the adversary some information that modifies his belief about X, possibly reducing his uncertainty.

Before describing more in detail the model, let us give the basic definitions that will be useful in the following.

6.1.1 Basic definitions

Let $\mathcal{S} = \langle \mathcal{X}, \mathcal{Y}, Act, \{M_a : a \in Act\}\rangle$ be an action-based randomisation mechanism, as defined in Definition 5.1.1, where $\mathcal{X}, \mathcal{Y}$ and Act are respectively the sets of secrets, observations and *actions* (or *queries*) and, for each $a \in Act$, M_a is a stochastic matrix of dimensions $|\mathcal{X}| \times |\mathcal{Y}|$.

We measure information leakage as the *average reduction in uncertainty*. A central point of our framework is that we work with a *generic* measure of uncertainty, $U(\cdot)$. Formally, $U(\cdot)$ is a a real-valued function over the set of probability distributions on $\mathcal{X}$, which represents possible beliefs of the adversary. Just two properties are assumed of $U(\cdot)$: concavity and continuity.

Definition 6.1.1 (Uncertainty) *Let $\mathcal{P}(\mathcal{X})$ be the set of all probability distributions on $\mathcal{X}$. A function $U : \mathcal{P}(\mathcal{X}) \to \mathbb{R}$ is an* uncertainty measure *if it is concave and continuous over $\mathcal{P}(\mathcal{X}) \subseteq \mathbb{R}^{|\mathcal{X}|}$.*

The role of concavity can be intuitively explained as follows. Suppose the secret is generated according to either a distribution $p(\cdot)$ or to another distribution $q(\cdot)$, the choice depending from a coin toss, with probability λ of getting head. The coin toss introduces *extra randomness* in the generation process. Therefore, the overall uncertainty of the adversary about the secret, $U\big(\lambda \cdot p + (1-\lambda) \cdot q\big)$, should be *no less* than the average uncertainty of the two original generation processes considered separately, that is $\lambda U\big(p\big) + (1-\lambda) U\big(q\big)$. As a matter of fact, most uncertainty measures in QIF do satisfy this concavity. Continuity is a technical requirement that comes into play only in Theorem 6.3.5.

Example 24 *The following entropy functions, and variations thereof, often considered in the quantitative security literature as measures of the difficulty or effort necessary to a passive adversary to identify a secret X, where X is a random variable over $\mathcal{X}$ distributed according to some $p(\cdot)$, are easily proven to be uncertainty measures in our sense:*

- Shannon entropy*: $H(p) \triangleq -\sum_{x \in \mathcal{X}} p(x) \log p(x)$, with $0 \log 0 = 0$;*
- Error probability entropy*: $E(p) \triangleq 1 - \max_{x \in \mathcal{X}} p(x)$;*
- Guessing entropy*: $G(p) \triangleq \sum_{i=1}^{n} i \cdot p(x_i)$ with $p(x_1) \geq p(x_2) \geq \ldots \geq p(x_n)$.*

As we have already mentioned, the adversary here can rely on the past observations in order to choose the next query, aiming to minimise his uncertainty. In order to make this choice, he follows a certain strategy, that can be seen as a function that associates each sequence of observations with a certain query.

Definition 6.1.2 (Strategy) *A* strategy *is a partial function $\sigma : \mathcal{Y}^* \to Act$ such that $\mathrm{dom}(\sigma)$ is non-empty and prefix-closed. A strategy is* finite *if $\mathrm{dom}(\sigma)$ is finite. The* length *of a finite strategy is defined as $\max\{l \geq 0 : y^l \in \mathrm{dom}(\sigma)\} + 1$.*

For each integer $n \geq 0$ we let $y^n, w^n, z^n, \ldots$ range over sequences in $\mathcal{Y}^n$; given $y^n = (y_1, \ldots, y_n)$ and $0 \leq j \leq n$, we let y^j denote the first j components of y^n, $(y_1, \ldots, y_j)$.

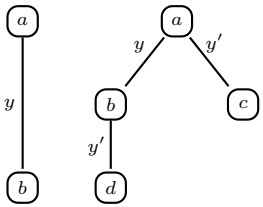

Figure 13: Two strategy trees.

Definition 6.1.3 (Truncation) *Given a strategy σ and an integer $n \geq 0$, the* truncation *of σ at level n, denoted as $\sigma\backslash n$, is the finite strategy $\sigma_{|\cup_{0\leq i\leq n}\mathcal{Y}^i}$.*

A finite strategy of length l is *complete* if $\mathrm{dom}(\sigma) = \cup_{0\leq i\leq l-1}\mathcal{Y}^i$. A strategy σ is *non-adaptive* if whenever y^n and w^n are two sequences of the same length then $\sigma(y^n) = \sigma(w^n)$. That is, the decision of which action to play next only depends on the number of past actions.

Remark 6.1.4 *Finite non-adaptive strategies are necessarily complete.*

We note that strategies can be described as trees, with nodes labelled by actions and arcs labelled by observations, in the obvious way. Any non-adaptive strategy also enjoys a simpler representation as a finite or infinite list of actions: we write $\sigma = [a_1, \ldots, a_i, \ldots]$ if $\sigma(y^{i-1}) = a_i$, for $i = 1, 2, \ldots$.

Example 25 *Strategies $\sigma = [\varepsilon \mapsto a, y \mapsto b]$ and $\sigma' = [\varepsilon \mapsto a, y \mapsto b, y' \mapsto c, yy' \mapsto d]$ can be represented as in Figure 13. Note that the height of the tree is one less than the length of the strategy.*

6.1.2 Adaptive quantitative information flow

Informally, we consider an adversary who repeatedly queries a system, according to a predefined *finite* strategy. At some point, the strategy will terminate, and the adversary will have collected a sequence of observations $y^n = (y_1, \ldots, y_n)$. Note that both the length n and the probability of the individual observations y_i, hence of the whole y^n, will in general depend both on X and on the strategy played by the adversary. In other words, the distribution $p(\cdot)$ of X *and* the strategy σ together induce a probability distribution on a subset of all observation sequences: the

ones that may arise as a result of a complete interaction with the system, according to the played strategy.

Formally, let $p(\cdot)$ be any given probability distribution over $\mathcal{X}$, which we will often refer to as the prior. For each finite strategy σ, we define a joint probability distribution $p_\sigma(\cdot)$ on $\mathcal{X} \times \mathcal{Y}^*$, depending on σ and on $p(\cdot)$, as follows.

Definition 6.1.5 (Joint probability) *For each finite strategy σ, let $p_\sigma(x, \varepsilon) \triangleq 0$ and, for each $j \geq 0$:*

$$p_\sigma(x, y_1, \ldots, y_j, y_{j+1}) \triangleq \begin{cases} p(x) \cdot p_{a_1}(y_1|x) \cdot \cdots \cdot p_{a_j}(y_j|x) p_{a_{j+1}}(y_{j+1}|x) \\ \qquad \textit{if } y^j \in \mathrm{dom}(\sigma), y^j y_{j+1} \notin \mathrm{dom}(\sigma) \\ 0 \qquad \textit{otherwise} \end{cases}$$

where $a_i = \sigma(y^{i-1})$ for $i = 1, \ldots, j+1$.

In case $\sigma = [a]$, a single action strategy, we will often abbreviate $p_{[a]}(\cdot)$ as $p_a(\cdot)$. Note that the support of $p_\sigma(\cdot)$ is finite, in particular $\mathrm{supp}(p_\sigma) \subseteq \mathcal{X} \times \{y^j y : j \geq 0, y^j \in \mathrm{dom}(\sigma), y^j y \notin \mathrm{dom}(\sigma)\}$. This definition applies also to the non adaptive model described in the previous chapter. In the non adaptive case, indeed, untrusted inputs can be viewed as actions taken by the adversary and the sequence $\mathbf{u}$, containing all possible values for the untrusted input, coincides with the strategy that plays all actions in $\mathcal{U}$ once. For each $s \in \mathcal{S}$ such that $p(s) > 0$, the probability $p_{\mathbf{u}}(\mathbf{o}|s)$ defined in Definition 5.1.2 corresponds to the ratio $\frac{p_{\mathbf{u}}(s,\mathbf{o})}{p(s)}$, where the numerator is the joint probability defined above.

Let (X, Y) be a pair of random variables with outcomes in $\mathcal{X} \times \mathcal{Y}^*$, jointly distributed according to $p_\sigma(\cdot)$: here X represents the secret and Y represents the sequence of observations obtained upon termination of the strategy. As before, we shall often use such shortened notations as: $p_\sigma(x|y^n)$ for $\Pr(X = x|Y = y^n)$, $p_\sigma(y^n)$ for $\Pr(Y = y^n)$, and so on. Explicit formulas for computing these quantities can be easily derived from the definition of $p_\sigma(\cdot)$ and using Bayes rule. We will normally keep the dependence of (X, Y) from $p(\cdot)$ and σ implicit. When different strategies are being considered at the same time and we want to stress that we are considering Y according to the distribution induced by a specific σ, we will write it as Y_σ.

Consider a prior $p(\cdot)$ and a *finite* strategy σ, and the corresponding pair of random variables (X, Y).

Definition 6.1.6 (Average and conditional uncertainty and information gain) *The following quantities respectively express* average uncertainty, conditional uncertainty *and* information gain *about X, that may result from interaction according to strategy σ (by convention, we let here y^n range over sequences with $p_\sigma(y^n) > 0$):*

$$\begin{aligned} U(X) &\triangleq U\big(p\big) \\ U(X|Y) &\triangleq \sum_{y^n} p_\sigma(y^n) U\big(p_\sigma(\cdot|y^n)\big) \\ I(X;Y) &\triangleq U(X) - U(X|Y). \end{aligned} \tag{6.1}$$

Note that, in the case of Shannon entropy, $I(X;Y)$ coincides with the familiar mutual information (see Definition 2.1.8), traditionally measured in bits. In the case of error entropy, $I(X;Y)$ is the additive leakage (see Definition 3.1.4), also called *advantage* in the cryptographic literature, see e.g. (DS05) and references therein.

In the rest of the section, unless otherwise stated, we let $U(\cdot)$ be an arbitrary uncertainty function. The following fact about $I(X;Y)$ follows from the concavity of $U(\cdot)$ and Jensen's inequality, plus routine calculations on probability distributions.

Lemma 6.1.7 *$I(X;Y) \geq 0$. Moreover $I(X;Y) = 0$ if X and Y are independent.*

Given the above definitions, the concept of adaptive QIF can now be defined quite simply.

Definition 6.1.8 (QIF **under adaptive adversaries)** *Let $\mathcal{S}$ be a system and $p(\cdot)$ be a prior over $\mathcal{X}$.*

1. *For a* finite *strategy σ, let $I_\sigma(\mathcal{S},p) \triangleq I(X;Y)$.*
2. *For an* infinite *strategy σ, let $I_\sigma(\mathcal{S},p) \triangleq \lim_{l\to\infty} I_{\sigma\backslash l}(\mathcal{S},p)$.*
3. *(*Maximum IF *under $p(\cdot)$) $I_\star(\mathcal{S},p) \triangleq \sup_\sigma I_\sigma(\mathcal{S},p)$.*

Note that $l' \geq l$ implies $I_{\sigma\backslash l'}(\mathcal{S},p) \geq I_{\sigma\backslash l}(\mathcal{S},p)$, hence the limit in (2) always exists. Taking the distribution that achieves the maximum leakage, we can define an analog of capacity (see Definition 2.1.10).

Definition 6.1.9 (Adaptive secrecy capacity)

$$C(\mathcal{S}) \triangleq \sup_{p(\cdot)\in\mathcal{P}(\mathcal{X})} I_\star(\mathcal{S},p).$$

6.1.3 Attack trees

It is sometimes useful to work with a pictorial representation of the adversary's attack steps, under a given strategy and prior. This can take the form of a tree, where each node represents an adversary's *belief* about the secret, that is, a probability distribution over $\mathcal{X}$. The tree describes the possible evolutions of the belief, depending on the strategy and on the observations. We formally introduce such a representation below: it will be extensively used in the examples in Section 6.1.4. Note that *attack* trees are different from *strategy* trees

Definition 6.1.10 (History and updated probability) *A* history *is a sequence $h \in (Act \times \mathcal{Y})^*$. Let $h = (a_1, y_1, \ldots, a_n, y_n)$ be such a history. Given a prior $p(\cdot)$, we define the* update of $p(\cdot)$ after h*, denoted by $p^h(\cdot)$, as the distribution on $\mathcal{X}$ defined by*

$$p^h(x) \quad \triangleq \quad p_{\sigma_h}(x|y^n) \tag{6.2}$$

where $\sigma_h = [a_1, \ldots, a_n]$, provided $p_{\sigma_h}(y^n) > 0$; otherwise $p^h(\cdot)$ is undefined.

Definition 6.1.11 (Attack trees) *The* attack tree *induced by a strategy σ and a prior $p(\cdot)$ is a tree with nodes labeled by probability distributions over $\mathcal{X}$ and arcs labeled with pairs (y, λ) of an observation and a probability.*

This tree is obtained from the strategy tree of σ as follows. First, note that, in a strategy tree, each node is identified by a unique history. Given the strategy tree for σ: (a) for each $y \in \mathcal{Y}$ and each node missing an outgoing y-labelled arc, attach a new y-labelled arc leading to a new node; (b) label each node of the resulting tree by $p^h(\cdot)$, where h is the history identifying the node, if $p^h(\cdot)$ is defined, otherwise remove the node and its descendants, as well as the incoming arc; (c) label each arc from a node h to a child hay in the resulting tree with $\lambda = p^h_a(y)$ - to be parsed as $(p^h)_{[a]}(y)$. This is the probability of observing y under a prior $p^h(\cdot)$ when submitting action a.

The concept of attack tree is demonstrated by a few examples in the next section. Here, we just note the following easy to check facts.

Remark 6.1.12 *For each leaf h of the attack tree:*

- *the label of the leaf is $p^h(\cdot) = p_\sigma(\cdot|y^n)$, where y^n is the sequence of observations in h;*

- *if we let π_h be the product of the probabilities on the edges from the root to the leaf, then $\pi_h = p_\sigma(y^n)$;*
- *each y^n such that $p_\sigma(y^n) > 0$ is found in the tree.*

As a consequence, for a finite *strategy, taking* (6.1) *into account, the uncertainty of X given Y can be computed from the attack tree as:*

$$U(X \mid Y) \quad = \quad \sum_{h \text{ a leaf}} \pi_h U(p^h)\,. \tag{6.3}$$

6.1.4 Examples

We present here a few instances of the framework introduced in the previous section. We emphasise that these examples are quite simple and only serve to illustrate our main definitions. In the rest of the section, we shall use the following notation: we let $\mathrm{u}\{x_1, \ldots, x_k\}$ denote the uniform distribution on $\{x_1, \ldots, x_k\}$.

Example 26 *An attacker gets hold of the table shown in Figure 14, which represents a fragment of a hospital database. Each row of the table contains: a numerical id followed by the ZIP code, age, discharge date and disease of an individual that has been recently hospitalised. The table does not contain personal identifiable information. The attacker gets to know that a certain target individual, John Doe (JD), has been recently hospitalised. However, the attacker is ignorant of the corresponding id in the table and any information about JD, apart from his name. The attacker's task is to identify JD, i.e. to find JD's id in the table, thus learning his/her disease. The attacker is in a position to ask a source, perhaps the hospital* DB*, queries concerning non sensitive information (ZIP code, age and discharge date) of any individual, including JD, and compare the answers with the table entries.*[1]

This situation can be modeled quite simply as an action-based mechanism $\mathcal{S}$, as follows. We pose: $Act = \{\mathrm{ZIP}, \mathrm{Age}, \mathrm{Date}\}$; $\mathcal{X} = \{1, \ldots, 10\}$, the set of possible id's, and $\mathcal{Y} = \mathcal{Y}_{\mathrm{ZIP}} \cup \mathcal{Y}_{\mathrm{Age}} \cup \mathcal{Y}_{\mathrm{Date}}$, where $\mathcal{Y}_{\mathrm{ZIP}} = \{z_1, z_2, z_3\}$, $\mathcal{Y}_{\mathrm{Age}} = \{30, 31, 65, 66, 67, 68\}$ and $\mathcal{Y}_{\mathrm{Date}} = \{d_1, d_2, d_3\}$. The conditional probability matrices reflect the behaviour of the source when queried about ZIP code, age and discharge date of an individual. We assume that the source is truthful, hence answers will match the entries of the table. For example, $p_{\mathrm{Age}}(y|1) = 1$ if $y = 65$ and 0 otherwise; $p_{\mathrm{ZIP}}(y|2) = 1$ if $y = z_1$, 0 otherwise; and so on.

[1] That this is unsafe is of course well-known from database security: the present example only serves the purpose of illustration.

id	ZIP	Age	Date	Disease
1	z_1	65	d_2	Hearth disease
2	z_1	65	d_2	Flu
3	z_1	67	d_2	Short breath
4	z_1	68	d_1	Obesity
5	z_1	68	d_1	Hearth disease
6	z_3	66	d_2	Hearth disease
7	z_3	67	d_2	Obesity
8	z_3	31	d_2	Short breath
9	z_2	30	d_3	Hearth disease
10	z_2	31	d_3	Obesity

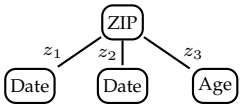

Figure 14: Medical Database and strategy tree of Example 26.

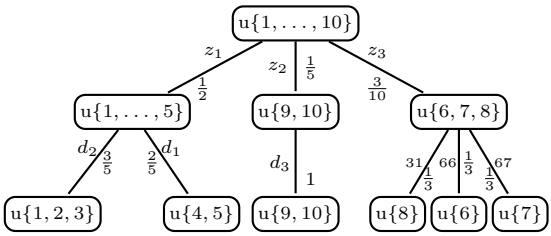

Figure 15: The attack tree of Example 26.

Note that this defines a deterministic *mechanism. Finally, since the attacker has no clues about JD's id, we set the prior to be the uniform distribution on* $\mathcal{X}$, $p(\cdot) = \mathrm{u}\{1, \ldots, 10\}$.

Assume now that, for some reason - maybe for the sake of privacy - the number of queries to the source about an individual is limited to two. Figure 14 displays a possible attacker's strategy σ*, of length 2. Figure 15 displays the corresponding attack tree, under the given prior. Note that the given strategy is not in any sense optimal. Assume we set* $U(\cdot) = H(\cdot)$*, Shannon entropy, as a measure of uncertainty. Using* (6.3)*, we can compute* $I_\sigma(\mathcal{S}, p) = H(X) - H(X|Y) = \log 10 - \frac{3}{10}\log 3 - \frac{2}{5} \approx 2.45$ *bits. With* $U(\cdot) = E(\cdot)$*, the error entropy, we have* $I_\sigma(\mathcal{S}, p) = E(X) - E(X|Y) = 0.5$ *bits.*

Example 27 (noisy version) *Consider a version of the previous mechanism where the public source queried by the attacker is not entirely truthful. In particular, for security reasons, whenever queried about age of an individual, the*

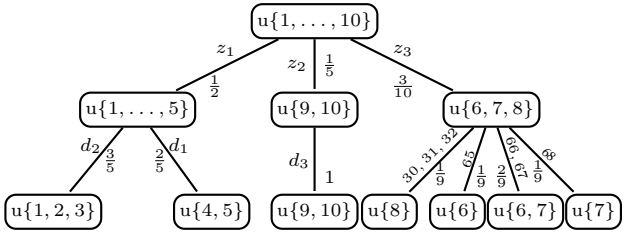

Figure 16: The attack tree of Example 27. Leaves with the same label and their incoming arcs have been coalesced.

source adds a random offset $r \in \{-1, 0, +1\}$ to the real answer. The only difference from the previous example is that the conditional probability matrix $p_{\mathrm{Age}}(\cdot|\cdot)$ is not deterministic anymore. For example, for $x = 1$, we have

$$p_{\mathrm{Age}}(y|1) \quad = \quad \begin{cases} \frac{1}{3} & \text{if } y \in \{64, 65, 66\} \\ 0 & \text{otherwise} \end{cases}$$

(also note that we have to insert $29, 32, 64$ and 69 as possible observations into $\mathcal{Y}_{\mathrm{Age}}$). Figure 16 shows the attack tree induced by the strategy σ of Figure 14 and the uniform prior in this case. If $U(\cdot) = H(\cdot)$ we obtain $I_\sigma(\mathcal{S}, p) = \log 10 - \frac{3}{10}\log 3 - \frac{8}{15} \approx 2.31$ bits; if $U(\cdot) = E(\cdot)$, instead, $I_\sigma(\mathcal{S}, p) = \frac{13}{30} \approx 0.43$ bits.

Example 28 (cryptographic devices) *As we have seen in previous sections, a cryptographic device can be abstractly modeled as a function f taking pairs of a key and a message into observations, thus, $f : \mathcal{K} \times \mathcal{M} \to \mathcal{Y}$. Assume the attacker can choose the message $m \in \mathcal{M}$ fed to the device, while the key k is fixed and unknown to him. This clearly yields an action-based mechanism $\mathcal{S}$ where $\mathcal{X} = \mathcal{K}$, $Act = \mathcal{M}$ and $\mathcal{Y}$ are the observations. If we assume the observations noiseless, then the conditional probability matrices are defined by*

$$p_m(y|k) \quad = \quad 1 \qquad \text{if and only if} \qquad f(k, m) = y\,.$$

We obtain therefore a deterministic mechanism. This is the way, for example, modular exponentiation is modeled (see Examples 10,13). More realistically, the observations will be noisy, due e.g. to the presence of "algorithmic noise". For example, assume $\mathcal{Y} \subseteq \mathbb{N}$ is the set of possible Hamming weights of the ciphertexts (this is related to power analysis attacks, see e.g. (KSWH00)). Then we may set

$$p_m(y|k) \quad = \quad \Pr(f(k, m) + N = y)$$

where N is a random variable modelling noise. For example, in the model of DES *S-Boxes considered in Example 20, $\mathcal{K} = \mathcal{M} = \{0,1\}^6$, while $\mathcal{Y} = \{0,1,2,\ldots\}$ is the set of observations: the (noisy) Hamming weight of the outputs of the target S-Box. In this case, N is taken to be the cumulative weight of the seven S-Boxes other than the target one.*

6.2 Comparing adaptive and non-adaptive strategies

Conceptually, we can classify systems into two categories, depending on the size of the set Act. Informally, the first category consists of systems with a huge - exponential, in the size of any reasonable syntactic description - number of actions. The second category consists of systems with an "affordable" number of actions. In the first category, we find, for instance, complex cryptographic hardware, possibly described via boolean circuits or other "succinct" notations (cf. the public key exponentiation algorithms considered in (KB07)). In the second category, we find systems explicitly described by tables, such as databases (Examples 26 and 27) and S-Boxes (Example 28).

6.2.1 Systems in succinct form

We argue that the analysis of such systems is in general an intractable problem, even if restricted to simple special instances of the *non-adaptive* case. We consider the problem of deciding if there is a finite strategy over a given time horizon yielding an information flow exceeding a given threshold. This decision problem is of course simpler than the problem of finding an optimal strategy over a finite time horizon: indeed, any algorithm for finding the optimal strategy can also be used to answer the first problem. We give some definitions.

Definition 6.2.1 (systems in boolean forms) *Let t, u, v be nonnegative integers. We say a mechanism $\mathcal{S} = \langle \mathcal{X}, \mathcal{Y}, Act, \{M_a : a \in Act\}\rangle$ is in (t,u,v)-*boolean form *if $\mathcal{X} = \{0,1\}^t$, $Act = \{0,1\}^u$, $\mathcal{Y} = \{0,1\}^v$ and there is a boolean function $f : \{0,1\}^{t+u} \to \{0,1\}^v$ such that for each $x \in \mathcal{X}$, $y \in \mathcal{Y}$ and $a \in Act$, $p_a(y|x) = 1$ if and only if $f(x,a) = y$. The* size *of $\mathcal{S}$ is defined as the syntactic size of the smallest boolean formula for f.*

It is not difficult to see that the class of boolean forms coincides, up to suitable encodings, with that of deterministic systems.

Definition 6.2.2 (Adaptive Bounding Problem in succinct form, ABPS) *Given a mechanism $\mathcal{S}$ in a (t, u, v)-boolean form, a prior distribution $p(\cdot)$, $l \geq 1$ and $T \geq 0$, decide if there is a strategy σ of length $\leq l$ such that $I_\sigma(\mathcal{S}; p) > T$.*

In the next theorem, we shall assume, for simplicity, the following reasonable properties of $U(\cdot)$: if $p(\cdot)$ concentrates all the probability mass on a single element, and $q(\cdot)$ is the uniform distribution, then $0 = U(p) < U(q)$. A slight modification of the argument also works without this assumption. The theorem says that even length 1 (hence non-adaptive) strategies are difficult to assess.

Theorem 6.2.3 *Assume $U(\cdot)$ satisfies the above stated property. Then the* ABPS *is* NP*-hard, even if fixing $t = v = l = 1$, and $T = 0$.*

PROOF We reduce from the satisfiability problem for boolean formulae. Let $\phi(z_1, \ldots, z_u) = \phi(\tilde{z})$ be an arbitrary boolean formula with u free boolean variables $z_1, \ldots, z_u$. We show how to build in polynomial time out of $\phi(\tilde{z})$ a mechanism $\mathcal{S}$ in $(1, u, 1)$-boolean form, and a prior $p(\cdot)$, with the following property: there is a length 1 strategy σ such that $I_\sigma(\mathcal{S}, p) > 0$ if and only if $\phi(\tilde{z})$ is satisfiable. Take $\mathcal{X} = \mathcal{Y} = \{0, 1\}$ and $Act = \{0, 1\}^u$. Let $\mathcal{S}$ be the mechanism defined by the boolean function $f(x, z_1, \ldots, z_u) = x \wedge \phi(\tilde{z})$. Let $p(\cdot)$ be the uniform prior on $\mathcal{X} = \{0, 1\}$. Now, if there is an action $\tilde{b} = (b_1, \ldots, b_u) \in Act$ such that $\phi(\tilde{b}) = 1$ ($\phi(\tilde{z})$ is satisfiable) then clearly we will have that $Y = X \wedge \phi(\tilde{b})$ is logically equivalent to X, hence $U(X|Y) = 0$. Consequently, setting $\sigma = [\varepsilon \mapsto \tilde{b}]$, we will have that $I_\sigma(\mathcal{S}, p) = U(X) - U(X|Y) > 0$. On the other hand, if $\phi(\tilde{z})$ is not satisfiable, then for any $\tilde{b} \in Act$ we will have that $Y = X \wedge \phi(\tilde{b})$ is logically equivalent to 0, hence $U(X|Y) = U(X)$. Consequently, for any $\sigma = [\varepsilon \mapsto \tilde{b}]$, we will have $I_\sigma(\mathcal{S}, p) = U(X) - U(X|Y) = 0$. □

6.2.2 General systems

The following results, which apply in general, are particularly interesting for systems with a moderate number of actions. The following theorem essentially says that, up to an expansion factor bounded by $|Act|$, non-adaptive strategies are as efficient as adaptive ones. Note that, for a strategy σ, the number of distinct actions that appear in σ is $|\mathrm{range}(\sigma)|$.

Theorem 6.2.4 *For each finite strategy σ of length l it is possible to build a non-adaptive finite strategy σ' of length $|\mathrm{range}(\sigma)| \times l$, such that*

$$I_{\sigma'}(\mathcal{S}, p) \geq I_\sigma(\mathcal{S}, p).$$

PROOF Let $\mathrm{range}(\sigma) = \{a_1, \ldots, a_h\}$ and let σ' be any non-adaptive strategy that plays each of $a_1, \ldots, a_h$ for l times, for example, $\sigma' = [a_1, \ldots, a_h, \ldots, a_1, \ldots, a_h]$ (l times); note that the length of σ' is $h \times l$, as required. For any y^j ($j \leq l$), we shall denote by $\sigma' - y^j$ the non-adaptive strategy of length $h \times l - j$ obtained by removing from σ', seen as a list, j actions $b_1, \ldots, b_j$, where $b_1 = \sigma(\varepsilon), \ldots, b_j = \sigma(y^{j-1})$.

Denote by Y_σ and $Y_{\sigma'}$ the random variables on $\mathcal{Y}^*$ corresponding to σ and σ', respectively. We will show that $U(X|Y_\sigma) \geq U(X|Y_{\sigma'})$. Take any $x \in \mathcal{X}$ such that $p(x) > 0$ and $y^j \in \mathrm{dom}(p_\sigma)$. We note that, for any sequence y^{hl-j}, and for an appropriate interleaving of the two sequences y^{hl-j} and y^j, that here we denote by just y^{hl-j}, y^j, we have that

$$p_{\sigma'-y^j}(y^{hl-j}|x)p_\sigma(y^j|x) = p_{\sigma'}(y^{hl-j}, y^j|x)\,. \tag{6.4}$$

From (6.4), it follows that

$$\begin{aligned} p_\sigma(y^j|x) &= \sum_{y^{hl-j}} p_{\sigma'-y^j}(y^{hl-j}|x)p_\sigma(y^j|x) \\ &= \sum_{y^{hl-j}} p_{\sigma'}(y^{hl-j}, y^j|x)\,. \end{aligned} \tag{6.5}$$

Now, for any x and y^j such that $p(x) > 0$ and $p_\sigma(y^j) > 0$, we have the following.

$$\begin{aligned} p_\sigma(x|y^j) &= \frac{p_\sigma(y^j|x)p(x)}{p_\sigma(y^j)} \\ &= \sum_{y^{hl-j}} p_{\sigma'}(y^{hl-j}, y^j|x)\frac{p(x)}{p_\sigma(y^j)} \qquad (6.6) \\ &= \sum_{y^{hl-j}} \frac{p_{\sigma'}(x|y^{hl-j}, y^j)p_{\sigma'}(y^{hl-j}, y^j)}{p(x)}\frac{p(x)}{p_\sigma(y^j)} \\ &= \sum_{y^{hl-j}} p_{\sigma'}(x|y^{hl-j}, y^j)\frac{p_{\sigma'}(y^{hl-j}, y^j)}{p_\sigma(y^j)} \qquad (6.7) \end{aligned}$$

where in (6.6) we have applied (6.5). It is an easy matter to show that $\sum_{y^{hl-j}} \frac{p_{\sigma'}(y^{hl-j}, y^j)}{p_\sigma(y^j)} = 1$ (this is basically a consequence of (6.4); we leave the details to the interested reader). Thus (6.7) shows that $p_\sigma(\cdot|y^j)$ can be expressed as a convex combination of the distributions $p_{\sigma'}(\cdot|y^{hl-j}, y^j)$,

for $y^{hl-j} \in \mathcal{Y}^{hl-j}$. Using this fact, the concavity of $U(\cdot)$ and Jensen's inequality, we arrive at the following.

$$U(p_\sigma(\cdot|y^j)) \quad \geq \quad \sum_{y^{hl-j}} U(p_{\sigma'}(\cdot|y^{hl-j}, y^j)) \frac{p_{\sigma'}(y^{hl-j}, y^j)}{p_\sigma(y^j)} \,. \tag{6.8}$$

We finally can compute the following lower-bound for $U(X|Y_\sigma)$.

$$\begin{aligned} U(X|Y_\sigma) &= \sum_{y^j} p_\sigma(y^j) U(p_\sigma(\cdot|y^j)) \\ &\geq \sum_{y^j} p_\sigma(y^j) \sum_{y^{hl-j}} \frac{p_{\sigma'}(y^{hl-j}, y^j)}{p_\sigma(y^j)} U(p_{\sigma'}(\cdot|y^{hl-j}, y^j)) \quad (6.9) \\ &= \sum_{y^j} \sum_{y^{hl-j}} p_{\sigma'}(y^{hl-j}, y^j) U(p_{\sigma'}(\cdot|y^{hl-j}, y^j)) \\ &= \sum_{y^{hl}} p_{\sigma'}(y^{hl}) U(p_{\sigma'}(\cdot|y^{hl})) \;=\; U(X|Y_{\sigma'}) \end{aligned}$$

where the inequality (6.9) follows from (6.8). □

We can give a better upper bound for deterministic systems.

Proposition 6.2.5 *If the mechanism $\mathcal{S}$ is deterministic, then the upper-bound in the previous theorem can be simplified to* $|\mathrm{range}(\sigma)|$.

PROOF Let σ be any finite non-adaptive strategy for $\mathcal{S}$. Suppose there is an action a that occurs at least twice in σ, seen as a tuple of actions, and let σ_- be the non-adaptive strategy obtained by removing the first occurrence of a from σ, seen as a list. Assume the two a's occur at position i and j, $i < j$, of σ. Since $\mathcal{S}$ is deterministic, it is easily seen that, for each $y^n = (y_1, \ldots, y_n)$, if $y_i \neq y_j$ then $p_\sigma(y^n) = 0$. On the other hand, whenever $y_i = y_j$, then $p_\sigma(y^n) = p_{\sigma_-}(y^{n-1})$ and $p_\sigma(x|y^n) = p_{\sigma_-}(x|y^{n-1})$, where by y^{n-1} we denote here the sequence obtained by removing y_i from y^n. This shows that $U(X|Y_\sigma) = U(X|Y_{\sigma_-})$. Repeating this elimination step, we can eventually get rid of all the duplicates in σ, while preserving the value of $I_\sigma(\mathcal{S}, p)$. Applying this fact to the strategy σ' defined in the proof of Theorem 6.2.4, we can come up with a strategy σ'' of length $|\mathrm{range}(\sigma)|$ such that $I_{\sigma''}(\mathcal{S}, p) = I_{\sigma_-}(\mathcal{S}, p)$. □

Example 29 *We reconsider Example 26. For the adaptive strategy σ defined in Figure 14, we have already shown that, for $U(\cdot) = H(\cdot)$, $I_\sigma(\mathcal{S}, p) \approx 2.45$.*

Consider now the non-adaptive strategy $\sigma' = [\mathrm{ZIP}, \mathrm{Date}, \mathrm{Age}]$*, which is just one action longer than* σ*. The corresponding attack tree is reported in Figure 17: the final partition obtained with* σ' *is finer than the one obtained with* σ*. In fact,* $I_{\sigma'}(\mathcal{S}, p) = \log 10 - \frac{2}{5} \approx 2.92 > I_\sigma(\mathcal{S}, p) \approx 2.45$*.*

The results discussed above are important from the point of view of the analysis. They entail that, for systems with a moderate number of actions, analysing adaptive strategies is essentially equivalent to analysing non-adaptive ones. The latter task can be much easier to accomplish. Results on asymptotic rate of convergence of non-adaptive strategies have already been discussed in Chapter 5. They permit to analytically assess the resistance of a mechanism as the length of the considered strategies grows. The following result, which covers the case of error entropy, is adapted from Theorem 5.2.4. Assume $Act = \{a_1, \ldots, a_k\}$, and for each $c_i \in \mathcal{X}/\equiv$, let $\pi_i \triangleq \max_{x \in c_i} p(x)$.

Proposition 6.2.6 *For each* $n \geq 1$*, consider the non-adaptive strategy* $\sigma_n = [a_1, \ldots, a_k, \ldots, a_1, \ldots, a_k]$ *(*n *times) and let* $(X, Y^n) \sim p_{\sigma_n}(\cdot)$*. Then, there are positive constants* γ *and* ρ*, only depending on the matrices* $p_a(\cdot|\cdot)$*, such that*

$$(1 - \sum_i \pi_i) \leq E(X|Y^n) \leq (1 - \sum_i \pi_i) + \gamma 2^{-n\rho}.$$

6.3 Maximum leakage

In this section we show that the class of adaptive and non adaptive strategies induce the same maximum leakage. For truly probabilistic mechanisms, strategies achieving maximum leakage are in general infinite. A key notion is that of indistinguishability, x and x' are indistinguishable if, no matter what strategy the adversary will play, he cannot tell them apart.

Definition 6.3.1 (Indistinguishability in the adaptive model) *Two states* $x, x' \in \mathcal{X}$ *are indistinguishable if they satisfy the following equalities:*

$$x \equiv x' \qquad \textit{if and only if} \qquad \textit{for each finite } \sigma : \; p_\sigma(\cdot|x) = p_\sigma(\cdot|x').$$

Despite being based on a universal quantification over all finite strategies, indistinguishability is in fact quite easy to characterise, also computationally. For each $a \in Act$, consider the equivalence relation defined by

$$x \equiv_a x' \qquad \text{if and only if} \qquad p_a(\cdot|x) = p_a(\cdot|x').$$

Lemma 6.3.2 *$x \equiv x'$ if and only if for each $a \in Act$, $p_a(\cdot|x) = p_a(\cdot|x')$. In other words,*

$$\equiv \; = \; \bigcap_{a \in Act} \equiv_a .$$

Now, consider $\mathcal{X}/\equiv$, the set of equivalence classes of $\equiv$, and let c ranges over this set. Let $[X]$ be the random variable whose outcome is the equivalence class of X according to $\equiv$. Note that $p(c) \triangleq \Pr([X] = c) = \sum_{x \in c} p(x)$. We consistently extend our I-notation by defining

$$U(X \,|\, [X]) \triangleq \sum_c p(c) U(p(\cdot \,|\, [X] = c))$$

and

$$I(X \,;\, [X]) \triangleq U(X) - U(X \,|\, [X]) .$$

More explicitly, $p(\cdot|[X] = c)$ denotes the distribution over $\mathcal{X}$ that yields $p(x)/p(c)$ for $x \in c$ and 0 elsewhere; we will often abbreviate $p(\cdot|[X] = c)$ just as $p(\cdot|c)$. Note that $I(X \,;\, [X])$ expresses the information gain about X when the attacker gets to know the indistinguishability class of the secret. As expected, this is an upper-bound to the information that can be gained playing any strategy.

Theorem 6.3.3 *$I_\star(\mathcal{S}, p) \leq I(X \,;\, [X])$.*

PROOF Fix any finite strategy σ and prior $p(\cdot)$. It is sufficient to prove that $U(X|Y) \geq U(X \,|\, [X])$. The proof exploits the concavity of U. First, we note that, for each x and y^j of nonzero probability, we have (c below ranges over $\mathcal{X}/\equiv$):

$$p_\sigma(x|y^j) \;\; = \;\; \sum_c \frac{p_\sigma(x, y^j, c)}{p_\sigma(y^j)} \;\; = \;\; \sum_c p_\sigma(c|y^j) p_\sigma(x|y^j, c) . \qquad (6.10)$$

By (6.10), concavity of $U(\cdot)$ and Jensen's inequality

$$U(p(\cdot|y^j)) \;\; \geq \;\; \sum_c p_\sigma(c|y^j) U(p_\sigma(\cdot|y^j, c)) . \qquad (6.11)$$

Now, we can compute as follows (as usual, y^j below runs over sequences of nonzero probability):

$$\begin{aligned} U(X|Y) \;\; &= \;\; \sum_{y^j} p_\sigma(y^j) U(p_\sigma(\cdot|y^j)) \\ &\geq \;\; \sum_{y^j, c} p_\sigma(y^j) p_\sigma(c|y^j) U(p_\sigma(\cdot|y^j, c)) \qquad (6.12) \end{aligned}$$

$$
\begin{aligned}
&= \sum_{y^j,c} p_\sigma(y^j) p_\sigma(c|y^j) U(p(\cdot|c)) && (6.13)\\
&= \sum_{c} \big(\sum_{y^j} p_\sigma(y^j, c) \big) U(p(\cdot|c)) \\
&= \sum_{c} p(c) U(p(\cdot|c)) \;\; = \;\; U(X \,|\, [X])
\end{aligned}
$$

where: (6.12) is justified by (6.11); and (6.13) follows from the fact that, for each x, $p_\sigma(x|y^j, c) = p(x|c)$ (once the equivalence class of the secret is known, the observation y^j provides no further information about the secret). □

6.3.1 Deterministic case

As to the maximal achievable information, we start our discussion from deterministic mechanism.

Proposition 6.3.4 *Let $\mathcal{S}$ be deterministic. Let $\sigma = [a_1, \ldots, a_k]$ be a non-adaptive strategy that plays all actions in Act once. Then*

$$I_\star(\mathcal{S}, p) = I_\sigma(\mathcal{S}, p).$$

PROOF Let $(X, Y) \sim p_\sigma(\cdot)$. We prove that $U(X \,|\, Y) = U(X \,|\, [X])$. We first note that for each $c \in \mathcal{X}/\equiv$ there is exactly one sequence y_c^k such that $p_\sigma(y_c^k|c) = 1$: this follows from $\mathcal{S}$ being deterministic. Moreover, if $c \neq c'$ then $y_c^k \neq y_{c'}^k$: otherwise, it would follow that $p_{a_i}(y|c) = p_{a_i}(y|c')$ for each $a_i \in Act$ and $y \in \mathcal{Y}$, contrary to Lemma 6.3.2 (note that $p(\cdot|c)$ is the same as $p(\cdot|x)$, for any $x \in c$). These facts can be used to show, through easy manipulations, that $p(x|y_c^k) = p(x|c)$ for each x. As a consequence, one can compute as follows.

$$
\begin{aligned}
U(X|Y) &= \sum_{y^k} p_\sigma(y^k) U(p_\sigma(\cdot|y^k)) \\
&= \sum_{c} p(c) \sum_{y^k} p_\sigma(y^k|c) U(p_\sigma(\cdot|y^k)) \\
&= \sum_{c} p(c) U(p_\sigma(\cdot|y_c^k)) \\
&= \sum_{c} p(c) U(p_\sigma(\cdot|c)) \\
&= U(X \,|\, [X])\,. \qquad \square
\end{aligned}
$$

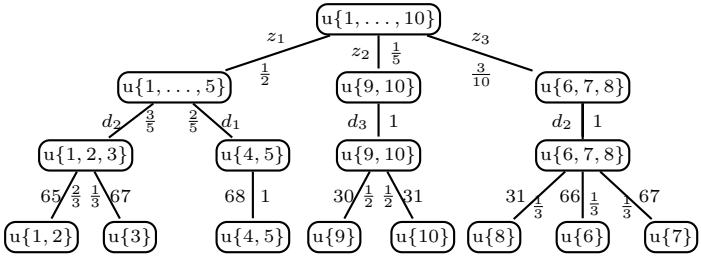

Figure 17: The attack tree corresponding to the the non-adaptive strategy $[\mathrm{ZIP}, \mathrm{Date}, \mathrm{Age}]$ for Example 26.

Hence, in the deterministic case, the maximal gain in information is obtained by a trivial brute-force strategy where all actions are played in any fixed order. It is instructive to observe such a strategy at work, under the form of an attack tree. The supports of the distributions that are at the same level constitute a partition of $\mathcal{X}$: more precisely, the partition at level i ($1 \leq i \leq k$) is given by the equivalence classes of the relation $\cap_{j=1}^{i} \equiv_{a_j}$. An example of this fact is illustrated by the attack tree in Figure 17, relative to the non-adaptive strategy $[\mathrm{ZIP}, \mathrm{Date}, \mathrm{Age}]$ for the mechanism in Example 26. This fact had been already observed in (KB07) for the restricted model considered there. Indeed, one would obtain the model of (KB07) by stripping the probabilities off the tree in Figure 17.

6.3.2 Probabilistic case

The general probabilistic case is slightly more complicated. Essentially, any non-adaptive strategy where each action is played infinitely often achieves the maximum information gain. The next theorem considers one such strategy.

Theorem 6.3.5 *There is a total, non-adaptive strategy σ such that $I_\sigma(\mathcal{S}, p) = I(X; [X])$. Consequently, $I_\star(\mathcal{S}, p) = I(X\,;\,[X])$.*

In order to prove Theorem 6.3.5, we introduce some terminology and concepts from the information-theoretic Method of Types, introduced in Section 2.4. Given $n \geq 1$ and a sequence $y^n \in \mathcal{Y}^n$, recall that $t_{y^n}(\cdot)$ denote the type of y^n (see Definition 2.4.3). We will often abbreviate $H(t_{y^n})$ as $H(y^n)$, and $D(t_{y^n} \| q)$ as $D(y^n \| q)$, where $D(\cdot\|\cdot)$ is the Kullback-Leibler distance (see Definition 2.4.1), thus denoting the type by a corresponding

sequence, when no confusion arises. Given $\varepsilon > 0$ and a probability distribution $q(\cdot)$ on $\mathcal{Y}$, the "ball" of n-sequences whose type is within distance ε of $q(\cdot)$ is defined thus:

$$B^{(n)}(q, \varepsilon) \triangleq \{y^n : D(y^n||q) \leq \varepsilon\}.$$

We shall also make use of the following new terminology about sequences. Assume $|Act| = k$. Given a sequence $y^n = (y_1, y_2, \ldots, y_n)$ and an integer $j = 1, \ldots, k$, we shall denote by $y^n(j)$ the subsequence $(y_j, y_{k+j}, y_{2k+j}, \ldots)$, obtained by taking the symbols of y^n at position $j, k+j, 2k+j, \ldots$. In the rest of the section, unless otherwise stated, we let σ be the infinite non-adaptive strategy that plays actions $a_1, \ldots, a_k, a_1, a_2, \ldots$, in a lock-step fashion: $\sigma(y^j) \triangleq a_{(j \bmod k)+1}$. For any $n \geq 1$, we let σ_n be the truncation at level n of σ: $\sigma_n \triangleq \sigma \backslash n$. For a prior $p(\cdot)$, let $p_{\sigma_n}(\cdot)$ be the resulting joint probability distribution on $\mathcal{X} \times \mathcal{Y}^n$: note that, for each x, the support of $p_{\sigma_n}(\cdot|x)$ is included in $\mathcal{Y}^n$. Let (X, Y^n) be jointly distributed according to $p_{\sigma_n}(\cdot)$: here we have introduced the superscript ^{n} to remember explicitly the dependence of Y from n. Let us define the set of sequences y^n where the type of each sub-sequence $y^n(i)$ is within ε distance of the distribution $p_{a_i}(\cdot|x)$, thus:

$$\hat{B}^{(n)}(x, \varepsilon) \triangleq \{y^n : D\big(y^n(i)||p_{a_i}(\cdot|x)\big) \leq \varepsilon \text{ for } i = 1, \ldots, k\} \quad (6.14)$$

Furthermore, we define the following quantities depending on a given sequence y^n and $x \in \mathcal{X}$:

$$\hat{H}(y^n) \triangleq \sum_{i=1}^{k} H(y^n(i)) \quad (6.15)$$

$$\hat{D}(y^n||p_{\sigma_n}(\cdot|x)) \triangleq \sum_{i=1}^{k} D(y^n(i)||p_{a_i}(\cdot|x)). \quad (6.16)$$

Finally, for each sequence $y^m \in \mathcal{Y}^m$ and for each action $a \in Act$, we let

$$p_a^m(y^m|x) = \prod_{i=1}^{m} p_a(y_i|x) \quad (6.17)$$

(this is the probability of generating y^m with i.i.d. extractions obeying distribution $p_a(\cdot|x)$).

Lemma 6.3.6 *Let n be a multiple of k and $x \in \mathcal{X}$. Then*

$$p_{\sigma_n}(y^n|x) = 2^{-\frac{n}{k}[\hat{H}(y^n) + \hat{D}(y^n||p_{\sigma_n}(\cdot|x))]}.$$

PROOF

$$p_{\sigma_n}(y^n|x) = \prod_{j=1}^{k} p_{a_j}^{\frac{n}{k}}(y^n(j)|x) \tag{6.18}$$

$$= \prod_{j=1}^{k} 2^{-\frac{n}{k}\left(H(y^n(j))+D\left(y^n(j)\|p_{a_j}(\cdot|x)\right)\right)} \tag{6.19}$$

$$= 2^{-\frac{n}{k}\sum_{j=1}^{k}\left(H(y^n(j))+D\left(y^n(j)\|p_{a_j}(\cdot|x)\right)\right)}$$

$$= 2^{-\frac{n}{k}\left(\hat{H}(y^n)+\hat{D}(y^n\|p_{\sigma_n}(\cdot|x))\right)} \tag{6.20}$$

where: (6.18) follows from re-arranging factors and the definition of $p_{a_j}^{\frac{n}{k}}(\cdot)$; (6.19) follows from Theorem 2.4.6; in (6.20) we have applied Definitions (6.15) and (6.16). □

Below, for a set A and a distribution $q(\cdot)$, we let $q(A)$ denote $\sum_{a\in A} q(a)$.

Lemma 6.3.7 *Let n be a multiple of k, $x \in \mathcal{X}$ and $\varepsilon > 0$. Then*

$$p_{\sigma_n}(\hat{B}^{(n)}(x,\varepsilon)|x) \geq 1 - 2^{-\frac{n}{k}\varepsilon} C\Big(\frac{n}{k}+1\Big)^{k|\mathcal{Y}|}$$

for some constant C not depending on n.

PROOF Let $m = n/k$. We give a lower bound on the probability of $\hat{B}^{(n)}(x,\varepsilon)$ as follows.

$$\begin{aligned}
p_{\sigma_n}(\hat{B}^{(n)}(x,\varepsilon)|x) &= \sum_{y^n\in\hat{B}^{(n)}(x,\varepsilon)} p_{\sigma_n}(y^n|x) \\
&= \sum_{y^n\in\hat{B}^{(n)}(x,\varepsilon)} \prod_{i=1}^{k} p_{a_i}^m(y^n(i)|x) \\
&= \prod_{i=1}^{k} \sum_{y^m\in B^{(m)}(p_{a_i}^m(\cdot|x),\varepsilon)} p_{a_i}(y^m|x) \qquad (6.21) \\
&= \prod_{i=1}^{k} p_{a_i}^m\left(B^{(m)}(p_{a_i}(\cdot|x),\varepsilon)\,|\,x\right) \\
&\geq \prod_{i=1}^{k} \left(1 - 2^{-m\varepsilon}(m+1)^{|\mathcal{Y}|}\right) \qquad (6.22)
\end{aligned}$$

$$
\begin{aligned}
&= \left(1 - 2^{-m\varepsilon}(m+1)^{|\mathcal{Y}|}\right)^k \\
&= 1 + \sum_{i=1}^{k} \binom{k}{i} (-1)^i 2^{-mi\varepsilon} (m+1)^{|\mathcal{Y}|i} \\
&\geq 1 - 2^{-m\varepsilon} C (m+1)^{k|\mathcal{Y}|} \qquad (6.23)
\end{aligned}
$$

where: the equality (6.21) follows from the definition of $\hat{B}^{(n)}(x,\varepsilon)$ (Equation 6.14); the inequality (6.22) follows from Theorem 2.4.8 (Equation 2.5); in (6.23), $C = k \cdot \max_i \binom{k}{i}$ (note that $\binom{k}{i}$ is maximum when $i = \lceil k/2 \rceil$). □

Lemma 6.3.8 *Let $x, x' \in \mathcal{X}$, with $x \not\equiv x'$. Let $n \geq 1$. Then there is $\varepsilon > 0$ such that $\hat{B}^{(n)}(x, 2\varepsilon) \cap \hat{B}^{(n)}(x', 2\varepsilon) = \emptyset$.*

PROOF It is well-known that given any two distinct distributions $p(\cdot)$ and $q(\cdot)$, there is $\varepsilon > 0$ such that $B^{(n)}(p, 2\varepsilon) \cap B^{(n)}(q, 2\varepsilon) = \emptyset$ (this is a consequence of Pinsker's inequality, (CT06, Lemma 11.6.1)). Thus, choose $\varepsilon > 0$ such that, for some j, $B^{(n)}\big(p_{a_j}(\cdot|x), 2\varepsilon\big) \cap B^{(n)}\big(p_{a_j}(\cdot|x'), 2\varepsilon\big) = \emptyset$: the wanted statement follows from the definition of $\hat{B}^{(n)}(x, 2\varepsilon)$ and $\hat{B}^{(n)}(x', 2\varepsilon)$. □

We are now set to prove Theorem 6.3.5.

PROOF [Theorem 6.3.5] Using the notation previously introduced, we shall prove that, as $n \to \infty$

$$
U(X|Y^n) \longrightarrow L \qquad \text{for some } L \leq U(X|[X])\,. \qquad (6.24)
$$

This will imply the thesis, as then $I_\sigma(\mathcal{S}, p) \geq I(X; [X])$, which, by virtue of Theorem 6.3.3, implies $I_\sigma(\mathcal{S}, p) = I(X; [X])$.

Let the equivalence classes of $\equiv$ be $c_1, \ldots, c_K$. For each $i = 1, \ldots, K$, choose a representative $x_i \in c_i$ of nonzero probability (if it exists; otherwise class c_i is just discarded). We can compute as follows.

$$
\begin{aligned}
U(X|Y^n) &= \sum_{y^n, x} p(x) p_{\sigma_n}(y^n|x) U(p_{\sigma_n}(\cdot|y^n)) \\
&= \sum_{x} p(x) \sum_{y^n} p_{\sigma_n}(y^n|x) U(p_{\sigma_n}(\cdot|y^n)) \\
&\leq \sum_{x} p(x) U\Big(\sum_{y^n} p_{\sigma_n}(y^n|x) p_{\sigma_n}(\cdot|y^n)\Big) \qquad (6.25) \\
&= \sum_{c_i} \sum_{x \in c_i} p(x) U\Big(\sum_{y^n} p_{\sigma_n}(y^n|x_i) p_{\sigma_n}(\cdot|y^n)\Big)
\end{aligned}
$$

$$= \sum_{c_i} p(c_i) U(\underbrace{\sum_{y^n} p_{\sigma_n}(y^n|x_i) p_{\sigma_n}(\cdot|y^n)}_{\triangleq q_i^n(\cdot)})$$

$$= \sum_{c_i} p(c_i) U(q_i^n) \tag{6.26}$$

where the inequality in (6.25) stems from the concavity of U and Jensen's inequality. We will show that there is a sub-sequence of indices $\{n_j\}$ such that for each $i = 1, \ldots, K$,

$$q_i^{n_j}(\cdot) \quad \longrightarrow \quad p(\cdot|c_i) \tag{6.27}$$

(according to any chosen metrics in $\mathcal{P}(\mathcal{X})$). This will imply (6.24): in fact, by virtue of the continuity of U, we will have, on the chosen sub-sequence, $\sum_{c_i} p(c_i) U(q_i^{n_j}) \to \sum_{c_i} p(c_i) U(p(\cdot|c_i)) = U(X|[X])$. Hence, by virtue of (6.26), on the chosen sub-sequence and hence on every sequence, we will have $U(X|Y^n) \to L \leq U(X|[X])$, which is (6.24).

In order to prove (6.27), take any $n \geq 1$ that is a multiple of k, and choose any $\varepsilon > 0$ such that $\hat{B}^{(n)}(x, 2\varepsilon) \cap \hat{B}^{(n)}(x', 2\varepsilon) = \emptyset$ whenever $x \not\equiv x'$ (the existence of such an ε is guaranteed by Lemma 6.3.8). Consider a generic $x \in c_i$ such that $p(x) > 0$. We have the following lower bound for $q_i^n(x)$.

$$q_i^n(x) =$$

$$= \sum_{y^n} \frac{p_{\sigma_n}(y^n|x_i) p_{\sigma_n}(y^n|x_i) p(x)}{\sum_{x'} p_{\sigma_n}(y^n|x') p(x')} \tag{6.28}$$

$$= \sum_{y^n} \frac{p_{\sigma_n}(y^n|x_i)}{\frac{p(c_i)}{p(x)} + \sum_{x' \not\equiv x_i} \frac{p_{\sigma_n}(y^n|x') p(x')}{p_{\sigma_n}(y^n|x_i) p(x_i)}}$$

$$\geq \sum_{y^n \in \hat{B}^{(n)}(x,\varepsilon)} \frac{p_{\sigma_n}(y^n|x_i)}{\frac{p(c_i)}{p(x)} + \sum_{x' \not\equiv x_i} 2^{-\frac{n}{k}[\hat{D}(y^n||p_{\sigma_n}(\cdot|x')) - \hat{D}(y^n||p_{\sigma_n}(\cdot|x))]} \frac{p(x')}{p(x)}} \tag{6.29}$$

$$\geq \sum_{y^n \in \hat{B}^{(n)}(x,\varepsilon)} \frac{p_{\sigma_n}(y^n|x_i)}{\frac{p(c_i)}{p(x)} + \sum_{x' \not\equiv x_i} 2^{-n\varepsilon} \frac{p(x')}{p(x)}} \tag{6.30}$$

$$= p_{\sigma_n}\big(\hat{B}^{(n)}(x_i, \varepsilon) \,|\, x_i\big) \frac{1}{\frac{p(c_i)}{p(x)} + 2^{-n\varepsilon} C'} \tag{6.31}$$

$$\geq \frac{1 - 2^{-\frac{n}{k}\varepsilon} C (\frac{n}{k} + 1)^{k|\mathcal{Y}|}}{\frac{p(c_i)}{p(x)} + 2^{-n\varepsilon} C'} \tag{6.32}$$

where: (6.28) follows from the definition of $q_i^n(x)$ and an application of Bayes rule, and from the fact that $p_{\sigma_n}(y^n|x) = p_{\sigma_n}(y^n|x_i)$; (6.29) follows from a simple union bound and from Lemma 6.3.6; (6.30) follows from the fact that, by assumption, $\hat{B}^{(n)}(x, 2\varepsilon) \cap \hat{B}^{(n)}(x', 2\varepsilon) = \emptyset$ (also note that $\hat{B}^{(n)}(x, 2\varepsilon) = \hat{B}^{(n)}(x_i, 2\varepsilon)$); (6.31) follows by definition of $\hat{B}^{(n)}(x, \varepsilon) = \hat{B}^{(n)}(x_i, \varepsilon)$; here C' is a suitable constant, not depending on n; (6.32) follows from Lemma 6.3.7.

Now, let $\{n_j\}$ be a sequence of indices such that, for each x and i, $q_i^{n_j}(x)$ converges to a limit, say $L_i(x)$ (such a sub-sequence must exist, by Bolzano-Weierstrass). The inequality

$$q_i^n(x) \geq \frac{1 - 2^{-\frac{n}{k}\varepsilon} C(\frac{n}{k} + 1)^{k|\mathcal{Y}|}}{\frac{p(c_i)}{p(x)} + 2^{-n\varepsilon} C'}$$

which holds for each n that is a multiple of k, implies that these limits satisfy $L_i(x) \geq \frac{p(x)}{p(c_i)}$. Since point-wise convergence for each x implies convergence of $q_i^{n_j}(\cdot)$ to a probability distribution, we have that, for each i and x, actually equality must hold: $L_i(x) = \frac{p(x)}{p(c_i)}$. Thus, for each $i = 1, \ldots, K$, $q_i^{n_j}(\cdot) \to p(\cdot|c)$, which proves (6.27). □

Theorem 6.3.9 *The following formulae holds, where $K = |\mathcal{X}/\equiv|$.*

- *For $U = H$ (Shannon entropy), $C(\mathcal{S}) = \log K$.*
- *For $U = E$ (Error entropy), $C(\mathcal{S}) = 1 - \frac{1}{K}$.*

PROOF Let x_i be any representative of class c_i, for $i = 1, \ldots, K$.

- $U = H$. By the symmetry of mutual information in the case of Shannon entropy, we have

$$\begin{aligned} I(X; [X]) &= H([X]) - \underbrace{H([X]|X)}_{=0} = H([X]) \\ &= -\sum_{c_i} p(c_i) \log p(c_i) \leq \log K \end{aligned}$$

  where the last inequality follows from the property of Shannon entropy that $H(q) \leq \log |\mathrm{supp}(q)|$, for any distribution q. On the other hand, if we take the distribution $p(\cdot)$ defined as $p(x_i) = \frac{1}{K}$, and $p(x) = 0$ elsewhere, we can easily compute that $I(X; [X]) = \log K$.

- $U = E$. Let $p(\cdot)$ be any prior and assume without loss of generality that $p(x_i) = \max_{x \in c_i} p(x)$ for each i, and furthermore that $p(x_1) = \max_x p(x)$. By easy manipulations, we have:

$$\begin{aligned} I(X;[X]) &= E(X) - E(X|[X]) \\ &= (1 - p(x_1)) - (1 - \sum_{c_i} p(c_i) \frac{p(x_i)}{p(c_i)}) \\ &= \sum_{i=1}^{K} p(x_i) - p(x_1) \quad = \quad \sum_{i=2}^{K} p(x_i)\,. \end{aligned}$$

  Now it is easily checked that the last term in this chain is $\leq 1 - \frac{1}{K}$: this is done by separately considering the two cases $\max_x p(x) = p(x_1) \leq \frac{1}{K}$ and $\max_x p(x) = p(x_1) > \frac{1}{K}$. On the other hand, if we take, as above, the distribution $p(\cdot)$ defined as $p(x_i) = \frac{1}{K}$, and $p(x) = 0$ elsewhere, we can easily compute that $I(X;[X]) = 1 - \frac{1}{K}$.

  □

Example 30 *Consider the mechanism defined in Example 26. One has the following capacities: for $U(\cdot) = H(\cdot)$, $C(\mathcal{S}) = \log 8 = 3$, while for $U(\cdot) = E(\cdot)$, $C(\mathcal{S}) = \frac{7}{8} = 0.875$.*

6.4 Computing finite strategies

We show that $I_\sigma(\mathcal{S}, p)$ can be expressed recursively, in terms of a Bellman-type equation. This allows for calculation of optimal finite strategies based on standard algorithms, such as backward induction.

6.4.1 A Bellman equation

Let us introduce some terminology.

Definition 6.4.1 (y-derivative) *For each y, the* y-derivative *of σ, denoted σ_y, is the function defined thus, for each $y^j \in \mathcal{Y}^*$:*

$$\sigma_y(y^j) \triangleq \sigma(yy^j).$$

Remark 6.4.2 *Note that if σ has length $l > 1$, then σ_y is a strategy of height $\leq l - 1$. For $l = 1$, σ_y is the empty function.*

Recall that according to (6.2), for $h = ay$, we have[2]

$$p^{ay}(x) \quad = \quad p_a(x|y)$$

By convention, we let $I_\sigma(\cdots)$ denote 0 when σ is empty. Moreover, we write $I_{[a]}(\cdots)$ as $I_a(\cdots)$.

Lemma 6.4.3 *Let $p(\cdot)$ be any prior on $\mathcal{X}$. Let σ be a strategy with $\sigma(\varepsilon) = a$. Then*

$$I_\sigma(\mathcal{S}; p) = I_a(\mathcal{S}; p) + \sum_y p_a(y) I_{\sigma_y}(\mathcal{S}; p^{ay}).$$

We introduce some additional notation to be used in the proof of Lemma 6.4.3. Let l denote the length of a strategy σ, and let (X, Y) be distributed according to $p_\sigma(\cdot)$. We can decompose Y as the concatenation of the first observation and whatever sequence of observations is left, thus: $Y = Y_1 \cdot Y_s$. Here, Y_1 takes values on $\mathcal{Y}$, while Y_s takes values on a subset of $\cup_{0 \le j \le l} \mathcal{Y}^j$ - in particular, if $l = 1$, Y_s takes on the value ε with probability 1. In what follows, we denote the marginal distribution of Y_1 under σ just as $p_\sigma(y)$, and that of Y_s as $p_\sigma(y^j)$, for generic y and y^j.

PROOF [Lemma 6.4.3] It is an easy matter to prove the following equations. For each prior $p(\cdot)$, finite strategy σ with $\sigma(\varepsilon) = a$, sequence y^j, observation y, one has (below, y and y^j run over elements of nonzero probability; moreover, for any prior $p(\cdot)$, history h and strategy σ, the term p^h_σ is to be parsed as $(p^h)_\sigma$):

$$p_\sigma(y) \quad = \quad p_a(y) \tag{6.33}$$

$$p_\sigma(x|y) \quad = \quad p_a(x|y) \; = \; p^{ay}(x) \tag{6.34}$$

$$p_\sigma(x|yy^j) \quad = \quad p^{ay}_{\sigma_y}(x|y^j) \tag{6.35}$$

$$p_\sigma(y^j|y) \quad = \quad p^{ay}_{\sigma_y}(y^j)\,. \tag{6.36}$$

By applying equalities (6.33), (6.34), (6.35) and (6.36) above as appropriate, we have:

$$\begin{aligned} I_\sigma(\mathcal{S}, p) \; &= \; I(X;Y) \; = \\ &= \; \big[U(X) - U(X|Y_1)\big] \; + \; \big[U(X|Y_1) - U(X|Y)\big] \end{aligned}$$

[2] In terms of a given prior $p(\cdot)$ and of the matrices of $\mathcal{S}$, this can be also expressed as: $p^{ay}(x) = \frac{p_a(y|x)p(x)}{\sum_{x'} p_a(y|x')p(x')}$.

$$
\begin{aligned}
&= \big[U(p) - \sum_y p_\sigma(y) U(p_\sigma(\cdot|y))\big] + \\
&\quad + \big[\sum_y p_\sigma(y) U(p_\sigma(\cdot|y)) - p_\sigma(y, y^j) U(p_\sigma(\cdot|yy^j))\big] \\
&= \big[U(p) - \sum_y p_\sigma(y) U(p_\sigma(\cdot|y))\big] + \\
&\quad + \big[\sum_y p_\sigma(y) U(p_\sigma(\cdot|y)) - p_\sigma(y) p_\sigma(y^j|y) U(p_\sigma(\cdot|yy^j))\big] \\
&= \big[U(p) - \sum_y p_\sigma(y) U(p_\sigma(\cdot|y))\big] + \\
&\quad + \sum_y p_\sigma(y) \big[U(p_\sigma(\cdot|y)) - \sum_{y^j} p_\sigma(y^j|y) U(p_\sigma(\cdot|yy^j))\big] \\
&= \big[U(p) - \sum_y p_a(y) U(p_a(\cdot|y))\big] + \\
&\quad + \sum_y p_a(y) \big[U(p^{ay}) - \sum_{y^j} p^{ay}_{\sigma_y}(y^j) U(p^{ay}_{\sigma_y}(\cdot|y^j))\big] \\
&= I_a(\mathcal{S}; p) \; + \; \sum_y p_a(y) I_{\sigma_y}(\mathcal{S}; p^{ay}).
\end{aligned}
$$

□

Let us say that a strategy σ of length l is *optimal* for $\mathcal{S}$, $p(\cdot)$ and l if it maximises $I_\sigma(\mathcal{S}, p)$ among all strategies of length l.

Corollary 6.4.4 (Bellman-type equation for optimal strategies) *There is an optimal strategy $\sigma^\star$ of length l for $\mathcal{S}$ and $p(\cdot)$ that satisfies the following equation*

$$
I_{\sigma^\star}(\mathcal{S}; p) = \max_a \; \big\{ I_a(\mathcal{S}; p) \; + \sum_{y:\, p_a(y) > 0} p_a(y) I_{\sigma^\star_{a,y}}(\mathcal{S}; p^{ay}) \big\} \tag{6.37}
$$

where $\sigma^\star_{a,y}$ is an optimal strategy of length $l-1$ for $\mathcal{S}$ and $p^{ay}(\cdot)$.

Corollary 6.4.4 allows one to employ dynamic programming or backward induction to compute optimal finite strategies. We discuss this briefly in the next subsection.

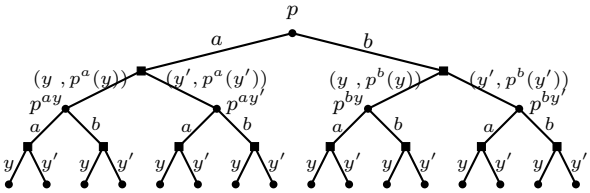

Figure 18: The first few levels of a MDP induced by a prior $p(\cdot)$ and a mechanism with $Act = \{a, b\}$ and $\mathcal{Y} = \{y, y'\}$. Round nodes are decision nodes and square nodes are probabilistic nodes. For the sake of space, labels of the last level of arcs and nodes are only partially shown.

6.4.2 Markov Decision Processes and backward induction

A mechanism $\mathcal{S}$ and a prior $p(\cdot)$ induce a *Markov Decision Process* (MDP), where all possible attack trees are represented at once. Backward induction amounts to recursively computing the most efficient attack tree out of this MDP, limited to a given length. More precisely, the MDP $\mathcal{M}$ induced by $\mathcal{S}$ and a prior $p(\cdot)$ is an (in general, infinite) tree consisting of *decision* nodes and *probabilistic* nodes. Levels of decision nodes alternate with levels of probabilistic nodes, starting from the root which is a decision node. Decision nodes are labeled with probability distributions over $\mathcal{X}$, edges outgoing decision nodes with actions, and edges outgoing probabilistic nodes with pairs (y, λ) of an observation and a real, in such a way that (again, we identify nodes with the corresponding history):

- a decision node corresponding to history h is labeled with $p^h(\cdot)$, if this is defined, otherwise the node and its descendants are removed, as well as the incoming edge;
- for any pair of consecutive edges leading from a decision node h to another decision node hay, for any $a \in Act$ and $y \in \mathcal{Y}$, the edge outgoing the probabilistic node is labeled with $(y, p^h_a(y))$.

Figure 18 shows the first few levels of such a MDP.

In order to compute an optimal strategy of length $l \geq 1$ by backward induction, one initially prunes the tree at l-th decision level (the root is at level 0) and then assigns *rewards* to all leaves of the resulting tree. Moreover, each probabilistic node is assigned an *immediate gain*. Rewards are then gradually propagated from the leaves up to the root, as follows:

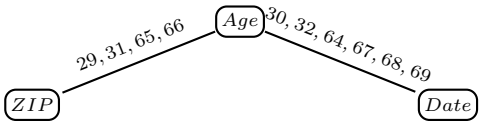

Figure 19: A Shannon entropy optimal strategy for Example 27. Leaves with the same label and their incoming arcs have been coalesced.

- each probabilistic node is assigned as a reward the sum of its immediate gain and the *average* reward of its children, average computed using the probabilities on the outgoing arcs;

- each decision node is assigned the *maximal* reward of its children; the arc leading to the maximising child is marked or otherwise recorded.

Eventually, the root will be assigned the maximal achievable reward. Moreover, the paths of marked arcs starting from the root will define an optimal strategy of length l. We can apply this strategy to our problem, starting with assigning rewards 0 to each leaf node h, and immediate gain $I_a(\mathcal{S}, p^h)$ to each a-child of any decision node h. The correctness of the resulting procedure is obvious in the light of Corollary 6.4.4.

In a crude implementation of the above outlined procedure, the number of decision nodes in the MDP will be bounded by $(|\mathcal{Y}| \times |Act|)^{l+1} - 1$ (probabilistic nodes can be dispensed with, at the cost of moving incoming action labels to outgoing arcs). Assuming that each distribution is stored in space $O(|\mathcal{X}|)$, the MDP can be built and stored in time and space $\mathcal{O}(|\mathcal{X}| \times (|\mathcal{Y}| \times |Act|)^{l+1})$. This is also the running time of the backward induction outlined above, assuming $U(\cdot)$ can be computed in time $O(|\mathcal{X}|)$ (some straightforward optimisations are possible here, but we will not dwell on this). By comparison, the running time of the exhaustive procedure outlined in (KB07, Theorem 1), for deterministic systems, runs in time $O(l \times |Act|^{r^l} \times |\mathcal{X}| \times \log |\mathcal{X}|)$, where r is the maximal number of classes in any relation $\equiv_a$; since r can be as large as $|\mathcal{Y}|$, this gives a worst-case running time of $O(l \times |Act|^{|\mathcal{Y}|^l} \times |\mathcal{X}| \times \log |\mathcal{X}|)$.

Example 31 *Applying backward induction to the mechanism of Example 27 with $U(\cdot) = H(\cdot)$ and $l = 2$, one gets the optimal strategy σ shown in Figure 19; this yields $I_\sigma(\mathcal{S}, p) \approx 2.4$ bits.*

In the general case, unfortunately, backward induction is quite memory-inefficient, even for a moderate number of observations or actions.

6.5 Concluding remarks

We have proposed a general information-theoretic model for the analysis of confidentiality under adaptive attackers. Within this model, we have proven several results on the limits of such attackers, on the relations between adaptive and non-adaptive strategies, and on the problem of searching for optimal finite strategies.

Adaptive attacks like chosen plaintext or ciphertext attacks are often considered in the literature on block ciphers, see (Gol04) and references therein. While such attacks can be easily modeled within our framework, further investigation is in order to understand the exact relationship between our security notions and those considered in that context, such as indistinguishability under chosen plaintext attack (IND-CPA) and variations thereof. A well-known difficulty is that, in an information-theoretic setting like ours, one cannot easily reason about the adversary's computing power.

Chapter 7

Conclusion

In this thesis we presented models to analyse a variety of statistical attacks in a uniform fashion. This permits the assessment of systems security against various kinds of attackers both at the global level and at the level of specific partitions of the secrets and in presence of both passive and active adversaries. In particular, we give precise bounds for the probability of misclassification on the part of the attacker, characterising both the limit value and the rate of convergence of the error probability as a function of the number of independent observations. In all analysed scenarios, we considered probabilistic systems and re-execution one-try attacks, defining suitable security metrics for their safety, studying their asymptotic behaviour (so to have an upper bound) and their rate of convergence to predefined error thresholds.

In Chapter 3, we analysed the case of a single passive eavesdropper. We considered two possible scenarios: the adversary tackled in Section 3.1 directly targets the value of the secret, while the one considered in Section 3.2 is only interested to check whether a certain property related to the secret holds true. In both cases, we showed that the asymptotic behaviour of the security metrics can be determined in a simple way from the channel matrix, we provided simple and tight bounds on them, as function of the number of observations, and we discussed feasible methods to evaluate the rate of convergence. The second scenario allowed us to focus also on the qualitative aspect of the analysis of information flow, differently from most of the previous works, that only consider the quantitative one, ignoring what is leaked.

In Chapter 4, we extended these results to a more sophisticated eaves-

dropping scenario, where the attacker is still passive, but this time can perform a (noisy) observation at each state of the computation, so collecting a sequence of observations for each execution of the system. In this case, we modeled systems as Hidden Markov Models, where the hidden sequence is given by the states traversed by the system, while the visible one is given by the corresponding observations. In particular, we proposed an algorithm to compute in an easy way the limit value of the security metric. This model allowed us to represent more faithfully scenarios where the attacker can collect observations from multiple sources at different times, such as when he has a number of local collaborators.

Turning to more complex attack scenarios, in Chapters 5 and 6, we analysed what happens when we are faced with active adversaries. In Chapter 5, we tackled the non-adaptive case, where the attacker can interact with the system, meaning that, for example, he can control part of the input, but the choice of the latter does not depend on previous observations. We extended the previous model even to this case. In particular, due to the computational difficulty of directly computing bounds and rate of security metrics, we proposed a sub-optimal, but reasonable efficient, strategy, that gives a lower bound on the real rate of convergence. We also made the first step about the integrity issue, proposing a method to detect if an adversary is exercising any undue influence on a deployed system, depending on the outcomes we obtain. Moreover, we analysed the problem of declassification policies from a quantitative point of view, quantifying how serious is a violation of the policy, if there is any. Finally, in Chapter 6, we addressed the adaptive case, where the attacker at each stage of computation can query the system and, from the obtained answer, update his/her knowledge about the secret and choose the query for the next stage. Besides extending the model also to this scenario, we provided several results on the limits of such attackers, comparing adaptive and non-adaptive strategies, and on the problem of searching optimal finite strategies.

There are several directions worth being pursued, starting from the present work. First of all, experiments and simulations with realistic protocols and programs may be useful to asses at a practical level the theoretical results of our study. For example, we believe that HMMs are relevant to security in peer-to-peer overlays. Consider, for instance, Tor, the most widely deployed technology for providing anonymity for users communicating over the internet. We could analyse Timing Attacks on it, applying our model with HMM's, where sequential observations are given by the local times that a message needs to go from one node to

another. We should lightly modify our model, because here we are no longer able to separate single observations (we have only the total time spent by the message to reach the receiver) and, then, even single observations have a variable length.

The application both of the view model and the adaptive one to sparse datasets prompts a connection to database privacy issues that deserves further attention. Concerning the adaptive case, it would be interesting to monitor a possible attack on a database, measuring the amount of information the attacker collects after each query, updating his belief. It would also be interesting to test the analytical model and decision strategy described in Section 5.4 against real-world, deployed systems.

Concerning the adaptive scenario, we would like to implement and experiment with the search algorithm described in Section 6.4. Adaptive querying of dataset, possibly specified via some query description language, might represent an ideal ground for evaluation of such algorithms. Then, we would like to investigate worst-case variations of the framework described in Chapter 6: an interesting possibility is to devise an adaptive version of Differential Privacy (BPSW06; DMNS06), a popular framework to define and enforce privacy for statistics on sensitive data, or one of its variants (BP12b). Relations with *entropic security* (DS05) and other notions related to block cipher cryptography, as outlined in Section 1.2, deserve further investigation.

Finally, in order to make easier the application of our models, it would be interesting to explore the field of Formal Verification, providing a symbolic definition of metrics like the information leakage or the error probability, and developing a type system to statically and automatically measure them.

References

[AAP12] Mário S. Alvim, Miguel E. Andrés, and Catuscia Palamidessi. Quantitative information flow in interactive systems. *Journal of Computer Security*, 20(1):3–50, 2012. 5

[AARR02] Dakshi Agrawal, Bruce Archambeault, Josyula R. Rao, and Pankaj Rohatgi. The em side-channel(s). In *Revised Papers from the 4th International Workshop on Cryptographic Hardware and Embedded Systems*, CHES '02, pages 29–45. Springer-Verlag, 2002. 6

[APvRS10] Miguel E. Andrés, Catuscia Palamidessi, Peter van Rossum, and Geoffrey Smith. Computing the leakage of information-hiding systems. In *TACAS*, volume 6015 of *Lecture Notes in Computer Science*, pages 373–389. Springer, 2010. 69

[BCO04] Eric Brier, Christophe Clavier, and Francis Olivier. Correlation power analysis with a leakage model. In Marc Joye and Jean-Jacques Quisquater, editors, *CHES*, volume 3156 of *Lecture Notes in Computer Science*, pages 16–29. Springer, 2004. 6

[BCP09] Christelle Braun, Konstantinos Chatzikokolakis, and Catuscia Palamidessi. Quantitative notions of leakage for one-try attacks. *Electr. Notes Theor. Comput. Sci.*, 249, 2009. 4, 5, 34, 39, 41

[BK08] Michael Backes and Boris Köpf. Formally bounding the side-channel leakage in unknown-message attacks. In *ESORICS*, volume 5283 of *Lecture Notes in Computer Science*, pages 517–532. Springer, 2008. 5, 7, 64

[BMS10] Béatrice Bérard, John Mullins, and Mathieu Sassolas. Quantifying opacity. In *QEST*, pages 263–272. IEEE Computer Society, 2010. 5, 56

[Bor09] Michele Boreale. Quantifying information leakage in process calculi. *Inf. Comput.*, 207(6):699–725, 2009. 4

[BP12a] Michele Boreale and Francesca Pampaloni. Quantitative multirun security under active adversaries. In *QEST*, pages 158–167. IEEE Computer Society, 2012. viii, 9, 94

[BP12b] Michele Boreale and Michela Paolini. Worst- and average-case privacy breaches in randomization mechanisms. In *IFIP TCS*, volume 7604 of *Lecture Notes in Computer Science*, pages 72–86. Springer, 2012. 125

[BPP11a] Michele Boreale, Francesca Pampaloni, and Michela Paolini. Asymptotic information leakage under one-try attacks. In *FOSSACS*, volume 6604 of *Lecture Notes in Computer Science*, pages 396–410. Springer, 2011. viii, 9, 43, 44

[BPP11b] Michele Boreale, Francesca Pampaloni, and Michela Paolini. Quantitative information flow, with a view. In *ESORICS*, volume 6879 of *Lecture Notes in Computer Science*, pages 588–606. Springer, 2011. viii, 9, 45

[BPar] Michele Boreale and Francesca Pampaloni. Quantitative information flow under generic leakage functions and adaptive adversaries. In *FORTE*, *Lecture Notes in Computer Science*. Springer, 2014, to appear. viii, 10

[BPPar] Michele Boreale, Francesca Pampaloni, and Michela Paolini. Asymptotic information leakage under one-try attacks (full version). *Math. Structure in Computer Science*, to appear. viii, 9, 43, 44

[BPSW06] Michele Bugliesi, Bart Preneel, Vladimiro Sassone, and Ingo Wegener, editors. *Automata, Languages and Programming, 33rd International Colloquium, ICALP 2006, Venice, Italy, July 10-14, 2006, Proceedings, Part II*, volume 4052 of *Lecture Notes in Computer Science*. Springer, 2006. 125

[BS11] Arnar Birgisson and Andrei Sabelfeld. Multi-run security. In *ESORICS*, volume 6879 of *Lecture Notes in Computer Science*, pages 372–391. Springer, 2011. 5, 9, 70, 71, 80, 81

[BV08] Thomas Baignères and Serge Vaudenay. The complexity of distinguishing distributions (invited talk). In *ICITS*, volume 5155 of *Lecture Notes in Computer Science*, pages 210–222. Springer, 2008. 37

[Cha81] David Chaum. Untraceable electronic mail, return addresses, and digital pseudonyms. *Commun. ACM*, 24(2):84–88, 1981. 31

[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. 4

[CPP08a] Konstantinos Chatzikokolakis, Catuscia Palamidessi, and Prakash Panangaden. Anonymity protocols as noisy channels. *Inf. Comput.*, 206(2-4), 2008. 4, 5, 7, 8, 27, 33

[CPP08b] Konstantinos Chatzikokolakis, Catuscia Palamidessi, and Prakash Panangaden. On the bayes risk in information-hiding protocols. *Journal of Computer Security*, 16(5), 2008. 4, 5, 7, 43, 55

[CS04] Imre Csiszár and Paul C. Shields. Information theory and statistics: A tutorial. *Foundations and Trends in Communications and Information Theory*, 1(4), 2004. 88, 90, 91

[CS10] Michael R. Clarkson and Fred B. Schneider. Quantification of integrity. In *CSF*, pages 28–43. IEEE Computer Society, 2010. 6, 70

[Csi98] Imre Csiszár. The method of types. *IEEE Transactions on Information Theory*, 44(6):2505–2523, 1998. 22

[CT06] Thomas M. Cover and Joy A. Thomas. *Elements of information theory (2. ed.)*. Wiley, 2006. 21, 23, 33, 37, 38, 88, 114

[Dan03] George Danezis. Statistical disclosure attacks. In Dimitris Gritzalis, Sabrina De Capitani di Vimercati, Pierangela Samarati, and Sokratis K. Katsikas, editors, *SEC*, IFIP Conference Proceedings, pages 421–426. Kluwer, 2003. 31

[DMNS06] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating Noise to Sensitivity in Private Data Analysis. In *Theory of Cryptography, Third Theory of Cryptography Conference, TCC 2006, New York, NY, USA, March 4-7, 2006, Proceedings*, Lecture Notes in Computer Science, pages 265–284. Springer, 2006. 125

[DS05] Yevgeniy Dodis and Adam Smith. Entropic security and the encryption of high entropy messages. In *TCC*, volume 3378 of *Lecture Notes in Computer Science*, pages 556–577. Springer, 2005. 99, 125

[GM82] Joseph A. Goguen and José Meseguer. Security policies and security models. In *IEEE Symposium on Security and Privacy*, pages 11–20, 1982. 1

[GMO01] Karine Gandolfi, Christophe Mourtel, and Francis Olivier. Electromagnetic analysis: Concrete results. In *Proceedings of the Third International Workshop on Cryptographic Hardware and Embedded Systems*, CHES '01, pages 251–261, 2001. 6

[Gol04] Oded Goldreich. *The Foundations of Cryptography - Volume 2, Basic Applications*. Cambridge University Press, 2004. 122

[GRS96] David M. Goldschlag, Michael G. Reed, and Paul F. Syverson. Hiding routing information. In *Information Hiding*, volume 1174 of *Lecture Notes in Computer Science*, pages 137–150. Springer, 1996. 8, 65

[JO05] Marc Joye and Francis Olivier. Side-channel analysis. In Henk C. A. van Tilborg, editor, *Encyclopedia of Cryptography and Security*. Springer, 2005. 6

[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. 5, 7, 70, 104, 111, 121

[KD09] Boris Köpf and Markus Dürmuth. A provably secure and efficient countermeasure against timing attacks. In *CSF*, pages 324–335. IEEE Computer Society, 2009. 5, 7

[KJJ99] Paul C. Kocher, Joshua Jaffe, and Benjamin Jun. Differential power analysis. In *Proceedings of the 19th Annual International Cryptology Conference on Advances in Cryptology*, CRYPTO '99, pages 388–397. Springer-Verlag, 1999. 6

[Koc96] Paul C. Kocher. Timing attacks on implementations of diffie-hellman, rsa, dss, and other systems. In *CRYPTO*, volume 1109 of *Lecture Notes in Computer Science*, pages 104–113. Springer, 1996. 6

[KS10] Boris Köpf and Geoffrey Smith. Vulnerability bounds and leakage resilience of blinded cryptography under timing attacks. In *CSF*, pages 44–56. IEEE Computer Society, 2010. 5, 41

[KSWH00] John Kelsey, Bruce Schneier, David Wagner, and Chris Hall. Side channel cryptanalysis of product ciphers. *J. Comput. Secur.*, 8(2,3):141–158, August 2000. 6, 30, 103

[LJ97] C. C. Leang and D. H. Johnson. On the asymptotics of m-hypothesis bayesian detection. *IEEE Transactions on Information Theory*, 43(1):280–282, 1997. 37

[Mas94] James L. Massey. Guessing and entropy. In *Proceedings of the 1994 IEEE International Symposium on Information Theory*, page 204, 1994. 16, 17

[MSZ04] Andrew C. Myers, Andrei Sabelfeld, and Steve Zdancewic. Enforcing robust declassification. In *CSFW*, pages 172–186. IEEE Computer Society, 2004. 80, 81

[Rab89] Lawrence R. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. In *Proceedings of the IEEE*, pages 257–286, 1989. 57, 64, 69

[Rén61] Alfréd Rényi. On measures of entropy and information. In *Proceedings of 4th Berkeley Symposium on Mathematics, Statistics and Probability*, volume 1, pages 547–561, 1961. 20

[RR98] Michael K. Reiter and Aviel D. Rubin. Crowds: Anonymity for web transactions. *ACM Trans. Inf. Syst. Secur.*, 1(1):66–92, 1998. 3, 7, 8, 28, 44

[SM03] Andrei Sabelfeld and Andrew C. Myers. Language-based information-flow security. *IEEE Journal on Selected Areas in Communications*, 21(1):5–19, 2003. 1, 80, 81

[Smi09] Geoffrey Smith. On the foundations of quantitative information flow. In *FOSSACS*, volume 5504 of *Lecture Notes in Computer Science*, pages 288–302. Springer, 2009. 4, 5, 8, 18, 20, 34, 40, 71

[SMY09] François-Xavier Standaert, Tal Malkin, and Moti Yung. A unified framework for the analysis of side-channel key recovery attacks. In *EUROCRYPT*, volume 5479 of *Lecture Notes in Computer Science*, pages 443–461. Springer, 2009. 6, 7, 30