Quantitative Approaches to Information Hiding

Christelle Braun

ÉCOLE POLYTECHNIQUE
Thèse de doctorat
Spécialité Informatique

Quantitative Approaches to Information Hiding

Présentée par :
Christelle Braun

Soutenue publiquement le 17 mai 2010 devant le jury composé de :

- Rapporteurs :
  - Michele Boreale
  - MohamamdReza Mousavi
- Directeur de thèse:
  - Catuscia Palamidessi
- Examinateurs :
  - Gérard Boudol
  - Philippe Darondeau
  - Josée Desharnais
  - Steve Kremer
  - Geoffrey Smith

Contents

Chapter 1

Introduction

1.1 Motivations

During the last decade, internet activities have become an important part of many people's lives. As the number of these activities increases, there is a growing amount of personal information about the users that is stored in electronic form and that is usually transferred using public electronic means. This makes it feasible, and often easy, to collect, transfer and process a huge amount of information about a person. As a consequence, the need for mechanisms to protect such information is compelling.

This motivated research on information hiding, i.e., the problem of preventing secrets from being learnt by an adversary. The term information-hiding does not have a precise connotation in literature: we use it in a general sense, to represent a class of problems that can be modeled in the same way by the approaches proposed in this thesis. Two prominent research areas that we intend to address, in particular, are:

- *information flow*, which studies the leakage of classified information via public outputs in programs and systems, and
- *anonymity protocols*, which aim at guaranteeing the anonymity of the users, so that an adversary cannot discover the identity of a user performing a specific action.

Even if these problems have different security concerns, we will see that they can be appropriately modeled in a common framework as information-hiding systems, whose security strength (and hence vulnerability) can be defined and measured in the same way.

To this purpose, it is crucial to rely on quantitative approaches which allow one to assess how much a given information-hiding system can be trusted and to give a relative scale on which different security systems can be compared.

Classifying the sensitivity of data is actually at the essence of information-flow security, which precisely aims at distinguishing the different levels of security held by system components and to adapt the behaviour of the protocol to these degrees. The importance of identifying and protecting sensitive information regularly hits the headlines whenever some big data theft is achieved by hackers, such as more than 40 million credit and debit card numbers stolen in the United States in 2008 [oJ08]. The classifications of objects and messages according to their sensitivity has proven to be

able to provide key solutions to such security threats in today's current applications. A web-scripting language enforcing information-flow policies was for instance designed in 2005 to secure the interfacing with databases in online information systems [LZ05]. This work was based on previous type systems such as Jif [Mye99] (which extends Java) and FlowCaml [SR03] (which extends Caml), both already used in practice to disallow information flow from high to low security levels in sensitive systems.

More recently, the system SABRE (Security Architecture for BRowser Extensions) was developed to analyze JavaScript-based browser extensions (JSEs) by tracking in-browser information flow [DG09]. JSEs are widely used today to enhance the look and feel of common web browsers, which usually have to execute them in privileged mode to allow their full functionality. Malicious JSEs represent however a considerable threat that may, without specific protection, greatly compromise the security of the system. Systems such as SABRE allow in such cases to benefit from functional JSEs while detecting their potential information flow violations.

In 2007, a dynamic analysis system based on information-flow analysis was also specifically developed to detect spyware, i.e., malicious code installed surreptitiously on a computer, which tracks and reports back specific actions of the machine's user [EKK+07]. By analyzing the flow of sensitive information processed by the web browser, this tool was able to classify unknown components as benign programs or spyware and provide comprehensive reports on their behavior. The same year, the system Panorama [YSE+07] was proposed, to detect and analyze malware in a broader context (e.g., keyloggers, password thieves, network sniffers, stealth backdoors, spyware and rootkits) by capturing their typical information access and processing behavior. As a case study, Panorama could detect that even Google Desktop, a popular local file system search tool, may send back sensitive information to remote servers in certain settings.

On the other hand, an increasing number of applications are today more concerned about obfuscating the actual identity of the sender (and/or receiver) of a message than hiding the transmitted information itself. This situation corresponds to anonymity protocols which constitute the second main instance of information-hiding systems we will consider in this thesis. Recently, the demand for anonymity has particularly grown, followed by the development of various anonymization strategies and tools. In the following, we give a few concrete examples which reflect this evolution and justify our motivation for giving in this thesis a particular importance to anonymity systems. Not surprisingly, good as well as bad purposes motivate the need for anonymity. On the one hand, malicious users such as criminals or terrorists may communicate anonymously to prevent law-enforcement bodies from identifying them. On the other hand, anonymity may also be used to counter government censorship which, over the internet, can easily be achieved on a large scale by restricting or blocking transmissions from and/or to specific IP addresses. In countries controlled by repressive governments, enforcement of anonymity allows users to communicate freely with the rest of the world. Recently, the 2006 OpenNet Initiative [ONI], a research project conducted by the universities of Harvard, Cambridge, Oxford and Toronto, studied government censorship in 46 countries and concluded that 25 of them filtered to some extent communications concerning e.g., political or religious positions. People may also anonymously denounce abuse from their employers or report crimes without fear of retaliation. Anonymous tips are widely used nowadays as a source of information by newspapers, and many countries even have laws protecting the anonymity of a person giving tips to a newspapers. Furthermore, legal protections also exist in most countries for the anonymity in communication with priests, doctors, etc. Anonymity

may also increase the objectivity of a message, since the receiver cannot based her interpretation on properties of the sender such as her age, sex or nationality. Relieved from the fear of being identified, an individual may also dare to speak more freely about private problems and find medical or psychological support to solve them. A research showed indeed that, as expected, anonymous participants disclose significantly more information about themselves that non-anonymous users [Joi01].

Pseudonymity is a variant of anonymity, in which the true identity to be hidden is replaced by another one. This strategy may be preferred to perfect anonymity, for instance when two pseudonyms wish to communicate with each others repeatedly and/or for a long time without revealing their true identity, or when a pseudonym sends several messages whose common origin is not part of the secret information. On the other hand, the security provided by pseudonymity is usually weaker than anonymity, since aggregation of information about the same pseudonym from different sources may reveal part of its identity. Pseudonymity is particularly relevant in social networks such as Facebook [Fac] or Twitter [Twi], which are becoming world-wide social phenomena today. Each user, or *avatar*, is characterized by her pseudonym and profile, the set of information she is willing to share with the other members of the online community she belongs to. The privacy and anonymity of social networks were recently analyzed [NS09], which revealed that the information one may learn (even with very little or no effort) on the true identity of a user can significantly exceed the profiled data: one third of the users who could be verified to have accounts on both Twitter and Flickr [Fli] could be re-identified in the anonymous Twitter graph with only a 12% error rate.

In general, the main issue which limits network anonymity is the need to include in every transmitted message the accurate destination address (so that it can be routed to the expected receivers) as well as truthful source information to achieve reliability. In practice, for internet communication, this data is encoded in the IP address of a node, which represents the address of its computer and thus specifies the locations of the source in the topology of the network. This IP number is usually logged along with the host name (logical name of the source). Even when the users connect to the internet with a temporary IP number assigned to them for a single session only, these temporary numbers are also in general logged by the ISP (Internet Service Provider), which makes it possible, with the ISP's complicity, to know who used a certain IP number at a certain time. The anonymity tools currently available aim therefore at preventing the observers of an online communication from seeing the IP address of the participants. Most applications rely on *proxies*, i.e., intermediary computers to which messages are forwarded and which appear then as senders of the communication, thus hiding the original initiator. While a proxy server is today easy to implement and maintain, *single-hop architectures*, in which all users enter and leave through the same proxy, create a single point of failure which can soon significantly threaten the security of the network. *Multi-hop architectures* have therefore been developed to increase the security of the system. In daisy-chaining anonymization for instance, a user's traffic hops deliberately via a series of participating nodes (changed for every new communication) before reaching the intended receiver, which prevents any single entity from identifying the user. Anonymouse [Ans], FilterSneak [Fil] and Proxify [Pro] are famous free web based proxies, while Anonymizer [Ane] is currently one of the leading commercial solution.

Unfortunately, adding an anonymity layer to network communication is usually achieved at the cost of a decrease in performance. Today, the trade-off between anonymity and performance requirements in real-life large networks is an active field of

research, which has to be scaled on a case-by-case basis. Anonymous email services for instance have usually strong requirements on anonymity, but few or no constraints on performance. On the other hand, an online streaming video application may select in priority a path of participants through the network that maximizes bandwidth and minimizes jitter, even if a lower degree of anonymity results and has consequently to be tolerated.

Anonymity plays also a key role in electronic voting. The Caltech/MIT Voting Technology Project (VTP) [VTP] was established in 2000 to "prevent a recurrence of the problems that threatened the 2000 U.S. Presidential Election" and is still actively developing better voting technologies, improving election administration and deepening scientific research in this area. The last decade has namely seen a significant move to e-voting systems in some countries such as the United States. Besides standard rules that have to be satisfied by any paper-based voting system, e-voting has to meet specific requirements before the trio key goals of anonymity, auditability and integrity are achieved. Existing solutions were recently reviewed and their flaws analyzed in [Ren09], which led the authors to the development of *HandiVote*, a new simple, anonymous, and auditable electronic voting which outperforms the limitations of previous systems. One of its main advantages is the possibility for the users to vote by mobile phones or even old-fashioned landline telephones, thus considerably extending the portion of the population which may take part to democratic votes.

Another application which has gained an increasing interest in the last decades is biometric access control. The European Biometrics Forum (EBF) [EBF] is an independent European organisation supported by the European Commission, which was created in 2003 to promote biometrics towards the Policy, Public, Industry and Research audiences. The consortium focuses on the two main functionalities of biometrics today: on the one hand, *identifying biometrics* aims at establishing or verifying the user's true identity and is used, e.g., in border or airport controls. On the other hand, *anonymous biometrics* is used when reliable authentication is required but the user's identity may have to be kept secret. Recently, a new Anonymous Biometric Access Control (ABAC) was for instance proposed, which verifies membership of a user without knowing her true identity [YLZC09]. This system was validated on iris biometrics experiments, illustrating a practical implementation of an anonymous biometric system.

We believe that our framework is general enough to be applicable to a wide spectrum of security systems in use today, which actually tend to increasingly require both information protection and anonymity enhancing technologies.

These concerns are not new, and we will see in the next section the various existing research that has been performed in the field of information-flow security and anonymity, in Section 1.2.1 and Section 1.2.2 respectively.

1.2 Related Work

1.2.1 Information Flow

1.2.1.1 Qualitative Information Flow

Multilevel Security Systems In 1976, Bell and La Padula introduced a security formalism called *multilevel security systems*, in which all components of a security system were classified into *subjects* and *objects* [BLP76]. Objects consisted of passive entities such as files, while subjects were active components such as users or

processes. Processes were further divided into trusted and untrusted entities. In this context, the security policy of the system consisted on a set of rules applied to untrusted subjects, where these latter had to follow the "read down and write up" condition, i.e., untrusted processes were only allowed to read from objects of lower or equal security level and to write to objects of greater or equal security level.

This model was further developed to represent discrete event systems in which information was typically classified into different security levels. In these models, all the sensitive information contained in a process entered in the form of discrete inputs labeled with a security level indicating their sensitivity, and left the process in the form of labeled outputs. An *event* consisted of an input or an output, and the *view* of a security level l corresponded to the events labeled with a level less or equal than l. All other events were said to be *hidden* from l. In this formalism, a *trace* represented a temporally ordered series of events that corresponded to one possible execution of the system.

Usually, only two different security levels were distinguished: *low* information corresponded to public, observable data, while *high* information represented sensitive data that had to be kept secret from observers. The goal of *secure information flow analysis* was then to prevent programs from leaking their high inputs to their low outputs. Under the settings of information flow, *confidentiality* (i.e., ensuring that information is accessible only to the users authorized to have access to it) corresponded to the absence of flows of information from secret inputs to publically observable outputs. Similarly, *integrity* (i.e., ensuring that the data remains valid) meant that there was no flow from a possibility tainted source, low, to an untainted one, high. In this thesis, we will primarily focus on the security goal of confidentiality.

Information flow is still tremendously relevant today as it addresses for instance issues of the everyday use of the internet: the goal of a spyware (trojan horse) is precisely to get access to high data and to leak this information to an attacker legally restricted to access only low data.

Noninterference Even if the Bell-LaPadula model represented a fundamental advance in the protection of systems security, it suffered from several drawbacks. First, it only considered confidentiality as security goal, and thus did neither enforce integrity nor availability. Furthermore, it lacked flexibility because access rules for subjects could not be changed once they had been defined. Last but not least, the model did not prevent the existence of *covert channels*, i.e., transmissions of information using non legitimate data transfer mechanisms that were neither designed nor intended for this purpose. Typical examples of covert channels include *storage channels* and *timing channels*. With a storage channel, one process (the sender) allocates some specific storage location and the other (the receiver) checks for the evidences of this allocation. With a timing channel, the sender influences the timing (e.g. CPU-time) of an event visible to the receiver. In terms of access control, a low subject could for instance send an object to a higher compromised subject who could then reclassify the object to his own (higher) security level or leave it in the lower security level, thus defining two different states that could be observed by the lower user and used to encode one bit. If, for instance, the object was reclassified, then the lower subject would not be allowed to access the object anymore, contrary to the situation in which the object was not reclassified and its access still permitted to the lower user.

These issues motivated the development of stronger multilevel security systems mainly based on the enforcement of *noninterference*, a property stating that security

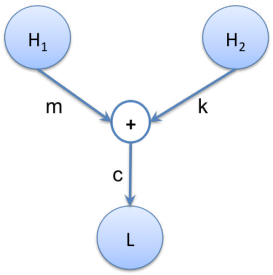

Figure 1.1: Noninterference is not satisfied, although the adversary cannot learn the secret.

can only be reached if high information does not interfere with low data output from the system. In other words, the low outputs are completely independent from the high inputs and any kind of information leakage is forbidden. Goguen and Meseguer gave an extensive study of noninterference [GM82] which they informally defined as follows:

> One group of users, using a certain set of commands, is *noninterfering* with another group of users if what the first group does with those commands has no effect on what the second group of users can see.

They proposed a new model which strictly distinguished between the system and the security policies applying on it. On the one hand, the system was modelled as a generalized automaton called a *capability system*. It consisted of both an ordinary state machine component and a capability machine component which kept track of what actions were permitted to which users. On the other hand, security policies were given as a set of noninterference assertions that declared which information flows were forbidden. They were further divided into *static* policies that always held and *dynamic* policies that depended on the state of the capability component of a system and were thus handled using *conditional* noninterference assertions.

Nondeducibility Goguen and Meseguer's model was however restricted to deterministic systems, in which outputs were entirely determined by the inputs and noninterference held if there was no noticeable difference in the outputs inferred by a removal of the high inputs. In a nondeterministic system however, removing high inputs and re-executing the protocol could lead to an (unrelated) change in the low outputs due to nondeterminism only, but this change would violate noninterference, albeit not revealing a security flaw. In many situations, the requirement of noninterference actually proved to be too strong for the security level needed in a system. Consider for instance the system illustrated in Figure 1.1, in which a high-level user H_1 sends a message m which is xor-ed by a string k issued by a second high-level user H_2 before being transmitted to a low-level receiver L. In this case, noninterference does not hold because high-level information is leaked from the system (in form of the cypher text c), even if the low-level user, because of the encryption, cannot learn anything from the message.

Most of the following research was therefore aimed at weakening noninterference and extending it to nondeterministic systems. In 1986, Sutherland proposed

a slightly weaker definition of security than noninterference, called *nondeducibility* [Sut86]. According to this definition, information flow was interpreted as deducibility and a system was secure if it was impossible for a user to *deduce* the actions of other users having a higher security status, thus being closer in spirit of the consideration of security as "nondisclosure of information". In the previous example of an encrypted high-level message transmitted to a low-level user, nondeducibility was enforced since the output message c did not increase the receiver's knowledge. In a deterministic system, nondeducibility held therefore whenever a low-level user could not select, from the outputs, a preferred input whose occurence was more likely than the others. This definition could be generalized to nondeterministic system, by interpreting nondeducibility as the impossibility for a low-level subject to rule out any consistent high input from the low outputs of the system.

Restrictiveness In 1990, McCullough, an expert in network systems, argued that Sutherland's model was not appropriate for the protection of security in real-life systems which often showed nondeterministic and asynchronous behaviours. In particular, McCullough described examples of nondeterministic systems which satisfied nondeducibility while violating basic security requirements [McC90]. Consider for instance a nondeterministic system which fills in the spaces between messages with strings of random bits to prevent attackers from analyze the flow of traffic on the system. Nondeducibility would be preserved even if a low-level user could read unencrypted high messages. If the attacker reads e.g., "We attack tomorrow", then she might guess that the string belonged to the high message, which might be true with high probability. However, the attacker would not be able to deduce with certainty that the sentence was not generated by chance in the string of random bits, hence enforcing nondeducibility. This criticism could be seen as an early attempt to advocate for the use of a quantitative definition, since the security of the system relied on "how much" an attacker could deduce about the high information given the observables, and nondeducibility held as soon as some information concerning high variables could not be deduced with certainty.

In order to address this issue, McCullough's first contribution was to define *generalized noninterference*, a notion of noninterference that could be extended to nondeterministic systems [McC87]. McCullough also pointed out the crucial importance of *hook-up security*, i.e., composability in multi-users systems, a property which has today become all the more relevant with the increasing complexity of computer systems and will therefore be specifically addressed in this thesis. Composability is a desirable property for several reasons. First, complex systems cannot today be built and maintained effectively in the long run if they cannot be decomposed into smaller components that can be handled separately and added or removed when necessary without perturbating the whole system. This is particularly relevant in open systems, which usually do not have a permanent notion of "entire system" and may evolve quickly. On the other hand, composability makes it much easier to check that a given property holds for the global system if it is sufficient to check it at the scale of the components of the system. With the degree of complexity reached by today's computer systems, composability has become a necessary requirement for building a modular and flexible system.

McCullough's contribution was to show that the three main approaches to system security that had been used so far, i.e., the Bell-LaPadula (access control), noninterference and nondeducibility all failed to be composable. Moreover, McCullough

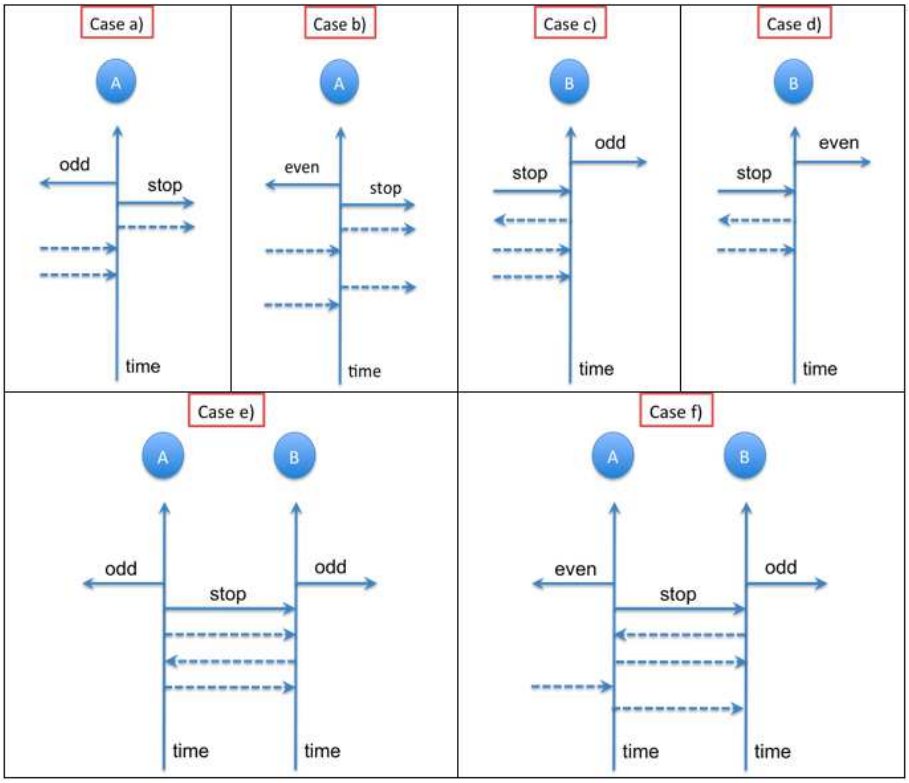

Figure 1.2: Generalized noninterference is not composable

recognized that his generalized noninterference was not hooked-up secure either, and was thus not yet a satisfactory security property for multi-users systems. The example provided by McCullough in [McC90] to show that generalized noninterference was not composable is illustrated on Figure 1.2.

The system consists of a component A represented on a vertical time axis in Figure 1.2 (case a and case b), which can exchange low-level and high-level inputs and outputs. Low-level inputs and outputs are represented as solid arrows, while high-level inputs and outputs are represented as dashed arrows.

The component is subject to a random number of high-level inputs (from its left) and high-level outputs (to its right) possibly occuring after some delays. At one point in time, the low message `stop` is output to the right and is immediately followed by another low message `odd` or `even` (called *parity message*) which is output to the left, leaking the parity of the number of high-level events (inputs and outputs) that have occured so far (i.e., before the output of `stop`). Since the high-level outputs may be delayed and the two low outputs can occur at any time one after the other, all high-level inputs may not have been output when the two low outputs occur.

This system is noninterference secure: if there had been an even number of high-level inputs, the output would be either {`stop, odd`} if an odd number of high-level outputs occured so far (case a in Figure 1.2), or {`stop, even`} if an even number of high-level outputs occured so far (case b in Figure 1.2). An odd number

of high-level inputs would lead to the symmetrical situation and therefore a change in the high inputs would not affect the low-level outputs, as required by noninterference (i.e., whatever the high inputs may be, an adversary observing the low outputs of the protocol can never rule out any of the two possibilities {`stop, odd`} or {`stop, even`}).

We now consider the component B which is similar to A except that the high-level outputs occur to the left, the parity messages are output to the right and the `stop` message is now a low-level input that comes from the left. The component B works similarly to A and it is easy to see that it satisfies noninterference as well (as illustrated on cases c and d in Figure 1.2).

Cases e and f in Figure 1.2, show that it is now possible to connect A to the left-hand side of B so that the outputs to the right of A become inputs to the left of B and the outputs to the left of B become inputs to the right of A. We assume here that B cannot receive any high-level input from the outside world. Under these settings, the combined system does not satisfy noninterference anymore: since the high-level outputs from A are input to B and vice-versa, one can easily see that A and B both output the same parity messages as long as no input to A from the outside occured (case e in Figure 1.2). However, an adversary can deduce with certainty that some high-level input occured from the outside if A and B do not output the same parity messages (case f in Figure 1.2). Therefore a change in the high inputs could affect the low-level outputs of the combined system, which violates generalized noninterference.

In order to enforce composability of security in nondeterministic systems, McCullough defined a new security property called *restrictiveness*, equivalent to noninterference for deterministic systems but extended to nondeterministic systems. The main problem with the previous approach was the need for a machine which behaves the same whether a low-level input was preceded by a low-level, a high-level or no input. The component B in the previous example for instance does not satisfy this property, since it outputs *even* if no high-level input occured, and *odd* otherwise. McCullough modelled the system as a state machine and specified a set of rules that had to be fulfilled by the state machine to satisfy *restrictiveness*, i.e., to prevent any high-level information from affecting the behavior of the system, as viewed by a low-level user. The important notion of *view* in this model was defined as an equivalence relation on states and input sequences. Then McCullough could prove that restrictiveness was composable. In the previous example, the fact that a high-level input followed by `stop` did not have the same effect as `stop` alone violated restrictiveness for component B.

Forward correctability In 1988, Johnson and Thayer argued that hook-up security as defined by McCullough was often stronger than needed in typical real-life applications such as basic text editors [JT88]. In order to justify their statement, they first defined a weaker notion of security called *n-forward-correctability*, based on the notions of *perturbation* and *correction*. A perturbation is a sequence of events (which may not necessarily be a valid trace) obtained from a valid trace by inserting or deleting high-level inputs. A correction consists then of a valid trace which is obtained from a sequence of events by inserting or deleting high-level outputs only. An event system is n-forward-correctable iff for any trace α and any perturbation α' obtained by inserting or deleting a single high-level input before at most n low-level inputs preceding a high-input-free segment γ , there is a correction of α' supported in γ. Johnson and Thayer proved that McCullough's hook-up security is equivalent to ω-

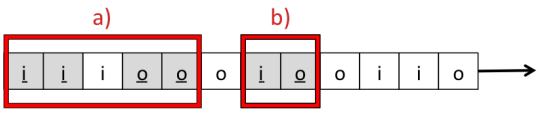

Figure 1.3: Valid trace for the forward correctability example

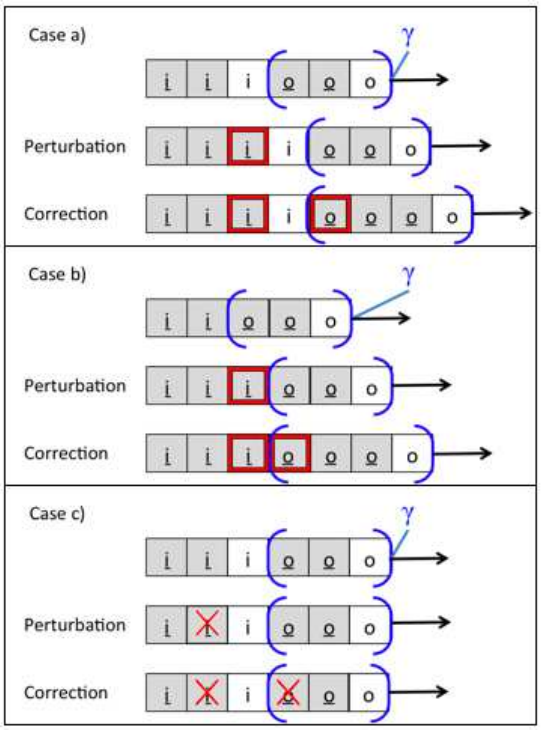

Figure 1.4: Forward correctability does not imply hook-up security

forward-correctability, i.e., n-forward-correctability for any integer n. In other words, it is possible to correct a perturbation after an arbitrarily long sequence of low-level inputs. As shown in [JT88], this is strictly stronger than simple forward-correctability ($n = 1$), because of the following relation:

$$\forall m, n, \text{s.t. } m > n \quad m\text{-forward correctability} \quad \not\Rightarrow \quad n\text{-forward correctability} \tag{1.1}$$

This proposition can be highlighted by a simple example in the case $n = 0$, $m = \omega$, which shows that 0-forward correctability (i.e., simple forward-correctability) does not imply ω-forward correctability. We consider a low-level monitoring task checking whether messages between two high-level processes have been transferred (without being itself able to read the content of the messages).

In order to comply with the specification saying that any high input (a message released by the sender process) has indeed been output (received by the receiver process), it is required that every low output (an aknowledgment of the monitoring task)

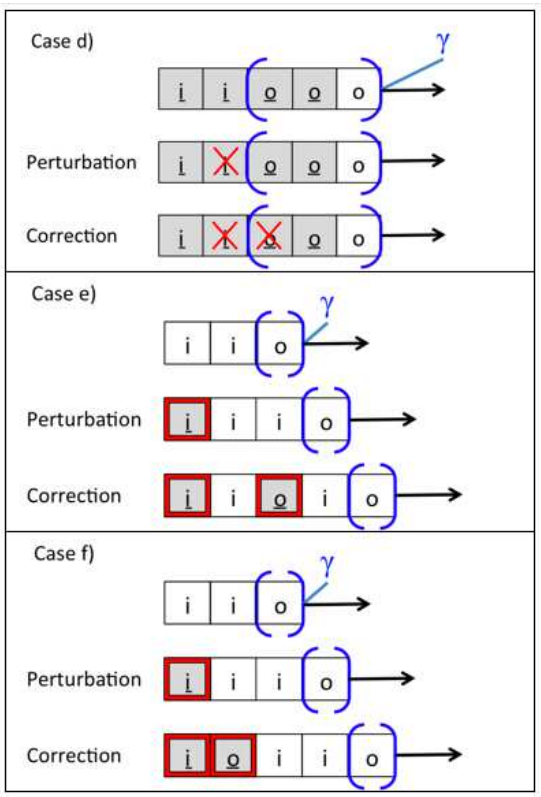

Figure 1.5: Forward correctability does not imply hook-up security (cont.)

is preceded by an equal number (possibly zero) of high inputs and high outputs. Any high output o_h occurs therefore in a segment of the form $i_h^+[i_l]o_h^+$, i.e., a finite number of high inputs followed by a finite number of high outputs, with possibly one low input between the two sequences (see the segments a and b in the trace given as example on Figure 1.3). In this figure, high inputs i_h (resp. low inputs i_l) appear as grey (resp. white) squares labeled $\underline{i}$ (resp. i), while high outputs o_h (resp. low outputs o_l) appear as grey (resp. white) squares labeled $\underline{o}$ (resp. o). A perturbation of this system can be performed by inserting or deleting a high input i_h closely before a high-level-free final segment γ (i.e., immediately before γ or before a low-level input immediately preceding γ). Then a correction must be applied in case a low output is contained in the final segment in order to restablish the balance between high inputs and high outputs before the low output occurs.

This can be achieved by inserting a high output o_h immediately after an i_h (case b in Figure 1.4) or after a low input if it occured right after an i_h (case a in Figure 1.4). On the other hand, a perturbation may also come from the deletion of a i_h preceding a high-level-free final segment γ with or without a low level input preceding γ (cases c in Figure 1.4 and case d in Figure 1.5 respectively). Again, if the final segment contains a high output, a correction must be applied, as illustrated in the figure. Since all these corrections are performed *inside* the sequence γ, the system satisfies 0-forward correctability (case b in Figure 1.4 and case d in Figure 1.5) and 1-forward

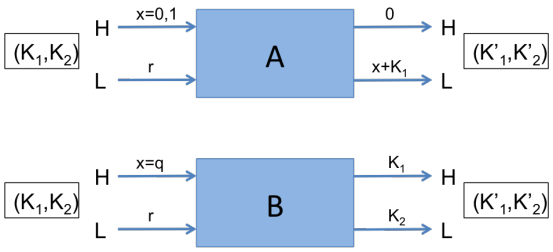

Figure 1.6:

correctability (cases a and case c in Figure 1.4).

On the other hand, it is not McCullough hook-up secure (i.e., ω-forward correctable): consider the perturbation $i_h i_l i_l o_l$, where i_h has been inserted in the trace $i_l i_l o_l$ (cases e and f in Figure 1.5). The only possible corrections are $i_h o_h i_l i_l o_l$ (case f) and $i_h i_l o_h i_l o_l$ (case e). However, these corrections are performed *outside* the sequence γ, therefore the system is not 2-forward correctable and thus not ω-forward correctable.

Johnson and Thayer argued that requiring forward correctability is strong enough for a large set of security systems, and they proved that this property is also composable, i.e., two hooked-up systems which are individually forward correctable lead to a combined system which is forward-correctable as well.

Nondeducibility on strategies Two years later, Wittbold and Johnson pursued the refinement of security properties and proposed *nondeducibility on strategies* (NoS) [WJ90], an extension of Sutherland's deducibility theory (called nondeducibility of inputs NoI in the following), weaker than forward correctability but strong enough to avoid the drawbacks of NoI and to be applicable to nondeterministic systems that may be networked. In particular, the following example shows how a system which satisfies NoI can be used to transmit information through a covert channel that is prevented when the system satisfies NoS. Consider the nondeterministic state-machine pictured on Figure 1.6. At each step of the computation, the system is in a state (K_1, K_2) hidden from the high transmitter H and to the low receiver L. H and L both give an input (resp. x and r), where $x \in \{0, 1, q\}$. As illustrated on Figure 1.6, if $x \in \{0, 1\}$, then transition A occurs. Otherwise transition B occurs. The outputs O_H and O_L given respectively to H and L in each case are written above the output arrows on the figure. Note that K_1' and K_2' correspond to random updates of K_1 and K_2.

Since H determines the outputs by his input x, he can learn at any step the value of K_1 by giving $x = q$ as input. Then, if he sends as next input the value $x \in \{0, 1\}$, this value is encoded into $x \oplus K_1$ (random pad) and the result of this operation received as output by L, as illustrated in case A on the Figure 1.6. If H repeats this protocol and L ignores one bit out of two (namely the bit output to L when K_1 is output to H, corresponding to the transition B on the Figure 1.6), any encrypted message can be successfully transmitted from H to L without L ever learning the value of H's initial inputs (since L never learns K_1 and is therefore unable to decrypt the message). This system is thus nondeducible on inputs because L cannot learn H's input string. However, L deduces information on H's input strategy, which means that this system

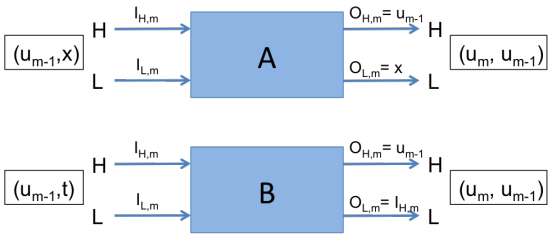

Figure 1.7: Nondeducibility on strategies does not imply forward correctability

violates NoS.

After providing a formal definition of this notion, Wittbold and Johnson related it to forward correctability by proving the following relation:

$$\text{forward correctability} \not\Leftrightarrow \text{NoS} \tag{1.2}$$

i.e., forward correctability is strictly stronger than NoS, which can again be understood from an example given in [WJ90]. Consider the state machine with two consecutive states $s_{m-1} = (u_{m-1}, v_{m-1})$ and $s_m = (u_m, v_m)$. Again, one low-level user L and and one high-level user H give inputs I_L and I_H and receive outputs O_L and O_H respectively. In a trace, a state transition is represented by the tuple $(I_{L,m}, I_{H,m}, O_{L,m}, O_{H,m})$.

For each m we have:

$$\begin{aligned} &u_{m-1} = v_m = O_{H,m} \\ &\text{If } v_{m-1} = x \in \{0,1\}, \text{ then } O_{L,m} = x \\ &\text{If } v_{m-1} = t, \text{ then } O_{L,m} = I_{H,m} \end{aligned} \tag{1.3}$$

This defines two kinds of transitions A and B depending on the value of v_{m-1}, as illustrated on Figure 1.7. Consider now the following trace, illustrated on Figure 1.8:

$$(t,0)(r,0,0,t)(0,t)(r,1,1,0)(1,0) \tag{1.4}$$

And its perturbation, illustrated by the crossed input on Figure 1.8:

$$(t,0)(r,0,0,t)(0,t)(r,0,1,0)(1,0) \tag{1.5}$$

This perturbation affects the input of H in case it is given as output to L. The only way to correct this perturbation and thus avoid a change in the output to L is to modify $u_{m-1} = v_m$ from t to 0 or 1, hence avoiding the propagation of the perturbation in the low output. However, this correction violates forward correctability, because it is performed *before* the perturbated input. On the other hand, there is no deduction about H's behaviour that could be made by L from its view of the system, so NoS is not violated and this security property is more appropriate than forward correctability in this case to reflect the desirable security requirement satisfied by the system.

As explained in [WJ90], the difference between the two notions comes from the fact that a system violates forward correctability whenever a deduction can be made by a low-level user L from the entire input/output history of the computation, possibly

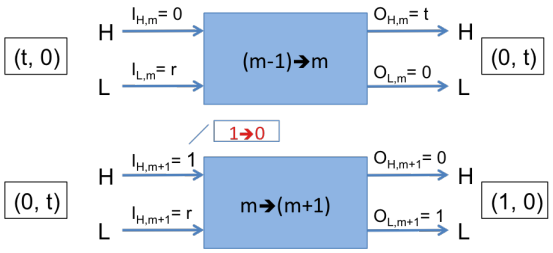

Figure 1.8: Nondeducibility on strategies does not imply forward correctability (cont.)

including the high user's history which, in reality, remains hidden from L. In our example, L can deduce that the output he gets should match the high input *as long as he knows that the high output previously received by H was t*. However, L is usually not allowed to see the output history of H, and therefore in this case the requirement for forward correctability is too strong. On the other hand, NoS only requires a low user to be prevented from making deductions based on his own input/output history, i.e., on what is indeed visible to him in reality.

In [WJ90], Wittbold and Johnson also insisted on the necessity to adopt a probabilistic approach, rather than a possibilistic one, in order to guarantee the protection of security in a nondeterministic context. They motivated their argument by the example of the two matrices on Figures 1.9 and 1.10, where each row represents a possible strategy π_i of the high user (i.e., a secret) and each column stands for a possible view o_j of the low-level user:

	o_1	o_2	...	o_m
π_1	p	i	...	p
π_2	p	p	...	i
...	...	...	...	...
π_n	p	p	...	p

Figure 1.9: Possibilistic approach: for each strategy π_i each low view is either possible (p) or impossible (i) [WJ90]

The first matrix, which corresponds to the possibilistic approach, is binary and each element $m_{ij} \in \{p, i\}$ specifies whether the view o_j is possible (p) or not (i), given that the strategy of the high-level user was π_i. The second matrix, on the other hand, corresponds to the probabilistic approach, and each coefficients gives the *likelihood* $p_{ij} = p(o_j|\pi_i)$ of the view o_j given the strategy π_i.

In the first case, NoS is satisfied iff there are no i entries in the matrix, i.e., all views are possible. In a probabilistic context, this is equivalent to saying that all likelihoods are positive. In the second case however, NoS is satisfied iff all likelihoods are the same for a given strategy, i.e., all the rows are equal, which is a significantly stronger requirement (and corresponds to our notion of strong anonymity as explained further on in this thesis).

	o_1	o_2	...	o_m
π_1	p_{11}	p_{12}	...	p_{1m}
π_2	p_{21}	p_{22}	...	p_{2m}
...	...	...	...	...
π_n	p_{n1}	p_{n2}	...	p_{nm}

Figure 1.10: Probabilistic approach: p_{ij} is the likelihood of the view o_j given the strategy π_i [WJ90]

Nondeducibility on composition An interesting extension to nondeducibility on strategies called *nondeducibility on composition* was developed in [FGM95] and proved to be specifically suitable for process calculi. In that paper, a variant of Milner's CCS is used to compare different definitions of noninterference. The notion of strategy expressed in the previous approach is here formalized as a high-level process which interacts with the system. Security of the system is achieved if such interactions are not observable by low-level users. Several other papers have explored this property [FG96, FG97, BFPR02, FR02], and real-time and probabilistic information flow analyses were performed in [FGM03] and [ABG04] on suitable extensions of CCS. Along the same line of research, noninterference was also expressed in the setting of CSP calculus [BHR84, Ros95a, RWW94, Rya91]. A comparison between this formalization of noninterference and nondeducibility on composition and an overview on several noninterference notions from a process algebraic perspective were given respectively in [Foc96] and [RS99].

On the other hand, several extensions to the π-calculus were developed to address more specifically security concerns. In particular, the spi-calculus, defined by Abadi and Gordon [AG97, AG99], used cryptographic primitives to describe and analyze authentication protocols. Hennessy and Rieley proposed an extension of the asynchronous π-calculus in which a variety of security properties were captured using types [HR02]. In their work, the multi-level security and access control properties were defined through the typing system which prevented implicit information flow from high-level to low-level processes.

Restricted composition All the aforementioned approaches to composability aimed at modifying security properties so that they could be preserved under arbitrary compositions. Another strategy consists in restricting composition so that a given security property can be preserved. Zakinthnos and Lee developed restricted forms of composition [ZL95, ZL96, ZL98] and proved that McCullough's generalized noninterference is preserved by composition if the combined system is *feedback-free*, i.e., for all connected pair (c_i, c_j) of components (meaning that at least one output from c_i is an input to c_j), then (c_j, c_i) is not connected. This can be restated as the requirement that no *feedback loop* occurs in the combined system, where a feedback loop to a component i occurs if there exists a trace which starts and ends in i. Zakinthnos and Lee also also showed that generalized noninterference could be preserved by composition even in the presence of feedback loops as long as a delay component was inserted into all feedback loops that involved high-level events. More precisely, these

components needed to delay any feedback to after the next low-level event. Another interesting result related to compositionality in [ZL98] is the derivation of conditions for *emergent* properties, i.e., properties that are not necessarily satisfied by all individual components but hold when these components are composed if certain conditions are fulfilled. Mantel used a generalization of the zipping lemma [JT88] to explore this phenomenon [Man02].

1.2.1.2 Quantitative Information Flow

In many security systems, a small information leakage may be acceptable, but cannot be handled by the aforementioned (qualitative) models. Guessing a password in an access control protocol releases for instance inevitably part of the high information, because the search space has become smaller whatever the result of the guess is. Similarly, in an anonymous voting protocol, an observer should be able to see the number of votes for each candidate but not the identity of the voters. Several approaches, reviewed in Sabelfeld and Sands [SS05], have been developed to address these violations of noninterference.

In the following, we review the strategies that have been developed to address this issue by *quantifying* the notion of noninterference, i.e., measuring the amount of interference between high and low information occuring in a system. This allows, depending on the specificities of the system under concern, to set a threshold corresponding to the maximal acceptable level of interference, thus giving much more flexibility than did the qualitative notion of interference developed so far.

Early approaches for quantifying information flow were focused on the detection of covert channels between processes in multi-user systems, and Shannon's information theory was the preferred approach to tackle this problem, as it could be used to prevent information flow between processes which were not explicitly covered by the access rules of the security policy.

History-free approach In 1982, Denning explored the use of information theory as the basis for a quantitative analysis of information flow in programs [Den82]. She identified the *quantity of leakage* from a state s to a state s' of the system as the decrease in uncertainty about the high information in state s, resulting from the knowledge of the low information in s'. Denning used the conditional entropy $H(h_s|l_s)$ to quantify the uncertainty about the high data in state s (denoted h_s) given the low data in state s (l_s). Then, the existence of an information flow from high to low is equivalent to the condition:

$$M_1 \quad = \quad H(h_s|l_s) - H(h_s|l_{s'}) \quad > \quad 0 \tag{1.6}$$

i.e., the uncertainty about h_s given l_s after the execution of the protocol is smaller than before the execution. If this condition holds (i.e., if M_1 is positive) then M_1 quantifies the amount of information leaked from high to low.

However, this definition suffered from a flaw identified by Clark et al. in [CHM07]: the lack of consideration of the history of low inputs. Clark et al. considered the example of the two following programs, where x is a high-level integer variable which takes a value in -16,...,15 with uniform distribution, and y is a low variable initialized to $\texttt{abs}(x)$ (the absolute value of x) in the initial state s. Variables (e.g., x) in state s'

are primed (x') while variables in state s'' are double-primed (x'').

$$\begin{array}{lll} (A) & \{s\} & \textbf{if } x = 0 \textbf{ then } y = 1 \textbf{ else } y = 2 \quad \{s'\} \\ (B) & \{s\} & \textbf{if } x < 0 \textbf{ then } y = 1 \textbf{ else } y = 2 \quad \{s''\} \end{array} \tag{1.7}$$

We have:

- $H(h_s|l_s) = H(x|y = \texttt{abs}(x)) = (1/32)H(x|y = 0) + (31/32)H(x|y > 0) = (31/32)\log 2 = 31/32$
- $H(h_s|l_{s'}) = (1/32)H(x|y' = 1) + (31/32)H(x|y' = 2) = (31/32)\log 31 = 4.8 > (31/32)$
- $H(h_s|l_{s''}) = (1/2)H(x|y'' = 1) + (1/2)H(x|y'' = 2) = \log 16 = 4 > (31/32)$

Since $H(h_s|l_s) < H(h_s|l_{s'})$ and $H(h_s|l_s) < H(h_s|l_{s''})$, M_1 is strictly negative for (A) and for (B), therefore Denning's calculation gives that there is no leakage from high to low in these programs. However, intuitively one would expect that there is leakage in (B) because the value of y'' (i.e., y in state s'') specifies the sign of x, a new piece of information that was not available in the previous state s. On the other hand, the value of y' in (A) only specifies whether x is different from zero or not, an information which could already be deduced in the previous state from the value of $y = \texttt{abs}(x)$. Thus there is no new knowledge in state s' compared to state s, and we would therefore expect a measure of leakage to give zero in this case.

It is easy to see that the problem comes from the low-level information $y = \texttt{abs}(x)$ in state s, which gives a lot of information about x (i.e., at most two possible values remain for x when y is known) while y' or y'' gives less information about x, *assuming that the value of y in state s has been "forgotten"*. However, in most real systems, the attacker is able to keep track of the previous low-level values (i.e., he has a memory), which lead to Clark et al.'s refinement of Denning's condition for the existence of information flow from high to low [CHM07]:

$$M_2 \quad = \quad H(h_s|l_s) - H(h_s|l_{s'}, l_s) \quad > \quad 0 \tag{1.8}$$

Since $H(X|Y, Z) \leq H(X|Y)$ for any random variable X, Y, Z (conditioning reduces entropy), we have $M_1 \leq M_2$. The quantity M_2 corresponds to conditional mutual information, to which we will come back later in this chapter.

Applied to the programs (A) and (B), this yields:

- $H(h_s|l_s) = H(x|y = \texttt{abs}(x)) = 31/32$
- $H(h_s|l_{s'}, l_s) = (1/32)H(x|y' = 1, y = \texttt{abs}(x)) + (31/32)H(x|y' = 2, y = \texttt{abs}(x)) = (31/32)\log 2 = 31/32$
- $H(h_s|l_{s''}, l_s) = (1/2)H(x|y'' = 1, y = \texttt{abs}(x)) + (1/2)H(x|y'' = 2, y = \texttt{abs}(x)) = 0$

The quantity M_2 is therefore equal to 0 in (A) and to 31/32 in (B), so there is leakage in (B) but not in (A), which complies with our intuition. We will come back to Clark et al.'s approach [CHM07] later on in this section.

Millen's Model Millen made an important step in the connection between information flow models and Shannon information theory, by establishing a formal analogy between noninterference and mutual information. He modelled the multi-user computer system as a channel with a sequence of inputs W (possibly coming from several users) and an output Y at the end of the computation [Mil87]. In his approach, the random variable X was a subsequence of W which represented the high input from a user U and $\bar{X}$ was the *complement* of X, i.e., the subsequence of W consisting of inputs from users other than U. Millen showed that for deterministic systems, if X was non-interfering with Y, then $I(X;Y) = 0$, provided that X and $\bar{X}$ were independent. The quantity $I(X;Y)$ was the classical Shannon's *mutual information* between X and Y[Sha48] and represented the information flow between the high input X and the output Y. In other words, under the assumption that X did not depend on the other inputs, noninterference was a sufficient condition for the absence of information flow. Millen also provided a counter-example to show that this condition was not necessary, i.e., the information flow could be zero even when X was *not* non-interfering with Y. His argument actually relied on the same observation that had lead Sutherland to define one year before nondeducibility as security property: the impossibility for noninterference to make a difference between the true eavesdropping of high-level data by a low-level user (where the information was leaked, e.g., through a covert channel), and the obtention of the encrypted version of a high-level message from which nothing could be deduced by the low-level user. In both cases, there was a strictly positive interference between high and low users but in the second case the mutual information between them was zero since the low-level user could not extract any information from the leaked data (assuming he did not know the decryption key).

Flow Model In 1990, McLean argued that a better distinction between allowed and forbidden information flows in a program was possible when the notion of time was introduced in the model, so that causal relations were specified explicitly [CM90]. He developed the security model called Flow Model (FM), which took time into account and viewed information as flowing from a high-level user H to a low-level user L only if H assigned values to objects in a state that preceded the state in which L made its assignment. Therefore, only certain classes of dependency between H and L were considered security violations, which gave more expressiveness in the specification of the system security properties than previous models did and was suitable to prevent information flows in systems with memory.

However, McLean's model was highly general and abstract, and lacked therefore the potential to be applied to the analysis of real and complex systems which were thriving in the early ninethies. Moreover, its equation defining security was dependent on the input probabilities which are usually unknown and we will in this thesis, as explained later in more details, try to abstract from them whenever possible.

Trade-off between general abstract models and simple restricted ones Gray proposed in 1991 a general purpose probabilistic state machine extending Millen's model, with the purpose to bridge the gap between the two main categories of models that had been developed so far in the field of information flow security [III91]: on the one hand, general models such as McLean's Flow Model (FM) [CM90] which were appropriate to evaluate security models but remained very abstract and thus hardly applicable to real systems, and models such as Millen's [Mil87, Mos91, WJ90] on the other hand, which were focused on concrete examples but tended to be so simple that they

could neither be used to describe real-life complex systems, such as computers with a general purpose memory. Gray proposed therefore to make a "trade-off" between these two tendencies and presented a general framework for the study of information leakage based on a probabilistic state machine and resembling Millen's model (in particular for the use of channels, inputs and outputs), but with a probabilistic rather than a nondeterministic transition function. More precisely, the transition from state s to state s' after an input I and yielding output O was specified in Gray's model by a probability $T(s, I, s', O)$, while a transition in Millen's model was given in terms of the two random variables W (representing, as described in the previous paragraph, an unknown interleaving between input X and all other inputs $\bar{X}$) and $Y = out(t, y)$ the output to user y in t, the state at the conclusion of the trial.

Furthermore, Gray partitioned the channels into two sets H and L representing respectively the channels connected to high and low processes. The generality of Gray's model came from the assumption that the only information about the system and its environment that was directly accessible to the high (resp. low) environment were the inputs and outputs that had previously occured on the high (resp. low) channels. So if the high environment obtained information about the low environment, it had to obtain it indirectly through its interaction with the system, and vice-versa (i.e., there was no direct communication exterior to the system). This allowed the environment external to H (resp. L) to use feedback from the system and to have memory of what the system had already done on channels in H (resp. L), thus capturing many real systems. This approach could be related to Wittbold and Johnson's nondeducibility on strategies due to feedback (see Paragraph 1.2.1.1).

After specifying individual security properties for the system and the environment, Gray defined a general probability measure P to reason about the probabilistic behaviour of both the system and the environment and used it to specify a condition for information flow security:

$$\begin{array}{rcl} P(L^I \cap L^O \cap H^I \cap H^O) > 0 & \Rightarrow & \\ P(l|L^I \cap L^O \cap H^I \cap H^O) & = & P(l|L^I \cap L^O) \end{array} \tag{1.9}$$

where L^I (resp. L^O) is the occurence of a particular input (output) history on the channels in L. The same holds for the channels in H and l is a final output event occuring on the channels in L.

This condition meant that the probability of a low output l could depend on previous low events $L^I \cap L^O$ but not on previous high events $H^I \cap H^O$. As noted by Gray, the output history on high channels H^O was necessary even if one could first believe that the following weaker condition was sufficient:

$$\begin{array}{rcl} P(L^I \cap L^O \cap H^I) > 0 & \Rightarrow & \\ P(l|L^I \cap L^O \cap H^I) & = & P(l|L^I \cap L^O) \end{array} \tag{1.10}$$

However, this condition did not rule out the existence of a covert channel. Consider for instance an encryption scenario in which the high inputs H^I are encrypted and thus hidden from the low outputs L^O. Now if the high environment can gain knowledge (e.g., through the high outputs H^O) on the encryption method (e.g., by observing some probabilistic pattern), then it could modify its input value accordingly and successfully transmit information to the low environment. In this case Condition 1.10 would hold but would not be sufficient to guarantee information flow security.

Another interesting contribution of Gray was his explicit definition of the capacity of the channel from H to L, which generalized the definition initially given by Shannon for memoriless discrete noisy channels [Sha48]. In order to account for the fact that the covert channels he considered could have memory and feedback, Gray calculated the mutual information between the low output at time t and the entire history of high inputs and outputs from time 0 through time $t-1$. Additionally, Gray used conditional mutual information in order to take into account the knowledge of the low inputs and low outputs in the history. He obtained following channel capacity from H to L:

$$\begin{array}{lcl} C & \equiv & \lim_{n\to\infty} C_n \\ \text{where} & & \\ C_n & \equiv & \\ & & \max_{H,L} \frac{1}{n} \sum_{i=1}^{n} I(\text{In\_Seq\_Event}_{H,i}, \text{Out\_Seq\_Event}_{H,i}; \\ & & \text{Final\_Out\_Event}_{L,i} | \text{In\_Seq\_Event}_{L,i}, \text{Out\_Seq\_Event}_{L,i}) \end{array} \tag{1.11}$$

In this definition, $\textit{In\_Seq\_Event}_{A,t}$ represents the occurence of a particular input history on the channels in A up to and including time $t-1$. The other terms in the mutual information are defined similarly. Gray showed that under this definition, the absence of information flow from H to L implies $C = 0$.

Information escape McIver and Morgan used an expression of channel capacity very similar to Equation 1.11, but based on a new notion of flow quantity in the context of program refinement and for a sequential programming language enriched with probabilities [MM03]. Their security goal differed from most traditional approaches of information flow, which usually focused on the protection of the initial values (only) of the high information. This is what McIver and Morgan called *weakly secure*. On the other hand, their security goal was to ensure that the privacy of the high variables was *maintained* (continuously) along the execution of the program:

> A system comprising operations Op is secure provided that if the value of $High$'s variables are not known (to Low) initially, then they cannot be inferred at any later time during use.

This requirement was stronger than weakly secure, and the authors showed that security of final values implied security of initial values for standard deterministic programs. McIver and Morgan defined the flow quantity from H to L (called *information escape* in [MM03]) as the difference given by:

$$H(h|l) - H(h'|l') \tag{1.12}$$

where h and l were respectively the high security and low security partitions of the store at the start of the program, and h' and l' the high security and low security partitions of the store at the end of the program.

The authors defined the channel capacity as the least upper bound over all possible input distributions of the information escape, and showed that security of the program was equivalent to the channel capacity being equal to zero.

Moreover, one of their theorems provided interesting alternative equivalent formulations to their notions of security for a program P:

1. P is secure

2. P is a secure permutation of the high variables

3. P is uniform preserving

The second statement means that a program can only permute high values. The third statement means that from the point of view of a low observer, the execution of the program gives no information about the probability distribution of the high variables. Since complete ignorance of h is equivalent to the uniform distribution, the high variables should appear as uniformly distributed to the low observer during the whole execution of the program. In other words, the program preserves maximal entropy on high values.

This approach avoided the need to keep track of the whole history like in Gray's model (see Equation 1.11), but suffered unfortunately from similar problems as other history-free approaches such as Denning's (see Section 1.2.1.2) and could be at best applied to adversaries without memory.

Another limitation of this model was illustrated by Clark et al. [CHM07] who gave the example of a program which swaps h and l (both independent of each other and belonging to the same data set with uniformly distributed elements), using a (high) temporary variable `temp`:

```
temp:=l;
l:=h;
h:=temp;
```

Intuitively, we would expect to find some positive information flow from h to l because at the end of the computation, the sensitive value h is entirely revealed in l. However, the calculation of the information escape according to McIver and Morgan [MM03] gives:

$$H(h|l) - H(h'|l') \quad = \quad H(h|l) - H(l|h) \quad = \quad H(h) - H(l) \quad = \quad 0 \qquad (1.13)$$

Another interesting contribution of McIver and Morgan, still in [MM03], was their definition of *demonic nondeterminism*, an interesting approach to distinguish between probabilistic choices that could not be influenced (so-called *probabilistic nondeterminism*), and nondeterminism that could be resolved by a scheduler (the *demon*) and that could be seen as underspecification in distributed systems.

The authors integrated these notions into their security model expressed in the probabilistic guarded command language: probabilistic information was specified as usual by probability distributions, while demonic behaviour was described by subsets of possibilities. Programs were then described by functions from initial states to sets of distributions over final states, with the degree of nondeterminism as well as the probabilistic information recorded in the multiplicity of the result set. The authors investigated then the influence of restricting the power of the demon making the nondeterministic choices, such that it could see all data, only low data, or no data. Very recently, McIver, Meinicke and Morgan also considered probabilistic noninterference (without demonic choice) and considered the compositional closure of order relations based on the Bayes risk **??**.

Information Flow Cardinality Lowe proposed in the following year a definition of quantity of information flow using the process algebra timed CSP and based on the notion of *information flow cardinality* [Low04]. This quantity represents the number

n of behaviours of a high-level user that a low-level user can distinguish, and corresponds therefore to the number $\log n$ of bits of information that can be passed from the high-level to the low-level user.

Static analysis for quantifying information flow Recently, Clark, Hunt and Malacaria presented a static analysis for quantifying information flow [CHM07] that followed several previous publications [CHM02, CHM05a] and clarified different metrics of security used by other authors such as Millen [Mil87] and Gray [III91]. They provided lower and upper bounds on the amount of information flow, expressed as

$$I(L^{out}; H^{in}|L^{in})$$

i.e., the mutual information flow from high inputs to low outputs, given that the adversary had control over the low inputs. As described in Paragraph 1.2.1.2, they also identified a flaw in Denning's approach and corrected it by introducing the notion of *memory* of previous values. They then showed that their approach coincided with this modified definition. This work was recently reconsidered by Malacaria [Mal07] who added a definition of security of looping constructs.

Additionally, the authors of [CHM07] mentioned the independent work of Di Pierro, Hankin and Wiklicky who measured interference and derived a quantitative measure of the similarity between agents written in a probabilistic concurrent constraint language [DPHW02]. However, as opposed to [CHM07], no measure of information was provided in their work. Very important in my thesis will also be the recent work of Smith [Smi09] who introduced a new foundation based on a concept of vulnerability, measuring uncertainty by applying Renyis min-entropy rather than Shannon entropy. We defer a detailed discussion on this approach to Chapter 5 in this thesis.

An important point in Clark et al.'s model was that neither the lower nor the upper bound on the amount of information flow depended on the input distribution and therefore the latter was an upper bound on the channel capacity of the program, i.e., it was secure against the *worst-case* attack. In the next section, we will however see that it may be of interest to weaken the security required in a system in order to better fit the expected attacker model and thus to ensure an acceptable security level while using less resources and reducing the complexity of the system.

1.2.1.3 Consideration of the attacker model

The dependence on the input distribution is a fundamental issue in the definition of security protocols. Ideally, the degree of protection guaranteed by a protocol should be high enough to protect against the expected attacker model without raising the complexity of the system higher than what can be handled effectively. This is all the more relevant in today's cryptosystems such as RSA which rely on computational-theoretic (also known as cryptographic) security rather than information-theoretic security, i.e., the security only holds because decryption is intractable with current technologies. In real systems, a trade-off between security and complexity must therefore be met, which depends on the specific application of the protocol under concern. In this section, we will review some of the approaches which have been developed to take into account the attacker model in the security protocol.

Entropy measures Köpf and Basin expressed an attacker's remaining uncertainty about a secret as a function of the number of side-channel measurements made [KB07].

Several information-theoretic entropy measures were proposed to quantify the remaining uncertainty of the attacker. One of the motivations of the authors was to address a wide range of systems with different models of attackers, thus requiring different types of entropy. As discussed in [Mas94, Pli00, Cac97], the different notions are partially incomparable. Still, Köpf and Basin gave interesting intuitive meanings of the different definitions in terms of guesses of the attacker. The first measure $H(X)$, *Shannon Entropy* [Sha48], corresponded to a lower bound for the average number of binary questions that needed to be asked to determine the value of the random variable X. In case the attacker had already some knowledge, the *conditional entropy* $H(X|Y)$ was used, which expressed the remaining uncertainty of the attacker with prior knowledge Y. The second measure, the *Guessing Entropy* $G(X)$ [Mas94] corresponded to the average number of questions of the kind "does $X = x$ hold?" that had to be asked to guess the value of X. A generalization of this notion gave the third measure *Marginal Guesswork* $W_\alpha(X)$ [Pli00] which, for a fixed $\alpha \in [0, 1]$, quantified the number of questions of the kind "does $X = x$ hold?" that had to be asked until the value of X was correctly determined with a chance of success given by α. Conditional entropies could also be defined for guessing entropy and marginal guesswork. While lower bounds for $G(X)$ could be given in terms of $H(X)$, there was no general upper bound for $G(X)$ in terms of $H(X)$ and it was proven that no general inequality related Shannon entropy with marginal guesswork [Pli00].

Belief-based approach Recently, Clarkson et al. defined as information leakage the difference between the a priori accuracy of the guess of the attacker, and the a posteriori one, after the attacker had made his observation [CMS08]. The accuracy of the guess was defined as the Kullback-Leibler distance between the *belief* (which was a weight attributed by the attacker to each input hypothesis) and the true distribution on the hypotheses. The reliability of this probabilistic belief-based approach (compared to e.g., a worst-case strategy) is however hard to evaluate, as different attackers may have different beliefs.

Absolute leakage and rate of leakage A recent effort in the specification of the attacker model was subsequently given by Boreale who studied the quantitative models of information leakage in process calculi using pi-calculus, and who defined two different quantitative notions of information leakage differing essentially in the assumptions made on the power of the attacker [Bor09].

First, an attacker with full control over the process was assumed, who knew the program code and could conduct any number of tries over it. Moreover, the attacker could know the probability distribution of the inputs (given as random variable X) as well as other "side information" publicly available and modelled as random variable Z. This corresponded therefore to a worst-case of attacker, i.e., an attacker with unlimited computational resources which allowed to define security guarantees independent from the computational power of the actual attacker. The leakage of interest was in this case the *absolute leakage*, i.e., the average amount of information that was leaked to the attacker by the program under these assumptions. As expected, and following the earlier results of Millen [Mil87] and Gray [III91], absolute leakage coincided with the conditional mutual information $I = H(X|Y) - H(X|Y, Z)$, where $Z = P(X, Y)$ represented the outputs, i.e., the "observational behaviour" of the protocol. As described previously, Clark et al. also considered conditional mutual information as measure of leakage [CHM07] but the main difference here was the con-

sideration of concurrent programs by Boreale rather than sequential (and necessarily terminating) computations.

A more realistic situation was then considered, in which the resources of the attacker were limited and a notion of *cost* was introduced. More precisely, an attacker could only perform a predefined number n of tries over the protocol, each yielding a binary answer representing success of failure. The leakage of interest in this case was the *leakage rate*, i.e., the maximal number of bits of information that could be obtained per experiment over the protocol.

Boreale also studied the relation between these notions of leakage and proved that they were consistent, i.e the absolute leakage coincided with the maximum amount of information about the inputs that could be extracted by repeated experiments on the protocol, and that the cost the adversary had to pay to achieve it was at least the absolute leakage divided by the rate of leakage.

Interestingly, Boreale related his notions of leakage with the probability of error and the guesswork of the adversary. The relations between absolute leakage and these notions could be easily derived from well-known inequalities involving Shannon's conditional entropy (from which mutual information could be deduced): Fano's inequality [CT06] for a lower bound on the probability of error of the attacker and Massey's [Mas94] and Jensen's [CT06] inequalities for a lower bound on conditional guesswork.

The author also considered compositionality of leakage, and proved that for both notions a global system could not have a greater leakage (rate) than its individual subsystems, with the exception of parallel composition in the case of leakage rate.

Other previous work on the rate of leakage The notion of rate of leakage had already been introduced in previous works, albeit not that explicitly. Volpano and Smith considered the problem of trying to guess the k-bit value of a secret s using well-typed programs written in a deterministic programming language with $match$ queries [VS00]. Under this formalism, the authors proved that no well-typed program running in time bounded by a polynomial in k could deduce s. Furthermore, if a probability distribution could be specified for s, then choosing the uniform distribution made the probability that a well-typed polynomial-time program could deduce s goes to zero as k increased.

The notion of process similarity developed by Di Pierro, Wiklicky and Hankin [DPHW02] and already mentioned previously replaced the traditional notion of (absolute) indistinguishability of processes [RS99] by a quantitative measure of their behavioural difference. Therefore, two behaviours though distinguishable, could still be considered as effectively non-interfering as long as their difference was below a specified threshold. This gave a notion of distance between behaviours which was connected to the number of tries necessary to distinguish them and thus came close to Boreale's notion of rate of leakage.

1.2.2 Anonymity

1.2.2.1 Definition and Protocols

In my thesis, I will particularly focus on information flow in the context of *anonymity* protocols. Anonymity means that the identity of the user performing a certain action is maintained secret. It is an *information-hiding* property different from *confidentiality* (also known as *secrecy*) which consists in keeping secret the content of a message,

and from *privacy*, which is more general and deals with the protection of personal data. One can roughly say that "confidentiality deals with data, while privacy deals with people", and anonymity can be seen as a specific part of privacy.

Halpern and O'Neill describe in more details the distinction between anonymity and other information-hiding properties [HO03, HO05]. Several different formal definitions and protocols for anonymity have been defined in the past. They are mainly based on process-calculus [SS96, RS01], epistemic logic [SS99, HO05] or "function views" [HS04].

In this thesis we will focus on the process calculus approach, which has already widely been used in the field of security [AG99, AL00, Low97, Ros95b, Sch96]. In the following, we will briefly review existing work on possibilistic (i.e., nondeterministic) and probabilistic approaches to anonymity, before motivating our choice to use both formalisms together and giving an overview of research related to this "combined" approach.

1.2.2.2 Possibilistic versus probabilistic approaches

Possibilistic approach The possibilistic (i.e., purely nondeterministic) approach [SS96, RS01] is based on the so-called *principle of confusion*: a system is anonymous if the set of the possible outcomes is saturated with respect to the intended anonymous users. This means that for any observable trace produced by an anonymous user during the computation, there must exist for each other anonymous user an alternative computation which produces the same trace (modulo the identity of the anonymous users). In this approach, a distinction is made between total lack of anonymity and "some" anonymity, but all protocols that provide anonymity to some extent, from the least to the maximal degree, are considered equivalent. However, this is insufficient in most security systems in which it is desirable to be able to measure more precisely "how much" anonymity is preserved. In other words, we are not only interested in knowing whether events are possible or impossible (possibilistic approach), but rather in determining what is their likelihood to occur.

Probabilistic approach Probabilistic definitions of anonymity have already been investigated by several authors ([Cha88, HO03, BP05, RR98, CP05]), who distinguished different strengths of anonymity, which will be described later in this section.

Several probabilistic anonymity protocols have also been developed in the past, such as the Dining Cryptographers [Cha88], Crowds [RR98], Onion Routing [SGR97] and Freenet [CSWH01]. We will focus here on Crowds and the Dining Cryptographer, which will be used as running example through this thesis.

Crowds Crowds was presented by Reiter and Rubin as an anonymity protocol for web transactions [RR98]. It involves a public network represented as a set of nodes (the crowd) which may send messages to each other. The security goal consists in ensuring that any sender remains anonymous, even to the receiver of the message. In order to achieve this goal, a message sent over the network is forwarded randomly until it reaches the receiver.

More precisely, a node which wants to send a message chooses randomly (with uniform probability) a node (possibly himself) in the crowd and sends the message to him. Upon reception of the message, the following node tosses a biased coin and forwards the message if heads is obtained. Otherwise, the message is sent directly to the final receiver.

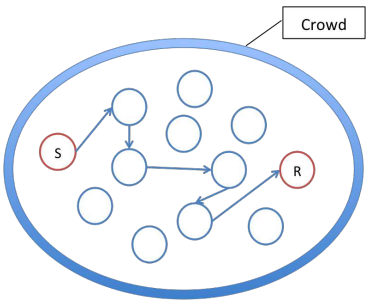

Figure 1.11: The Crowds protocol (S stands for the sender and R for the receiver of a message)

In this scenario, the receiver of a message can never determine who is the initial sender of the message, who remains therefore anonymous. Even if a node may know who was the previous node in the path, he cannot determine whether this node initiated or just forwarded the message.

This protocol is illustrated on Figure 1.11. As we will see later, the security goal defined by Reiter and Rubin for Crowds is *probable innocence*, which states that a node appears equally likely to have initiated the message as not to have. The protocol was extended to the case in which some of the nodes are corrupted, i.e., they can collaborate in order to reveal the identity of the initiator of the message.

Dining Cryptographers In the Dining Cryptographers (DC) protocol, proposed by Chaum, and illustrated in Figure 1.12, three cryptographers and a master are dining together and they agree that only one of the four, secretly chosen by the master, will have to pay the bill [Cha88]. At the end of the dinner, the master secretly tells to each cryptographer whether he has to be the payer or not. The cryptographers would like then to find out whether the payer is the master or if it is one of them, but without discovering which cryptographer is the payer in this latter case (i.e., they want the payer to remain anonymous if it is one of the cryptographers).

Chaum gave a solution to this problem: each cryptographer tosses a coin and the result is visible only to him and his right neighbour. Each cryptographer announces then ”agree” if the two coins he can see (his own coin and his left neighbour's) are both head or both tail, and ”disagree” otherwise. However, if one cryptographer is the payer, he lies and says the opposite.

It can be proven that if the number of ”disagree” is even, then the master is paying. Otherwise, one of the cryptographer is the payer, but his identity is unknown. This result is easy to understand, since if the master is paying (i.e., all cryprographers tell the truth), there will be either zero ”disagree” if the three coins give the same result, or two ”disagree” if one of the coin differ from the two others. If one cryptographer is paying (and thus lies), this will add one to the previous sum, which gives indeed an odd number of ”disagree” in this case.

Combining nondeterminism and probabilities In this thesis we will follow the framework described by Bhargava and Palamidessi in [BP05, Pal06, BP09]. While previous formal definitions of anonymity were either nondeterministic or purely prob-

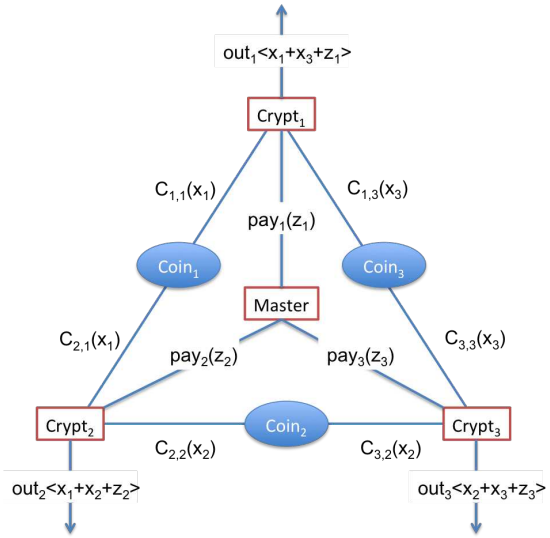

Figure 1.12: The Dining Cryptographers protocol

abilistic, the authors of those papers consider the most general situation in which the users have a nondeterministic behaviour while the protocol follows a probability distribution.

The choice of nondeterminism to characterize the users' behaviour is motivated by the fact that usually nothing is known about the relative frequency by which each user performs the anonymous actions. Moreover, in several systems, this behaviour may change and be irregular over time. This is particularly relevant in distributed and concurrent systems where there may not be a global notion of time. In this case, the sequence of actions performed by an agent (e.g. a, b, c) may be seen as a different sequence (e.g. a, c, b, b, a, c) by other agents, depending on their relative space-time locations in the system. Nondeterminism may provide a convenient formalization in such contexts.

It has long been discussed whether nondeterminism could be assimilated to a probabilistic behaviour where the probability distribution is unknown, but the general agreement today is that nondeterminism does not follow a probability law whatsoever.

On the other hand, we are interested in anonymity protocols (such as Crowds or the Dining Cryptographers) which rely on random mechanisms to add noise and obfuscate the identity of the users. These strategies can usually be described by a probability law, hence the choice of a probabilistic model for the anonymity protocol.

This approach clearly separates the considerations of the protocol from the assumptions on the users, which models effectively the situation occuring in most real systems, where we want to have guarantees on the security of the anonymity protocol whoever the users may be.

Probabilistic Automata An appropriate formal description of this model requires therefore the capability to express both nondeterminism and probability. Many models proposed in the literature combine these two approaches, and one of the most

general is the formalism of the *probabilistic automata* proposed by Segala and Lynch [SL95]. In [BP05] the notion of anonymity is formulated in terms of observables for processes in the probabilistic π-calculus, whose semantics is based on probabilistic automata. A function called *scheduler* is used to resolve nondeterminism, and the problem of the choice of the scheduler will be addressed later in this thesis.

1.2.2.3 Degrees of anonymity

Anonymity Metrics Several anonymity metrics have already been proposed in the past, which can mainly be classified as metrics based on the anonymity set, information theoretic entropy metrics and probability-based metrics.

Anonymity set metrics The simplest metrics rely on the *anonymity set* as defined by Chaum for the Dining Cryptographer [Cha88], i.e., the size of the set of users that may have sent a message through the system. The larger the set of potential initiators of the message, the stronger the anonymity of the sender. Therefore, the degree of anonymity was defined directly as the size n of the anonymity set, or as $\log(n)$. Unfortunately, these metrics only work when the adversary considers the a priori probability distribution of the possible senders to be uniform, an assumption we will try to relieve in this thesis.

Information theoretic entropy metrics An information theoretic entropy metric was then proposed by Danezis and Serjantov in 2002 [SD02a] in order to overcome the limitations of the previous approach and quantify anonymity in case of an arbitrary distribution of potential senders. Here, the uncertainty of an adversary is quantified in terms of Shannon entropy $H(S)$ where S is the probability distribution on the communication participants regarding which one is the sender in a communication. Here, we have $0 \leq H(S) \leq \log n$, where $H(S) = 0$ corresponds to knowledge of the sender (i.e., $p_i = 1$ if the sender is the user i) and $H(S) = \log n$ represents strong anonymity (i.e., S is the uniform distribution).

Diaz et al. [DSCP02a] proposed a normalization of this metric by defining the degree of anonymity as $H(S)/\log n$ so that it varies between 0 and 1.

The main drawback of these metrics is their explicit reliance on the knowledge of an adversary, which may complicate the comparison of different real-life anonymity systems. Diaz and Sassaman [DSD04] addressed this issue by statistical analysis, i.e., by calculating for a large volume of traffic data gathered in two anonymity systems the maximum and minimum observed entropies over a long time period.

Probability-based metrics: the anonymity hierarchy The large flexibility given by probabilities compared to possibilistic approaches, already mentioned previously, was also used by several authors to define different degrees of anonymity, thus scaling the gap between the least and maximal anonymity that occurs with a possibilistic approach. In the next paragraph, we will see how this approach was used to define the so-called *anonymity hierarchy*.

The Anonymity Hierarchy Reiter and Rubin were the first in [RR98] to give a hierarchy of anonymity degrees, which, as they explain, adds a third aspect to the two other classical properties of an anonymity system, namely the anonymous communication model and the attacker model, both defined in the eighties by Pfitzmann and

Waidner [PW87]. An anonymous communication model may be characterized either by *sender anonymity*, where the identity of the sender is hidden while the receiver and the message may not be, or by *receiver anonymity* where the identity of the receiver is hidden, or by *unlinkability of sender and receiver*, where only the communication between sender and receiver is hidden but possibly not their identities. On the other hand, the attacker can be either modelled as an eavesdropper that may observe messages exchanged in the system, or as collaborations of some senders, receivers, and other parties, or variations of these.

In order to complement these two properties, Reiter and Rubin defined a continuum of degrees of beliefs, ranging from *absolute secrecy* (no observable effect for the attacker) to *provably exposed* (the attacker can identify the sender and prove his identity to other parties). In between these two extrema and in decreasing degree of anonymity, the following three intermediary notions are defined:

- *Beyond suspicion*: From the attackers point of view, the sender appears no more likely to be the originator of the message than any other potential sender in the system.
- *Probable innocence*: From the attackers point of view, the sender appears no more likely to be the originator of the message than to not be the originator.
- *Possible innocence*: From the attackers point of view, there is a nontrivial probability that the real sender is someone else.

Initially, these levels were tailored to Crowds (i.e., they depended in [RR98] on symmetries inherent to Crowds that do not necessarily occur in a more general system) and it was shown that the degree of anonymity provided by the protocol for the sender is *probable innocence*.

Strong anonymity Reiter and Rubin's hierarchy was a first step towards a real quantification of anonymity [RR98]. However, as mentioned previously, it was restricted to systems satisfying the assumptions in Crowds. Following work in this field aimed therefore at generalizing the anonymity hierarchy, starting from the definition of *strong anonymity*.

This notion had already been defined by Chaum who proved that his Dining Cryptographer protocol satisfies *strong anonymity*, under the assumption that the coins are fair [Cha88]. Intuitively, strong anonymity occurs when the observables (here the answers of the cryptographers) do not give additional knowledge to the observer on the secret information (here the identity of the payer). This notion of anonymity describes the ideal situation in which the protocol does not leak any information concerning the identity of the user. In [Cha88], it was formulated as the condition $p(a|o) = p(a)$, where a is a secret action and o is an observable.

In the subsequent research, there have been basically two points of view to express strong anonymity. The first one corresponds to Chaum's definition [Cha88] and expresses "probabilistic noninterference", i.e., the fact that the attacker does not learn anything about an anonymous action from the execution of the protocol, or in other words that the a priori probability $p(a)$ of an anonymous action a is equal to its a posteriori probability $p(a|o)$ after the observation of o. In [CP05] it was observed that this condition is equivalent to the equality of the *likelihoods* of all anonymous events:

$$\forall a, a', o,\ p(o|a) \quad = \quad p(o|a') \tag{1.14}$$

The second point of view, adopted by Halpern and O'Neill [HO03, HO05] considers the lack of confidence of the attacker and requires that after an observation o, all the a posteriori probabilities of the anonymous events $p(a|o)$ (where a is an anonymous action) be equal, so that the attacker cannot have confidence in one hypothesis more than in another. Formally, this means:

$$\forall a, a', o,\ p(a|o) \quad = \quad p(a'|o) \tag{1.15}$$

The notion of strong anonymity was called *conditional anonymity* by Halpern and O'Neill in [HO03]. It is also equivalent to the anonymity level called *beyond suspicion* in Reiter and Rubin's hierarchy [RR98].

It corresponds to the situation in which all anonymous events produce the same observables events with the same probability, which prevents the attacker from deducing anything concerning the anonymous events. This definition was adopted by Chatzikokolakis and Palamidessi, because it meets the security goal without relying on the distribution of the anonymous events [CP05].

Chatzikokolakis also studied compositionality of strong anonymity and proved that if S_1 and S_2 are two anonymity systems, then the composition $S_1; S_2$ is an anonymity system which satisfies strong anonymity if and only if S_1 and S_2 satisfy both strong anonymity [Cha07].

Probable Innocence Besides the notion of strong anonymity, a weaker notion is needed to characterize systems in which some leakage is allowed and which could correspond to the (informal) notion of "probable innocence" in Reiter and Rubin's hierarchy, defined as the situation in which the probability that the initiator sends the message to an attacker is at most 1/2. Halpern and O'Neill reinterpreted more formally this notion which reflects the quantification of an observer's uncertainty [HO05], and thus comes close to the recent work of Clarkson et al. previously mentioned with their notion of attacker's *belief* [CMS08]. A definition of anonymity in the process algebra CSP is also given in [HO05], as well as definitions of information hiding using function views.

In [CP05], a generalized notion of probable innocence is proposed, which combines both approaches, i.e., expresses a limit both on the attackers confidence and on the probability of detection. Moreover, it relaxes the two assumptions that were the main drawbacks of the former definitions: it does not depend on the probability distribution of the anonymous events as Halpern and O'Neill's approach [HO05] and holds for systems without the symmetries present in Crowds [RR98]. On the other hand, it still reduces to the definition in [HO05] when the anonymous events have a uniform distribution and to the definition in [RR98] when the system is given the same symmetries as in Crowds, as expected from a correct modelization.

Notice that [CP05] contains a proof showing that as expected, the generalized probable innocence remains indeed weaker than strong anonymity (i.e., the latter implies the former).

1.2.2.4 Anonymity Protocols as Noisy Channel

Information-Theoretic Approach to Anonymity The work on information flow decribed in Paragraph 1.2.1 made wide use of the representation of security protocols as noisy channels where noninterference is formalized as the converse of channel capacity. Similarly, there have been various attempts to define the anonymity degree in terms of entropy and mutual information [SD02b, DSCP02b, ZB05, DPW07].

Moscowitz et al. related channel capacity to anonymity, by proposing a method where non-perfect anonymity could be used to create covert communication [MNCM03, MNS03].

Chatzikokolakis, Palamidessi and Panangaden defined the notion of *conditional capacity*, a generalization of Shannon capacity very useful for protocols in which some loss of anonymity is permitted [CPP08a]. This occurs for instance in an election protocol in which the number or votes for each candidate should be known, but without revealing the individual choice of each voter.

The authors of [CPP08a] also considered the problem of the effective computation of the channel capacity. This is not a trivial issue since there is no analytical formula available for this calculation and numerical algorithms such as the Arimoto-Blahut algorithm [CT06] can only converge asymptotically to the capacity. It was shown in [CPP08a] that symmetries of the channel, which are very common in real systems may be exploited to compute the capacity more easily. A simple operation which involves only one row of the matrix and which can be computed in time linear with the number of observables is provided to calculate the capacity of a channel matrix with symmetries.

Hypothesis Testing and Bayes Risk In [CPP08b], the properties of the channel matrix (e.g., its symmetries) were also related to the inferences that could be made by an attacker about the anonymous events from the channel matrix (i.e., the likelihood probabilities) and the observables. The amount of anonymous information that can be obtained by the attacker with this so-called *hypothesis testing* strategy is captured by the *probability of error*, i.e., the probability that the attacker makes a wrong guess. Typically, the attacker will follow the Bayesian method and apply the MAP (Maximum Aposteriori Probability) criterion which, as the name says, dictates that one should choose the hypothesis with the maximum aposteriori probability for the given observation, and which is provably the best strategy for the attacker. "Best" means that this strategy induces the smallest probability of error in the guess of the hypothesis. The probability of error, in this case, is also called *Bayes risk*. In [CPP08a], the authors proposed to define the *degree of protection* provided by a protocol as the Bayes risk associated to the matrix and we will pursue in this direction later on in this thesis.

Recently, Smith used a notion closely related to the Bayes risk and called *vulnerability* to argue that Renyi's *min-entropy* was a better measure of uncertainty than the traditionally used Shannon entropy when the attacker attempts to guess correctly the secret in one try [Smi07, Smi09]. He shows an example of two programs in which the mutual information is about the same, but the probability of making the right guess, after having observed the output, is much higher in one program than in the other. In this case, Renyi's min-entropy allows to clearly distinguish between the two situations, which justifies Smith's approach.

1.2.3 Belief Logics

We conclude this section with a brief review of the literature on belief logics, which will be helpful to understand our motivation for developing in this thesis a new modal logic with error control to express dynamic belief of agents in security systems, and to formalize security properties such as probabilistic anonymity and oblivious transfer.

The literature on belief logics is large due to their wide applicability in philosophical logic, artificial intelligence, and information security. However, all belief logics

we are aware of and which will be presented below are either only about belief without error control, or *static* belief with error control. In situations involving uncertainties or subjective and potentially changing quantities, error control, i.e., the specifications of lower and/or upper bounds for a variable quantity, is particularly important, as it allows to restrict the potential error within a precise margin and thus to turn an imprecise quantity into a precise variability domain of this same quantity. In the context of security, error control is particularly useful to assess with more accuracy the power of an attacker and thus to determine the degree of security enforced in the system.

Static belief with error control is introduced in [HO05] in the form of a functional symbol (term constructor) $\Pr_i(\phi)$ to be used in atomic formulas $\Pr_i(\phi) \leq \alpha$ that are true in a certain state by definition if and only if the probability according to agent i that ϕ is true is at most α in that state. The probability value results from a probability measure applied to the set of all those states that are indistinguishable from the current state to agent i and where ϕ is true. The authors then obtain a formalization of probabilistic anonymity for the dining cryptographers that mixes static knowledge (as a modality) and static belief. The logic is static, i.e., it does not have a temporal fragment. A fortiori, the belief in the logic is static (not possibly evolving). Also, the authors do not explicitly account for the possible presence of a scheduler.

In [HP05], the authors introduce what they call *randomized*, *explicit* (or *algorithmic*) belief. The intuition is that a randomized knowledge algorithm returning "Yes" to a query about a fact ϕ provides evidence for ϕ being true. The algorithm always returns either "Yes", "No", or "Don't know", and the return value "Yes" may depend on the outcome of coin tosses. The authors' motivation for the algorithmic modeling of belief is the resource-boundedness of real agents, which are thus identified with algorithms. The authors define measurable upper and lower *weights of evidence* assigned to hypotheses given observations. Such a weight is not a probability measure, but rather "a prescription for how to update a prior probability on the hypotheses into a posterior probability on those hypotheses, after having considered the observations made".

A formalization of non-probabilistic anonymity for the dining cryptographers is presented in [LP07], expressed in a modal logic combining knowledge and time. Hence, their notion of knowledge is dynamic, yet not enhanced with probability: it really is knowledge, which is necessarily true, and not belief, which possibly is false.

In [DMO07], the authors present a formalization of non-probabilistic anonymity for the dining cryptographers expressed in the μ-calculus with knowledge. Hence, the same comments apply as for [LP07]. Additionally, their logic is, as ours, closely tied to a process calculus.

Internalized probability in our logic is based on the construct $[\phi]_p$ introduced in [PS07] to represent probabilistic statements. The operator $[\phi]_p$ is true whenever the probability of the states that satisfy the formula ϕ is at least p. A different probabilistic extension of Hennessy-Milner logic is the one of [LS91, DEP98], where they consider a probabilistic variant $\langle a \rangle_p$ of the modal operator $\langle a \rangle$. Intuitively, $\langle a \rangle_p \phi$ means that a process can perform an a-transition and go *with probability at least* p to a state that satisfies ϕ. As showed in [PS07], the operator $[\phi]_p$ is more expressive, because $\langle a \rangle_p \phi$ can be represented as $\langle a \rangle [\phi]_p$. Furthermore, Parma and Segala have shown that the operator $[\phi]_p$ is necessary for characterizing (probabilistic) bisimulation in systems that allow both probabilistic and non-deterministic branching from the same state, which turns out to be the case for CCS_p.

1.3 Contribution

In Chapter 3, we focus on protocols for information-hiding which typically use randomized primitives to obfuscate the link between the observables and the information to be protected. The degree of protection provided by such a protocol can be expressed in terms of the probability of error associated to the inference of the secret information. The best approximation of this value is achieved by applying the so-called Maximum Aposteriori Probability (MAP) rule which requires the input distribution to be known. This assumption is however often too strong for the applications we consider. Therefore, we distinguish in this chapter two different cases: the scenario in which the input distribution is known, in which case we consider the Bayes risk as probability of error, and the one in which we have no information on the input distribution, or it changes over time. In this second scenario, we consider as degree of protection the probability of error associated to the Maximum Likelihood rule, averaged on all possible input distributions. It turns out that such average is equal to the value of the probability of error on the point of uniform distribution, which is much easier to compute.

In Section 3.2, we consider a probabilistic process algebra called CCS$_p$for the specication of information-hiding protocols, and we investigate which constructs in the language can be used safely in the sense that by applying them to a term, the degree of protection provided by the term does not decrease. This provides a criterion to build specications in a compositional way, while preserving the degree of protection. We do this study for both the Bayesian and the Maximum Likelihood (ML) approaches.

We then apply in Section 3.6 these compositional methods to the example of the Dining Cryptographers, and we are able to strengthen the strong anonymity result by Chaum. Namely we show that we can have strong anonymity even if some coins are unfair, provided that there is a spanning tree of fair ones. This result is obtained by adding processes representing coins to the specication and using the fact that this can be done with a safe construct.

In Chapter 4, we introduce in Section 4.2 a novel modal logic, namely the *doxastic μ-calculus with error control* (DμCEC), and propose a formalization of *probabilistic anonymity* and *oblivious transfer* in the logic, and the validation of these formalizations on implementations formalized in probabilistic CCS. The distinguishing feature of our logic is to provide a combination of *dynamic* operators for belief (whence the attribute "doxastic") and for internalized probability, with a *control* on the possible error of apprehension of the perceived reality. As described in Section 1.2.3, existing works in this field are, to the best of our knowledge, restricted to (dynamic) belief without error control, or to static belief with error control. We show some application examples of our logic to the dining cryptographers [Cha88] (Section 4.3), and to oblivious transfer [Rab81] for single bits and entire strings (Section 4.4). In both cases, we specify the protocol in CCS$_p$. Dynamicity is useful for the logical formalization of the original intuition of probabilistic anonymity and oblivious transfer, which is an invariant with respect to *a priori* and *a posteriori* stances of apprehension of the perceived reality (cf. Sections 4.3 and 4.4). The new operators we define allow to clearly distinguish between the subjective notion of *confidence*, i.e., the qualification of an agent's belief (that something is the case) and the objective notion of *certainty*, i.e., a qualification of something being the case. In our framework, both qualifications are also quantitative thanks to the mentioned error control in terms of a lower and upper probability bound.

Finally, we consider the notion of information leakage, or vulnerability of the system, which has been related in some approaches to the concept of mutual informa-

tion of the channel. A recent work of Smith has shown, however, that if the attack consists in one single try, then the mutual information and other concepts based on Shannon entropy are not suitable, and he has proposed to use Rényi's min-entropy instead [Smi09]. In Chapter 5, we consider and compare two different possibilities of defining the leakage, both based on the Bayes risk, which was already defined previously and happens to be closely related to Rényi min-entropy.

We propose to formalize the notion of leakage as the "difference" between the probability of error *a priori* (before observing the output) and *a posteriori* (using the output to infer the input via the aforementioned MAP rule). We argue that there are at least two natural ways of defining this difference: one, that we call *multiplicative* in Paragraph 5.3.3, corresponds to Smith's proposal. The other, which we present in Paragraph 5.3.4 and call *additive*, is new. In both cases, we show that it is relatively easy to find the suprema, which is nice in that it allows us to consider the worst case of leakage. The worst case is also interesting because it abstracts from the input distribution, which, as previously mentioned, is usually unknown, or (in the case of anonymity) may depend on the set of users. In Section 5.6 we compare both measures of leakage before discussing and illustrating our results.

1.4 Publications

The main results of this thesis have previously been the subject of several scientific publications. Chapter 3 is based on the article *Compositional methods for information-hiding* published in the proceedings of FOSSACS 2008 [BCP08]. The logical approach developed in Chapter 4 was described in the article *A quantitative doxastic logic for probabilistic processes and applications to information-hiding* that has been published in the Journal of Applied Non-Classical Logics in 2009 [KPS+10]. Finally, the results in Chapter 5 appeared in the article *Quantitative notions of leakage for one-try attacks* published in the proceedings of the MFPS 25 Conference [BCP09].

Chapter 2

Preliminaries

In this chapter, we first give a brief introduction to probability spaces and probabilistic automata. Then, we present the probabilistic process algebra CCS_p, an extension of standard CCS ([Mil89]) obtained by adding probabilistic choice. In the remaining, we give some insights on notions from information theory such as Rényi entropies, Shannon entropy, and mutual information. Finally, convexity and corner points are reviewed to complete this overview over the preliminary notions necessary to understand the content of this thesis.

2.1 Probability spaces

Let Ω designate a set. A *σ-field* (also *σ-algebra*) over Ω is a collection $\mathcal{F}$ of subsets of Ω closed under complement and countable union and such that $\Omega \in \mathcal{F}$. If $\mathcal{B}$ is a collection of subsets of Ω then *the σ-field generated by $\mathcal{B}$* is defined as the smallest σ-field containing $\mathcal{B}$ (its existence is ensured by the fact that the intersection of an arbitrary set of σ-fields containing $\mathcal{B}$ is still a σ-field containing $\mathcal{B}$). A *probability measure* on $\mathcal{F}$ is a function $\mu\colon \mathcal{F} \to [0, \infty]$ such that

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

We denote by $supp(\mu) \stackrel{\text{def}}{=} \{x \in \Omega \mid \mu(\{x\}) > 0\}$ the *support set* of μ. A *probability space* is a tuple $(\Omega, \mathcal{F}, \mu)$ where Ω is a set, called the *sample space*, $\mathcal{F}$ is a σ-field on Ω and μ is a probability measure on $\mathcal{F}$. The elements of a σ-field $\mathcal{F}$ are also called *events*. For $x \in \Omega$, we denote by $\delta(x)$ (called the *Dirac measure* on x) the probability measure on $\mathcal{F}$ such that $\delta(x)(\{y\}) = 1$ if $y = x$, and $\delta(x)(\{y\}) = 0$ otherwise. If $c_1, \ldots, c_n$ are convex coefficients (namely $c_i \geq 0$ for all i and $\sum_i c_i = 1$), and $\mu_1, \ldots, \mu_n$ are probability measures, we denote by $\sum_i c_i \mu_i$ the probability measure defined as $(\sum_i c_i \mu_i)(A) \stackrel{\text{def}}{=} \sum_i c_i \mu_i(A)$. If A, B are events then $A \cap B$ is also an event. If $\mu(A) > 0$ then we can define the *conditional probability* $p(B \mid A)$, meaning "the probability of B given A", as

$$p(B \mid A) \stackrel{\text{def}}{=} \frac{\mu(A \cap B)}{\mu(A)}$$

Note that $p(\,\cdot \mid A)$ is a new probability measure on $\mathcal{F}$. In continuous probability spaces, where many events have zero probability, it is possible to generalize the concept of conditional probability to allow conditioning on such events. However, this is not necessary for the purpose of this thesis. Thus we will use the above "traditional" definition of conditional probability and make sure that we never condition on events of zero probability. A probability space and the corresponding probability measure are called *discrete* if Ω is countable and $\mathcal{F} = 2^{\Omega}$. In this case, we can construct μ from a function $p\colon \Omega \to [0,1]$ satisfying $\sum_{x\in\Omega} p(x) = 1$ by assigning $\mu(\{x\}) = p(x)$. The set of all discrete probability measures with sample space Ω will be denoted by $Disc(\Omega)$.

2.2 Probabilistic Automata

In this section we introduce the probabilistic automata of [SL95, Seg95] following a notation that is similar to the one used in [Seg06].

A *probabilistic automaton* M is a tuple $(St, s_{init}, Act, \mathcal{T})$ where St is a set of states, $s_{init} \in St$ is the *initial state*, Act is a set of actions and $\mathcal{T} \subseteq St \times Act \times Disc(St)$ is a *transition relation*. Intuitively, if $(s, a, \mu) \in \mathcal{T}$ then there is a transition from the state s performing the action a and leading to a distribution μ over the states of the automaton. The idea is that the choice of transition among the available ones in $\mathcal{T}$ is performed non-deterministically, and the choice of the target state among the ones allowed by μ (i.e. those states s' such that $\mu(s') > 0$) is performed probabilistically. Note that in general from a state there can be two transitions with the same action leading to two different distributions. A probabilistic automaton M is *fully probabilistic* if from each state of M there is at most one transition available.

An *execution fragment* h of a probabilistic automaton is a (possibly infinite) alternating sequence $s_0 a_1 s_1 a_2 s_2 \ldots$ of states and actions, such that for each i there is a transition $(s_i, a_{i+1}, \mu_i) \in \mathcal{T}$ and $\mu_i(s_{i+1}) > 0$. The concatenation of a finite execution fragment $h_1 = s_0 \ldots a_n s_n$ and an execution fragment $h_2 = s_n a_{n+1} s_{n+1} \ldots$ is the execution fragment $h_1 \cdot h_2 = s_0 \ldots a_n s_n a_{n+1} s_{n+1} \ldots$. A finite execution fragment h_1 is a prefix of h, written $h_1 \leq h$, if there is an execution fragment h_2 such that $h = h_1 \cdot h_2$. We use $fst(h), lst(h)$ to denote the first and last state of a finite execution fragment h respectively.

An *execution* (or *history*) h is an execution fragment such that $fst(h) = s_{init}$. An execution h is maximal if it is infinite or there is no transition from $lst(h)$ in $\mathcal{T}$. We denote by $exec^*(M)$, $exec^{\perp}(M)$, and $exec(M)$ the set of all the finite, all the non maximal, and all the executions of the probabilistic automaton M, respectively.

A *scheduler* for a probabilistic automaton $M = (St, s_{init}, Act, \mathcal{T})$ is a total function

$$\zeta\colon exec^{\perp}(M) \to \mathcal{T}$$

such that $\zeta(h) = (s, a, \mu) \in \mathcal{T}$ implies that $s = lst(h)$. The role of the scheduler is to resolve nondeterminism: when we are in state s the scheduler selects a transition among the ones available in $\mathcal{T}$ for s, and it can base its decision on the history of the execution that has led to s.

The above definition actually corresponds to a restricted class of schedulers called the Dirac non-halting schedulers. These schedulers choose a transition each time one is available (while in general a scheduler may choose to stop even if a transition is available). This restriction will allow us to simplify the definition of the probability measures induced by the scheduler, since it reduces the sample space. Besides this

constraint, we also impose that a scheduler does not use randomization in resolving nondeterminism, while in general a scheduler may be randomized.

.

The *execution tree* of M under the scheduler ζ, denoted by $etree(M, \zeta)$, is a fully probabilistic automaton $M' = (St', s_{init}, Act, \mathcal{T}')$ such that $St' \subseteq exec^*(M)$, and $(h, a, \mu') \in \mathcal{T}'$ if and only if $\zeta(h) = (lst(h), a, \mu)$ for some μ, and $\mu'(has) = \mu(s)$. Intuitively, $etree(M, \zeta)$ is produced by unfolding the executions of M and resolving all non-deterministic choices using ζ.

Given a probabilistic automaton $M = (St, s_{init}, Act, \mathcal{T})$ and a scheduler ζ we can define the probability space $(\Omega_M, \mathcal{F}_M, p_M)$ on the maximal executions of M induced by ζ as follows:

- $\Omega_M \stackrel{\text{def}}{=} exec(M) \setminus exec^{\perp}(M)$ (the set of all the maximal executions of M).
- Given a finite execution h, the cone with prefix h is defined as $C_h \stackrel{\text{def}}{=} \{h' \in \Omega_M | h \leq h'\}$. Define $\mathcal{F}$ as the σ-field generated by the set of all cones of M.
- Define the probability of a cone C_h, where $h = s_0 a_1 s_1 \dots a_n s_n$, as

$$p(C_h) \stackrel{\text{def}}{=} \prod_{i=1}^{n} \mu_i(s_i)$$

  where, for each i, $\zeta(s_0 a_1 s_1 \dots a_{i-1} s_{i-1}) = (s_{i-1}, a_i, \mu_i)$. We define $p_{M,\zeta}$ as the measure extending p to $\mathcal{F}$ (see [Seg95] for more details).

Remark 2.2.1. *The σ-field used in [SL95] considers the sample space $\Omega = exec(M)$ to account for the termination at non-maximal executions. Since here we require that the schedulers are total, the support of the measure $p_{M,\zeta}$ needs not to include elements of $exec^{\perp}(M)$. Note that the σ-field defined in this chapter coincides with the sub-σ-field not containing $exec^{\perp}(M)$ of the standard σ-field on probabilistic-automata induced by total schedulers.*

Convention Given a probabilistic automaton M and a scheduler ζ, we will denote $p_{M,\zeta}$ by p_ζ whenever M is clear from the context.

2.3 CCS with probabilistic choice

In this section, we consider an extension of standard CCS ([Mil89]) obtained by adding probabilistic choice. The resulting calculus CCS$_p$ can be seen as a simplified version of the probabilistic π-calculus presented in [HP00, PH05] and is similar to the one considered in [DPP05, CP07b]. As in those calculi, computations have both a probabilistic and a nondeterministic nature.

We consider a finite set A of actions and a set $\bar{A}$ of *complementary* actions such that, for each $a \in A$ there is a complementary action $\bar{a} \in \bar{A}$ with $\bar{\bar{a}} = a$. The whole set of actions $Act = A \cup \bar{A} \cup \{\tau\}$ corresponds to the union of A, $\bar{A}$ and the action $\tau \notin A \cup \bar{A}$ which represents the *invisible action* and does not have a complementary action. Usually the number of elements in A is assumed to be at most countable, so it can be in general either finite or infinite. We will restrict this condition to the finite case when necessary in the remainder of the thesis.

$$\text{PROB}\ \frac{}{\sum\!\!\!\!\!\!\bigcirc_i\ p_i\, T_i \xrightarrow{\tau} \sum_i p_i\, \delta(T_i)} \qquad \text{ACT}\ \frac{j \in I}{\biguplus_I a_i.T_i \xrightarrow{a_j} \delta(T_j)}$$

$$\text{PAR1}\ \frac{T_1 \xrightarrow{a} \mu}{T_1 \mid T_2 \xrightarrow{a} \mu \mid T_2} \qquad \text{PAR2}\ \frac{T_2 \xrightarrow{a} \mu}{T_1 \mid T_2 \xrightarrow{a} T_1 \mid \mu}$$

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

$$\text{COM}\ \frac{T_1 \xrightarrow{a} \delta(T_1') \quad T_2 \xrightarrow{\overline{a}} \delta(T_2')}{T_1 \mid T_2 \xrightarrow{\tau} \delta(T_1' \mid T_2')} \qquad \text{RES}\ \frac{T \xrightarrow{b} \mu \qquad b \neq a, \overline{a}}{(\nu a)T \xrightarrow{b} (\nu a)\mu}$$

Table 2.1: The semantics of CCS_p.

2.3.1 Syntax

T	::=		*process term*
		$\sum\!\!\!\!\!\!\bigcirc_{i\in I}\ p_i\, T_i$	*probabilistic choice* $(\sum_{i\in I} p_i = 1)$
	|	$\biguplus_{i\in I}\ a_i.T_i$	*nondeterministic choice* $(\forall i, a_i \in Act)$
	|	$T \mid T$	*parallel composition*
	|	$(\nu a)\, T$	*restriction* $(a \in Act)$
	|	$!\, T$	*replication*

All the summations in the syntax are finite and the set I is a finite set of indices. The nil process is implicitely specified by a nondeterministic choice where the set of indices is empty. We will use the notation $T_1 \oplus_p T_2$ to represent a binary probabilistic choice $\sum\!\!\!\!\!\!\bigcirc_i\ p_i\, T_i$ with $p_1 = p$ and $p_2 = 1 - p$. Similarly we will use $a_1.T_1 \uplus a_2.T_2$ to represent a binary nondeterministic choice.

2.3.2 Semantics

The semantics of a given CCS_p term is a probabilistic automaton whose states are process terms, whose initial state is the given term, and whose transitions are those derivable from the rules in Table 2.1. We will use the notations (T, a, μ) and $T \xrightarrow{a} \mu$ interchangeably. We denote by $\mu \mid T$ the measure μ' such that $\mu'(T' \mid T) = \mu(T')$ for all processes T' and $\mu'(T'') = 0$ if T'' is not of the form $T' \mid T$, and similarly for $T \mid \mu$. Furthermore we denote by $(\nu a)\mu$ the measure μ' such that $\mu'((\nu a)T) = \mu(T)$, and $\mu'(T') = 0$ if T' is not of the form $(\nu a)T$.

We explain now briefly the meaning of the rules:

- ACT represents the execution of the action a_j in $\biguplus_I a_i.T_i$.

- PAR1 (resp. PAR2) represent the fact that in $T_1 \mid T_2$, the process T_1 (resp. T_2) can execute a step while T_2 (resp. T_1) stays idle (interleaving).
- COM represents a communication step between T_1 and T_2 in $T_1 \mid T_2$, which can take place when T_1 and T_2 are ready to perform complementary actions.
- RES filters out the transitions with label a or $\bar{a}$ from a process restricted on a.
- REP1 (resp. REP2) express the fact that $!T$ can spawn one (resp. two) copies of T and let these copies perform a step.
- PROB models internal probabilistic choice: a silent τ transition is available from the sum $\sum_i p_i T_i$ to a measure composed of the sum of the Dirac functions of its operands (the $\delta(T_i)$'s) weighted by the corresponding probabilities p_i. Note that a term T_i in the probabilistic sum $\sum_i p_i T_i$ generates exactly one (probabilistic) transition $T_i \xrightarrow{\tau} \delta(T_i)$.

Thus, all the rules of CCS$_p$ specialize to the ones of CCS except for PROB.These rules allow the occurence of nondeterminism in the execution path of an automaton, by involving nondeterministic choices or parallel processes. As explained in Section 2.2, schedulers will be used to resolve such nondeterminism.

Note that in the produced probabilistic automaton, all transitions to non-Dirac measures are silent. In other words, any non-silent transition performed by the automaton leads to a Dirac measure, i.e. is of the form $T \xrightarrow{a} \delta(T')$, which corresponds to a transition of a non-probabilistic automaton (i.e. a standard labeled transition system where, given the current process and the next action, only one transition is possible).

It is interesting to observe that the resulting automaton is consistent with the definition of alternating automaton of [PS07]: a probabilistic automaton is *alternating* if the states that enable a non-Dirac transition enable only one transition. We call *probabilistic* those states that enable non-Dirac transitions, and *nondeterministic* all the other states. In other words, a probabilistic state enables at most one transition while a nondeterministic state may enable several transitions with the constraint that the target measure of each of these transitions is a Dirac measure.

2.4 Rényi Entropies, Shannon Entropy, and Mutual Information

In this thesis, we will be interested in evaluating the amount of information that can be deduced about a random variable, given the knowledge of another random variable. We recall now the most important notions which were defined for this purpose.

Rényi entropies [Rén60] are a family of functions representing the uncertainty associated to a random variable. The Rényi entropy of order α, with $\alpha \geq 0$ and $\alpha \neq 1$, is defined as

$$H_\alpha(X) = \frac{1}{1-\alpha} \log \left(\sum_{i=1}^{n} p(x_i)^\alpha \right)$$

where X is a random variable ranging over the set $\{x_1, \ldots, x_n\}$ and $p(x_i)$ is the probability of x_i. In the case of a uniform distribution all the Rényi entropies of are equal to $\log n$. Otherwise the entropies are weakly decreasing as a function of α. The

following are some particular cases:

$$\begin{array}{lll} \alpha = 0 & H_0(X) = \log |X| = \log n & \text{Hartley entropy} \\ \alpha \to 1 & H_1(X) = -\sum_i p(x_i) \log p(x_i) & \text{Shannon entropy} \\ \alpha \to \infty & H_\infty(X) = -\log \max_i p(x_i) & \text{min-entropy} \end{array}$$

We will be particularly interested in Shannon entropy, initially introduced by Claude Shannon in 1948 [Sha48] (we will often write H for H_1 in the remaining). In particular, Shannon conditional entropy of X given Y represents the average residual entropy of X once the value of Y is known, and it is defined as

$$\begin{array}{rcl} H_1(X|Y) & = & \sum_y p(y) H_1(X|Y=y) \\ & = & -\sum_{ij} p(x_i, y_j) \log p(x_i|y_j) \\ & = & H_1(X,Y) - H_1(Y) \end{array}$$

where $H_1(X,Y)$ represents the entropy of the conjunction of X and Y.

The mutual information of X and Y represents the correlation of information between X and Y. It is defined as

$$I(X;Y) = H_1(X) - H_1(X|Y) = H_1(X) + H_1(Y) - H_1(X,Y)$$

It is possible to show that $I(X;Y) \geq 0$, with $I(X;Y) = 0$ iff X and Y are independent. For more details, we refer the reader to [CT06].

For min-entropy, the definition of $H_\infty(X|Y)$ has been subject to controversy in the literature: some authors, e.g., [Cac97], generalize the aforementioned definition of $H_1(X|Y)$ to min-entropy, which leads to:

$$\begin{array}{rcl} H_\infty(X|Y) & = & \sum_y p(y) H_\infty(X|Y=y) \\ & = & -\sum_j p(y_j) \log \max_i p(x_i|y_j) \end{array} \tag{2.1}$$

Other authors, e.g., [DORS08, Smi09], use the following definition:

$$H_\infty(X|Y) = -\log \sum_j \max_i (p(y_j|x_i) p(x_i)) \tag{2.2}$$

The motivation for this second definition will become clear in Chapter 5.

2.5 Convexity and Corner Points

Finally, we recall here some basic notions of convexity. Let $\mathbb{R}$ be the set of real numbers. The elements $\lambda_1, \lambda_2, \ldots, \lambda_k \in \mathbb{R}$ constitute a set of *convex coefficients* if, for every $i \in \{1, \ldots, k\}$, $\lambda_i \geq 0$ and $\sum_k \lambda_k = 1$. Given a vector space V, a *convex combination* of $\vec{v}_1, \vec{v}_2, \ldots, \vec{v}_k \in V$ is any vector of the form $\sum_i \lambda_i \vec{v}_i$ where $\lambda_1, \lambda_2, \ldots, \lambda_k \in \mathbb{R}$ are a set of convex coefficients. A subset S of a vector space is *convex* if every convex combination of vectors in S is in S.

In the following we will denote by $D^{(n)}$ the domain of probability distributions of dimension n. It is easy to see that, for every n, $D^{(n)}$ is convex.

Given a convex subset S of a vector space V, and a function $f : S \to \mathbb{R}$, we say that the function f is *convex* if for any $\vec{v}_1, \vec{v}_2, \ldots, \vec{v}_k \in S$ and any set of convex

coefficients $\lambda_1, \lambda_2, \ldots, \lambda_k \in \mathbb{R}$, we have $f(\sum_i \lambda_i \vec{v}_i) \leq \sum_i \lambda_i f(\vec{v}_i)$. A function f is *concave* if its opposite $-f$ is convex.

We now introduce (with a slight abuse of terminology) the concept of *convex base*.

Given a subset S of V, the *convex hull* of S, which we will denote by $ch(S)$, is the smallest convex set containing S. $ch(S)$ is the set of convex combinations of nonempty finite subsets of S. Since the intersection of convex sets is convex, it is clear that $ch(S)$ always exists.

Given two vector sets S and U, we say that U is a convex base for S if $U \subseteq S$ and $S \subseteq ch(U)$.

In the following, for a given vector $\vec{v} = (v_1, v_2, \ldots, v_n)$, we will use the notation $(\vec{v}, f(\vec{v}))$ to denote the vector (with one additional dimension) $(v_1, v_2, \ldots, v_n, f(\vec{v}))$. Similarly, given a vector set S in a n-dimensional space, we will use the notation $(S, f(S))$ to represent the set of vectors $\{(\vec{v}, f(\vec{v})) \mid \vec{v} \in S\}$ in an $(n+1)$-dimensional space. The notation $f(S)$ represents the image of S under f, i.e. $f(S) = \{f(\vec{v}) \mid \vec{v} \in S\}$.

Given a vector set S, a convex base U of S, and a function $f : S \to \mathbb{R}$, we say that U is a set of *corner points* of f if $(U, f(U))$ is a convex base for $(S, f(S))$. We also say that f is *convexly generated* by $f(U)$.

In other words, if U is a set of corner points of f, then for every $\vec{v} \in S$, there are elements $\vec{u}_1, \vec{u}_2, \ldots, \vec{u}_k$ in U and $\lambda_1, \lambda_2, \ldots, \lambda_k$ in $\mathbb{R}$ such that $\vec{v} = \sum_i \lambda_i \vec{u}_i$ and $f(\vec{v}) = \sum_i \lambda_i f(\vec{u}_i)$.

Chapter 3

The process calculus approach

3.1 Introduction

Recently it has been observed that at an abstract level information-hiding protocols can be viewed as *channels* in the information-theoretic sense. A channel consists of a set of input values $\mathcal{S}$, a set of output values $\mathcal{O}$ (the observables) and a transition matrix which gives the conditional probability $p(o|s)$ of producing o as the output when s is the input. In the case of privacy preserving protocols, $\mathcal{S}$ contains the secret information that we want to protect and $\mathcal{O}$ the facts that the attacker can observe. This framework allows us to apply concepts from information theory to reason about the knowledge that the attacker can gain about the input by observing the output of the protocol.

In the information-hiding systems we consider in this thesis, the attacker finds himself in the following scenario: he cannot directly detect the information of interest, namely the actual value of the random variable $S \in \mathcal{S}$, but he can discover the value of another random variable $O \in \mathcal{O}$ which depends on S according to a known conditional distribution. This kind of situation is quite common also in other disciplines, like medicine, biology, and experimental physics, to mention a few. The attempt to infer S from O is called *hypothesis testing* (the "hypothesis" to be validated is the actual value of S), and it has been widely investigated in statistics. One of the most used approaches to this problem is the Bayesian method, which consists in assuming that the *a priori* probability distribution of the hypotheses is known, and deriving from that (and from the matrix of the conditional probabilities) the a posteriori distribution after a certain fact has been observed. It is well known that the best strategy for the adversary is to apply the MAP (Maximum Aposteriori Probability) criterion, which, as the name says, dictates that one should choose the hypothesis with the maximum a posteriori probability for the given observation. "Best" means that this strategy induces the smallest probability of error in the guess of the hypothesis. The probability of error, in this case, is also called *Bayes risk*. In [CPP07], the *degree of protection* provided by a protocol was defined as the Bayes risk associated to the matrix.

A major problem with the Bayesian method is the requirement that the *a priori* distribution is known, which is by far not the common case in security applications. It may be sometimes possible to approximate the *a priori* distribution by statistical inference, but in most situations the input distribution is not known beforehand, and may actually change over time. Thus other methods need to be considered, which do not depend on the *a priori* distribution. Such a well-known alternative is the method

based on the so-called *Maximum Likelihood* (ML) criterion. Given an hypothesis s, the likelihood of an outcome o corresponds to the probability $p(o|s)$ and the ML criterion dictates to choose the highest such probability for this hypothesis.

For the large majority of the security protocols considered in this thesis, the ML rule will provide a convenient approximation to the MAP rule. We will see in this chapter that this approximation particularly holds when the protocol is repeated a very large number of times, since a sufficient number of repetitions allows one to effectively "factor out" the *a priori* probabilities in the calculation of the preferred hypothesis.

In the next section we present the variant of CCS$_p$ used in this chapter. Section 3.3 shows how to model protocols and process terms as channels. Section 3.4 discusses hypothesis testing and presents some properties of the probability of error. Section 3.5 characterizes the constructs of CCS$_p$ which are safe, in the sense that applying them do not decrease the security of the protocol. Finally Section 3.6 applies previous results to find a new property of the Dining Cryptographers.

3.2 CCS$_p$ with secret and observable actions

In this section, we use a variant of the calculus CCS$_p$ introduced in Section 2.3. We make here a distinction between *observable* and *secret* actions, introduced for the purpose of specifying information-hiding protocols. More precisely, we assume that the set of actions Act is partitioned into a set Sec of *secret actions* s, a set Obs of *observable actions* o, and the silent action τ, i.e., $Act = Sec \cup Obs \cup \{\tau\} = A \cup \bar{A} \cup \{\tau\}$.

Furthermore, for any secret $s \in Sec$ (resp. $o \in Obs$) we assume that the complementary action satisfies $\overline{s} \in Sec$ (resp. $\overline{o} \in Obs$).

The syntax corresponds to the one given in Paragraph 2.3.1, with two differences: all actions in the nondeterministic choice are observables, and an additional choice called *secret choice* is introduced, which only differs from nondeterministic choice by the fact that all actions are secrets. More precisely, the nondeterministic choice in the syntax given in Paragraph 2.3.1 is replaced by the two following choices:

$$\boxplus_i s_i.T_i \quad \textit{secret choice} \quad (s_i \in Sec)$$
$$\boxplus_i r_i.T_i \quad \textit{nondeterministic choice} \quad (r_i \in Obs \cup \{\tau\})$$

The semantics of CCS$_p$ described in Paragraph 2.3.2 is similar in this chapter.

The distinction between the two kind of labels (secrets and observables) influences the notion of scheduler for CCS$_p$: the secret actions are assumed to be *inputs* of the system, namely they can only be performed if the input matches them. Hence some choices are determined, or influenced, by the input. In particular, a secret choice with different guards is entirely decided by the input. The scheduler has only to resolve the residual nondeterminism which occurs in nondeterministic choices (where the labels are observables) and in parallel composition of processes.

In the following, we use the notation $X \rightharpoonup Y$ to represent the partial functions from X to Y, and $h_{|Sec}$ to represent the projection of a sequence of actions h on Sec. If for instance $h = s_1 s_2 o_1 s_3 o_2 o_3$ with $\forall i$, $s_i \in Sec$ and $o_i \in Obs$, then $h_{|Sec} = s_1 s_2 s_3$.

We now adapt the notion of scheduler defined in Section 2.2 to secret choices.

Definition 3.2.1. *Let T be a process in CCS$_p$ and M be the probabilistic automaton*

generated by T. A scheduler is a function

$$\zeta : Sec^* \to exec^*(M) \rightharpoonup \mathcal{T}$$

such that:

(i) if $s = s_1 s_2 \dots s_n$ and $h_{|Sec} = s_1 s_2 \dots s_m$ with $m \leq n$, and

(ii) there exists a transition $(lst(h), a, \mu)$ such that, if $a \in Sec$ then $a = s_{m+1}$

then $\zeta(s)(h)$ is defined, and it is one of such transitions. We say that the scheduler ζ is compatible *with the input s on h. We will write $\zeta_s(h)$ for $\zeta(s)(h)$.*

In other words, a scheduler can only determine the outcome of a secret choice if there exists a transition from the current state involving a secret action which matches the next secret in the input sequence. Moreover, we require that the scheduler always executes a transition if one is possible, which differs from the definition of scheduler used in probabilistic automaton, where the scheduler can decide to stop, even if a transition is allowed. This means that the schedulers we consider here always perform *maximal executions*, i.e., they only stop when no next transition is possible.

We now adapt the definition of *execution tree* from the notion found in probabilistic automata. In our case, the execution tree depends not only on the scheduler, but also on the input. Given an input s and a scheduler ζ, it corresponds to the fully probabilistic automaton which is obtained by removing from the execution tree $exec^*(M)$ of the initial automaton all transitions that are not chosen by ζ_s.

Definition 3.2.2. *Let $M = (St, T, Act, \mathcal{T})$ be the probabilistic automaton generated by a CCS$_p$ process T, where St is the set of processes reachable from T. Given an input s and a scheduler ζ, the* execution tree *of T for s and ζ, denoted by $etree(T, s, \zeta)$, is a fully probabilistic automaton $M' = (St', T, Act, \mathcal{T}')$ such that $St' \subseteq exec(M)$, and $(h, a, \mu') \in \mathcal{T}'$ if and only if $\zeta_s(h) = (lst(h), a, \mu)$ for some μ, and $\mu'(has) = \mu(s)$.*

3.3 Modeling protocols for information-hiding

In this section we propose an abstract model for information-hiding protocols, and we show how to represent this model in CCS$_p$.

3.3.1 Protocols as channels

As mentioned in the introduction of this chapter, we view protocols as *channels* in the information-theoretic sense [CT06]. The secret information that the protocol is trying to conceal constitutes the input of the channel, and the observables constitute the outputs. The set of the possible inputs and that of the possible outputs will be denoted by $\mathcal{S}$ and $\mathcal{O}$ respectively. We assume that $\mathcal{S}$ and $\mathcal{O}$ are of finite cardinality m and n respectively. We also assume a discrete probability distribution over the inputs, which we will denote by $\vec{\pi} = (\pi_{s_1}, \pi_{s_2}, \dots, \pi_{s_m})$, where π_s is the probability of the input s.

To fit the model of the channel, we assume that at each run, the protocol is given exactly one secret s_i to conceal. This is not a restriction, because the s_i's can be complex information like sequences of keys or tuples of individual data. During the run, the protocol may use randomized operations to increase the level of uncertainty

about the secrets and obfuscate the link with the observables. It may also have internal interactions between internal components, or other forms of nondeterministic behavior, but let us rule out this possibility for the moment, and consider a purely probabilistic protocol. We also assume there is exactly one output from each run of the protocol, and again, this is not a restrictive assumption because the elements of $\mathcal{O}$ can be structured data.

Given an input s, a run of the protocol will produce each $o \in \mathcal{O}$ with a certain probability $p(o|s)$ which depends on s and on the randomized operations performed by the protocol. Note that $p(o|s)$ depends only on the probability distributions on the mechanisms of the protocol, and not on the input distribution. The probabilities $p(o|s)$, for $s \in \mathcal{S}$ and $o \in \mathcal{O}$, constitute a $m \times n$ array M which is called the *matrix* of the channel, where the rows are indexed by the elements of $\mathcal{S}$ and the columns are indexed by the elements of $\mathcal{O}$. We will use the notation $(\mathcal{S}, \mathcal{O}, M)$ to represent the channel.

Note that the input distribution $\vec{\pi}$ and the probabilities $p(o|s)$ determine a distribution on the output. We will represent by $p(o)$ the probability of $o \in \mathcal{O}$. Thus both the input and the output can be considered *random variables*. We will denote these random variables by S and O.

If the protocol contains some forms of nondeterminism, like internal components giving rise to different interleaving and interactions, then the behavior of the protocol, and in particular the output, will depend on the scheduling policy. We can reduce this case to previous (purely probabilistic) scenario by assuming a scheduler ζ which resolves the nondeterminism entirely. Of course, the conditional probabilities, and therefore the matrix, will depend on ζ, too. We will express this dependency by using the notation M_ζ.

3.3.2 Process terms as channels

A given CCS$_p$ term T can be regarded as a protocol in which the input is constituted by sequences of secret actions, and the output by sequences of observable actions. We assume that only a finite set of such sequences is relevant. This is certainly true if the term is terminating, which is usually the case in security protocols, as each session is supposed to terminate in finite time.

Thus the set S could be, for example, the set of all sequences of secret actions up to a certain length (for example, the maximal length of executions) and analogously O could be the set of all sequences of observable actions up to a certain length. To be more general, we will just assume $\mathcal{S} \subseteq_{fin} Sec^*$ and $\mathcal{O} \subseteq_{fin} Obs^*$.

Definition 3.3.1. *Given a term T and a scheduler $\zeta : \mathcal{S} \to exec^*(M) \to \mathcal{T}$, the matrix $M_\zeta(T)$ associated to T under ζ is defined as the matrix such that, for each $s \in \mathcal{S}$ and $o \in \mathcal{O}$, $p(o|s)$ is the probability of the set of the maximal executions in $etree(T, s, \zeta)$ whose projection in Obs is o.*

The following remark may be useful to understand the nature of the above definition:

Remark 3.3.2. *Given a sequence $s = s_1 s_2 \dots s_h$, consider the term*

$$T' = (\nu Sec)(\bar{s}_1.\bar{s}_2.\dots.\bar{s}_h.0 \mid T)$$

Given a scheduler ζ for T, let ζ' be the scheduler on T' that chooses the transition

$$((\nu Sec)(\bar{s}_j.\bar{s}_{j+1}.\dots.\bar{s}_h.0 \mid U), r, (\nu Sec)(\bar{s}_j.\bar{s}_{j+1}.\dots.\bar{s}_h.0 \mid \mu))$$

if ζ_s chooses (U, r, μ), with $(r \notin Sec)$, and it chooses

$$((\nu Sec)(\bar{s}_j.\bar{s}_{j+1}.\ldots.\bar{s}_h.0 \mid U), \tau, (\nu Sec)(\delta(\bar{s}_{j+1}.\bar{s}_{j+2}.\ldots.\bar{s}_h.0 \mid U')))$$

if ζ_s chooses $(U, s_j, \delta(U'))$.

Note that ζ' is a "standard" scheduler, i.e., it does not depend on an input sequence.

We have that each element $p(o|s)$ in $M_\zeta(T)$ is equal to the probability of the set of all the maximal executions of T', under ζ', whose projection in Obs gives o.

3.4 Inferring the secrets from the observables

In this section we discuss possible methods by which an adversary can try to infer the secrets from the observables, and consider the corresponding probability of error, that is, the probability that the adversary draws the wrong conclusion. We regard the probability of error as a representative of the degree of protection provided by the protocol, and we study its properties with respect to the associated matrix.

We start by defining the notion of *decision function*, which represents the guess the adversary makes about the secrets, for each observable. This is a well-known concept, particularly in the field of *hypothesis testing*, where the purpose is to try to discover the valid hypothesis from the observed facts, knowing the probabilistic relation between the possible hypotheses and their consequences. In our scenario, the hypotheses are the secrets.

Definition 3.4.1. *A decision function for a channel $(\mathcal{S}, \mathcal{O}, M)$ is any function*

$$f : \mathcal{O} \to \mathcal{S}$$

Given a channel $(\mathcal{S}, \mathcal{O}, M)$, an input distribution $\vec{\pi}$, and a decision function f , the *probability of error* $\mathcal{P}(f, M, \vec{\pi})$ is the average probability of guessing the wrong hypothesis by using f, weighted on the probability of the observable (see for instance [CT06]). The probability that, given o, s is the wrong hypothesis is $1 - p(s|o)$ (with a slight abuse of notation, we use $p(\cdot|\cdot)$ to represent also the probability of the input given the output). Hence we have:

Definition 3.4.2 ([CT06])**.** *The probability of error is defined by*

$$\mathcal{P}(f, M, \vec{\pi}) \;=\; 1 - \sum_{\mathcal{O}} p(o)p(f(o)|o)$$

Given a channel $(\mathcal{S}, \mathcal{O}, M)$, the best decision function that the adversary can use, namely the one that minimizes the probability of error, is the one associated to the so-called MAP rule, which prescribes choosing the hypothesis s which has *Maximum Aposteriori Probability* (for a given $o \in \mathcal{O}$), namely the s for which $p(s|o)$ is maximum. The fact that the MAP rule represent the 'best bet' of the adversary is rather intuitive, and well known in the literature. We refer to [CT06] for a formal proof.

The MAP rule is used in the so-called *Bayesian approach* to hypothesis testing, and the corresponding probability of error is also known as *Bayes risk*. We will denote it by $\mathcal{P}_{MAP}(M, \vec{\pi})$. The following characterization is an immediate consequence of Definition 3.4.2 and of the Bayes theorem $p(s|o) = p(o|s)\pi_s/p(o)$.

$$\mathcal{P}_{MAP}(M,\vec{\pi}) \;=\; 1-\sum_{\mathcal{O}}\max_{s}(\pi_s p(o|s))$$

It is natural then to define the degree of protection associated to a process term as the infimum probability of error that we can obtain from this term under every scheduler (in a given class) which is compatible with the input on this term.

In the following, we assume the class of schedulers $\mathcal{A}$ to be the set of all the schedulers compatible with the given input $\mathcal{S}$ on the given term.

It turns out that the infimum probability of error on $\mathcal{A}$ is actually a minimum. In order to prove this fact, let us first define a suitable metric on $\mathcal{A}$.

Definition 3.4.3. *Consider a CCS$_p$ process T, and let M be the probabilistic automaton generated by T. We define a distance d between schedulers in $\mathcal{A}$ as follows:*

$$d(\zeta,\zeta') = \begin{cases} 2^{-m} & \text{if } m = \min\{|h| \mid h \in exec^*(M) \text{ and } \zeta(h) \neq \zeta'(h)\} \\ 0 & \text{if } \zeta(h) = \zeta'(h) \text{ for all } h \in exec^*(M) \end{cases}$$

where $|h|$ represents the length of h.

Note that M is finitely branching, both in the nondeterministic and in the probabilistic choices, in the sense that from every node T' there is only a finite number of transitions (T', a, μ) and μ is a finite summation of the form $\mu = \sum_i p_i\ \delta(T_i)$. Hence we have the following (standard) result:

Proposition 3.4.4. *$(\mathcal{A}, d)$ is a* sequentially compact *metric space, i.e., every sequence has a convergent subsequence (namely a subsequence with a limit in $\mathcal{A}$).*

We are now ready to show that there exists a scheduler that gives the minimum probability of error:

Proposition 3.4.5. *For every CCS$_p$ process T we have*

$$\inf_{\zeta\in\mathcal{A}}\ \mathcal{P}_{MAP}(M_\zeta(T),\vec{\pi}) \;=\; \min_{\zeta\in\mathcal{A}}\ \mathcal{P}_{MAP}(M_\zeta(T),\vec{\pi})$$

Proof. By Proposition 3.4.4, $(\mathcal{A}, d)$ is sequentially compact. Since the channel matrix is a continuous function of the distance on the schedulers (each pair of a secret and an observable, and the corresponding conditional probability is determined after a finite number of steps), and since $\mathcal{P}_{MAP}(M_\zeta(T),\vec{\pi})$ is a continuous function of the matrix, then $\mathcal{P}_{MAP}(M_\zeta(T),\vec{\pi})$ is a continuous function from $(\mathcal{A}, d)$ to $([0,1], d')$, where d' is the standard distance on real numbers. Consequently, $(\{\mathcal{P}_{MAP}(M_\zeta(T),\vec{\pi}) \mid \zeta \in \mathcal{A}\}, d')$ is also sequentially compact. Let $\{\zeta_n\}_n$ be a sequence such that for all n

$$\mathcal{P}_{MAP}(M_{\zeta_n}(T),\vec{\pi}) - \inf_{\mathcal{A}}\mathcal{P}_{MAP}(M_\zeta(T),\vec{\pi}) \leq 2^{-n}$$

We have that $\{\mathcal{P}_{MAP}(M_{\zeta_n}(T),\vec{\pi})\}_n$ is convergent and

$$\lim_n \mathcal{P}_{MAP}(M_{\zeta_n}(T),\vec{\pi}) \;=\; \inf_{\mathcal{A}}\mathcal{P}_{MAP}(M_\zeta(T),\vec{\pi})$$

Consider now a convergent subsequence $\{\zeta_{n_j}\}_j$ of $\{\zeta_n\}_n$. By continuity of $\mathcal{P}_{MAP}$, we have

$$\lim_n \mathcal{P}_{MAP}(M_{\zeta_n}(T),\vec{\pi}) \;=\; \lim_j \mathcal{P}_{MAP}(M_{\zeta_{n_j}}(T),\vec{\pi}) \;= \mathcal{P}_{MAP}(\lim_j M_{\zeta_{n_j}}(T),\vec{\pi})$$

which concludes the proof. □

Thanks to the previous proposition, we can define the degree of protection provided by a protocols in terms of the minimum probability of error.

Definition 3.4.6. *Given a CCS$_p$ process T, the protection $Pt_{MAP}(T)$ provided by T, in the Bayesian approach, is given by*

$$Pt_{MAP}(T, \vec{\pi}) \;=\; \min_{\zeta \in \mathcal{A}} \; \mathcal{P}_{MAP}(M_\zeta(T), \vec{\pi})$$

The problem with the MAP rule is that it assumes that the input distribution is known to the adversary. This is often not the case, so it is natural to try to approximate it with some other rule. One such rule is the so-called ML rule, which prescribes the choice of the s which has *Maximum Likelihood* (for a given $o \in \mathcal{O}$), namely the s for which $p(o|s)$ is maximum. The name comes from the fact that $p(o|s)$ is called the *likelihood* of s given o. We will denote the corresponding probability of error by $\mathcal{P}_{ML}(M, \vec{\pi})$. The following characterization is an immediate consequence of Definition 3.4.2 and of the Bayes theorem.

$$\mathcal{P}_{ML}(M, \vec{\pi}) \;=\; 1 - \sum_{\mathcal{O}} \pi_s \max_s(p(o|s))$$

Note that if the input distribution is the uniform distribution $\vec{\pi}_u = (\frac{1}{m}, \frac{1}{m}, \ldots, \frac{1}{m})$, then $\mathcal{P}_{MAP}$ and $\mathcal{P}_{ML}$ coincide:

$$\begin{array}{rcl} \mathcal{P}_{MAP}(M, \vec{\pi}_u) & = & 1 - \sum_{\mathcal{O}} \max_{s \in \mathcal{S}}(p(o|s)\pi_s) \\ & = & 1 - \sum_{\mathcal{O}} \max_{s \in \mathcal{S}}(p(o|s)\frac{1}{m}) \\ & = & 1 - \frac{1}{m} \sum_{\mathcal{O}} \max_{s \in \mathcal{S}}(p(o|s)) \\ & = & \mathcal{P}_{ML}(M, \vec{\pi}_u) \end{array} \tag{3.1}$$

It has been shown (see for instance [CPP08a]) that under certain conditions on the matrix, the ML rule approximates indeed the MAP rule, in the sense that by repeating the protocol the adversary can make the probability of error arbitrarily close to 0, with either rule.

We could now define the degree of protection provided by a term T under the ML rule as the minimum $\mathcal{P}_{ML}(M_\zeta(T), \vec{\pi})$, but it does not seem reasonable to give a definition that depends on the input distribution, since the main reason to apply a non-Bayesian approach is that indeed we do not know the input distribution. Instead, we define the degree of protection associated to a process term as the *average* probability of error with respect to all possible distributions $\vec{\pi}$:

Definition 3.4.7. *Given a CCS$_p$ process T, the protection $Pt_{ML}(T)$ provided by T, in the Maximum Likelihood approach, is given by*

$$Pt_{ML}(T) \;=\; \min_{\zeta \in \mathcal{A}} \; (m-1)! \int_{\vec{\pi}} \mathcal{P}_{ML}(M_\zeta(T), \vec{\pi}) \; d\vec{\pi}$$

In the above definition, $(m-1)!$ represents a normalization factor: $\frac{1}{(m-1)!}$ is namely the hyper-volume of the domain of all possible distributions $\vec{\pi}$ on $\mathcal{S}$, namely the $(m-1)$-dimensional space of points $\vec{\pi}$ (where m is the cardinality of $\mathcal{S}$) such that $\forall s, 0 \leq \pi_s \leq 1$ and $0 \leq \sum_{s \in S} \pi_s = 1$.

Fortunately, it turns out that this definition is equivalent to a much simpler one: the average value of the probability of error, under the Maximum Likelihood rule, can be obtained simply by computing $\mathcal{P}_{ML}$ on the uniform distribution $\vec{\pi}_u = (\frac{1}{m}, \frac{1}{m}, \ldots, \frac{1}{m})$.

Theorem 3.4.8. $Pt_{ML}(T) = \min_{\zeta \in \mathcal{A}} \mathcal{P}_{ML}(M_\zeta(T), \vec{\pi}_u)$

Proof. **Simplifications** First we note that the proof is trivial for $m = 1$, which corresponds to the situation in which there is only one secret. In this case an adversary always guesses correctly the secret and therefore the probability of error under the MAP rule and under the ML rule is zero. In the following we assume $m \geq 2$. Given a channel $(\mathcal{S}, \mathcal{O}, M)$ and an input distribution $\vec{\pi} = (\pi_1, \ldots, \pi_m)$ of cardinality m, the probability of error under the ML rule is characterized by the expression:

$$\mathcal{P}_{ML}(M, \vec{\pi}) = 1 - \sum_{\mathcal{O}} \pi_s \max_s(p(o|s)) = f_m(\vec{\pi})$$

where $f_m(\vec{\pi})$ is a function of the input distribution $\vec{\pi}$. Since the elements $\max_s(p(o|s))$ are coefficients of the channel matrix, $f_m(\vec{\pi})$ is of the form:

$$f_m(\vec{\pi}) = a_1\pi_1 + \ldots + a_m\pi_m$$

where the coefficients a_i are linear combinations of elements of the channel matrix. Therefore, $f_m(\vec{\pi})$ is linear.

With the additional constraint $\sum_{i=1\ldots m} \pi_i = 1$, the dependency on one of the m variables $\pi_1, \ldots, \pi_m$, for instance π_m, can be removed. Replacing π_m by the equivalent expression $1 - \sum_{i=1}^{m-1} \pi_i$ yields:

$$f_m(\vec{\pi}) = c_1\pi_1 + \ldots + c_{m-1}\pi_{m-1} + c_m$$

with

$$\begin{array}{rcl} c_1 & = & a_1 - a_m \\ c_2 & = & a_2 - a_m \\ \ldots & & \\ c_{m-1} & = & a_{m-1} - a_m \\ c_m & = & a_m \end{array}$$

Expression of the normalization factor The hyper-volume $V_m(X)$ of the domain $D_m(X)$ of all possible distributions $\vec{\pi}$ on $\mathcal{S}$ (where m is the cardinality of $\mathcal{S}$), i.e., the $(m-1)$-dimensional space of points $\vec{\pi}$ such that $\forall s, 0 \leq \pi_s \leq X$ and $0 \leq \sum_{s \in S} \pi_s = X$ is given by:

$$V_m(X) = \frac{X^{m-1}}{(m-1)!}$$

Induction hypothesis We will show by induction on m that following equality $\mathcal{H}_m$ holds for all $m \geq 2$:

$$\int_{D_m(X)} f_m(\vec{\pi}) d\vec{\pi} = V_m(X) f_m(\vec{\pi}_u(X)) \qquad (\mathcal{H}_m) \tag{3.2}$$

where $\vec{\pi}_u(X) = (\frac{X}{m}, \frac{X}{m}, \ldots, \frac{X}{m})$. Theorem 3.4.8 then follows by taking $X = 1$.

According to the aforementioned notations, $\mathcal{H}_m$ can be written as:

$$L_m(X) \;=\; R_m(X)$$

where

$$L_m(X) \;=\; \int_{x_{m-1}=0}^{X} \int_{x_{m-2}=0}^{X-x_{m-1}} \cdots \int_{x_1=0}^{X-x_{m-1}-\ldots-x_2} f_m(x_1,\ldots,x_{m-1})dx_1\ldots dx_{m-1}$$

and

$$R_m(X) \;=\; \frac{X^{m-1}}{(m-1)!}\Big(\sum_{i=1}^{m-1} c_i\frac{X}{m} + c_m\Big)$$

Base case: $m = 2$
We have:

$$\begin{aligned} L_2(X) &= \textstyle\int_{x_1=0}^{x_1=X} (c_1x_1 + c_2)dx_1 \\ &= \tfrac{c_1X^2}{2} + c_2X \\ &= X(\tfrac{c_1X}{2} + c_2) \\ &= R_2(X) \end{aligned}$$

Thus $\mathcal{H}_2$ holds.

Induction step: $\mathcal{H}_m \Rightarrow \mathcal{H}_{m+1}$
Consider

$$\begin{aligned} f_{m+1}(x) &= c_1x_1 + \ldots + c_mx_m + c_{m+1} \\ &= \textstyle\sum_{i=1}^{m} c_ix_i + c_{m+1} \\ &= f_m(x) - c_m + c_mx_m + c_{m+1} \end{aligned}$$

The left-hand side of $\mathcal{H}_{m+1}$ is given by:

$$L_{m+1}(Y) \;=\; \textstyle\int_{x_m=0}^{x_m=Y} \ldots \int_{x_1=0}^{x_1=Y-x_m-\ldots-x_2} f_{m+1}(x_1,\ldots,x_m)dx_1\ldots dx_m$$

The $m-1$ inner-most integrations can be resolved according to $\mathcal{H}_m$ (replacing X by $Y - x_m$) which leads to:

$$\begin{aligned} L_{m+1}(Y) &= \textstyle\int_{x_m=0}^{x_m=Y} V_m(Y-x_m) f_{m+1}(\frac{Y-x_m}{m},\ldots,\frac{Y-x_m}{m})dx_m \\ &= \textstyle\int_{x_m=0}^{x_m=Y} \frac{(Y-x_m)^{m-1}}{(m-1)!}(\sum_{i=1}^{m-1} c_i\frac{Y-x_m}{m} + c_mx_m + c_{m+1})dx_m \end{aligned}$$

Replacing $Y - x_m$ by Z leads to:

$$\begin{aligned}
L_{m+1}(Y) &= \int_{Z=0}^{Z=Y} \tfrac{Z^{m-1}}{(m-1)!}((\textstyle\sum_{i=1}^{m-1} c_i)\tfrac{Z}{m} + c_m(Y-Z) \\
&\quad +c_{m+1})dZ \\
&= \int_{Z=0}^{Z=Y} ((\tfrac{1}{m!}(\textstyle\sum_{i=1}^{m-1} c_i) - \tfrac{c_m}{(m-1)!})Z^m \\
&\quad +(\tfrac{c_m Y + c_{m+1}}{(m-1)!})Z^{m-1})dZ \\
&= (\tfrac{1}{m!}(\textstyle\sum_{i=1}^{m-1} c_i) - \tfrac{c_m}{(m-1)!})\tfrac{Y^{m+1}}{m+1} + (\tfrac{c_m Y + c_{m+1}}{(m-1)!})\tfrac{Y^m}{m} \\
&= \tfrac{(\sum_{i=1}^{m-1} c_i)+c_m}{(m+1)!}Y^{m+1} + \tfrac{c_{m+1}}{m!}Y^m \\
&= \tfrac{Y^m}{m!}(\textstyle\sum_{i=1}^{m} c_i \tfrac{Y}{m+1} + c_{m+1}) = R_{m+1}(Y)
\end{aligned}$$

Thus $\mathcal{H}_{m+1}$ holds.

This completes the proof for Theorem 3.4.8. □

The next corollary follows immediately from Theorem 3.4.8 and from the definitions of $\mathcal{P}_{MAP}$ and $\mathcal{P}_{ML}$ which imply that $\mathcal{P}_{ML}(M, \vec{\pi}_u) = \mathcal{P}_{MAP}(M, \vec{\pi}_u)$ (see Equation 3.1):

Corollary 3.4.9. $Pt_{ML}(T) = \min_{\zeta \in \mathcal{A}} \mathcal{P}_{MAP}(M_\zeta(T), \vec{\pi}_u)$

We conclude this section with some properties of $\mathcal{P}_{MAP}$, which will hold also for $\mathcal{P}_{ML}$ on the uniform distribution, because of Equation 3.1.

The next proposition shows that the probabilities of error are *concave* functions with respect to the space of matrices.

Proposition 3.4.10. *Consider a family of channels* $\{(\mathcal{S}, \mathcal{O}, M_i)\}_{i \in I}$, *and a family* $\{c_i\}_{i \in I}$ *of convex coefficients, namely* $0 \leq c_i \leq 1$ *for all* $i \in I$, *and* $\sum_{i \in I} c_i = 1$. *Then:*

$$\mathcal{P}_{MAP}(\sum_{i \in I} c_i M_i, \vec{\pi}) \geq \sum_{i \in I} c_i \mathcal{P}_{MAP}(M_i, \vec{\pi})$$

Proof. Consider $\forall i \in I, M_i = (p_i(o|s))_{s \in \mathcal{S}, o \in \mathcal{O}}$. Then:

$$\begin{aligned}
\mathcal{P}_{MAP}(\textstyle\sum_i c_i M_i, \vec{\pi}) &= 1 - \sum_o \max_s (\sum_i c_i\, p_i(o|s)\pi_s) & \\
&\geq 1 - \sum_o \sum_i c_i \max_s (p_i(o|s)\pi_s) & \text{(convexity of max)} \\
&= 1 - \sum_i \sum_o c_i \max_s (p_i(o|s)\pi_s) & \text{(positive summands)} \\
&= 1 - \sum_i c_i \sum_o \max_s (p_i(o|s)\pi_s) & \\
&= \sum_i c_i - \sum_i c_i \sum_o \max_s (p_i(o|s)\pi_s) & \text{(since } \textstyle\sum_{i \in I} c_i = 1) \\
&= \sum_i c_i (1 - \sum_o \max_s (p_i(o|s)\pi_s & \\
&= \sum_i c_i \mathcal{P}_{MAP}(M_i, \vec{\pi}) &
\end{aligned}$$

□

Corollary 3.4.11. *Consider a family of channels* $\{(\mathcal{S}, \mathcal{O}, M_i)\}_{i\in I}$*, and a family* $\{c_i\}_{i\in I}$ *of convex coefficients. Then:*

$$\mathcal{P}_{MAP}(\textstyle\sum_{i\in I} c_i\, M_i, \vec{\pi}) \;\geq\; \min_{i\in I}\; \mathcal{P}_{MAP}(M_i, \vec{\pi})$$

Proof.

$$\begin{array}{rcl} \mathcal{P}_{MAP}(\sum_{i\in I} c_i\, M_i, \vec{\pi}) & \geq & \sum_{i\in I} c_i\, \mathcal{P}_{MAP}(M_i, \vec{\pi}) \\ & \geq & \sum_{i\in I} c_i\, \min_{k\in I}\; \mathcal{P}_{MAP}(M_k, \vec{\pi}) \\ & = & \min_{i\in I}\; \mathcal{P}_{MAP}(M_i, \vec{\pi}) \end{array}$$

□

The next proposition shows that if we transform the observables, and collapse the columns corresponding to observables which have become the same after the transformation, the probability of error does not decrease.

Proposition 3.4.12 (Collapsing observables lowers leakage)**.** *Consider a channel* $(\mathcal{S}, \mathcal{O}, M)$*, where* M *has conditional probabilities* $p(o|s)$*, and a transformation of the observables* $f : \mathcal{O} \to \mathcal{O}'$*. Let* M' *be the matrix whose conditional probabilities are* $p'(o'|s) = \sum_{f(o)=o'} p(o|s)$ *and consider the new channel* $(\mathcal{S}, \mathcal{O}', M')$*. Then:*

$$\mathcal{P}_{MAP}(M', \vec{\pi}) \;\geq\; \mathcal{P}_{MAP}(M, \vec{\pi})$$

Proof. The result derives from:

$$\begin{array}{rcl} \sum_{o'\in\mathcal{O}'} \max_s(p'(o'|s)\pi_s) & = & \sum_{o'\in\mathcal{O}'} \max_s(\sum_{f(o)=o'} p(o|s)\pi_s) \\ & \leq & \sum_{o'\in\mathcal{O}'} \sum_{f(o)=o'} \max_s(p(o|s)\pi_s) \\ & = & \sum_{o\in\mathcal{O}} \max_s(p(o|s)\pi_s) \end{array}$$

□

The following propositions are from the literature.

Proposition 3.4.13 (Probabilistic noninterference minimizes leakage [CPP08a])**.** *Given* $\mathcal{S}$*,* $\mathcal{O}$*, let* M *be a matrix indexed on* $\mathcal{S}$*,* $\mathcal{O}$ *such that all the rows of* M *are equal, namely* $p(o|s) = p(o|s')$ *for all* $o \in \mathcal{O}$*,* $s, s' \in \mathcal{S}$*. Then,*

$$\mathcal{P}_{MAP}(M, \vec{\pi}) \;=\; 1 - \max_s\; \pi_s$$

Furthermore $\mathcal{P}_{MAP}(M, \vec{\pi})$ *is the maximum probability of error, i.e., for every other matrix* M' *indexed on* $\mathcal{S}$*,* $\mathcal{O}$ *we have:*

$$\mathcal{P}_{MAP}(M, \vec{\pi}) \;\geq\; \mathcal{P}_{MAP}(M', \vec{\pi})$$

Proposition 3.4.14 ([BP05])**.** *Given a channel* $(\mathcal{S}, \mathcal{O}, M)$*, the rows of* M *are equal (and hence the probability of error is maximum) if and only if* $p(s|o) = \pi_s$ *for all* $s \in \mathcal{S}$*,* $o \in \mathcal{O}$*.*

The condition $p(s|o) = \pi_s$ means that the observation does not give any additional information concerning the hypothesis. In other words, the *a posteriori* probability of s coincides with its *a priori* probability. The property $p(s|o) = \pi_s$ for all $s \in \mathcal{S}$ and $o \in \mathcal{O}$ was used as a definition of (strong) anonymity by Chaum [Cha88] and was called *conditional anonymity* by Halpern and O'Neill [HO05].

3.5 Safe constructs

In this section we investigate constructs of the language CCS$_p$ which are *safe* with respect to the protection of the secrets.

We start by giving some conditions that will allow us to ensure the safety of the parallel and the restriction operators.

Definition 3.5.1. *Consider a process term T, and the observables $o_1, o_2, \ldots, o_k$ such that*

(i) T does not contain any secret action, and

(ii) the observable actions of T are included in $o_1, o_2, \ldots, o_k$.

Then we say that T is safe for $Obs \setminus \{o_1, o_2, \ldots, o_k\}$.

The following theorem states our main results for Pt_{MAP}. Note that they are also valid for Pt_{ML}, because $Pt_{ML}(T) = Pt_{MAP}(T, \vec{\pi}_u)$.

Theorem 3.5.2. *The probabilistic choice, the nondeterministic choice, and a restricted form of parallel composition are safe constructs, namely, for every input probability π, and any terms $T_1, T_2, \ldots, T_h$, we have*

(1) $$Pt_{MAP}(\bigoplus_i p_i\, T_i, \vec{\pi}) \;\geq\; \sum_i p_i\, Pt_{MAP}(T_i, \vec{\pi}) \;\geq\; \min_i\, Pt_{MAP}(T_i, \vec{\pi})$$

(2) $$Pt_{MAP}(\Big\lfloor\!+\!\Big\rfloor_i o_i.T_i, \vec{\pi}) \;=\; \min_i Pt_{MAP}(T_i, \vec{\pi})$$

(3) $$Pt_{MAP}((\nu o_1, o_2, \ldots, o_k)\,(T_1 \mid T_2), \vec{\pi}) \;\geq\; Pt_{MAP}(T_2, \vec{\pi})$$
if T_1 is safe for $Obs \setminus \{o_1, o_2, \ldots, o_k\}$.

Proof. 1. By definition $Pt_{MAP}(\bigoplus_i p_i\, T_i, \vec{\pi}) = \min_{\zeta \in \mathcal{A}}\, \mathcal{P}_{MAP}(M_\zeta(\bigoplus_i p_i\, T_i), \vec{\pi})$. Let $\zeta_m = \text{minarg}_{\mathcal{A}} \mathcal{P}_{MAP}(M_\zeta(\bigoplus_i p_i\, T_i), \vec{\pi})$. The scheduler ζ_m corresponds to the worst case with respect to the protection of security, i.e., it always chooses the execution path which minimizes the probability of error. By definition of ζ_m, we have:

$$Pt_{MAP}(\bigoplus_i p_i\, T_i, \vec{\pi}) \;=\; \mathcal{P}_{MAP}(M_{\zeta_m}(\bigoplus_i p_i\, T_i), \vec{\pi})$$

Consider, for each i, the scheduler ζ_{m_i} defined as ζ_m on the i-th branch, except for the removal of the first state and the first τ-step (i.e., the probabilistic choice) from the execution fragments in the domain. Since ζ_m has no influence on the first transition corresponding to the probabilistic choice, it is easy to see that

$$M_{\zeta_m}(\bigoplus_i p_i\, T_i) = \sum_i p_i M_{\zeta_{m_i}}(T_i)$$

Thus we have:

$$\begin{aligned} \mathcal{P}_{MAP}(M_{\zeta_m}(\textstyle\bigoplus_i p_i\, T_i), \vec{\pi}) &= \mathcal{P}_{MAP}(\textstyle\sum_i p_i M_{\zeta_{m_i}}(T_i), \vec{\pi}) \\ &\geq \textstyle\sum_i p_i \mathcal{P}_{MAP}(M_{\zeta_{m_i}}(T_i), \vec{\pi}) \quad \text{(Prop. 3.4.10)} \end{aligned}$$

Finally, observe that if ζ_m is compatible with the input $\mathcal{S}$ on T, then ζ_{m_i} is compatible with $\mathcal{S}$ on T_i, i.e., all transitions labeled by a secret action that were allowed with ζ_m are still allowed with ζ_{m_i}. This comes from the fact that the only action "consumed" in the execution of the probabilistic choice is a τ-action and therefore this step does not change the input sequence, i.e., the input sequence given to every ζ_{m_i} is the same as the one given to ζ_m. This also means that every ζ_{m_i} is the worst-case scheduler in the subbranch starting with the process T_i and hence we have

$$\begin{array}{rcl} \mathcal{P}_{MAP}(M_{\zeta_m}(\sum_i p_i\, T_i), \vec{\pi}) & \geq & \sum_i p_i \mathcal{P}_{MAP}(M_{\zeta_{m_i}}(T_i), \vec{\pi}) \\ & \geq & \sum_i p_i Pt_{MAP}(T_i, \vec{\pi}) \end{array}$$

which concludes the proof of the first inequality.

The second inequality follows from Corollary 3.4.11.

2. Let $\zeta_m = \text{minarg}_{\mathcal{A}} \mathcal{P}_{MAP}(M_\zeta(\lfloor\!+\!\rfloor_k\, o_k.T_k), \vec{\pi})$. Let $\mathcal{A}_i$ be the class of schedulers that choose the i-th branch at the beginning of the execution, and define

$$\zeta_{n_i} = \text{minarg}_{\mathcal{A}_i} \mathcal{P}_{MAP}(M_\zeta(\Big\lfloor\!+\!\Big\rfloor_k\, o_k.T_k), \vec{\pi})$$

Since ζ_m is the worst-case scheduler for the whole execution tree and ζ_{n_i} is the worst-case scheduler for the subtree starting with T_i, ζ_m coincides with ζ_{n_j}, if ζ_m chooses the transition $\lfloor\!+\!\rfloor_k\, o_k.T_k \xrightarrow{o_j} \delta(T_j)$ as first step. The remaining question is now how to determine which first path is chosen by ζ_m. This will obviously be the path j in which the scheduler ζ_{n_j} leads to the minimal probability of error compared to the any other scheduler ζ_{n_k} in a path k. Therefore we have

$$\begin{array}{rcl} Pt_{MAP}(\lfloor\!+\!\rfloor_i\, o_i.T_i, \vec{\pi}) & = & \mathcal{P}_{MAP}(M_{\zeta_m}(\lfloor\!+\!\rfloor_k\, o_k.T_k), \vec{\pi}) \\ & = & \min_i\ \mathcal{P}_{MAP}(M_{\zeta_{n_i}}(\lfloor\!+\!\rfloor_k\, o_k.T_k), \vec{\pi}) \end{array}$$

Consider now, for each i, the scheduler ζ_{m_i} defined as as ζ_{n_i}, except for the removal of the first state and the first step from the execution fragments in the domain. Obviously ζ_{m_i} is still compatible with $\mathcal{S}$ on $\lfloor\!+\!\rfloor_k\, o_k.T_k$ (because the first step does not involve a secret action), and the observables of T_i are in one-to one correspondence with those of $\lfloor\!+\!\rfloor_k\, o_k.T_k$ via the bijective function $f_i(o_i o_{j_1} \ldots o_{j_k}) = o_{j_1} \ldots o_{j_k}$ which maps the observables of the process $o_i T_i$ to the observables of T_i. Furthermore, all the probabilities of the channel $M_{\zeta_{n_i}}(\lfloor\!+\!\rfloor_i\, o_i.T_i)$ are the same as those of $M_{\zeta_{m_i}}(T_i)$ modulo the renaming of o into $f(o)$. In other words, the scheduler does not increase the probability of error by another way that the choice of a transition.

$$\begin{array}{rcl} Pt_{MAP}(\lfloor\!+\!\rfloor_i\, o_i.T_i, \vec{\pi}) & = & \min_i\ \mathcal{P}_{MAP}(M_{\zeta_{n_i}}(\lfloor\!+\!\rfloor_k\, o_k.T_k), \vec{\pi}) \\ & = & \min_i\ \mathcal{P}_{MAP}(M_{\zeta_{m_i}}(T_i), \vec{\pi}) \\ & = & \min_i\ Pt_{MAP}(T_i, \vec{\pi}) \end{array}$$

which concludes the proof.

3. Let $\zeta_m = \text{minarg}_{\mathcal{A}} \mathcal{P}_{MAP}(M_\zeta((\nu o_1, o_2, \ldots, o_k)\,(T_1 \mid T_2)), \vec{\pi})$. Hence

$$Pt_{MAP}((\nu o_1, o_2, \ldots, o_k)\,(T_1 \mid T_2), \vec{\pi}) \quad = \\ \mathcal{P}_{MAP}(M_{\zeta_m}((\nu o_1, o_2, \ldots, o_k)\,(T_1 \mid T_2)), \vec{\pi})$$

The proof proceeds by constructing a set of series of schedulers whose limit with respect to the metric d in Definition 3.4.3 correspond to schedulers on the execution tree of T_2. Consider a generic node in the execution tree of $(\nu o_1, o_2, \ldots, o_k)\,(T_1 \mid T_2)$ under ζ_m, and let $(\nu o_1, o_2, \ldots, o_k)\,(T_1' \mid T_2')$ be the corresponding term in that node, where T_1' represents the evolution of T_1 and T_2' represents the evolution of T_2. Assume h to be the execution history up to that node. Let us consider separately the three possible kinds of transitions derivable from the operational semantics, i.e., a transition from T_1', a transition from T_2' or a synchronization between T_1' and T_2':

a) Transition from T_1'
We consider the step

$$(\nu o_1, o_2, \ldots, o_k)\,(T_1' \mid T_2') \xrightarrow{a} (\nu o_1, o_2, \ldots, o_k)\,(\mu \mid T_2') \tag{3.3}$$

due to a transition $T_1' \xrightarrow{a} \mu$. In this case, a cannot be a secret action because of the assumption that T_1 does not contain secret actions. On the other hand, the assumption that all the observable actions of T_1' are included in $\{o_1, o_2, \ldots, o_k\}$, and the fact that the transition does not "consume" any of them (as seen from the top-level restrictions) means that a cannot be an observable action either. Therefore, a must be τ. Assume that $\mu = \sum_i p_i\ \delta(T_{1i}')$. Then we have
$(\nu o_1, o_2, \ldots, o_k)\,(\mu \mid T_2') = \sum_i p_i\ \delta((\nu o_1, o_2, \ldots, o_k)\,(T_{1i}' \mid T_2'))$. Let us consider the tree obtained by replacing this distribution with
$\delta((\nu o_1, o_2, \ldots, o_k)\,(T_{1i}' \mid T_2'))$ (i.e., the tree obtained by pruning all alternatives except $(\nu o_1, o_2, \ldots, o_k)\,(T_{1i}' \mid T_2')$, and assigning to it probability 1). Let ζ_{mi} be the projection of ζ_m on the new tree (i.e., ζ_{mi} is defined as the projection of ζ_m on the histories h' such that if h is a proper prefix of h' then $h\tau(\nu o_1, o_2, \ldots, o_k)\,(T_{1i}' \mid T_2')$ is a prefix of h'). We have

$$\mathcal{P}_{MAP}(M_{\zeta_m}((\nu o_1, o_2, \ldots, o_k)\,(T_1 \mid T_2)), \vec{\pi}) \\ = \\ \mathcal{P}_{MAP}(\textstyle\sum_i p_i\ M_{\zeta_{mi}}((\nu o_1, o_2, \ldots, o_k)\,(T_1 \mid T_2)), \vec{\pi}) \\ \geq \qquad \text{(by Proposition 3.4.10)} \\ \textstyle\sum_i p_i\ \mathcal{P}_{MAP}(M_{\zeta_{mi}}((\nu o_1, o_2, \ldots, o_k)\,(T_1 \mid T_2)), \vec{\pi})$$

The transition given in Equation 3.3 does not have a correspondent in the execution tree of T_2 (since the execution in the parallel processes occurs on the side of T_1). However, the outcome of the transition in the side of T_1 may have an influence on the future computation in the execution tree of T_2: nothing prevents the execution of $(\nu o_1, o_2, \ldots, o_k)\,(T_{1i}' \mid T_2')$ from differing from the execution of $(\nu o_1, o_2, \ldots, o_k)\,(T_{1j}' \mid T_2')$ when $i \neq j$.

In other words, ζ_{mi} and ζ_{mj} may follow different paths in the execution tree of T_2. This obliges us to consider all different schedulers for T_2 which are associated to the various ζ_{mi}'s for different i's.

b) Transition from T_2'
We consider a step

$$(\nu o_1, o_2, \ldots, o_k)\,(T_1' \mid T_2') \xrightarrow{a} (\nu o_1, o_2, \ldots, o_k)\,(T_1' \mid \mu)$$

due to a transition $T_2' \xrightarrow{a} \mu$, with a not included in $\{o_1, o_2, \ldots, o_k\}$. In this case, the corresponding scheduler for T_2 will choose the same transition, i.e., $T_2' \xrightarrow{a} \mu$. This comes from the fact that the observables actions of T_1 are included in $\{o_1, o_2, \ldots, o_k\}$, which are not "consumed" in this transition. Therefore, the scheduler for T_2 cannot win anything by choosing another transition than $T_2' \xrightarrow{a} \mu$.

c) Synchronization between T_1' and T_2'
We consider a step

$$(\nu o_1, o_2, \ldots, o_k)\,(T_1' \mid T_2') \xrightarrow{\tau} (\nu o_1, o_2, \ldots, o_k)\,\delta(T_1'' \mid T_2'')$$

due to the transitions $T_1' \xrightarrow{a} \delta(T_1'')$ and $T_2' \xrightarrow{\bar{a}} \delta(T_2'')$. In this case a must be an observable o because of the assumption that T_1 does not contain secret actions. The corresponding scheduler for T_2 must choose the transition $T_2' \xrightarrow{\bar{a}} \delta(T_2'')$.

By considering the inequalities given by the transitions of type (a), we obtain

$$\begin{gathered}
\mathcal{P}_{MAP}(M_{\zeta_m}((\nu o_1, o_2, \ldots, o_k)\,(T_1 \mid T_2)), \vec{\pi}) \\
\geq \\
\textstyle\sum_i p_i\ \mathcal{P}_{MAP}(M_{\zeta_{mi}}((\nu o_1, o_2, \ldots, o_k)\,(T_1 \mid T_2)), \vec{\pi}) \\
\geq \\
\textstyle\sum_i p_i\ \sum_j q_j\ \mathcal{P}_{MAP}(M_{\zeta_{mij}}((\nu o_1, o_2, \ldots, o_k)\,(T_1 \mid T_2)), \vec{\pi}) \\
\geq \\
\textstyle\sum_i p_i\ \sum_j q_j\ \sum_h r_h\ \mathcal{P}_{MAP}(M_{\zeta_{mijh}}((\nu o_1, o_2, \ldots, o_k)\,(T_1 \mid T_2)), \vec{\pi}) \\
\geq \\
\ldots
\end{gathered}$$

Observe now that $\{\zeta_m, \zeta_{mi}, \zeta_{mij}, \zeta_{mijh}, \ldots\}$ is a converging series of schedulers whose limit $\zeta_{mijh\ldots}$ is isomorphic to a scheduler for T_2, except that some of the observable transitions in T_2 may be removed due to the restriction on $o_1, o_2, \ldots, o_k$. This removal determines a (usually non injective) mapping f on

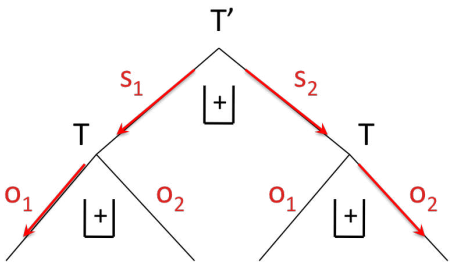

Figure 3.1: Secret choice does not preserve safety

the observables. Hence:

$$\begin{array}{c}
\mathcal{P}_{MAP}(M_{\zeta_m}((\nu o_1, o_2, \ldots, o_k)\,(T_1 \mid T_2)), \vec{\pi}) \\
\geq \\
\sum_i p_i \ \sum_j q_j \ \sum_h r_h \ \ldots \mathcal{P}_{MAP}(M_{\zeta_{mijh\ldots}}((\nu o_1, o_2, \ldots, o_k)\,(T_1 \mid T_2)), \vec{\pi}) \\
\geq \qquad \text{(by Proposition 3.4.12)} \\
\sum_i p_i \ \sum_j q_j \ \sum_h r_h \ \ldots \mathcal{P}_{MAP}(M_{\zeta_{mijh\ldots}}(T_2), \vec{\pi}) \\
\geq \\
\sum_i p_i \ \sum_j q_j \ \sum_h r_h \ \ldots \min_{\zeta \in \mathcal{A}} \mathcal{P}_{MAP}(M_\zeta(T_2), \vec{\pi})
\end{array}$$

Finally, observe that $\sum_i p_i = \sum_j q_j = \sum_h r_h = \ldots = 1$, hence

$$\mathcal{P}_{MAP}(M_{\zeta_m}((\nu o_1, o_2, \ldots, o_k)\,(T_1 \mid T_2)), \vec{\pi}) \geq \min_{\zeta \in \mathcal{A}} \mathcal{P}_{MAP}(M_\zeta(T_2), \vec{\pi})$$

which concludes the proof.

□

Unfortunately the safety property does not hold for the secret choice. The following is a counterexample, illustrated on Figure 3.1.

Example 3.5.3. *Let the set of secrets be* $Sec = \{s_1, s_2\}$ *and assume that the set* $\mathcal{S}$ *of possible input sequences does not contain the empty sequence. Let* $T = o_1.0 \boxplus o_2.0$. *Then* $Pt_{MAP}(T, \vec{\pi})$ *is maximum (i.e.,* $Pt_{MAP}(T, \vec{\pi}) = 1 - \max \vec{\pi}$*) because for every sequence* $s \in \mathcal{S}$ *we have* $p(o_1|s) = p(o_2|s)$. *Let* $T' = s_1.T \boxplus s_2.T$. *We can now define a scheduler* ζ_0 *such that, if the secret starts with* s_1, *it selects* o_1, *and if the secret starts with* s_2, *it selects* o_2. *Hence, under this scheduler,* $p(o_1|s_1 s) = p(o_2|s_2 s) = 1$ *while* $p(o_1|s_2 s) = p(o_2|s_1 s) = 0$. *Therefore*

$$\begin{array}{rcl}
\mathcal{P}_{MAP}(M_{\zeta_0}(T'), \vec{\pi}) & = & 1 - \sum_{\mathcal{O}} \max_{s \in \mathcal{S}}(p(o|s)\pi_s) \\
& = & 1 - \max_{s \in \mathcal{S}}(p(o_1|s_1 s)p_1) - \max_{s \in \mathcal{S}}(p(o_2|s_2 s)p_2) \\
& = & 1 - p_1 - p_2
\end{array}$$

where p_1 *and* p_2 *are the maximum probabilities of the secrets of the form* $s_1 s$ *and* $s_2 s$, *respectively. Note now that either* $\max \vec{\pi} = p_1$ *or* $\max \vec{\pi} = p_2$ *because of the*

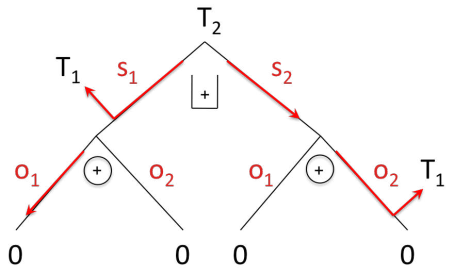

Figure 3.2: Without condition (i), parallel composition may not preserve safety

assumption that $\mathcal{S}$ does not contain the empty sequence. Let $\vec{\pi}$ be such that both p_1 and p_2 are positive. Then we have

$$\begin{array}{rcll} Pt_{MAP}(T', \vec{\pi}) & = & \min_\zeta \ \mathcal{P}_{MAP}(M_\zeta(T'), \vec{\pi}) & \textit{by definition} \\ & \leq & \mathcal{P}_{MAP}(M_{\zeta_0}(T'), \vec{\pi}) & \\ & = & 1 - p_1 - p_2 & \\ & < & 1 - \max \vec{\pi} & p_1 \textit{ and } p_2 \textit{ positive} \\ & = & Pt_{MAP}(T, \vec{\pi}) & \end{array}$$

which shows that the safety property is not satisfied.

The reason why we need the condition *(i)* in Definition 3.5.1 for the parallel operator is analogous to the case of secret choice. The following is a counterexample illustrated on Figure 3.2.

Example 3.5.4. *Let Sec and $\mathcal{S}$ be as in Example 3.5.3. Define $T_1 = s_1.0 \boxplus s_2.0$ and $T_2 = o_1.0 \boxplus o_2.0$. Clearly, $Pt_{MAP}(T_2, \vec{\pi}) = 1 - \max \vec{\pi}$. Consider now the term $T_1 \mid T_2$ and define a scheduler that first executes an action s in T_1 and then, if s is s_1, it selects o_1, while if s is s_2, it selects o_2. The rest proceeds like in Example 3.5.3, where $T' = T_1 \mid T_2$ and $T = T_2$.*

The reason why we need the condition *(ii)* in Definition 3.5.1 is that without it the parallel operator may create different interleavings, thus increasing the possibility of an adversary discovering the secrets. The following is a counterexample illustrated on Figure 3.3.

Example 3.5.5. *Let Sec and $\mathcal{S}$ be as in Example 3.5.3. Define $T_1 = o.0$ and $T_2 = s_1.(o_1.0 \oplus_{.5} o_2.0) \boxplus s_2.(o_1.0 \oplus_{.5} o_2.0)$. It is easy to see that $Pt_{MAP}(T_2, \vec{\pi}) = 1 - \max \vec{\pi}$ (here T_2 only involves probabilistic and secret choices, which do not leave any nondeterminism to be resolved by the scheduler). Consider the term $T' = T_1 \mid T_2$ and define a scheduler that first executes an action s in T_2 and then, if s is s_1, it selects first T_1 and then the continuation of T_2, while if s is s_2, it selects first the continuation of T_2 and then T_1. Hence, under this scheduler, $p(oo_1|s_1s) = p(oo_2|s_1s) = .5$ and also $p(o_1o|s_2s) = p(o_2o|s_2s) = .5$ while $p(oo_1|s_2s) = p(oo_2|s_2s) = 0$ and $p(o_1o|s_1s) = p(o_2o|s_1s) = 0$. Therefore $Pt_{MAP}(T', \vec{\pi}) \leq 1 - p_1 - p_2$ where p_1 and p_2 are the maximum probabilities of the secrets of the form s_1s and s_2s, respectively. Following*

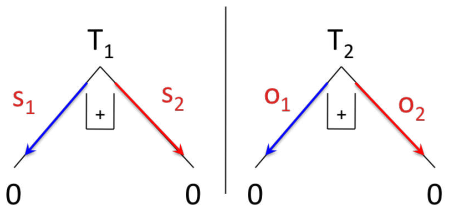

Figure 3.3: Without condition (ii), parallel composition may not preserve safety

the same reasoning as in Example 3.5.3, we have that for certain $\vec{\pi}$, $Pt_{MAP}(T', \vec{\pi}) \leq 1 - p_1 - p_2 < 1 - \max \vec{\pi} = Pt_{MAP}(T_2, \vec{\pi})$.

3.6 A case study: the Dining Cryptographers

In this section, we consider the Dining Cryptographers (DC) protocol already presented in Paragraph 1.2.2.2, and we show how to describe it in CCS$_p$. Then, we apply the results of the previous section to obtain a generalization of Chaum's strong anonymity result.

The cryptographers correspond to nodes of a so-called DC multigraph, in which the edges represent the coins and there may be several edges between two nodes (hence the name "multigraph"). Here the master is not explicitely represented. The theorem proven by Chaum in [Cha88] states that the DC is strongly anonymous if all the coins are fair, i.e., they give 0 and 1 with equal probability, and the DC multigraph is connected, namely there is a path between each pair of nodes. To state formally the property, let us denote by s the secret identity of the payer, and by o the collection of the declarations of the cryptographers.

Theorem 3.6.1 ([Cha88]). *If the DC multigraph is connected, and the coins are fair, then DC is strongly anonymous, namely for every* s *and* o, $p(s|o) = p(s)$ *holds.*

We can now represent a DC protocol involving n cryptographers as a noisy channel matrix M where each coefficient $m_{ij} = p(o_j|s_i)$ corresponds to the probability that the cryptographer i is the payer, given the observable o_j. There are n secrets and 2^n observables in the protocol: each observable $o_j = (o_{j1}, o_{j2}, \ldots, o_{jn})$ is a (binary) n-tuple of the possible answers of the cryptographers.

We are now going to show how to express the DC in CCS$_p$. We start by introducing a notation for value-passing in CCS$_p$, following standard lines.

$$\begin{array}{lrcl} \textit{Input} & c(x).T & = & \left\lfloor + \right\rfloor_v c_v.T[v/x] \\ \textit{Output} & \bar{c}\langle v \rangle & = & \bar{c}_v \end{array}$$

We define a process $Crypt_i$ and a process $Coin_h$ for each cryptographer and each coin respectively. An additional process $Collect$ is created whose purpose is to collect all the declarations of the cryptographers, and output them in the form of a tuple. To each cryptographer $Crypt_i$ is associated a secret action pay_i, which is zero unless the corresponding cryptographer is the payer. All the other actions are observables.

$$
\begin{array}{rcl}
Crypt_i & = & c_{i,i_1}(x_1) \,.\, \ldots \,.\, c_{i,i_k}(x_k) \,.\, pay_i(z_i) \,.\, \bar{out}_i\langle x_1 + \ldots + x_k + z_i\rangle \\
Coin_h & = & \bar{c}_{\ell,h}\langle 0\rangle \,.\, \bar{c}_{r,h}\langle 0\rangle.0 \;\oplus_{p_h}\; \bar{c}_{\ell,h}\langle 1\rangle \,.\, \bar{c}_{r,h}\langle 1\rangle.0 \\
Collect & = & out_1(y_1) \,.\, out_2(y_2) \,.\, \ldots \,.\, out_n(y_n) \,.\, \overline{outall}\langle y_1, y_2, \ldots, y_n\rangle \\
DC & = & (\nu\vec{c})(\nu\vec{out})(\prod_i Crypt_i \mid \prod_h Coin_h \mid Collect)
\end{array}
$$

Table 3.1: The dining cryptographers protocol expressed in CCS_p. The action $\bar{out}_i\langle x_1 + \ldots + x_k + z_i\rangle$ represents the output of the sum modulo 2 of $x_1, \ldots, x_k, z_i$ and the notation $\prod_i T_i$ stands for the parallel computation of all T_i processes.

The channel $c_{i,h}$ represents the communication channel between the cryptographer $Crypt_i$ and the coin $Coin_h$ if the index h is indeed the index of a coin in the system. Otherwise, $c_{i,h}$ is a communication channel "with the environment". We call this latter *external* channel. In the original definition of the DC there are no external channels, we have added them to prove a generalization of Chaum's result. They could be interpreted as a way for the environment to influence the computation of the cryptographers and hence to test the system, for the purpose of discovering the secret.

The protocol can then be described as the parallel composition of the cryptographers processes $Crypt_i$, of the coin processes $Coin_h$, and of the process $Collect$. See Table 3.1 for the DC protocol expressed in CCS_p.

We are now ready to state our generalization of Chaum's result, which states that all edges (i.e., coins) of the DC network are not required to achieve strong anonymity, as long as a spanning tree of fair coins connects all cryptographers of the network.

We recall that a *spanning tree* of a connected graph G is a tree composed of all the vertices and some (or perhaps all) of the edges of G. It is also a minimal set of edges that connect all vertices of the graph.

Theorem 3.6.2. *A DC is strongly anonymous if the DC multigraph has a spanning tree consisting of fair coins only.*

Proof. Consider the term DC in Table 3.1. Remove all the coins that do not belong to the spanning tree, and the corresponding restriction operators. Let T be the process term obtained this way. Let $\mathcal{A}$ be the class of schedulers which select the value 0 for all the external channels. This situation corresponds to the original formulation of Chaum and so we can apply Chaum's result (Theorem 3.6.1) and Proposition 3.4.14 to conclude that all the rows of the matrix M are the same and hence, by Proposition 3.4.13, $\mathcal{P}_{\scriptscriptstyle MAP}(M, \vec{\pi}) = 1 - \max_i \pi_i$.

Consider now one of the removed coins, h, and assume, without loss of generality, that $c_{\ell,h}(x)$, $c_{r,h}(x)$ are the first actions in the definitions of $Crypt_\ell$ and $Crypt_r$. Consider the class of schedulers $\mathcal{B}$ that selects value 1 for x in these actions. The matrix M' that we obtain is isomorphic to M: the only difference is that each column o is now mapped to a column $o + w$, where w is a tuple that has 1 in the ℓ and r positions, and 0 in all other positions, and $+$ represents the componentwise binary sum. Since this map is a bijection, we can apply Proposition 3.4.12 in both directions and derive that $\mathcal{P}_{\scriptscriptstyle MAP}(M', \vec{\pi}) = 1 - \max_i \pi_i$.

By repeating the same reasoning on each of the removed coins, we can conclude that $Pt_{MAP}(T, \vec{\pi}) = 1 - \max_i \pi_i$ for any scheduler ζ of T.

Consider now the term T' obtained from T by adding back the coin h:

$$T' = (\nu c_{\ell,h} c_{r,h})(Coin_h \mid T)$$

By applying Theorem 3.5.2 we can deduce that

$$Pt_{MAP}(T', \vec{\pi}) \geq Pt_{MAP}(T, \vec{\pi})$$

By repeating this reasoning, we can add back all the coins, one by one, and obtain the original DC. Hence we can conclude that

$$Pt_{MAP}(DC, \vec{\pi}) \geq Pt_{MAP}(T, \vec{\pi}) = 1 - \max_i \pi_i$$

and, since $1 - \max_i \pi_i$ is the maximum probability of error we have

$$Pt_{MAP}(DC, \vec{\pi}) = 1 - \max_i \pi_i$$

which concludes the proof. □

Interestingly, also the other direction of Theorem 3.6.2 holds. We report this result for completeness, however we have proved it by using traditional methods, not by applying the compositional methods of Section 3.5.

Theorem 3.6.3. *A DC is strongly anonymous only if the DC multigraph graph has a spanning tree consisting of fair coins only.*

Proof. By contradiction. Let G be the multigraph associated to the DC and let n be the number of vertices in G. Assume that G does not have a spanning tree consisting only of fair coins. Then it is possible to split G in two non-empty subgraphs, G_1 and G_2, such that all the edges between G_1 and G_2 are unfair. Let $(c_1, c_2, \ldots, c_m)$ be the vector of coins corresponding to these edges. Since G is connected, we have that $m \geq 1$.

Let a_1 be a vertex in G_1 and a_2 be a vertex in G_2. By strong anonymity, for every observable o we have

$$p(o \mid a_1) = p(o \mid a_2) \tag{3.4}$$

Observe now that $p(o \mid a_1) = p(o + w \mid a_2)$ where w is a binary vector of dimension n containing 1 exactly twice, in correspondence of a_1 and a_2, and $+$ is the binary sum. Hence (3.4) becomes

$$p(o + w \mid a_2) = p(o \mid a_2) \tag{3.5}$$

Since, by construction, G_1 and G_2 are two non-empty subgraphs, part of the elements in the n-tuple o correspond to edges in G_1 and others to edges in G_2. Let d be the binary sum of all the elements of o in G_1, and d' be the binary sum of all the elements of $o + w$ in G_1. Since in G_1 w contains 1 exactly once, we have $d' = d + 1$. Hence Equation 3.5, being valid for all o's, implies

$$p(d + 1 \mid a_2) = p(d \mid a_2) \tag{3.6}$$

Because of the way o, and hence d, are calculated, and since the contribution of the edges internal to G_1 is 0, and a_2 (the payer) is not in G_1, we have that

$$d = \sum_{i=1}^{m} c_i$$

from which, together with Equation 3.6, and the fact that the coins are independent from the choice of the payer, we derive

$$p(\sum_{i=1}^{m} c_i = 0) = p(\sum_{i=1}^{m} c_i = 1) = 1/2 \tag{3.7}$$

The last step is to prove that $p(\sum_{i=1}^{m} c_i = 0) = 1/2$ implies that one of the c_i's is fair, which will give us a contradiction. We prove this by induction on m. The property obviously holds for $m = 1$. Let us now assume that we have proved it for the vector $(c_1, c_2, \ldots, c_{m-1})$. Observe that $p(\sum_{i=1}^{m} c_i = 0) = p(\sum_{i=1}^{m-1} c_i = 0)p(c_m = 0) + p(\sum_{i=1}^{m-1} c_i = 1)p(c_m = 1)$. From Equation 3.7 we derive

$$p(\sum_{i=1}^{m-1} c_i = 0)p(c_m = 0) + p(\sum_{i=1}^{m-1} c_i = 1)p(c_m = 1) = 1/2 \tag{3.8}$$

Now, it is easy to see that Equation 3.8 has only two solutions: one in which $p(c_m = 0) = 1/2$, and one in which $p(\sum_{i=1}^{m-1} c_i = 1) = 1/2$. In the first case we are done, in the second case we apply the induction hypothesis. □

3.7 Conclusion and future work

In this chapter we have investigated the properties of the probability of error associated to a given information-hiding protocol, and the CCS_pconstructs that are safe, i.e., that are guaranteed not to decrease the protection of the protocol. Then we have applied these results to strengthen a result of Chaum: the dining cryptographers are strongly anonymous if and only if they have a spanning tree of fair coins.

In the future, we would like to extend our results to other constructs of the language. This is not possible in the present setting, as the examples after Theorem 3.5.2 show. The problem is related to the scheduler: the standard notion of scheduler is too powerful and can leak secrets, by depending on the secret choices that have been made in the past (problem of the *omniscient scheduler*). All the examples after Theorem 3.5.2 are based on this kind of problem. This problem has already been considered in [CPP07], where a language-based solution was used to restrict the power of the scheduler. We are planning to investigate whether such approach could be exploited here to guarantee the safety of more constructs.

Chapter 4

The logical approach

4.1 Introduction

In this chapter, we propose a modal logic, called *doxastic μ-calculus with error control* (DμCEC), for expressing properties based on belief, such as "the execution of the protocol does not increase the belief about the identity of the culprit" (anonymity), and "Alice believes with degree of confidence $1/2$ that Bob has received the bit 0" (a feature of the oblivious transfer), thus expressing notions that were not captured by the approach based on process calculus developed in Chapter 3. In this chapter, we express security protocols in terms of DμCEC logical formulae interpreted as processes of the specification formalism CCS_p(CCS with probabilistic internal choice). This language was already presented in Chapter 2 and used in Chapter 3 to formalize information-hiding protocols and to study their compositionality.

Contrary to Chapter 3, we will not need in this chapter to distinguish between secret and nondeterministic choices and we will therefore use the initial version of the language.

The distinguishing feature of our logic is to provide a combination of *dynamic* operators for belief (whence the attribute "doxastic") with a *control* on the possible error of apprehension of the perceived reality, and for internalized probability. Both operators are dynamic (non-monotonic) thanks to the possibility of combining them with temporal operators, and are parameterized with a lower and upper probability bound (the error control).

Dynamicity is useful for the logical formalization of the original intuition of probabilistic anonymity and oblivious transfer, namely the invariance between the *a priori* and the *a posteriori* stances of apprehension of the perceived reality (cf. Section 4.3 and Paragraph 4.4.2). The belief operator is used to express that an agent a believes with *confidence* of at least l and at most u that a state of affairs ϕ is the case. The operator for internalized probability is used to express that a state of affairs ϕ is the case with *certainty* of at least l and at most u. Note that confidence is a qualification of an agent's belief (that something is the case), whereas certainty is just a qualification of something being the case: confidence has a subjective (belief) connotation whereas certainty has an objective (truth) connotation. In our framework, both qualifications are also quantitative thanks to the mentioned error control in terms of a lower and upper probability bound.

Our motivation for developing such a logic relies on its multiple advantages compared to the approaches based on the expression of the properties directly on the un-

derlying formalism used to model the protocol. First, the logic resides on a higher level and therefore allows to reason more deeply about the properties of the protocol, by highlighting their subtelties. Moreover, using a logic to express properties leads to a specification that is independent from the (lower-level) formalism used for representing the protocol. Last but not least, the logic is more expressive, in particular thanks to the possible distinction between subjective and objective uncertainty, i.e. between belief and probabilistic truth.

As an example, consider the property of strong anonymity that, in Paragraph 1.2.2.3 and following [CP05], we expressed as the equality of the likelihoods of all anonymous events. Intuitively we intend such likelihoods to represent the subjective uncertainty of an adversary, but, having only one form of probability to express them made it in that case impossible to distinguish between belief and probabilistic truth. Here however we are able to make such distinction, and we express strong anonymity in terms of belief (see Section 4.3). We come back to the Dining Cryptographers (DC) network (slightly modified compared to the example in Section 3.6) to illustrate our approach, and additionally consider Oblivious Transfer [Rab81] for single bits and entire strings. The properties of the Oblivious Transfer, which were already analyzed in [CP07a], are represented here by using both belief and probabilistic truth (see Section 4.4).

We start this chapter by introducing in Section 4.2 the *doxastic μ-calculus with error control* (DμCEC). Sections 4.3 and 4.4 are dedicated to the formalization and validation of probabilistic anonymity and oblivious transfer. Finally, Section 4.5 concludes the chapter with an assessment of achievements and future work.

4.2 A quantitative doxastic logic and its interpretation in CCS$_p$

4.2.1 Modal operators for belief and truth

In this section we propose an extension DμCEC of Nielsen's *μ-calculus with past* [Nie98] suitable for expressing information-hiding properties and for reasoning about security protocols.

Recall that the μ-calculus has modal operators $\langle a \rangle$ that express the *future capabilities* of a process: the formula $\langle a \rangle \phi$ means that a process can perform the action a and evolve into a new process that satisfies ϕ. In addition to these, Nielsen's calculus contains also their *past* counterparts $\langle \overleftarrow{a} \rangle$: the formula $\langle \overleftarrow{a} \rangle \phi$ means that the process is the outcome of an a-transition from another process which satisfies ϕ.

We extend Nielsen's calculus in two ways:

1. We internalize probabilistic truth in the form of probabilistic statements which are based on Parma and Segala's probabilistic extension [PS07] of Hennessy-Milner logic. They consider constructs like $[\phi]_p$, where p is a parameter representing a probability. Formulas are interpreted on probability measures, and the meaning of $[\phi]_p$ is that the set of states that satisfy ϕ has probability *at least* p with respect to the given measure. We actually consider constructs like $P_p^q(\phi)$, meaning that the set of states that satisfy ϕ has probability *at least* p *and at most* q. The operator $P_p^q(\phi)$ could be approximated by $[\phi]_p \wedge \neg[\phi]_{q+\epsilon}$, but we prefer to have the former as a primitive because in some examples we need exact probabilities.

2. We add belief in the form of doxastic operators ${}_i\mathcal{B}_p^q$ which represent the degree of confidence of agents about the truth of formulas in DμCEC. Intuitively the

formula ${}_i\mathcal{B}^q_p(\phi)$ means that the agent i estimates that there is a probability *at least* p *and at most* q that the process satisfies ϕ.

Remark 4.2.1. *Note that in general* ${}_i\mathcal{B}^q_p(P^1_1(\phi))$ *and* ${}_i\mathcal{B}^1_1(P^q_p(\phi))$ *are not equivalent, neither are* $P^1_1({}_i\mathcal{B}^q_p(\phi))$ *and* $P^q_p({}_i\mathcal{B}^1_1(\phi))$*. The intuitive meaning of these four formulae is the following:*

- ${}_i\mathcal{B}^q_p(P^1_1(\phi))$*: Agent* i *believes with probability at least* p *and at most* q *that* ϕ *is always satisfied (i.e. satisfied in all states of the protocol).*
- ${}_i\mathcal{B}^1_1(P^q_p(\phi))$*: Agent* i *knows (i.e. has no doubt) that a state in which* ϕ *is satisfied has a probability at least* p *and at most* q *to occur.*
- $P^1_1({}_i\mathcal{B}^q_p(\phi))$*: In all states, agent* i *believes with probability at least* p *and at most* q *that* ϕ *is satisfied.*
- $P^q_p({}_i\mathcal{B}^1_1(\phi))$*: There is a probability at least* p *and at most* q *that a state occurs in which agent* i *knows that* ϕ *is satisfied.*

For instance, if ${}_i\mathcal{B}^1_1(P^{1/2}_{1/2}(\phi))$ *holds, the agent* i *knows that* ϕ *is satisfied in half of the states of the protocol, while if* ${}_i\mathcal{B}^{1/2}_{1/2}(P^1_1(\phi))$ *holds, agent* i *is never sure about anything. On the other hand if* $P^{1/2}_{1/2}({}_i\mathcal{B}^1_1(\phi))$ *holds, we may find a state in which agent* i *knows that* ϕ *is false, while if* $P^1_1({}_i\mathcal{B}^{1/2}_{1/2}(\phi))$ *holds, agent* i *is never sure about anything.*

4.2.2 Syntax of DμCEC

The syntax of DμCEC is given by the following grammar, where p and q are constant between 0 and 1:

$$\phi ::= \top \mid X \mid \phi \wedge \phi \mid \neg\phi \mid \langle a \rangle \phi \mid \overleftarrow{\langle a \rangle}\phi \mid P^q_p(\phi) \mid {}_i\mathcal{B}^q_p(\phi) \mid lfp_X \phi(X)$$

where $lfp_X \phi(X)$ is a *least fixpoint formula* where the variable X is assumed to occur positively in ϕ.

The use of fixpoint operators in modal logics of programs goes back mainly to Pratt [Pra81], Emerson and Clarke [EC80] and Kozen [Koz83] and was motivated by the need to have a semantics for recursion, which could then provide an effective way of expressing all the usual operators of temporal logics. In particular, formula such as "always ϕ" or "there exists a path on which ϕ eventually holds" can be easily expressed with fixpoint operators. The positivity requirement on the fixpoint operator allows to ensure by a syntactic means that $\phi(X)$ represents a functional monotonic in X, and so has unique minimal and maximal fixpoint. We refer the interested reader to [BS] for more details on modal μ-calculi.

To define formally what it means that X occurs positively in a formula, we use the standard notion of *context* $C[\ ]$ and define the concepts of *positive* and *negative* context as follows.

Definition 4.2.2. *For a context $C[\,]$, the properties of being positive and being negative are defined inductively as follows:*

$[\,]$	*is*	*positive*	
$C[\,] \wedge \phi$	*is*	*positive*	*if $C[\,]$ is positive*
$\phi \wedge C[\,]$	*is*	*positive*	*if $C[\,]$ is positive*
$pop(C[\,])$	*is*	*positive*	*if $C[\,]$ is positive with $pop = \langle\!\langle a \rangle\!\rangle, \overleftarrow{\langle\!\langle a \rangle\!\rangle}, P^1_p$, or ${}_i\mathcal{B}^1_p$*
$nop(C[\,])$	*is*	*positive*	*if $C[\,]$ is negative with $nop = \neg, P^p_0$, or ${}_i\mathcal{B}^p_0$*
$lfp_X C[X]$	*is*	*positive*	*($C[\,]$ must be positive)*

and

$C[\,] \wedge \phi$	*is*	*negative*	*if $C[\,]$ is negative*
$\phi \wedge C[\,]$	*is*	*negative*	*if $C[\,]$ is negative*
$pop(C[\,])$	*is*	*negative*	*if $C[\,]$ is negative with $pop = \langle\!\langle a \rangle\!\rangle, \overleftarrow{\langle\!\langle a \rangle\!\rangle}, P^1_p$, or ${}_i\mathcal{B}^1_p$*
$nop(C[\,])$	*is*	*negative*	*if $C[\,]$ is positive with $nop = \neg, P^p_0$, or ${}_i\mathcal{B}^p_0$*

Definition 4.2.3. *X occurs positively in ϕ if $\phi = C[X]$ for some positive context $C[\,]$.*

Remark 4.2.4. *Informally, a variable X occurs positively if it is not within the scope of an odd numer of negations, that is if we use the actual value of X and not its negations. There exists another implicit kind of negation that occurs when we impose an upper bound to the probability of a positive occurrence of X or when we impose a lower bound to the probability of a negative occurrence of X. This leads to the relative definition of operators P^q_p.*

4.2.3 CCS$_p$ revisited

We want to use the logic DμCEC to express properties of processes written in CCS$_p$, and we will therefore provide an interpretation of our logic in this language, in the form of a *satisfaction relation* $\models$ between CCS$_p$ and DμCEC.

We will use in this chapter the CCS$_p$ variant described in Chapter 2. Again, the whole set of actions is represented by $Act = A \cup \bar{A} \cup \{\tau\}$, but here we assume that the number of channel names (i.e. the elements in A) is finite (rather than countable in the general case). This restriction allows us to express certain operators as syntactic sugar, notably the operators $\langle\!\langle \cdot \rangle\!\rangle$, $\overleftarrow{\langle\!\langle \cdot \rangle\!\rangle}$, and $\boxdot$ of Table 4.2, thus simplifying the theory. The finiteness assumption is not really a restriction in the context of this chapter, because we are interested in analysing properties of programs, that, being finite syntactic entities, can only contain a fixed number of channel names.

4.2.4 Interpretation of DμCEC

We are now ready to define a satisfaction relation between the language CCS$_p$ and our logic DμCEC. In standard Hennessy-Milner logic, and in μ-calculus, satisfaction is usually defined with respect to processes. Here we need to interpret the doxastic operators ${}_i\mathcal{B}^q_p$, and for this purpose we must consider not just the current process, but the whole (finite) history of the execution, because of the dynamic nature of our notion of belief. Furthermore, as explained before, in order to interpret the formulas $P^q_p(\phi)$ we need to consider probabilistic measures. In conclusion, we are going to take as domain the set of discrete distributions $Disc(exec^*(M))$, where $exec^*(M)$ is

the set of finite histories generated by the probabilistic automaton M underlying the CCS$_p$ semantics (as defined in Section 2.2). Note that by definition, all histories in $exec^*(M)$ start with the same initial state s_{init}, the initial state of the automaton M. Given a history h, an action a, and a probability distribution on states μ, we denote by $ha\mu$ the extension of μ to the histories of the form haP, for every CCS$_p$ process P. Namely:

$$(ha\mu)(h') \stackrel{\text{def}}{=} \begin{cases} \mu(P) & \text{if } h' = haP \\ 0 & \text{otherwise} \end{cases}$$

The interpretation of the operators ${}_i\mathcal{B}_p^q$ is based on an epistemic accessibility relation $\equiv_i$ on finite histories. Intuitively $h_1 \equiv_i h_2$ represents the fact that the histories h_1 and h_2 are indistinguishable to an agent i. $\equiv_i$ is usually chosen to be an equivalence relation as induced by the local view of i. We assume that the local view is only restricted to actions (while the states, represented by processes, remain hidden to i), hence we consider the projection of histories on actions (*traces*). Intuitively, the trace of h is the string of the actions in h, i.e. what is left in h after we remove all the states. More formally:

Definition 4.2.5. *Given a finite history h, the trace of h is defined inductively as follows:*

- $trace(P) = \epsilon$ *(the empty trace)*
- $trace(haP) = trace(h)a$

We assume that in general an agent has only a *partial view* on actions. Formally, this can be represented by introducing the following abstraction function:

Assumption 4.2.6. *For every agent i we assume a function $\mathcal{V}_i \colon A \to A \cup \{\epsilon\}$ which represents i's* view *on actions.*

We can now define formally the accessibility relation on traces and histories. We use for simplicity the same symbol $\equiv_i$ to denote both relations.

Definition 4.2.7. *Let a, b be actions and t, t' be traces.*

- *For every agent i, the relation $\equiv_i$ on traces is defined inductively as follows:*
  - $\epsilon \equiv_i \epsilon$
  - $ta \equiv_i t'b$ *if either* $\mathcal{V}_i(a) = \mathcal{V}_i(b)$ *and* $t \equiv_i t'$
    or
    $\mathcal{V}_i(a) = \epsilon$ *and* $t \equiv_i t'\mathcal{V}_i(b)$
    or
    $\mathcal{V}_i(b) = \epsilon$ *and* $t\mathcal{V}_i(a) \equiv_i t'$
- *For every agent i, the relation $\equiv_i$ on histories is defined as follows:*

$$h \equiv_i h' \text{ if and only if } trace(h) \equiv_i trace(h')$$

We can now define the interpretation of DμCEC with respect to the process terms of CCS$_p$. We only consider the closed formulas of DμCEC, namely only the formulas in which all variable occurrences are bound.

$\mu \models \top$		
$\mu \models \phi_1 \wedge \phi_2$	:iff	$\mu \models \phi_1$ and $\mu \models \phi_2$
$\mu \models \neg\phi$	:iff	$\mu \not\models \phi$
$\mu \models \langle\!\langle a \rangle\!\rangle \phi$	:iff	for every $h \in supp(\mu)$ there exists η and a transition $lst(h) \xrightarrow{a} \eta$ such that $ha\eta \models \phi$
$\mu \models \overleftarrow{\langle\!\langle a \rangle\!\rangle} \phi$	:iff	there exists h such that $lst(h) \xrightarrow{a} \mu$ and $\delta(h) \models \phi$
$\mu \models P_p^q(\phi)$	:iff	$p \leq \mu([\![\phi]\!]) \leq q$
$\mu \models {}_i\mathcal{B}_p^q(\phi)$	:iff	for every $h \in supp(\mu)$ and for every scheduler ζ for $fst(h)$ we have $p \leq p_\zeta(\downarrow [\![\phi]\!] \mid \downarrow [h]_{\equiv_i}) \leq q$
$\mu \models lfp_X \phi(X)$	:iff	$\mu \in \bigcap\{D_X \subseteq Disc(exec^*(M)) \mid \forall \eta \in Disc(exec^*(M))$ if $\eta \models \phi(X := D_X)$ then $\eta \models D_X\}$
$\mu \models D_X$	:iff	$\mu \in D_X$
$P \models \phi$	:iff	$\delta(P) \models \phi$

Table 4.1: Definition of satisfaction for the closed formulas in DμCEC. $\mu \in Disc(exec^*(M))$, where $exec^*(M)$ is the set of finite histories generated by the probabilistic automaton M underlying the CCS$_p$ semantics.

Definition 4.2.8. *The relation $\models$ on $Disc(exec^*(M))$ and on the closed formula of DμCEC is defined according to the clauses in Table 4.1. In the table, $[\![\ ]\!]$ is defined as $[\![\phi]\!] \stackrel{\text{def}}{=} \{h \mid \delta(h) \models \phi\}$, while p_ζ represents the probability measure on $etree(P, \zeta)$ (see section 2.2), and $[h]_{\equiv_i}$ is the equivalence class of h with respect to $\equiv_i$. Finally, if H is a set of executions, $\downarrow H$ represents the maximal executions with prefix in H, i.e. $\downarrow H \stackrel{\text{def}}{=} \{h \in \Omega_P \mid \exists h' \in H \text{ s.t. } h' \leq h\}$*

In the definition of $\mu \models {}_i\mathcal{B}_p^q(\phi)$, the idea is that the probability that the process satisfies ϕ *given any h' indistinguishable from h in i's view* is between p and q. We quantify over all possible schedulers because in general i does not know what is the scheduler, except for the partial view it has on h.

The auxiliary "hybrid formulas" D_X ("auxiliary" because they do not exist in the syntax of the language, and "hybrid" because X represents a set of executions) are introduced to define the semantics of lfp_X.

The semantic correspondent of $lfp_X \phi(X)$ (i.e. the set of distributions that satisfy $lfp_X \phi(X)$) is the *least fixed point* of a transformation $\mathcal{T}_\phi \colon 2^{Disc(exec^*(M))} \to 2^{Disc(exec^*(M))}$ defined as follows:

$$\mathcal{T}_\phi(D) \stackrel{\text{def}}{=} \{\mu \mid \mu \models \phi(X := D)\}$$

It can be proved that if X occurs positively in $\phi(X)$ then $\mathcal{T}_\phi$ is monotonic on the lattice $(2^{Disc(exec^*(M))}, \subseteq)$ which, by the Theorem of Knaster-Tarski, implies the existence of the least and greatest fixed points.

The core of the proof is Theorem 4.2.10 below.

Definition 4.2.9 (Monotonicity)**.** *For any formula ϕ in DμCEC, let $\{\!|\phi|\!\}$ denote the set $\{\mu \mid \mu \models \phi\}$. An n-ary operator op in DμCEC is* monotonic. *if for all $\phi_1, \phi_2 \ldots, \phi_n$, $\psi_1, \psi_2 \ldots, \psi_n$, we have that $\{\!|\phi_1|\!\} \subseteq \{\!|\psi_1|\!\}, \{\!|\phi_2|\!\} \subseteq \{\!|\psi_2|\!\}, \ldots, \{\!|\phi_n|\!\} \subseteq \{\!|\psi_n|\!\}$ implies $\{\!|op(\phi_1, \phi_2, \ldots, \phi_n)|\!\} \subseteq \{\!|op(\psi_1, \psi_2, \ldots, \psi_n)|\!\}$. It is* antimonotonic *if for all $\phi_1, \phi_2, \ldots, \phi_n, \psi_1, \psi_2, \ldots, \psi_n$, we have that $\{\!|\phi_1|\!\} \subseteq \{\!|\psi_1|\!\}, \{\!|\phi_2|\!\} \subseteq \{\!|\psi_2|\!\}, \ldots, \{\!|\phi_n|\!\} \subseteq \{\!|\psi_n|\!\}$ implies $\{\!|op(\psi_1, \psi_2, \ldots, \psi_n)|\!\} \subseteq \{\!|op(\phi_1, \phi_2, \ldots, \phi_n)|\!\}$.*

Theorem 4.2.10. *The operators $\wedge$, $\langle a \rangle$, $\overleftarrow{\langle a \rangle}$, P^1_p, ${}_i\mathcal{B}^1_p$ and lfp_X are monotonic. The operators $\neg$, P^p_0 and ${}_i\mathcal{B}^p_0$ are antimonotonic.*

Proof. The proof proceeds by case analysis. We consider here only the operators that are used in this paper, i.e. those which appear in the scope of a *lfp* or *gfp* operator in Table 4.2. In the following, we assume $\{\!|\phi|\!\} \subseteq \{\!|\psi|\!\}$, $\{\!|\phi_1|\!\} \subseteq \{\!|\psi_1|\!\}$, and $\{\!|\phi_2|\!\} \subseteq \{\!|\psi_2|\!\}$.

$\wedge$) Let $\mu \models \phi_1 \wedge \phi_2$. Then by definition $\mu \models \phi_1$ and $\mu \models \phi_2$. From the hypotheses we have $\{\!|\phi_i|\!\} \subseteq \{\!|\psi_i|\!\}$, i.e. $(\mu \models \phi_i) \Rightarrow (\mu \models \psi_i)$ for $i \in \{1, 2\}$, hence $\mu \models \psi_1 \wedge \psi_2$.

$\neg$) Let $\mu \models \neg\psi$. Then by definition $\mu \not\models \psi$. From the hypotheses we have $\{\!|\phi|\!\} \subseteq \{\!|\psi|\!\}$, i.e. $(\mu \not\models \psi) \Rightarrow (\mu \not\models \phi)$, hence $\mu \models \neg\phi$.

$\langle a \rangle$) Let $\mu \models \langle a \rangle \phi$. Then by definition for every $h \in supp(\mu)$ there exists η and a transition $lst(h) \xrightarrow{a} \eta$ such that $h a \eta \models \phi$. From the hypotheses we have $\{\!|\phi|\!\} \subseteq \{\!|\psi|\!\}$, i.e. $(h a \eta \models \phi) \Rightarrow (h a \eta \models \psi)$, hence $\mu \models \langle a \rangle \psi$.

$\overleftarrow{\langle a \rangle}$) Let $\mu \models \overleftarrow{\langle a \rangle} \phi$. Then by definition for every $h \in supp(\mu)$ there exists h' such that $lst(h') \xrightarrow{a} \mu$ and $\delta(h') \models \phi$. From the hypotheses we have $\{\!|\phi|\!\} \subseteq \{\!|\psi|\!\}$, i.e. $(\delta(h') \models \phi) \Rightarrow (\delta(h') \models \psi)$, hence $\mu \models \overleftarrow{\langle a \rangle} \psi$.

${}_i\mathcal{B}^1_p$) Let $\mu \models {}_i\mathcal{B}^1_p(\phi)$. Then, by definition, for every $h \in supp(\mu)$ and for every scheduler ζ for $fst(h)$ we have $p \leq p_\zeta(\downarrow [\![\phi]\!] \mid \downarrow [h]_{\equiv_i}) \leq 1$. From the hypotheses we have $\{\!|\phi|\!\} \subseteq \{\!|\psi|\!\}$, therefore $p_\zeta(\downarrow [\![\phi]\!] \mid \downarrow [h]_{\equiv_i}) \leq p_\zeta(\downarrow [\![\psi]\!] \mid \downarrow [h]_{\equiv_i})$. Hence $p \leq p_\zeta(\downarrow [\![\phi]\!] \mid \downarrow [h]_{\equiv_i}) \leq p_\zeta(\downarrow [\![\psi]\!] \mid \downarrow [h]_{\equiv_i}) \leq 1$, and therefore $\mu \models {}_i\mathcal{B}^1_p(\psi)$.

□

Corollary 4.2.11. *If X occurs positively in $\phi(X)$ then $\mathcal{T}_\phi$ is monotonic on the lattice $(2^{Disc(exec^*(M))}, \subseteq)$.*

Corollary 4.2.12. *If X occurs positively in $\phi(X)$ then the set of fixed points of $\mathcal{T}_\phi$ forms a sublattice of $(2^{Disc(exec^*(M))}, \subseteq)$. In particular, there exists a least and a greatest fixed point.*

4.2.5 Relation with standard (KD45) belief

Following standard Kripke semantics (see e.g. [Eme90]), we can construct a Kripke frame $\mathcal{F}$, defined over $Disc(exec^*(M))$, the non-empty set of discrete distributions with sample space $exec^*(M)$ (where $exec^*(M)$ is the set of finite histories generated by the probabilistic automaton M underlying the CCS$_p$ semantics), and with accessibility relation T, the transition relation of M. Our goal in this section is to discuss the relation of DμCEC with standard $KD45$ belief.

We recall that given a modal operator B and logical formulae ϕ, ψ in a Kripke frame $\mathcal{F} = (W, R)$ over a set W and with accessibility relation R, the logic of belief $KD45$ is the logic generated by the set of the four following axioms:

- Axiom K: $B(\phi \to \psi) \to (B(\phi) \to B(\psi))$
- Axiom D: $B(\phi) \to \neg B(\neg\phi)$
  This axiom states that one cannot believe a contradiction, and requires R to be serial.
- Axiom 4: $B(\phi) \to BB(\phi)$
  This axiom states that belief is positively introspective, and requires R to be transitive.
- Axiom 5: $\neg B(\phi) \to B(\neg B(\phi))$ This axiom states that belief is negatively introspective and requires R to be Euclidean.

Our operators ${}_i\mathcal{B}^q_p$ and P^q_p satisfy probabilistic analogues of the axioms of standard belief and truth, in the sense expressed by Theorems 4.2.13 and 4.2.15 below. In the following, the operator $\to$ stands for Boolean (material) implication, see Table 4.2, and $\models \phi$ means that $\mu \models \phi$ holds for all μ.

Theorem 4.2.13. *For any $p, q, r, s \in [0, 1]$, any formulas ϕ, ψ in DμCEC, and any agent i, the following hold.*

K) $\models {}_i\mathcal{B}^q_p(\phi \to \psi) \to ({}_i\mathcal{B}^s_r(\phi) \to {}_i\mathcal{B}^q_t(\psi))$ *where* $t = \max\{0, p + r - 1\}$

D) $\models {}_i\mathcal{B}^0_0(\bot)$ *("i does not believe false")*

4) $\models {}_i\mathcal{B}^q_p(\phi) \to {}_i\mathcal{B}^1_r({}_i\mathcal{B}^q_p(\phi))$

5) $\models \neg{}_i\mathcal{B}^q_p(\phi) \to {}_i\mathcal{B}^1_r(\neg{}_i\mathcal{B}^q_p(\phi))$

Proof. **K)** Assume $\mu \models {}_i\mathcal{B}^q_p(\phi \to \psi)$ and $\mu \models {}_i\mathcal{B}^s_r(\phi)$. Then, for every $h \in supp(\mu)$ and for every scheduler ζ for $fst(h)$ we have $p \leq p_\zeta(\downarrow[\![\phi \to \psi]\!] \mid \downarrow[h]_{\equiv_i}) \leq q$ and $r \leq p_\zeta(\downarrow[\![\phi]\!] \mid \downarrow[h]_{\equiv_i}) \leq s$. Observe now that:

$$\begin{aligned} p_\zeta(\downarrow[\![\phi \to \psi]\!] \mid \downarrow[h]_{\equiv_i}) &= p_\zeta(\downarrow[\![\neg\phi]\!] \mid \downarrow[h]_{\equiv_i}) + p_\zeta(\downarrow[\![\psi]\!] \mid \downarrow[h]_{\equiv_i}) \\ &\quad - p_\zeta(\downarrow[\![\neg\phi \wedge \psi]\!] \mid \downarrow[h]_{\equiv_i}) \\ &\leq p_\zeta(\downarrow[\![\neg\phi]\!] \mid \downarrow[h]_{\equiv_i}) + p_\zeta(\downarrow[\![\psi]\!] \mid \downarrow[h]_{\equiv_i}) \\ &= (1 - p_\zeta(\downarrow[\![\phi]\!] \mid \downarrow[h]_{\equiv_i})) + p_\zeta(\downarrow[\![\psi]\!] \mid \downarrow[h]_{\equiv_i}) \end{aligned}$$

Hence

$$\begin{aligned} p_\zeta(\downarrow[\![\psi]\!] \mid \downarrow[h]_{\equiv_i}) &\geq p_\zeta(\downarrow[\![\phi \to \psi]\!] \mid \downarrow[h]_{\equiv_i}) + p_\zeta(\downarrow[\![\phi]\!] \mid \downarrow[h]_{\equiv_i}) - 1 \\ &\geq p + r - 1 \end{aligned}$$

Hence $p_\zeta(\downarrow[\![\psi]\!] \mid \downarrow[h]_{\equiv_i}) \geq \max\{0, p + r - 1\}$.

On the other side, observe that we have

$$\begin{aligned} q &\geq p_\zeta(\downarrow[\![\phi \to \psi]\!] \mid \downarrow[h]_{\equiv_i}) \\ &= p_\zeta(\downarrow[\![\psi]\!] \mid \downarrow[h]_{\equiv_i}) + p_\zeta(\downarrow[\![\neg\phi]\!] \mid \downarrow[h]_{\equiv_i}) - p_\zeta(\downarrow[\![\neg\phi \wedge \psi]\!] \mid \downarrow[h]_{\equiv_i}) \\ &\geq p_\zeta(\downarrow[\![\psi]\!] \mid \downarrow[h]_{\equiv_i}) \end{aligned}$$

D) This statement follows immedialy from the observation that for every scheduler ζ and every history h we have $p_\zeta(\downarrow[\![\bot]\!] \mid \downarrow[h]_{\equiv_i}) = p_\zeta(\downarrow[\![\bot]\!]) = 0$.

4) Assume $\mu \models {}_i\mathcal{B}^q_p(\phi)$. Then, for every $h \in supp(\mu)$ and for every scheduler ζ for $fst(h)$ we have $p \leq p_\zeta(\downarrow[\![\phi]\!] \mid \downarrow[h]_{\equiv_i}) \leq q$. Hence, for every $h \in supp(\mu)$ we have $\delta(h) \models {}_i\mathcal{B}^q_p(\phi)$, from which we derive that,for every scheduler ζ for $fst(h)$, $p_\zeta(\downarrow[\![{}_i\mathcal{B}^q_p(\phi)]\!] \mid \downarrow[h]_{\equiv_i}) = 1$ holds. Therefore $\mu \models {}_i\mathcal{B}^1_r({}_i\mathcal{B}^q_p(\phi))$.

5) Similar to the proof of **(4)**.

□

For **(4)** and **(5)**, when $r = 1$ the implication holds also in the other direction, which means that belief can be "flattened" for certain probabilities, as shown in the following proposition.

Proposition 4.2.14. *For every agent i and every formula ϕ, the following hold*

1. $\models {}_i\mathcal{B}^1_1({}_i\mathcal{B}^q_p(\phi)) \to {}_i\mathcal{B}^q_p(\phi)$
2. $\models {}_i\mathcal{B}^1_1(\neg {}_i\mathcal{B}^q_p(\phi)) \to \neg {}_i\mathcal{B}^q_p(\phi)$

Proof. 1. Assume $\mu \models {}_i\mathcal{B}^1_1({}_i\mathcal{B}^q_p(\phi))$. Then, for every $h \in supp(\mu)$ and for every scheduler ζ for $fst(h)$ we have $p_\zeta(\downarrow[\![{}_i\mathcal{B}^q_p(\phi)]\!] \mid \downarrow[h]_{\equiv_i}) = 1$. Hence, for every $h \in supp(\mu)$ we have $\delta(h) \models {}_i\mathcal{B}^q_p(\phi)$, from which we derive that, for every scheduler ζ for $fst(h)$, $p \leq p_\zeta(\downarrow[\![{}_i\mathcal{B}^q_p(\phi)]\!] \mid \downarrow[h]_{\equiv_i}) \leq q$ holds. Therefore $\mu \models {}_i\mathcal{B}^q_p(\phi)$.

2. The proof is similar.

□

For probabilistic truth we have the following

Theorem 4.2.15. *For any $p, q, r, s \in [0, 1]$, any formulas ϕ, ψ in DμCEC, and any agent i, the following hold.*

K) $\models P^q_p(\phi \to \psi) \to (P^s_r(\phi) \to P^q_t(\psi))$ *where* $t = \max\{0, p + r - 1\}$

D) $\models P^0_0(\bot)$

4) $\models P^q_p(\phi) \to P^q_p(P^1_r(\phi))$ *if* $r > 0$

5) $\models \neg P_p^q(\phi) \to P_p^q(\neg P_0^r(\phi))$ *if* $r < 1$

Proof. **K)** Similar to the proof of **(K)** in Theorem 4.2.13.

D) Similar to the proof of **(D)** in Theorem 4.2.13.

4) Assume $\mu \models P_p^q(\phi)$. Then $p \leq \mu([\![\phi]\!]) \leq q$. Observe that $h \in [\![\phi]\!]$ if and only if $\delta(h)([\![\phi]\!]) = 1$, or equivalently, for $r > 0$, $r \leq \delta(h)([\![\phi]\!]) \leq 1$. By definition this is equivalent to $\delta(h) \models P_r^1(\phi)$, which holds if and only if $h \in [\![P_r^1(\phi)]\!]$. Therefore $p \leq \mu([\![P_r^1(\phi)]\!]) \leq q$.

5) Similar to the proof of **(4)**.

□

For **(4)** and **(5)**, the implication holds also in the other direction, meaning that also the probabilistic truth can be "flattened" for certain probabilities.

Proposition 4.2.16. *For every agent i and every formula ϕ, the following hold*

1. $\models P_p^q(P_r^1(\phi)) \to P_p^q(\phi)$ *if* $r > 0$.
2. $\models P_p^q(\neg P_0^r(\phi)) \to \neg P_p^q(\phi)$ *if* $r < 1$.

Proof. 1. Assume $\mu \models P_p^q(P_r^1(\phi))$. Then $p \leq \mu([\![P_r^1(\phi)]\!]) \leq q$. Following the same reasoning as in the proof of Theorem 4.2.15 (4), we have that (for $r > 0$) $h \in [\![P_r^1(\phi)]\!]$ if and only if $h \in [\![\phi]\!]$. Therefore $\mu \models P_p^q(\phi)$.

2. The proof is similar.

□

Finally, we want to point out that the following formulas hold, meaning that our belief operators behave well with respect to probability measures. The proof is immediate.

Proposition 4.2.17. *For every agent i and every formula ϕ, the following hold*

- $\models {}_i\mathcal{B}_p^q(\phi) \leftrightarrow {}_i\mathcal{B}_{1-q}^{1-p}(\neg\phi)$
- $\models P_p^q(\phi) \leftrightarrow P_{1-q}^{1-p}(\neg\phi)$

We conclude this section by giving the definition of some derived operators in DμCEC. They are illustrated in Table 4.2.

4.3 Application: Dining Cryptographers

In this section, we consider a variant of the Dining Cryptographers protocol which is simpler than the version we presented in Chapter 3 in that we will restrict to $n = 3$ cryptographers and add explicitly the master, whose role is to choose the payer (himself or one of the three cryptographers).

In order to specify formally the protocol, we use CCS$_p$, the probabilistic version of CCS presented in Section 2.3, with a standard notation for value-passing:

$$\begin{array}{rrcl} \mathit{Input} & c(x)\,.\,P & = & \sum_v c_v\,.\,P[v/x] \\ \mathit{Output} & \overline{c}\langle v\rangle & = & \overline{c}_v \end{array}$$

$\bot$	$\stackrel{\text{def}}{=} \neg\top$	false
$\phi_1 \vee \phi_2$	$\stackrel{\text{def}}{=} \neg(\neg\phi_1 \wedge \neg\phi_2)$	Boolean disjunction
$\phi_1 \rightarrow \phi_2$	$\stackrel{\text{def}}{=} \neg\phi_1 \vee \phi_2$	Boolean (material) implication
$\boxed{a}\phi$	$\stackrel{\text{def}}{=} \neg\langle a\rangle\neg\phi$	after every a-transitions ϕ holds
$gfp_X\phi(X)$	$\stackrel{\text{def}}{=} \neg lfp\neg\phi(\neg X)$	greatest fixed point of $\lambda X.\phi(X)$. The variable X is assumed to occur positively in ϕ. Note that this implies that also $\neg X$ occurs positively in $\neg\phi$
$\langle a\rangle_p\phi$	$\stackrel{\text{def}}{=} \langle a\rangle P_p^1(\phi)$	there is an a-transition after which ϕ holds with probability at least p
$\langle\cdot\rangle\phi$	$\stackrel{\text{def}}{=} \bigvee_{a\in Act}\langle a\rangle\phi$	there is a transition after which ϕ holds
$\overleftarrow{\langle\cdot\rangle}\phi$	$\stackrel{\text{def}}{=} \bigvee_{a\in Act}\overleftarrow{\langle a\rangle}\phi$	there is a transition before which ϕ holds
$\boxed{\cdot}\phi$	$\stackrel{\text{def}}{=} \bigwedge_{a\in Act}\boxed{a}\phi$	after all transitions ϕ holds
$\langle a\rangle^*$	$\stackrel{\text{def}}{=} lfp_X.\langle a\rangle\top \vee \langle\cdot\rangle X$	it is possible to reach a state which has an a-transition
$\overleftarrow{\langle a\rangle}^*$	$\stackrel{\text{def}}{=} lfp_X.\overleftarrow{\langle a\rangle}\top \vee \overleftarrow{\langle\cdot\rangle}X$	there has been an a-transition in the past
$\boxed{\cdot}^*\phi$	$\stackrel{\text{def}}{=} lfp_X.\phi \wedge \boxed{\cdot}X$	ϕ holds now and at all points in all the possible futures
${}_I\mathcal{CB}_p^1\phi$	$\stackrel{\text{def}}{=} gfp_X(\bigwedge_{i\in I}{}_i\mathcal{B}_p^1(X\wedge\phi))$	ϕ is common belief among the agents in I

Table 4.2: Some derived operators in DμCEC.

$$
\begin{array}{rcl}
Master & \stackrel{\text{def}}{=} & (\overline{m}_0\langle 0\rangle \,.\, \overline{m}_1\langle 0\rangle \,.\, \overline{m}_2\langle 0\rangle) \oplus_p (\sum_0^2 p_i \overline{m}_{0+i}\langle 1\rangle \,.\, \overline{m}_{1+i}\langle 0\rangle \,.\, \overline{m}_{2+i}\langle 0\rangle) \\
Crypt_i & \stackrel{\text{def}}{=} & c_{i,i}(x_0) \,.\, c_{i,i-1}(x_1) \,.\, m_i(z_i) \,.\, \overline{pay}_i\langle z_i\rangle \,.\, \overline{out}_i\langle x_0 + x_1 + z_i\rangle \\
Coin_h & \stackrel{\text{def}}{=} & (\overline{c}_{h,h}\langle 0\rangle \,.\, \overline{c}_{h+1,h}\langle 0\rangle) \oplus_{p_h} (\overline{c}_{h,h}\langle 1\rangle \,.\, \overline{c}_{h+1,h}\langle 1\rangle) \\
Collect & \stackrel{\text{def}}{=} & out_0(y_0) \,.\, out_1(y_1) \,.\, out_2(y_2) \,.\, \overline{outall}\langle y_0, y_1, y_2\rangle \\
DC & \stackrel{\text{def}}{=} & (\nu \vec{c})(\nu \vec{m})(\nu \vec{out})(Master \mid \prod_i Crypt_i \mid \prod_h Coin_h \mid Collect)
\end{array}
$$

Table 4.3: The dining cryptographers protocol formalized in CCS$_p$ (addition and subtraction in the indices is performed modulo 3).

The protocol can now be described as the parallel composition of the coin processes $Coin_h$, $h \in \{0,1,2\}$, the cryptographer processes $Crypt_i$, $i \in \{0,1,2\}$, the master process $Master$, and a process $Collect$ whose purpose is again to avoid the problem of the omniscient scheduler (see discussion in Paragraph 3.7).

The CCS$_p$ terms expressing the protocol are given in Table 4.3.In this representation, the secret actions are $\overline{pay}_i\langle z_i\rangle$, and the observable actions is $\overline{outall}\langle y_0, y_1, y_2\rangle$. The constants p and p_i's represent the probability that the master pays, and the probability that cryptographer i pays, respectively. Note that we have the constraint $p + \sum_i p_i = 1$.

In the following we model the property of strong anonymity with respect to *external agents*.[1] We assume that, for every external agent i, the actions $\overline{pay}_j\langle b_j\rangle$ and $\overline{pay}_{j'}\langle b_{j'}\rangle$ are indistinguishable for i, namely for each agent j, j' and bit b_j, $b_{j'}$

$$\mathcal{V}_i(\overline{pay}_j\langle b_j\rangle) = \mathcal{V}_i(\overline{pay}_{j'}\langle b_{j'}\rangle)$$

i.e. the *view* that i has of $\overline{pay}_j\langle 0\rangle$ is the same as of $\overline{pay}_j\langle 1\rangle$, $\overline{pay}_{j'}\langle 0\rangle$ and $\overline{pay}_{j'}\langle 1\rangle$.

Strong anonymity can be expressed by the following class of formulas, where p is an arbitrary number in $[0,1]$, j stands for $\overline{pay}_j\langle 1\rangle$ with $j \neq i$, and the conjunction is taken over all external agents i:

$$\bigwedge_i \; \boxdot^*({}_i\mathcal{B}_p^p(\overleftarrow{\diamondsuit_j}^*) \rightarrow \boxdot^*{}_i\mathcal{B}_p^p(\overleftarrow{\diamondsuit_j}^*)) \tag{4.1}$$

Intuitively, this formula means that at every point of the execution, if Agent i attributes probability p to j (i.e. to Cryptographer j being the payer), then at every point in the future he will attribute to j the same probability. In other words, the observable events of the protocol do not help the agent to refine his estimation of the probability distribution on the secrets. This definition of strong anonymity corresponds to the one given originally by Chaum [Cha88], requiring the *a priori* probability of a secret event a to be equal to its *a posteriori* probability after an observation o, i.e. $p(a|o) = p(a)$.

It is possible to show that, if the coins are fair, the program illustrated in Table 4.3 satisfies the formula 4.1.

Proposition 4.3.1.

$$DC \models \bigwedge_i \; \boxdot^*({}_i\mathcal{B}_p^p(\overleftarrow{\diamondsuit_j}^*) \rightarrow \boxdot^*{}_i\mathcal{B}_p^p(\overleftarrow{\diamondsuit_j}^*))$$

[1] In order to model anonymity also with respect to internal agents we need quantification over probabilities. This is left as future work.

Proof. The proof proceeds by induction, by proving that, for every i, ${}_i\mathcal{B}^p_p(\overleftarrow{\langle\!\langle j \rangle\!\rangle^*}) \rightarrow \Box^* {}_i\mathcal{B}^p_p(\overleftarrow{\langle\!\langle j \rangle\!\rangle^*})$ is an invariant which holds at every step of the execution. □

The strong anonymity of the Dining Cryptographers with fair coins was also proved in [BP05]. One major difference with respect to that work is that here we use belief operators, which allows us to express the belief of a given agent. As a consequence, we can distinguish between the belief of internal agents and external ones. In [BP05] strong anonymity is expressed in terms of equality of the *likelihoods* of an observable o, that is the conditional probabilities of o given different culprits a and a', i.e. $p(o|a) = p(o|a')$. However, the relation between agents and observables is not formalized with this approach. An internal agent, for instance, observes more than an external one because he can see also the results of the adjacent coins. This is not a problem in the case of a complete ring where there is a direct link between all pairs of cryptographers (i.e. a fully connected network), which is indeed the case in our example. But if the ring were incomplete (i.e. missing at least one arc) then there would be a difference between external and internal agents, in the sense that strong anonymity would only hold for external agents, not for internal ones. With the approach in [BP05] we would not be able to express this difference formally. This is also related to the fact that an approach based simply on probabilities cannot distinguish between subjective uncertainty (belief) and objective uncertainty (truth), as already mentioned in the introduction.

4.4 Application: Oblivious Transfer

An oblivious transfer is a protocol by which an initiator sends some information to a responder, but remains oblivious (ignorant) as to what was recovered by the responder.

In this section, two variations of the oblivious transfer protocol are considered and specified in DμCEC. For each of them, we give the expression of the agents' post-belief holding after the execution of the protocol, and we give a specification in CCS$_p$ of an implementation for the second one.

4.4.1 Oblivious Transfer of one bit only

4.4.1.1 Description

The *Oblivious-Transfer-of-one-bit-only* protocol, OT_b, was first described in [Kil88]. In this protocol, a single secret bit b is transferred between the initiator (e.g. Alice) and the responder (e.g. Bob). At the end of the protocol, one of the following two events will have occurred, each with a probability $\frac{1}{2}$:

1. the responder Bob learns the value of b, or
2. the responder Bob gains no information about the value of b.

In both cases, at the end of the protocol, Bob knows which of these two events has occurred, while the initiator Alice learns nothing about that.

4.4.1.2 Specification

We express the communication between the agents with two actions s and r defined as follows:

$s \stackrel{\text{def}}{=} Send(Alice, b, Bob)$: Alice sends bit b to Bob

$r \stackrel{\text{def}}{=} Receive(Bob, b)$: Bob receives bit b

The OT_b protocol can be specified as follows:

$$OT_b \stackrel{\text{def}}{=} \langle s \rangle P_{1/2}^{1/2}(\langle r \rangle^*)$$

Intuitively, this formula means that after the bit was sent by Alice, there is a probability of $\frac{1}{2}$ that Bob eventually receives it.

The post-belief of the agents after the execution of the protocol can be expressed as:

$$PostBelief_b \stackrel{\text{def}}{=} \boxdot^*(\rho(r, s))$$

where

$$\rho(\alpha, \beta) \stackrel{\text{def}}{=} {}_{Alice}\mathcal{B}_{1/2}^{1/2}\overleftarrow{\langle \alpha \rangle}^* \wedge P_{1/2}^{1/2}({}_{Bob}\mathcal{K}\overleftarrow{\langle \beta \rangle}^*) \tag{4.2}$$

and ${}_a\mathcal{K}\phi \stackrel{\text{def}}{=} {}_a\mathcal{B}_1^1\phi$

This formula can be read as follows: Alice believes with degree of confidence $\frac{1}{2}$ that Bob has received the bit (subjective probability), while, with probability $\frac{1}{2}$, Bob knows the bit that Alice has sent (objective probability).

Note that the fact that Bob knows that a formula ϕ holds (${}_{Bob}\mathcal{K}\phi$) is expressed as the limit of belief, i.e. ${}_{Bob}\mathcal{B}_1^1\phi$.

4.4.2 The 1-out-of-2 Oblivious Transfer

4.4.2.1 Description

In the *1-out-of-2-Oblivious-Transfer* protocol, OT_2^1 [Kil88], the initiator Alice sends two secret strings u and v, of which the responder Bob receives exactly one. At the end of the protocol, the following three states of affairs hold:

1. Bob learns one of the two strings,
2. Bob gains no information about the other string, and (4.3)
3. Alice does not know which one of the two strings Bob knows.

4.4.2.2 Specification

In the following, with a slight abuse of notation we use the symbols u and v to represent the actions of *sending* the messages u and v respectively. Analogously we represent by $\underline{u}$ and $\underline{v}$ the complementary actions of *retrieving* u and v.

We now express the OT_2^1 protocol and the agents' post-beliefs in terms of DμCEC formulae.

The first requirement is that, after Alice sends the two strings, there is a probability $\frac{1}{2}$ that Bob retrieves u, and a probability $\frac{1}{2}$ that Bob retrieves v:

$$OT_2^1 \stackrel{\text{def}}{=} \langle u \rangle^* \langle v \rangle^* (P_{1/2}^{1/2}(\underline{u}) \wedge P_{1/2}^{1/2}(\underline{v}))$$

Secondly, we require that, after the execution of the protocol, Alice believes with degree of confidence $\frac{1}{2}$ that Bob has received the message u and with degree of confidence $\frac{1}{2}$ that Bob has received the message v (subjective probability), while Bob with probability $\frac{1}{2}$ knows the message that Alice has sent (objective probability):

$$PostBelief_1 \stackrel{\text{def}}{=} \boxdot^*(\rho(\underline{u}, v) \wedge \rho(\underline{v}, u))$$

where $\rho(\alpha, \beta)$ is defined in Equation 4.2.

Finally, we require that if Bob receives u, then he gains no further information about v, and viceversa if he receives v, then he gains no further information about u. This can be expressed with an invariant, like for the Dining Cryptographers:

$$PostBelief_2 \stackrel{\text{def}}{=} (\forall p \varrho_p(v, u)) \wedge (\forall q \varrho_q(u, v))$$

where

$$\varrho_p(\alpha, \beta) \stackrel{\text{def}}{=} {}_{Bob}\mathcal{B}^p_p(\overleftarrow{\diamondsuit^*}) \rightarrow (\boxdot^* {}_{Bob}\mathcal{K}(\overleftarrow{\diamondsuit^*}) \rightarrow {}_{Bob}\mathcal{B}^p_p(\overleftarrow{\diamondsuit^*}))$$

4.4.2.3 Implementation of the OT^1_2 protocol using a public-key cryptosystem

We consider here the implementation of the oblivious transfer OT^1_2 described in [EGL85]. In the following, $\mathcal{M}$ represents the message space and we assume that all the random choices of messages or bits are made with a uniform probability.

Let $\boxplus, \boxminus : \mathcal{M} \times \mathcal{M} \rightarrow \mathcal{M}$ denote two binary operators which satisfy the following:

1. For every $x \in \mathcal{M}$, the mapping $y \mapsto x \boxplus y$ is a permutation on $\mathcal{M}$.
2. For every $y \in \mathcal{M}$, the mapping $x \mapsto x \boxplus y$ is a permutation on $\mathcal{M}$.
3. For every $x, y \in \mathcal{M}$, $(x \boxplus y) \boxminus y = x$.

Furthermore, we assume that these operators are known by both agents. For instance, when using RSA as public-key cryptosystem, $x \boxplus y$ can be defined as the reduction modulo N (the RSA's modulus) of $x + y$ while $x \boxminus y$ can be defined as the reduction modulo N of $x - y$.

In our process calculus, the OT^1_2 protocol can be specified as the parallel composition of the initiator process $Init$ and of the responder process $Resp$. The initiator Alice wants to send one of the two strings u and v. She starts the communication by generating a public key/private key pair (e, d) and sending her public key along with two random messages m_0 and m_1 to the responder Bob. Bob chooses a random message m and a random bit r and sends back to Alice $z = E(m, e) \boxplus m_r$, where $E(m, e)$ denotes the encryption of the message m with the public key e. Similarly, $D(c, d)$ denotes the decryption of a string c with the private key d and we have $D(E(m, e), d) = m$.

Alice (who does not know r) computes both $e_0 = z \boxminus m_0$ and $e_1 = z \boxminus m_1$. Then, Alice decrypts with her private key d both e_0 and e_1, obtaining respectively d_0 and d_1. Only one of these two values, namely $d_r = D(E(m, e), d)$, is identical to the initial message m. This however cannot be determined by Alice since she does not know the value of r and m.

Alice chooses then a random bit s and transmits to Bob the tuple $(u \boxplus d_s, v \boxplus d_{1-s}, s)$. Depending on the choice of s, two independent situations may occur: either $s = r$, and thus $d_s = d_r = m$ and Bob can read u (by performing the operation $(u \boxplus d_s) \boxminus m$)

$$
\begin{array}{lcl}
\mathit{Init} & \stackrel{\text{def}}{=} & \sum\!\!\!\!\!\!\diamond_{m_0,m_1}\, p\ \overline{out}_1\langle e, m_0, m_1\rangle\,.\,in(z)\,. \\
 & & \quad \sum\!\!\!\!\!\!\diamond_{s\in\{0,1\}}\, 1/2\ \overline{out}_2\langle u \boxplus D(z \boxminus m_s, d), v \boxplus D(z \boxminus m_{1-s}, d), s\rangle\,.\,0 \\
\mathit{Resp} & \stackrel{\text{def}}{=} & out_1(e, x_0, x_1)\,.\sum\!\!\!\!\!\!\diamond_{r\in\{0,1\}}\, 1/2 \sum\!\!\!\!\!\!\diamond_{m}\, q\ \overline{in}\langle E(m, e) \boxplus x_r\rangle\,. \\
 & & \quad out_2(y_0, y_1, s)\,.\,\overline{out}\langle y_{s+r} \boxminus m\rangle\,.\,0 \\
POT_2^1 & \stackrel{\text{def}}{=} & \nu(\mathit{Init}|\mathit{Resp})
\end{array}
$$

Table 4.4: Implementation of the protocol OT_2^1 in CCS$_p$. The probabilities p and q represent the uniform probabilities over the space of message pairs (m_0, m_1) and the space of the messages m, respectively.

without learning anything about v, or $s = 1 - r$ and Bob can read v (by performing the operation $(v \boxplus d_{1-s}) \boxminus m$) without learning anything about u. Both events have equal probability to occur (due to the uniform probability on the random choice of r and s), which ensures that the first and second intended properties of the protocol (corresponding respectively to the first and second sentences in statements 4.3) are satisfied.

Moreover, since Alice only gets the information $z = E(m, e) \boxplus m_r$ and m is randomly chosen by Bob, z does not give Alice any information about r, which ensures that the third intended property of the protocol (corresponding to the third sentence in statements 4.3) is satisfied as well.

The protocol narration of OT_2^1 is as follows:

$$
\begin{array}{lllll}
1. & \mathit{Alice} & \xrightarrow{e, m_0, m_1} & & \mathit{Bob} \\
2. & \mathit{Bob} & \xrightarrow{E(m,e) \boxplus m_r} & & \mathit{Alice} \\
3. & \mathit{Alice} & \xrightarrow{u \boxplus d_s, v \boxplus d_{1-s}, s} & & \mathit{Bob}
\end{array}
$$

We describe POT_2^1, the implementation of the protocol OT_2^1 in CCS$_p$ in Table 4.4.

The unfolding of the CCS$_p$ terms representing the protocol OT_2^1 is illustrated in Table 4.5.

4.4.2.4 Verification

In this section we show that our protocol satisfies OT_2^1, $\mathit{PostBelief}_1$ and $\mathit{PostBelief}_2$.

The initial prefix in the formula for OT_2^1 specifies that eventually, the actions u and v occur (i.e. the messages u and v are sent): $POT_2^{1(5,6)} \models OT_2^1$, where the tuple $(5, 6)$ in the exponent represents the final state. This is indeed achieved in our protocol by the (synchronous) action out_2 performed in step $POT_2^{1(4,4)}$. The remaining part of the formula OT_2^1 is true if u and v are received each with a probability of exactly one half, which holds as explained beforehand in the protocol description.

On the contrary to OT_2^1, the prefixes of $\mathit{PostBelief}_1$ and $\mathit{PostBelief}_2$ are used to

Unfolding of the protocol OT_2^1. **Initiator**

$$
\begin{aligned}
Init^0 &:= \textstyle\sum_{m_0,m_1} p\ \overline{out}_1\langle e, m_0, m_1\rangle \,.\, in(z)\,. \\
&\qquad \textstyle\sum_{s\in\{0,1\}} 1/2\ \ \overline{out}_2\langle u \boxplus D(z \boxminus m_s, d), v \boxplus D(z \boxminus m_{1-s}, d), s\rangle \,.\, 0 \\
Init^1 &:= \overline{out}_1\langle e, m_0, m_1\rangle \,.\, in(z)\,. \\
&\qquad \textstyle\sum_{s\in\{0,1\}} 1/2\ \ \overline{out}_2\langle u \boxplus D(z \boxminus m_s, d), v \boxplus D(z \boxminus m_{1-s}, d), s\rangle \,.\, 0 \\
Init^2 &:= in(z)\,. \\
&\qquad \textstyle\sum_{s\in\{0,1\}} 1/2\ \overline{out}_2\langle u \boxplus D(z \boxminus m_s, d), v \boxplus D(z \boxminus m_{1-s}, d), s\rangle \,.\, 0 \\
Init^3 &:= \textstyle\sum_{s\in\{0,1\}} 1/2\ \overline{out}_2\langle u \boxplus D(z \boxminus m_s, d), v \boxplus D(z \boxminus m_{1-s}, d), s\rangle \,.\, 0 \\
Init^4 &:= \overline{out}_2\langle u \boxplus D(z \boxminus m_s, d), v \boxplus D(z \boxminus m_{1-s}, d), s\rangle \,.\, 0 \\
Init^5 &:= 0
\end{aligned}
$$

Responder

$$
\begin{aligned}
Resp^0 &:= out_1(e, x_0, x_1)\,.\textstyle\sum_{r\in\{0,1\}} 1/2 \sum_m q\ \overline{in}\langle E(m, e) \boxplus x_r\rangle\,. \\
&\qquad out_2(y_0, y_1, s)\,.\,\overline{out}\langle y_{s+r} \boxminus m\rangle\,.\,0 \\
Resp^1 &:= \textstyle\sum_{r\in\{0,1\}} 1/2 \sum_m q\ \overline{in}\langle E(m, e) \boxplus x_r\rangle\,. \\
&\qquad out_2(y_0, y_1, s)\,.\,\overline{out}\langle y_{s+r} \boxminus m\rangle\,.\,0 \\
Resp^2 &:= \textstyle\sum_m q\ \overline{in}\langle E(m, e) \boxplus x_r\rangle\,.\,out_2(y_0, y_1, s)\,.\,\overline{out}\langle y_{s+r} \boxminus m\rangle\,.\,0 \\
Resp^3 &:= \overline{in}\langle E(m, e) \boxplus x_r\rangle\,.\,out_2(y_0, y_1, s)\,.\,\overline{out}\langle y_{s+r} \boxminus m\rangle\,.\,0 \\
Resp^4 &:= out_2(y_0, y_1, s)\,.\,\overline{out}\langle y_{s+r} \boxminus m\rangle\,.\,0 \\
Resp^5 &:= \overline{out}\langle y_{s+r} \boxminus m\rangle\,.\,0 \\
Resp^6 &:= 0
\end{aligned}
$$

Protocol

$$POT_2^{1(i,j)} \stackrel{\text{def}}{=} \nu(Init^i | Resp^j)$$

Unfolding:

$$
\begin{aligned}
&POT_2^{1(0,0)} \xrightarrow{m_0,m_1} POT_2^{1(1,0)} \xrightarrow{\tau(out_1)} POT_2^{1(2,1)} \xrightarrow{r} POT_2^{1(2,2)} \xrightarrow{q} \\
&POT_2^{1(2,3)} \xrightarrow{\tau(in)} POT_2^{1(3,4)} \xrightarrow{s} POT_2^{1(4,4)} \xrightarrow{\tau(out_2)} POT_2^{1(5,5)} \xrightarrow{out} POT_2^{1(5,6)}
\end{aligned}
$$

Table 4.5: Unfolding of the protocol OT_2^1.

describe invariant properties that have therefore to hold at every step of the protocol:

$$\forall (i,j) \in \{(0,0),(1,0),(2,1),(2,2),(2,3),(3,4),(4,4),(5,5),(5,6)\}$$
$$POT_2^{1(i,j)} \models PostBelief_1 \wedge PostBelief_2$$

The first part of $PostBelief_1$, namely ${}_{Alice}\mathcal{B}_{1/2}^{1/2}(\overleftarrow{\Diamond}^*) \wedge {}_{Alice}\mathcal{B}_{1/2}^{1/2}(\overleftarrow{\Diamond}^*)$ describes the subjective knowledge of Alice and is the transcription of the third axiom in statements 4.3, while the remaining of the formula specifies an objective knowledge of Bob and corresponds to the first axiom in statements 4.3. Similarly, $PostBelief_2$ is the transcription of the second axiom in statements 4.3. We already saw that these axioms, and thus $PostBelief_1$ and $PostBelief_2$ hold at the end of the protocol. They also hold at the beginning of the protocol. From the description of the protocol, one can finally see that no step leads to a change of these beliefs. Therefore, $PostBelief_1$ and $PostBelief_2$ hold at each step of the protocol.

Note that for the sake of simplicity, several aspects of our description which were not directly necessary for our purposes, such as the cryptographic primitives or the fixpoint operators, have been left informal.

4.5 Conclusion and Future Work

In this chapter, we have achieved novel formalizations of probabilistic anonymity and oblivious transfer in a new modal logic, namely the doxastic μ-calculus with error control (DμCEC). Our formalizations can be validated on the protocol of the dining cryptographers, and on the protocols of 1-bit and 1-out-of-2-strings oblivious transfer. The intuitiveness of our formalizations is due, first, to our distinction between belief and internalized probabilistic truth, but also to the dynamicity of these notions, and finally to the introduction of lower and upper bounds (error control) therefore.

We have also shown that belief and internalized probabilistic truth satisfy a probabilistic analogue of standard KD45-belief, and that these notions can be flattened on certain, but different conditions.

As future work for DμCEC, we envisage the development of *tool-support*, its *axiomatization*, and the introduction of *cryptographic data types* and restricted *logical quantification* (over messages, including probability values).

Given the expressibility of oblivious transfer in DμCEC and the foundational power of oblivious transfer for modern cryptography [Kil88], we also believe that DμCEC can serve as a framework for comparing abstract cryptography based on Dolev-Yao message-passing and concrete cryptography based on bit-string message-passing, thus bringing a new approach to a problem that has received a lot of attention recently, see for intance the work of [CRZ07].

Chapter 5

Other notions based on Bayesian risk

5.1 Introduction

In the previous chapters we studied concepts from information theory that turned out to be quite convenient in developing quantitative theories for security problems such as secure information flow and anonymity. We considered in particular the notion of noisy channel to model protocols for information-hiding, where the input and the output of the channel represent respectively the information to be kept secret and the observable visible to an adversary. The noise of the channel is generated by the efforts of the protocol to hide the link between the secrets and the observable, often achieved by using randomized mechanisms.

Correspondingly, as explained in the introduction of this thesis, there have been various attempts to define the degree of leakage by using concepts based on Shannon entropy, notably the mutual information [ZB05, CHM05b, Mal07, MC08] and the related notion of capacity [MNS03, MNCM03, CPP08a].

In a recent work, however, Smith has shown that the concept of mutual information is not very suitable for modeling the information leakage in the situation in which the adversary attempts to guess the value of the secret in one single try [Smi07]. He shows an example of two programs in which the mutual information is about the same, but the probability of making the right guess, after having observed the output, is much higher in one program than in the other. In a subsequent paper [Smi09], Smith proposes to use a notion based on Rényi *min-entropy*.

The programs used by Smith in [Smi07] to motivate his new measure of leakage are the programs $P1$ and $P2$ given on Figure 5.1. The secret h is a uniformly distributed $8k$-bit integer with range $0 \leq h < 2^{8k}$ and $k \geq 2$. Initially, nothing is known about the $8k$-bit integer h, thus its Shannon entropy is $H(h) = 8k$. Note that since the programs are deterministic, $H(l|h) = 0$ and therefore the mutual information between the input and output is given by:

$$I(h;l) \quad = \quad H(h) - H(h|l) \quad = \quad H(l) - H(l|h) \quad = \quad H(l) \qquad (5.1)$$

Program $P1$ leaks the secret h entirely for $1/8$ of the values of h (i.e. whenever h is a multiple of 8), and leaks almost nothing otherwise (it only leaks the fact that h is not a multiple of 8). In the definition domain of h, there are 2^{8k-3} multiples of 8, i.e.

PROGRAM P1

1 **if** $h \mod 8 = 0$
2 **then** $l := h$
3 **else** $l := 1$

PROGRAM P2

1 $l := h \,\&\, 0^{7k-1}1^{k+1}$

Figure 5.1: When mutual information is a bad measure of leakage (example from [Smi07])

of the form $h = 8n$ with $0 \leq n < 2^{8k-3}$ so the Shannon entropy of l is:

$$\begin{aligned} H(l) &= -p(l=1)\log p(l=1) - 2^{8k-3}p(l=h)\log p(l=h) \\ &= -\tfrac{7}{8}\log\tfrac{7}{8} - 2^{8k-3}\tfrac{1}{2^{8k}}\log\tfrac{1}{2^{8k}} \\ &\approx 0.169 + k \end{aligned} \tag{5.2}$$

Program $P2$ always leaks the last $k+1$ bits of h, so $H(l) = k+1$.

Smith pointed out that the traditional entropy-based notion of leakage, which identifies the mutual information $I(h;l)$ with information leakage, fails here to provide a satisfactory notion of information leakage. According to Equation 5.1, we have namely $I(h;l) \approx k + 0.169$ for $P1$ and $I(h;l) = k+1$ for $P2$, which means that both programs leak about the same amount of information. More precisely, $P2$ leaks slightly more information than $P1$, meaning that $P1$ is safer in terms of anonymity than $P2$ with respect to this measure of leakage.

However, the probability for an attacker to guess h correctly in one try (i.e. the *vulnerability* of h [Smi07]) is about 1/8 with $P1$, whereas with $P2$ her probability of success is only $1/2^{7k-1}$. So if we consider the worst case scenario (i.e. the adversary guesses h in one try), the program $P2$ appears actually much safer than $P1$, which contradicts the result based on Shannon entropy.

Smith proposed therefore to use the aforementioned vulnerability as a measure of leakage, in order to capture effectively the intuitive notion of leakage highlighted in the example. More precisely, he uses *min-entropy* $H_\infty(h) = -\log V(h)$ instead of Shannon entropy and defines information leakage as the corresponding mutual information $I_\infty(h;l) = H_\infty(h) - H_\infty(h|l)$ where $H_\infty(h|l)$ is the *conditional min-entropy* as defined in Equation 2.2.

For $P1$ (resp. $P2$), this measure gives a leakage of $\log(2^{8k-3}+1) \approx 8k-3$ (resp. $k+1$), so $P2$ appears now much safer than $P1$, reflecting correctly our intuitive expectation.

Smith however noted that if we increase the amount of bits copied from h to l in $P2$, so that only the three last bits of the input remain unknown, then the vulnerability of $P2$ is equal to the vulnerability of $P1$ ($1/2^3 = 1/8$), although the nature of the threats are very different in the two programs: $P2$ gives a systematic leakage (of $8k-3$ bits) but never leaks h entirely, while $P1$ leaks either everything of h (with probability $1/8$) or nothing (with probability $7/8$).

We look at the problem from the point of view of the *probability of error*, and we propose to formalize the notion of leakage as the "difference" between the probability

of error *a priori* (before observing the output) and *a posteriori* (using the output to infer the input via the MAP rule described in Section 3.1). We argue that there are at least two natural ways of defining this difference: one, that we call *multiplicative*, corresponds to Smith's proposal. The other, which we call *additive*, is new.

In both cases, we show that it is possible to find the suprema, which is nice in that it allows us to consider the maximum leakage (i.e. the minimal level of security) with respect to our additive and multiplicative notions of leakage. However, Smith proved that the supremum of additive leakage is NP-complete (see the proof in the full version of the TACAS 2010 paper [AAP]), which means that the search for the supremum, while possible, may not be efficient. The maximum leakage is also interesting because it abstracts from the input distribution, which is usually unknown, or (in the case of anonymity) may depend on the set of users.

5.2 Mutual Information and Capacity

The examples given by Smith in [Smi09] and mentioned previously are deterministic, i.e. have the property that the input determines univocally the output. For such systems, it turns out that the issue observed for the mutual information (namely that two programs with different vulnerabilities have the same mutual information) does not arise in the case of the capacity. Surprisingly, indeed, Smith showed in [Smi09] that the Shannon capacity coincides with the "min-entropy capacity", i.e., the maximum (with respect to all input distributions) of the logarithm of the ratio between the a posteriori probability of making a right guess and the a priori one. Therefore, the difference of vulnerabilities between two programs (measured as the difference between their min-entropy capacities), will be systematically reflected in a difference between their Shannon capacities. We will come back on this point in the next section.

Unfortunately, this coincidence does not carry out to the more general case of probabilistic channels, and, worse yet, the notion of capacity suffers (in the general case) from the same problem as the mutual information. The following example illustrates the situation.

Example 5.2.1. *Consider the following channels:*

	y_1	y_2	y_3
x_1	$2/3$	$1/6$	$1/6$
x_2	$1/6$	$2/3$	$1/6$
x_3	$1/6$	$1/6$	$2/3$

Figure 5.2: Channel matrix A

Since A and B are symmetric channels, their capacities can be easily calculated using the formula [CPP08a]:

$$C = \log |\mathcal{O}| - H(\vec{r})$$

	y_1	y_2	y_3
x_1	$2/3$	$1/3$	0
x_2	0	$2/3$	$1/3$
x_3	$1/3$	0	$2/3$

Figure 5.3: Channel matrix B

where $|\mathcal{O}|$ is the cardinality of the observables and $H(\vec{r})$ is the entropy of a row of the matrix.

We have therefore

$$C(A) = \log 3 - (\frac{2}{3}\log\frac{3}{2} + \frac{1}{6}\log 6 + \frac{1}{6}\log 6) = \frac{1}{3}$$

and

$$C(B) = \log 3 - (\frac{2}{3}\log\frac{3}{2} + \frac{1}{3}\log 3) = \frac{2}{3}$$

However, under the uniform input distribution, the ratio R between the a posteriori and the a priori probability of making the right guess (see Section 5.3) is the same for both channels:

$$R(A) = R(B) = \frac{\sum_j \max_i p(y_j|x_i)p(x_i)}{\max_i p(x_i)} = \sum_j \max_i p(y_j|x_i) = 3\frac{2}{3} = 2$$

Therefore the capacity does not seem to correctly capture the notion of leakage for nondeterministic systems.

5.3 Towards a more suitable notion of leakage

In the following, we are interested in quantifying the *leakage* of a security protocol, i.e. the amount of information about the input that an adversary can learn by running the protocol and observing the resulting output.

5.3.1 Probabilities of a right guess

Before running the protocol, the probability that a given input x_i occurs depends only on the a priori distribution $\vec{\pi}$, and a rational adversary will therefore assume that the most probable input, called the *a priori probability of a right guess* $PR_i(\vec{\pi})$, will be the input having the maximum a priori probability, i.e.:

Definition 5.3.1. *The a priori probability of a right guess is defined as*

$$PR_i(\vec{\pi}) \quad = \quad \max_i \pi_i$$

After running the protocol and seeing the output, the adversary may revise his guess. An adversary applying the MAP rule, when observing output y_j, will choose as most probable input x_i the one for which the a posteriori probability $p(x_i|y_j)$ is the highest. The average of this value on all possible outputs gives the *a posteriori probability of a right guess* $PR_o(\vec{\pi})$, which is the complement of the Bayes risk.

Definition 5.3.2. *The a posteriori probability of a right guess is defined as*

$$PR_o(\vec{\pi}) \quad = \quad \sum_j \max_i (p(y_j|x_i)\pi_i)$$

In the rest of this chapter, we will consider only adversaries applying the MAP rule since this is the rule that gives the best result from the point of view of the adversary (see Section 3.1 for more details).

5.3.2 Leakage and uncertainty

Intuitively, the *leakage* is the amount of information learnt by the adversary by observing the output of the protocol. Following [Smi09], it seems natural to define it as the difference between the uncertainty about the input before observing the output, and the remaining uncertainty afterwards:

$$\text{Information Leaked} \quad = \quad \text{Initial Uncertainty} - \text{Remaining Uncertainty} \quad (5.3)$$

Now, the question is how to measure information, and (correspondingly) what do we actually mean by uncertainty. We consider here two possibilities. The first leads to a multiplicative notion of leakage, and it follows the proposal of Smith [Smi09]. The second leads to an additive notion, and it is new.

5.3.3 Multiplicative leakage

In relation to Equation (5.3), Smith [Smi09] measures the information in bits, and proposes to define the initial uncertainty as the *min-entropy* of X, $H_\infty(X)$, the instance of Rényi entropy [Rén60] obtained for $\alpha = \infty$ (see Paragraph 2.4). As for the remaining uncertainty, it would be natural to use the conditional *min-entropy* of X given Y. Unfortunately, as explained in Section 2.4, there is no agreement on what Rényi's generalization of Shannon's conditional entropy should be, even though there seem to be a consensus towards $\sum_y p(y) H_\alpha(X|Y = y)$ [Cac97]. Smith however proposes to use the definition of $H_\infty(X|Y)$ equivalent to the one given in [DORS08], which is

$$H_\infty(X|Y) = -\log PR_o(\vec{\pi})$$

In this way, Smith obtains a definition of leakage similar to the definition of mutual information, except that Shannon entropy is replaced by H_∞:

$$L(X;Y) \;=\; H_\infty(X) - H_\infty(X|Y) \;=\; \log \frac{PR_o(\vec{\pi})}{PR_i(\vec{\pi})}$$

We consider a similar definition for leakage, namely the ratio between $PR_o(\vec{\pi})$ and $PR_i(\vec{\pi})$, which coincides with Smith's notion apart from the absence of the logarithm. Furthermore, in general we want to abstract from the a priori distribution, and consider the worst case, hence we are particularly interested in the supremum of such ratio.

Definition 5.3.3. *We define the multiplicative leakage as*

$$\mathcal{L}_\times(\vec{\pi}) \;=\; \frac{PR_o(\vec{\pi})}{PR_i(\vec{\pi})}$$

Note that $PR_i(\vec{\pi}) > 0$ for every $\vec{\pi}$, hence $\mathcal{L}_\times(\vec{\pi})$ is always defined.
We will also use the notation $\mathcal{ML}_\times$ to represent the supremum of this quantity:

$$\mathcal{ML}_\times \;=\; \max_{\vec{\pi}}(\mathcal{L}_\times(\vec{\pi}))$$

5.3.4 Additive leakage

Another possible interpretation for Equation (5.3) is to consider the uncertainty as the probability of guessing the wrong input. The leakage then expresses how much the knowledge of the observable helps decreasing such probability. This leads to define the leakage as the difference between the probabilities of error before and after observing the output. As usual, we are particularly interested in the supremum of this difference.

Definition 5.3.4. *We define the additive leakage as*

$$\mathcal{L}_+(\vec{\pi}) \;=\; PR_o(\vec{\pi}) - PR_i(\vec{\pi})$$

We will also use the notation $\mathcal{ML}_+$ to represent the supremum of this quantity:

$$\mathcal{ML}_+ \;=\; \max_{\vec{\pi}}(\mathcal{L}_+(\vec{\pi}))$$

Proposition 5.3.5.

$$\forall \vec{\pi}, \mathcal{L}_+(\vec{\pi}) \geq 0$$

Proof

$$\begin{array}{rcll}
\mathcal{L}_+(\vec{\pi}) & = & PR_o(\vec{\pi}) - PR_i(\vec{\pi}) & \\
& = & \sum_j \max_i(p(y_j|x_i)\pi_i) - \max_i \pi_i & \\
& \geq & \sum_j p(y_j|x_{i_m})\pi_{i_m} - \pi_{i_m} & \text{where } \pi_{i_m} = \max_i \pi_i \\
& = & \sum_j p(y_j \wedge x_{i_m}) - \pi_{i_m} & \\
& = & \pi_{i_m} - \pi_{i_m} & \\
& = & 0 &
\end{array}$$

□

5.4 Properties of the mutiplicative leakage

In this section we consider the multiplicative leakage and we study its supremum. It turns out that the supremum is very easy to compute. In fact, it coincides with the value of the leakage in the point of uniform distribution, and it is equal to the sum of the maxima of the columns of the channel matrix corresponding to the protocol. This property was also discovered independently by Geoffrey Smith and Ziyuan Meng (personal communication).

Proposition 5.4.1.

$$\mathcal{ML}_\times = \mathcal{L}_\times(\vec{\pi_u}) = \sum_j \max_i p(y_j|x_i)$$

where $\vec{\pi_u}$ is the uniform distribution.

Proof

$$\begin{aligned}
\mathcal{L}_\times(\pi_1,\ldots,\pi_n) &= \tfrac{1}{\max_i \pi_i}\textstyle\sum_j \max_i(p(y_j|x_i)\pi_i)\\
&\leq \tfrac{1}{\max_i \pi_i}\textstyle\sum_j \max_i p(y_j|x_i)(\max_i \pi_i)\\
&= \textstyle\sum_j \max_i p(y_j|x_i)\\
&= n\textstyle\sum_j \max_i(p(y_j|x_i)\tfrac{1}{n})\\
&= \mathcal{L}_\times(\tfrac{1}{n},\ldots,\tfrac{1}{n})
\end{aligned}$$

Since this inequality holds for all input distributions $(\pi_1,\ldots,\pi_n)$, the leakage in the point of uniform distribution is the supremum of the multiplicative leakage (but other distributions may realize this supremum too).

□

5.5 Properties of the additive leakage

We turn now our attention to the additive leakage. We will see that the supremum is not always in the point of uniform distribution. However, we prove that it is in one of the corner points of PR_i. Since PR_i has a finite set of corner points, and their form is known, also the additive leakage is relatively easy to compute.

First we prove a general property concerning the relation between suprema, convexity, and corner points:

Proposition 5.5.1. *Consider two functions $f, g : D^{(n)} \to \mathbb{R}$ and suppose f has a set of corner points U, and g is convex. Define $h : D^{(n)} \to \mathbb{R}$ as $h = f + g$. If h has a maximum over U, then it has the same maximum over $D^{(n)}$.*

Proof
Let $\vec{u} \in U$ be such that $h(\vec{u})$ is maximum over U and let $\vec{w} \in D^{(n)}$ be arbitrary. Since $\vec{w} \in D^{(n)}$, there are elements $\vec{v}_1, \vec{v}_2, \ldots, \vec{v}_k$ in U and $c_1, c_2, \ldots, c_k$ in $\mathbb{R}$ with $\sum_i c_i = 1$ such that $\vec{w} = \sum_i c_i\vec{v}_i$ and $f(\vec{w}) = \sum_i c_i f(\vec{v}_i)$. Then

$$\begin{aligned}
h(\vec{w}) &= f(\vec{w}) + g(\vec{w})\\
&= f(\textstyle\sum_i c_i\vec{v}_i) + g(\sum_i c_i\vec{v}_i)\\
&= \textstyle\sum_i c_i f(\vec{v}_i) + g(\sum_i c_i\vec{v}_i)\\
&\leq \textstyle\sum_i c_i f(\vec{v}_i) + \sum_i c_i g(\vec{v}_i) && \text{since } g \text{ is convex}\\
&= \textstyle\sum_i c_i h(\vec{v}_i)\\
&\leq \textstyle\sum_i c_i h(\vec{u})\\
&= h(\vec{u}) && \text{since } \textstyle\sum c_i = 1
\end{aligned}$$

□

An example of Proposition 5.5.1 is illustrated in Figure 5.4.

We now show that $-PR_i$ and PR_o satisfy the hypotheses of Proposition 5.5.1. The necessary property for $-PR_i$ comes from a result in [CPP08b].

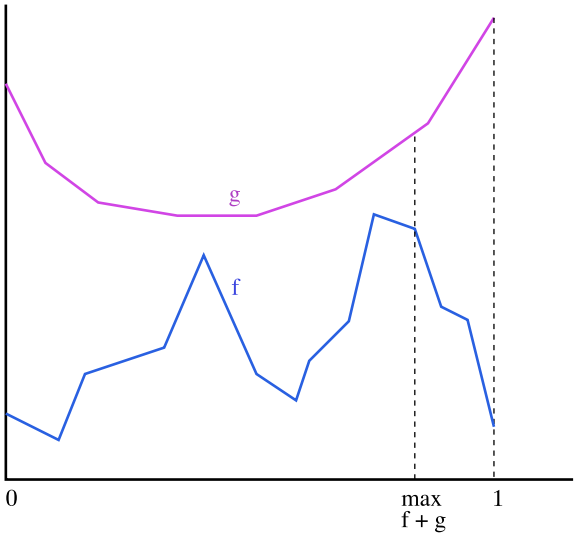

Figure 5.4: An illustration of Proposition 5.5.1

Proposition 5.5.2 ([CPP08b], Proposition 3.9)**.** *The function PR_i on $D^{(n)}$ is convexly generated by $(U, f(U))$ with $U = U_1 \cup U_2 \cup \ldots \cup U_n$ where, for each r, U_r is the set of all vectors that have value $1/r$ in exactly r components, and 0 everywhere else.*

Remark 5.5.3. *The cardinality $|U|$ of the set U is $2^n - 1$:*

$$\begin{aligned} |U| &= \textstyle\sum_{i=1}^{n} \binom{n}{i} \\ &= \textstyle\sum_{i=0}^{n} \binom{n}{i} - 1 \\ &= (1+1)^n - 1 \textit{ (Binôme de Newton)} \\ &= 2^n - 1 \end{aligned}$$

Remark 5.5.4. *The function $-PR_i$ has the same corner points as PR_i.*

We now prove that PR_o satisfies the necessary property.

Proposition 5.5.5. *PR_o is convex.*

Proof
Let $\vec{z}$ be the convex combination $\sum_i \lambda_i \vec{z_i}$ where the dimension of $\vec{z}, \vec{z_1}, \ldots, \vec{z_m}$ corresponds to the number of input variables and $\vec{z_1}, \ldots, \vec{z_m}$ is a set of corner points. The j^{th} component z_{k_j} of any corner point $\vec{z_k}$ corresponds to the input variable x_{k_j} chosen according to the MAP rule when the output variable y_j is obtained.

$$\begin{aligned}
PR_o(\textstyle\sum_i \lambda_i \vec{z}_i) &= \textstyle\sum_j \max_k \{p(y_j|x_k)(\sum_i \lambda_i \vec{z}_i)_k\} \\
&= \textstyle\sum_j p(y_j|x_{k_j})(\sum_i \lambda_i \vec{z}_i)_{k_j} \\
&= \textstyle\sum_j p(y_j|x_{k_j})(\sum_i \lambda_i z_{i,k_j}) \\
&= \textstyle\sum_j \sum_i \lambda_i p(y_j|x_{k_j}) z_{i,k_j} \\
&= \textstyle\sum_i \lambda_i \sum_j p(y_j|x_{k_j}) z_{i,k_j} \\
&\leq \textstyle\sum_i \lambda_i \sum_j \max_k p(y_j|x_k) z_{i,k} \\
&= \textstyle\sum_i \lambda_i PR_o(\vec{z}_i)
\end{aligned}$$

□

Corollary 5.5.6. *$\mathcal{ML}_+$ is reached on one of the corner points of PR_i.*

Proof
Since $-PR_i$ has a finite set of corner points U and PR_o is convex, $\mathcal{L}_+ = PR_o - PR_i$ has a maximum $\mathcal{ML}_+$ over U and Proposition 5.5.1 shows that this maximum is reached on a corner point of $-PR_i$, which correspond to the corner points of PR_i.
□

Remark 5.5.7. *In general $\mathcal{ML}_+$ is not reached on the point of uniform distribution.*

Example 5.5.8. *Consider the channel whose matrix is given in Figure 5.5.*

	y_1	y_2	y_3
x_1	1	0	0
x_2	0	$1-e$	e
x_3	0	$1-2e$	$2e$

Figure 5.5: Channel matrix ($e \in [0, 1/2]$)

The calculation of $\mathcal{L}_+$ on the distributions corresponding to the corner points gives:

Corner points	PR_o	PR_i	$\mathcal{L}_+$
$(1,0,0),(0,1,0),(0,0,1)$	1	1	0
$(1/2,1/2,0),(1/2,0,1/2)$	1	$1/2$	$1/2$
$(0,1/2,1/2)$	$(e+1)/2$	$1/2$	$e/2$
$(1/3,1/3,1/3)$	$(e+2)/3$	$1/3$	$(e+1)/3$

We have for every $e \in [0, 1/2[$,

$$0 = \mathcal{L}_+(1,0,0) \leq \mathcal{L}_+(0,1/2,1/2) < \mathcal{L}_+(1/3,1/3,1/3) < \mathcal{L}_+(1/2,1/2,0) = 1/2$$

and $\mathcal{L}_+(1/3,1/3,1/3) = \mathcal{L}_+(1/2,1/2,0) = 1/2$ for $e = 1/2$. Therefore if $e < 1/2$, $\mathcal{ML}_+ = 1/2$, reached on distributions that are different from the uniform distribution $(1/3,1/3,1/3)$.

Moreover, this remark holds also for symmetric matrices:

Remark 5.5.9. *Even in case of symmetric matrices, in general $\mathcal{ML}_+$ is not reached on the point of uniform distribution.*

Example 5.5.10. *Consider the channel whose matrix is given in Figure 5.6.*

	y_1	y_2	y_3	$\ldots$	y_{10}	y_{11}
x_1	0	1/10	1/10	$\ldots$	1/10	1/10
x_2	1/10	0	1/10	$\ldots$	1/10	1/10
$\ldots$	$\ldots$	$\ldots$	$\ldots$	$\ldots$	$\ldots$	$\ldots$
x_{10}	1/10	1/10	1/10	$\ldots$	0	1/10
x_{11}	1/10	1/10	1/10	$\ldots$	1/10	0

Figure 5.6:

Let $\vec{\pi} = (1/r, 1/r, \ldots, 1/r, 0, \ldots, 0)$ be the a priori distribution with an equal probability of $1/r$ for the r first inputs ($r \geq 2$). This distribution is a corner point of the matrix, and since the matrix is symmetric, any other corner point corresponding to a distribution containing r non-null probabilities of $1/r$ will give the same results for PR_i, PR_o and $\mathcal{L}_+$.

$$\begin{aligned} PR_i(\vec{\pi}) &= 1/r \\ PR_o(\vec{\pi}) &= \textstyle\sum_{1,\ldots,11}(1/r)(1/10) \\ &= 11/(10r) \\ \mathcal{L}_+(\vec{\pi}) &= PR_o(\vec{\pi}) - PR_i(\vec{\pi}) \\ &= 1/(10r) \end{aligned}$$

Therefore $\mathcal{ML}_+$ is reached when r has the smallest value, i.e. when $r = 2$. This corresponds to the distribution $(1/2, 1/2, 0, \ldots, 0)$ and gives $\mathcal{ML}_+ = 1/20$, while $\mathcal{L}_+(1/11, \ldots, 1/11) = 1/110$.

5.6 Comparison

In this section, we compare the two notions of leakage. We first compare them with respect to a specific distribution, and then we consider the comparison of their worst cases.

5.6.1 Comparison on a specific input distribution

If we consider a specific distribution, it comes out that the two notions are equivalent, in the sense that a program is better with respect to the additive notion if and only if it is better with respect to the multiplicative notion.

From Definitions 5.3.3 and 5.3.4, we can derive a direct relation between the additive and multiplicative leakages at a specific input distribution $\vec{\pi}$:

$$\mathcal{L}_+(\vec{\pi}) \quad = \quad PR_i(\vec{\pi})(\mathcal{L}_\times(\vec{\pi}) - 1) \tag{5.4}$$

Proposition 5.6.1. *Consider two programs P and P', and let $\mathcal{L}_+$ and ${\mathcal{L}_+}'$ be the additive measures of leakage for P and P', respectively. Analogously, let $\mathcal{L}_\times$ and ${\mathcal{L}_\times}'$ be the multiplicative measures of leakage for P and P', respectively. We have that, for every $\vec{\pi}$*

$$\mathcal{L}_+(\vec{\pi}) \ \leq \ {\mathcal{L}_+}'(\vec{\pi}) \ \Leftrightarrow \ \mathcal{L}_\times(\vec{\pi}) \ \leq \ {\mathcal{L}_\times}'(\vec{\pi})$$

Proof
Obvious from Relation 5.4 and from the fact that for all $\vec{\pi}$, $PR_i(\vec{\pi})$ is positive. □

5.6.2 Comparison of the worst cases

Another criterion of comparison is the worst case. Let us consider first two examples.

Example 5.6.2. *Let us consider a $2^k \times 2^k$ channel with the 2^k first natural numbers as inputs and outputs, i.e. $\mathcal{X} = \mathcal{Y} = \{0, \ldots, 2^k - 1\}$. Consider a random input variable X with values ranging in $\{0, 2^k - 1\}$.*

Consider the following program:

PROGRAM P(X)
1 ▷ *Input* X
2 ***if*** $X = 0$ *or* $X = 1$
3 ***then*** *Output* X
4 ***else*** *Output one of the values* $\{2, \ldots, 2^k - 1\}$ *chosen randomly according to the uniform distribution*

This program corresponds to a channel whose matrix is given in Figure 5.7.

Let us consider first $\mathcal{ML}_+$. Because of Corollary 5.5.6, we know that $\mathcal{ML}_+$ is reached on a corner point, i.e. a distribution of the form $(q_1, \ldots, q_{2^k})$ where each q_i is either 0 or $1/r$, and there are r elements with value $1/r$ in the distribution.

For every corner point $\vec{\pi}$ we have $PR_i(\vec{\pi}) = 1/r$, thus maximizing $\mathcal{L}_+$ for a given r is equivalent to maximizing PR_o. From the channel matrix, one can see that the maximum value of PR_o is reached on an input distribution where the two first elements are as high as possible.

Therefore, we can restrict our study to distributions of the form $(1/r, \ldots, 1/r, 0 \ldots, 0)$, i.e. distributions where the elements with value $1/r$ are the r first elements.

For $r = 1$, we have:

$$PR_i(1, 0, \ldots, 0) \quad = \quad 1$$

$$PR_o(1, 0, \ldots, 0) \quad = \quad 1$$

	0	1	2	...	2^k-1
0	1	0	0	...	0
1	0	1	0	...	0
2	0	0	p	...	p
...	...	...	...	...	...
2^k-1	0	0	p	...	p

Figure 5.7: Channel matrix ($p = 1/(2^k - 2)$)

Thus:

$$\mathcal{L}_+(1,0\ldots,0) = 0$$

$$\mathcal{L}_\times(1,0,\ldots,0) = 1$$

For $r = 2$, we have:

$$PR_i(1/2,1/2,0\ldots,0) = 1/2$$

$$PR_o(1/2,1/2,0,\ldots,0) = 1$$

Thus:

$$\mathcal{L}_+(1/2,1/2,0\ldots,0) = 1/2$$

$$\mathcal{L}_\times(1/2,1/2,0,\ldots,0) = 2$$

For $r \geq 3$, we have:

$$PR_i(1/r,1/r,1/r,0,\ldots,0) = 1/r$$

$$\begin{aligned} PR_o(1/r,1/r,1/r,0,\ldots,0) &= 1/r + 1/r + (2^k-2)(1/r)p \\ &= 3/r \end{aligned}$$

Thus:

$$\mathcal{L}_\times(1/r,1/r,1/r,0,\ldots,0) = 3$$

$$\mathcal{L}_+(1/r,1/r,1/r,0,\ldots,0) = 2/r$$

We observe that for $r \geq 3$, the value of $\mathcal{L}_+$ decreases when r increases. Since

$$\mathcal{L}_+(1/3,1/3,1/3,0,\ldots,0) = 2/3 > \mathcal{L}_+(1/2,1/2,0,\ldots,0)$$

$$= 1/2 > \mathcal{L}_+(1,0,\ldots,0) = 0$$

we have $\mathcal{ML}_+ = 2/3$ reached for $r = 3$.

In particular, $\mathcal{ML}_+ > \mathcal{L}_+(1/2^k,\ldots,1/2^k) = 1/2^{k-1}$ for all $k > 1$.

Concerning $\mathcal{L}_\times$, we have that, for $r \geq 3$, $\mathcal{L}_\times(1/r,1/r,1/r,0,\ldots,0) = 3 > \mathcal{L}_\times(1/2,1/2,0,\ldots,0) = 2 > \mathcal{L}_\times(1,0,\ldots,0) = 1$, thus $\mathcal{ML}_\times = 3$, reached on any distribution $(1/r,1/r,1/r,0,\ldots,0)$ with $r \geq 3$, and in particular on the uniform distribution, which confirms Proposition 5.4.1.

Example 5.6.3. *Let us consider the following program:*

PROGRAM P$'(X)$

1 $\triangleright$ *Input* X
2 *with probability* $3/2^k$ *Output* X
3 *with probability* $1 - 3/2^k$ *Output a value in* $\{0, 2^k - 1\}\backslash\{X\}$ *chosen randomly according to the uniform distribution*

This program corresponds to a channel whose matrix is given in Figure 5.8.

	0	1	2	...	$2^k - 1$
0	p_1	p_2	p_2	...	p_2
1	p_2	p_1	p_2	...	p_2
2	p_2	p_2	p_1	...	p_2
...	...	...	...	...	...
$2^k - 1$	p_2	p_2	p_2	...	p_1

Figure 5.8: Channel matrix ($p_1 = 3/2^k$ and $p_2 = (1 - (3/2^k))/(2^k - 1)$)

The symmetry of the matrix implies that we can restrict the study to the a priori distribution $\vec{\pi} = (1/r, 1/r, \ldots, 1/r, 0, \ldots, 0)$*, where the* r *elements with value* $1/r$ *are the first elements in the distribution.*

In this case, for $r \geq 1$*:*

$$PR_i(\vec{\pi}) = 1/r$$

$$\begin{aligned} PR_o(\vec{\pi}) &= r(p_1/r) + (2^k - r)(p_2/r) \\ &= p_1 + [(2^k/r) - 1]p_2 \end{aligned}$$

Finally:

$$\mathcal{L}_+{}'(\vec{\pi}) = p_1 - p_2 - \tfrac{2}{r(2^k-1)}$$

Thus $\mathcal{L}_+{}'$ *increases when* r *increases, and* $\mathcal{ML}_+{}' = 1/2^{k-1}$ *is reached for* $r = 2^k$ *(on the uniform distribution).*

$$\begin{aligned} \mathcal{L}_\times{}'(\vec{\pi}) &= rp_1 + (2^k - r)p_2 \\ &= r(p_1 - p_2) + 2^k p_2 \end{aligned}$$

Since $p_1 > p_2$*,* $\mathcal{L}_\times{}'$ *increases when* r *increases, and thus* $\mathcal{ML}_\times{}' = 3$ *is obtained for* $r = 2^k$ *(on the uniform distribution, which confirms Proposition 5.4.1).*

The programs P *and* P' *have therefore the same maximum multiplicative leakage* $\mathcal{ML}_\times = \mathcal{ML}_\times{}' = 3$*, but the maximum additive leakage is equal to* $2/3$ *for* P *and equal to* $1/2^{k-1}$ *for* P'*.*

The limitation of the multiplicative leakage highlighted with the previous examples is due to the fact that two different channel matrices lead to the same value of

$\mathcal{ML}_\times$ (reached for both protocols in the uniform input distribution) as soon as the sums of the maxima of their columns are the same. This can happen for two matrices with very different shapes, as highlighted with the previous examples. These differences may be further interpreted as differences in the levels of anonymity of both protocols, which would then not be appropriately captured by $\mathcal{ML}_\times$. In such cases, $\mathcal{ML}_+$ may be more discriminative than $\mathcal{ML}_\times$ and allow to distinguish between the two levels of anonymity, as was the case in the two previous examples.

We would like to investigate in the future whether this property holds in general, or if we can find also for the additive leakage two programs with the same value of $\mathcal{ML}_+$ (and same value of $\mathcal{ML}_\times$) but obvious differences in their degrees of anonymity.

Finally, we can also notice that in some cases $\mathcal{ML}_\times$ and $\mathcal{ML}_+$ may even give opposite results, as illustrated in the previous example by taking $p_1 = 4/2^k = 1/2^{k-2}$ and $p_2 = (1 - (1/2^{k-2}))/(2^k - 1)$ for the probabilities in the channel matrix of Program P' (given in Figure 5.8). In this case, we obtain for the maximum multiplicative leakage:

$$\mathcal{ML}_\times{}' \;=\; 4 \;>\; 3 \;=\; \mathcal{ML}_\times$$

and for the maximum additive leakage:

$$\mathcal{ML}_+{}' \;=\; \tfrac{3.2^{k-1}-1}{2^{k-1}(2^k-1)} \;<\; \tfrac{2}{3} \;=\; \mathcal{ML}_+$$

for all $k \geq 3$, which shows even more clearly the non-equivalence of the two different measures of leakage.

5.7 Conclusion

We have considered two notions of leakage related to the Bayes risk. One of them, which we call multiplicative, corresponds to the notion recently proposed by Smith based on Renyi min-entropy. The other, which we call additive, is new. We have shown that the two notions are equivalent in all distributions. If we consider the distributions that give the worst case for the leakage, however, then the two notions are different. In particular, the multiplicative one has the worst case always in correspondence of the uniform distribution, while this is not the case for the additive one. So we can consider the new notion as a criterion, in addition to the one of Smith, to help assessing the degree of protection offered by a protocol or a program.

Chapter 6

Conclusion

In this thesis, we considered different approaches for quantifying information hiding in communication protocols. First, we investigated a process-calculus approach and used a probabilistic extension of the language CCS to specify protocols and study their compositional safety, i.e. how constructs of the language may (or may not) be composed while preserving the security guarantees of the protocol. We showed that some of the constructs, such as secret choice or unrestricted parallel composition, do not preserve safety and that this property mainly comes from the unconstrained scheduler we used and whose behaviour may leak some information to the adversary.

Using process calculus was a rather natural approach to reason in the first hand on our information-hiding systems, whose behaviour can be modeled very conveniently as probabilistic automata. However, this formalism was unable to capture useful dynamic and temporal properties as well as subtle variations in the belief of agents using the communication protocols under concern. This motivated the use of modal logic which has a long history of successful applications in the field of verification of security properties in computer programs. For our purpose we specified a new logic which, as opposed to the existing ones, does combine both dynamicity of belief and error control. We were then able to express with accuracy e.g. *how much* an agent believes in a certain fact or a given property holds.

These two approaches provided us with a complementary, *dual* strategy to analyze the security of information-hiding protocols such as information flow or anonymity systems. We have been able to quantify efficiently the vulnerability of communication protocols by using the Bayes risk of a potential adversary. We investigated further this approach in Chapter 5 of this thesis, following a recent work of Smith [Smi09], by comparing two different ways of measuring the difference between the a posteriori and the a priori vulnerability of a protocol, interpreted as its degree of security. These two methods (multiplicative and additive) actually proved to be complementary and thus useful to evaluate the vulnerability of a wider spectrum of security protocols.

In order to illustrate our results, we used as running example the well-known anonymity protocol of the Dining Cryptographers and we believe that our results may already contribute to assess effectively the security of various information-hiding systems in use today.

We already outlined in each chapter several open questions that may be worth considering. In addition to these, we would be interested in the future in implemeting a probabilistic model checker able to automatically evaluate the security of various information-hiding systems. This idea was already investigated by Norman,

Palamidessi, Parker and Wu, who presented an implementation of model checking for probabilistic and stochastic extensions of the $\pi-$calculus [NPPW09]. The resulting compiler was actually split into two parts: translation from probabilistic $\pi-$calculus to the Probabilistic Symbolic Transition Graph (PSTG) and traduction from PSTG to PRISM. While the first part has already been achieved [MMC], the second step has not been completed yet, which significantly limits the applicability of the tool in its present form. We would like therefore to pursue the implementation of this automatic checker, and possibly add features related to our results, such as the quantification of the security guaranteed by the protocols given as input to the checker.

Another possible future line of research is the extension of our framework to quantum anonymity protocols, which were recently proposed in the context of *quantum information theory*, the generalization of information theory to the quantum world [CW05, HZBB05, VSC07]. These papers demonstrate that quantum information processing provides resources allowing anonymous communication of classical data with security features not achievable by classical cryptography. In particular, most of the quantum anonymity protocols rely on the existence of an *entangled state* shared between the participants, i.e. a global state identical to all of them but which may be modified locally by the sender depending e.g. on the value of the message which has to be anonymously transmitted. Devetak defined in 2003 the notion of *private classical capacity of a quantum channel* to quantify the amount of (classical) secret information that could be reliably transmitted over a quantum channel [Dev03]. In the same line of research, the security of the communication of quantum information (i.e. where messages are qubits rather than classical bits) was also investigated and it was shown that an information-theoretically secure anonymous transmission could be achieved [BBF+07]. We would be particularly interested in following these approaches to extend our results to the quantum world, by e.g. studying whether the use of a quantum channel may increase the safety (as defined in Section 3.5) of information-hiding protocols for classical data. Additionally, we would also like to consider the transmission of quantum information and study strategies for quantifying this communication.

Bibliography

[AAP] Mário Alvim, Miguel Andrés, and Catuscia Palamidessi. Information Flow in Interactive Systems. Technical report.

[ABG04] Alessandro Aldini, Mario Bravetti, and Roberto Gorrieri. A process-algebraic approach for the analysis of probabilistic noninterference. *J. Comput. Secur.*, 12(2):191–245, 2004.

[AG97] Martn Abadi and Andrew D. Gordon. Reasoning about cryptographic protocols in the spi calculus. In *In CONCUR'97: Concurrency Theory*, pages 59–73. Springer-Verlag, 1997.

[AG99] Martín Abadi and Andrew D. Gordon. A calculus for cryptographic protocols: The spi calculus. *Information and Computation*, 148(1):1–70, 10January 1999.

[AL00] Roberto M. Amadio and Denis Lugiez. On the reachability problem in cryptographic protocols. In *Proceedings of CONCUR 00*, volume 1877 of *Lecture Notes in Computer Science*. Springer, 2000. INRIA Research Report 3915, march 2000.

[Ane] Anonymizer. http://anonymizer.com.

[Ans] Anonymouse. http://anonymouse.org.

[BBF+07] Gilles Brassard, Anne Broadbent, Joseph Fitzsimons, Sébastien Gambs, and Alain Tapp. Anonymous quantum communication. In *ASIACRYPT*, pages 460–473, 2007.

[BCP08] Christelle Braun, Konstantinos Chatzikokolakis, and Catuscia Palamidessi. Compositional Methods for Information-Hiding. In Roberto Amadio, editor, *Foundations of Software Science and Computation Structures (FOSSACS)*, volume 4962 of *Lecture Notes in Computer Science*, Budapest Hongrie, 2008. Springer.

[BCP09] Christelle Braun, Konstantinos Chatzikokolakis, and Catuscia Palamidessi. Quantitative notions of leakage for one-try attacks. In *Proceedings of the 25th Conf. on Mathematical Foundations of Programming Semantics*, volume 249 of *Electronic Notes in Theoretical Computer Science*, pages 75–91. Elsevier B.V., 2009.

[BFPR02] Annalisa Bossi, Riccardo Focardi, Carla Piazza, and Sabina Rossi. A proof system for information flow security. In *PROC. OF INT. WORKSHOP ON LOGIC BASED PROGRAM DEVELOPMENT AND TRANSFORMATION, LNCS*, pages 199–218. Springer-Verlag, 2002.

[BHR84] S. D. Brookes, C. A. R. Hoare, and A. W. Roscoe. A theory of communicating sequential processes. *J. ACM*, 31(3):560–599, 1984.

[BLP76] DE Bell and LJ La Padula. Secure computer system: Unified exposition and multics interpretation, MTR-2997. *MITRE Corp., Bedford, MA*, 1976.

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

[BP05] Mohit Bhargava and Catuscia Palamidessi. Probabilistic anonymity. Technical report, INRIA Futurs and LIX, 2005. To appear in the proceedings of CONCUR 2005. Report version available at http://www.lix.polytechnique.fr/~catuscia/papers/Anonymity/report.ps.

[BP09] Romain Beauxis and Catuscia Palamidessi. Probabilistic and nondeterministic aspects of anonymity. *Theoretical Computer Science*, 410(41):4006–4025, 2009.

[BS] J. Bradfield and C. Stirling. Modal mu-calculi. *Handbook of Modal Logic*, 3:721–756.

[Cac97] C. Cachin. Entropy measures and unconditional security in cryptography, 1997.

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

[Cha07] Konstantinos Chatzikokolakis. *Probabilistic and Information-Theoretic Approaches to Anonymity*. PhD thesis, Laboratoire d'Informatique (LIX), École Polytechnique, Paris, October 2007.

[CHM02] David Clark, Sebastian Hunt, and Pasquale Malacaria. Quantitative analysis of the leakage of confidential data. 59(3), 2002.

[CHM05a] D. Clark, S. Hunt, and P. Malacaria. Quantified interference for a while language. In *Electronic Notes in Theoretical Computer Science 112*, pages 149 – 166. Elsevier, 2005.

[CHM05b] D. Clark, S. Hunt, and P. Malacaria. Quantitative information flow, relations and polymorphic types. *Journal of Logic and Computation, Special Issue on Lambda-calculus, type theory and natural language*, 18(2):181–199, 2005.

[CHM07] David Clark, Sebastian Hunt, and Pasquale Malacaria. A static analysis for quantifying information flow in a simple imperative language. *Journal of Computer Security*, 15(3):321–371, 2007.

[CM90] John Mclean Center and John Mclean. Security models and information flow. In *In Proc. IEEE Symposium on Security and Privacy*, pages 180–187. IEEE Computer Society Press, 1990.

[CMS08] Michael R. Clarkson, Andrew C. Myers, and Fred B. Schneider. Quantifying information flow with beliefs. 2008.

[CP05] Konstantinos Chatzikokolakis and Catuscia Palamidessi. Probable innocence revisited. Technical report, INRIA Futurs and LIX, 2005. http://www.lix.polytechnique.fr/~catuscia/papers/Anonymity/reportPI.pdf.

[CP07a] Konstantinos Chatzikokolakis and Catuscia Palamidessi. A framework for analyzing probabilistic protocols and its application to the partial secrets exchange. *Theoretical Computer Science*, 389(3):512 – 527, 2007. Semantic and Logical Foundations of Global Computing.

[CP07b] Konstantinos Chatzikokolakis and Catuscia Palamidessi. Making random choices invisible to the scheduler. In Luís Caires and Vasco Thudichum Vasconcelos, editors, *Proceedings of CONCUR'07*, volume 4703 of *Lecture Notes in Computer Science*, pages 42–58. Springer, 2007. http://www.lix.polytechnique.fr/~catuscia/papers/Scheduler/report.pdf.

[CPP07] Konstantinos Chatzikokolakis, Catuscia Palamidessi, and Prakash Panangaden. Probability of error in information-hiding protocols. In *Proceedings of the 20th IEEE Computer Security Foundations Symposium (CSF20)*, pages 341–354. IEEE Computer Society, 2007. http://www.lix.polytechnique.fr/~catuscia/papers/ProbabilityError/full.pdf.

[CPP08a] Konstantinos Chatzikokolakis, Catuscia Palamidessi, and Prakash Panangaden. Anonymity protocols as noisy channels. *Information and Computation*, 206(2-4):378 – 401, 2008. Joint Workshop on Foundations of Computer Security and Automated Reasoning for Security Protocol Analysis (FCS-ARSPA '06).

[CPP08b] Konstantinos Chatzikokolakis, Catuscia Palamidessi, and Prakash Panangaden. On the bayes risk in information-hiding protocols. *J. Comput. Secur.*, 16(5):531–571, 2008.

[CRZ07] Véronique Cortier, Michaël Rusinowitch, and Eugen Zalinescu. Relating two standard notions of secrecy. *Logical Methods in Computer Science*, 3(3), 2007.

[CSWH01] Ian Clarke, Oskar Sandberg, Brandon Wiley, and Theodore W. Hong. Freenet: A distributed anonymous information storage and retrieval system. *Lecture Notes in Computer Science*, 2009:46–66, 2001.

[CT06] Thomas M. Cover and Joy A. Thomas. *Elements of Information Theory (Wiley Series in Telecommunications and Signal Processing)*. Wiley-Interscience, 2006.

[CW05] Matthias Christandl and Stephanie Wehner. Quantum anonymous transmissions. *LNCS*, 3788:217, 2005.

[Den82] Dorothy E. Denning. *Cryptography and data security*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1982.

[DEP98] J. Desharnais, A. Edalat, and P. Panangaden. A logical characterization of bisimulation for labeled markov processes. *Logic in Computer Science, Symposium on*, 0:478, 1998.

[Dev03] I. Devetak. The private classical capacity and quantum capacity of a quantum channel, 2003.

[DG09] M. Dhawan and V. Ganapathy. Analyzing information flow in javascript-based browser extensions. pages 382 –391, dec. 2009.

[DMO07] F. Dechesne, M.R. Mousavi, and S. Orzan. Operational and epistemic approaches to protocol analysis: Bridging the gap. *Lecture Notes in Computer Science*, 4790:226, 2007.

[DORS08] Yevgeniy Dodis, Rafail Ostrovsky, Leonid Reyzin, and Adam Smith. Fuzzy extractors: How to generate strong keys from biometrics and other noisy data. *SIAM J. Comput.*, 38(1):97–139, 2008.

[DPHW02] Alessandra Di Pierro, Chris Hankin, and Herbert Wiklicky. Approximate non-interference. In *CSFW '02: Proceedings of the 15th IEEE workshop on Computer Security Foundations*, page 3, Washington, DC, USA, 2002. IEEE Computer Society.

[DPP05] Yuxin Deng, Catuscia Palamidessi, and Jun Pang. Compositional reasoning for probabilistic finite-state behaviors. In Aart Middeldorp, Vincent van Oostrom, Femke van Raamsdonk, and Roel C. de Vrijer, editors, *Processes, Terms and Cycles: Steps on the Road to Infinity*, volume 3838 of *Lecture Notes in Computer Science*, pages 309–337. Springer, 2005. http://www.lix.polytechnique.fr/~catuscia/papers/Yuxin/BookJW/par.pdf.

[DPW07] Yuxin Deng, Jun Pang, and Peng Wu. Measuring anonymity with relative entropy. pages 65–79. 2007.

[DSCP02a] Claudia Diaz, Stefaan Seys, Joris Claessens, and Bart Preneel. Towards measuring anonymity. pages 54–68. Springer-Verlag, 2002.

[DSCP02b] Claudia Diaz, Stefaan Seys, Joris Claessens, and Bart Preneel. Towards measuring anonymity. pages 54–68. Springer-Verlag, 2002.

[DSD04] Claudia Díaz, Len Sassaman, and Evelyne Dewitte. Comparison between two practical mix designs. In *ESORICS*, pages 141–159, 2004.

[EBF] European biometrics forum. http://www.eubiometricsforum.com.

[EC80] E. Allen Emerson and Edmund M. Clarke. Characterizing correctness properties of parallel programs using fixpoints. In *Proceedings of the 7th Colloquium on Automata, Languages and Programming*, pages 169–181, London, UK, 1980. Springer-Verlag.

[EGL85] Shimon Even, Oded Goldreich, and Abraham Lempel. A randomized protocol for signing contracts. *Commun. ACM*, 28(6):637–647, 1985.

[EKK+07] Manuel Egele, Christopher Kruegel, Engin Kirda, Heng Yin, and Dawn Song. Dynamic spyware analysis. In *ATC'07: 2007 USENIX Annual Technical Conference on Proceedings of the USENIX Annual Technical Conference*, pages 1–14, Berkeley, CA, USA, 2007. USENIX Association.

[Eme90] E. Allen Emerson. Temporal and modal logic. In *Handbook of Theoretical Computer Science*, pages 995–1072. Elsevier, 1990.

[Fac] Facebook. http://www.facebook.com.

[FG96] R. Focardi and R. Gorrieri. Automatic compositional verification of some security properties. *Lecture Notes in Computer Science*, 1055:167–186, 1996.

[FG97] Riccardo Focardi and Roberto Gorrieri. The compositional security checker: A tool for the verification of information flow security properties, 1997.

[FGM95] Riccardo Focardi, Roberto Gorrieri, and Fabio Martinelli. Classification of security properties (part ii: Network security), 1995.

[FGM03] R. Focardi, R. Gorrieri, and F. Martinelli. Real-time information flow analysis. *IEEE Journal on Selected Areas in Communications*, 21(1):20–35, 2003.

[Fil] Filtersneak. http://www.filtersneak.com.

[Fli] Flickr. http://www.flickr.com.

[Foc96] Riccardo Focardi. Comparing two information flow security properties. In *CSFW '96: Proceedings of the 9th IEEE workshop on Computer Security Foundations*, page 116, Washington, DC, USA, 1996. IEEE Computer Society.

[FR02] Riccardo Focardi and Sabina Rossi. Information flow security in dynamic contexts. In *CSFW '02: Proceedings of the 15th IEEE workshop on Computer Security Foundations*, page 307, Washington, DC, USA, 2002. IEEE Computer Society.

[GM82] Joseph A. Goguen and José Meseguer. Security policies and security models. In *IEEE Symposium on Security and Privacy*, pages 11–20, 1982.

[HO03] Joseph Y. Halpern and Kevin R. O'Neill. Anonymity and information hiding in multiagent systems. In *Proc. of the 16th IEEE Computer Security Foundations Workshop*, pages 75–88, 2003.

[HO05] Joseph Y. Halpern and Kevin R. O'Neill. Anonymity and information hiding in multiagent systems. *Journal of Computer Security*, 13(3):483–512, 2005.

[HP00] Oltea Mihaela Herescu and Catuscia Palamidessi. Probabilistic asynchronous π-calculus. In Jerzy Tiuryn, editor, *Proceedings of FOSSACS 2000 (Part of ETAPS 2000)*, volume 1784 of *Lecture Notes in Computer Science*, pages 146–160. Springer, 2000. http://www.lix.polytechnique.fr/~catuscia/papers/Prob_asy_pi/fossacs.ps.

[HP05] Joseph Y. Halpern and Riccardo Pucella. Probabilistic algorithmic knowledge. In *Logical Methods in Computer Science*, pages 118–130, 2005.

[HR02] Matthew Hennessy and James Riely. Information flow vs. resource access in the asynchronous pi-calculus. *ACM Trans. Program. Lang. Syst.*, 24(5):566–591, 2002.

[HS04] Dominic Hughes and Vitaly Shmatikov. Information hiding, anonymity and privacy: a modular approach. *Journal of Computer Security*, 12(1):3–36, 2004.

[HZBB05] Mark Hillery, Mario Ziman, Vladimir Buzek, and Martina Bielikova. Towards quantum-based privacy and voting, 2005.

[III91] James W. Gray III. Toward a mathematical foundation for information flow security. In *IEEE Symposium on Security and Privacy*, pages 21–35, 1991.

[Joi01] Adam N. Joinson. Self-disclosure in computer-mediated communication: The role of self-awareness and visual anonymity. *European Journal of Social Psychology*, 31(2):177–192, 2001.

[JT88] D.M. Johnson and F.J. Thayer. Security and the Composition of Machines. In *Proceedings of the Computer Security Foundations Workshop*, pages 72–89, 1988.

[KB07] Boris Koepf and David Basin. An information-theoretic model for adaptive side-channel attacks. In *Proc. 14th ACM Conference on Computer and Communications Security (CCS '07)*, pages 286–296, New York, NY, USA, 2007. ACM.

[Kil88] Joe Kilian. Founding cryptography on oblivious transfer. In *Proceedings of the 20th ACM Annual Symposium on the Theory of Computing*, pages 20–31, 1988.

[Koz83] Dexter Kozen. Results on the propositional mu-calculus. *Theor. Comput. Sci.*, 27:333–354, 1983.

[KPS+10] Simon Kramer, Catuscia Palamidessi, Roberto Segala, Andrea Turrini, and Christelle Braun. A quantitative doxastic logic for probabilistic processes and applications to information-hiding. *The Journal of Applied Non-Classical Logics*, 2010.

[Low97] Gavin Lowe. Casper: A compiler for the analysis of security protocols. In *Proceedings of 10th IEEE Computer Security Foundations Workshop*, 1997. Also in Journal of Computer Security, Volume 6, pages 53-84, 1998.

[Low04] Gavin Lowe. Defining information flow quantity. *J. Comput. Secur.*, 12(3,4):619–653, 2004.

[LP07] Alessio Lomuscio and Wojciech Penczek. Symbolic model checking for temporal-epistemic logics. *SIGACT News*, 38(3):77–99, 2007.

[LS91] Kim G. Larsen and Arne Skou. Bisimulation through probabilistic testing. *Inf. Comput.*, 94(1):1–28, 1991.

[LZ05] Peng Li and Steve Zdancewic. Practical information-flow control in web-based information systems. In *CSFW '05: Proceedings of the 18th IEEE workshop on Computer Security Foundations*, pages 2–15, Washington, DC, USA, 2005. IEEE Computer Society.

[Mal07] P. Malacaria. Assessing security threat of looping constructs. In *Proc. 34th ACM Symposium on Principles of Programming Languages*, pages 225–235, Jan 2007.

[Man02] Heiko Mantel. On the composition of secure systems. In *SP '02: Proceedings of the 2002 IEEE Symposium on Security and Privacy*, page 88, Washington, DC, USA, 2002. IEEE Computer Society.

[Mas94] James L. Massey. Guessing and entropy. In *In Proceedings of the 1994 IEEE International Symposium on Information Theory*, page 204, 1994.

[MC08] Pasquale Malacaria and Han Chen. Lagrange multipliers and maximum information leakage in different observational models. In *PLAS '08: Proceedings of the third ACM SIGPLAN workshop on Programming languages and analysis for security*, pages 135–146, New York, NY, USA, 2008. ACM.

[McC87] Daryl McCullough. Specifications for multi-level security and a hookup. *Security and Privacy, IEEE Symposium on*, 0:161, 1987.

[McC90] Daryl McCullough. A hookup theorem for multilevel security. *IEEE Trans. Softw. Eng.*, 16(6):563–568, 1990.

[Mil87] Jonathan K. Millen. Covert channel capacity. *Security and Privacy, IEEE Symposium on*, 0:60, 1987.

[Mil89] R. Milner. *Communication and Concurrency*. International Series in Computer Science. Prentice Hall, 1989.

[MM03] Annabelle McIver and Carroll Morgan. A probabilistic approach to information hiding. pages 441–460, 2003.

[MMC] Mmcsp compiler. http://www.cs.ucl.ac.uk/staff/p.wu/mmc_sp_manual.html.

[MNCM03] Ira S. Moskowitz, Richard E. Newman, Daniel P. Crepeau, and Allen R. Miller. Covert channels and anonymizing networks. In *WPES*, pages 79–88, 2003.

[MNS03] Ira S. Moskowitz, Richard E. Newman, and Paul F. Syverson. Quasi-anonymous channels. In *IASTED CNIS*, pages 126–131, 2003.

[Mos91] Ira S. Moskowitz. Variable noise effects upon a simple timing channel. *Security and Privacy, IEEE Symposium on*, 0:362, 1991.

[Mye99] Andrew C. Myers. Jflow: Practical mostly-static information flow control. In *In Proc. 26th ACM Symp. on Principles of Programming Languages (POPL*, pages 228–241, 1999.

[Nie98] Mogens Nielsen. Reasoning about the past. In Lubos Brim, Jozef Gruska, and Jiri Zlatuska, editors, *Proceedings of MFCS'98*, volume 1450 of *Lecture Notes in Computer Science*, pages 117–128. Springer-Verlag, 1998.

[NPPW09] G. Norman, C. Palamidessi, D. Parker, and P. Wu. Model checking probabilistic and stochastic extensions of the π-calculus. *IEEE Transactions on Software Engineering*, 35(2):209–223, 2009.

[NS09] Arvind Narayanan and Vitaly Shmatikov. De-anonymizing social networks. Mar 2009.

[oJ08] US Department of Justice. Retail hacking ring charged for stealing and distributing credit and debit card numbers from major u.s. retailers, 2008. http://www.justice.gov/opa/pr/2008/August/08-ag-689.html.

[ONI] Opennet initiative. http://opennet.net.

[Pal06] Catuscia Palamidessi. Probabilistic and nondeterministic aspects of anonymity. *Electronic Notes in Theoretical Computer Science*, 155:33 – 42, 2006. Proceedings of the 21st Annual Conference on Mathematical Foundations of Programming Semantics (MFPS XXI).

[PH05] Catuscia Palamidessi and Oltea M. Herescu. A randomized encoding of the π-calculus with mixed choice. *Theoretical Computer Science*, 335(2-3):373–404, 2005. http://www.lix.polytechnique.fr/~catuscia/papers/prob_enc/report.pdf.

[Pli00] John O. Pliam. On the incomparability of entropy and marginal guesswork in brute-force attacks. In *INDOCRYPT*, pages 67–79, 2000.

[Pra81] V. R. Pratt. A decidable mu-calculus: Preliminary report. In *SFCS '81: Proceedings of the 22nd Annual Symposium on Foundations of Computer Science*, pages 421–427, Washington, DC, USA, 1981. IEEE Computer Society.

[Pro] Proxify. http://proxify.com.

[PS07] Augusto Parma and Roberto Segala. Logical characterizations of bisimulations for discrete probabilistic systems. In *FoSSaCS*, pages 287–301, 2007.

[PW87] A Pfitzmann and M Waidner. Networks without user observability. *Comput. Secur.*, 6(2):158–166, 1987.

[Rab81] Michael O. Rabin. How to exchange secrets by oblivious transfer. *Technical Memo TR-81, Aiken Computation Laboratory, Harvard University*, 1981.

[Rén60] Alfred Rényi. On measures of entropy and information. In *Proceedings of the 4th Berkeley Symposium on Mathematics, Statistics, and Probability*, pages 547–561, 1960.

[Ren09] P. Renaud, K.;Cockshott. Handivote: Simple, anonymous, and auditable electronic voting. *Journal of Information Technology & Politics, Volume 6, Issue 1*, pages 60–80, 2009.

[Ros95a] A. W. Roscoe. Csp and determinism in security modelling. In *In Proc. IEEE Symposium on Security and Privacy*, pages 114–127. Society Press, 1995.

[Ros95b] A. W. Roscoe. Modelling and verifying key-exchange protocols using CSP and FDR. In *Proceedings of the 8th IEEE Computer Security Foundations Workshop*, pages 98–107. IEEE Computer Soc Press, 1995.

[RR98] Michael K. Reiter and Aviel D. Rubin. Crowds: anonymity for Web transactions. *ACM Transactions on Information and System Security*, 1(1):66–92, 1998.

[RS99] P.Y.A. Ryan and S.A. Schneider. Process algebra and non-interference. In *Journal of Computer Security*, pages 214–227, 1999.

[RS01] Peter Y. Ryan and Steve Schneider. *Modelling and Analysis of Security Protocols*. Addison-Wesley, 2001.

[RWW94] A. W. Roscoe, J. C. P. Woodcock, and L. Wulf. Non-interference through determinism. In *ESORICS '94: Proceedings of the Third European Symposium on Research in Computer Security*, pages 33–53, London, UK, 1994. Springer-Verlag.

[Rya91] P. Y. A. Ryan. A csp formulation of non-interference and unwinding. *Cipher: IEEE Comput. Soc. Tech. Comm. Newsl. Secur. Priv.*, pages 19–30, 1991.

[Sch96] S. Schneider. Security properties and CSP. In *Proceedings of the IEEE Symposium Security and Privacy*, 1996.

[SD02a] A. Serjantov and G. Danezis. Towards an information theoretic metric for anonymity. *Lecture Notes in Computer Science*, 2482:41–53, 2002.

[SD02b] Andrei Serjantov and George Danezis. Towards an information theoretic metric for anonymity. In Roger Dingledine and Paul F. Syverson, editors, *Proceedings of the workshop on Privacy Enhancing Technologies (PET) 2002*, volume 2482 of *Lecture Notes in Computer Science*, pages 41–53. Springer, 2002.

[Seg95] Roberto Segala. *Modeling and Verification of Randomized Distributed Real-Time Systems*. PhD thesis, Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, June 1995. Available as Technical Report MIT/LCS/TR-676.

[Seg06] Roberto Segala. Probability and nondeterminism in operational models of concurrency. In Christel Baier and Holger Hermanns, editors, *Proceedings of the 17th International Conference on Concurrency Theory (CONCUR)*, volume 4137 of *Lecture Notes in Computer Science*, pages 64–78. Springer, 2006.

[SGR97] Paul F. Syverson, David M. Goldschlag, and Michael G. Reed. Anonymous connections and onion routing, 1997.

[Sha48] Claude E. Shannon. *A Mathematical Theory of Communication*. CSLI Publications, 1948.

[SL95] Roberto Segala and Nancy Lynch. Probabilistic simulations for probabilistic processes. *Nordic Journal of Computing*, 2(2):250–273, 1995. An extended abstract appeared in *Proceedings of CONCUR '94*, LNCS 836: 481-496.

[Smi07] Geoffrey Smith. Adversaries and information leaks (tutorial). *Lecture Notes in Computer Science*, 4912:383–400, 2007.

[Smi09] Geoffrey Smith. On the foundations of quantitative information flow. In *FOSSACS*, pages 288–302, 2009.

[SR03] Vincent Simonet and Inria Rocquencourt. Flow caml in a nutshell. In *Proceedings of the first APPSEM-II workshop*, pages 152–165, 2003.

[SS96] Steve Schneider and Abraham Sidiropoulos. CSP and anonymity. In *Proc. of the European Symposium on Research in Computer Security (ESORICS)*, volume 1146 of *Lecture Notes in Computer Science*, pages 198–218. Springer, 1996.

[SS99] Paul F. Syverson and Stuart G. Stubblebine. Group principals and the formalization of anonymity. In *World Congress on Formal Methods (1)*, pages 814–833, 1999.

[SS05] Andrei Sabelfeld and David Sands. Dimensions and principles of declassification. In *CSFW '05: Proceedings of the 18th IEEE workshop on Computer Security Foundations*, pages 255–269, Washington, DC, USA, 2005. IEEE Computer Society.

[Sut86] D. Sutherland. A model of information. In *Proc. of the 9th National Computer Security Conference*, 1986.

[Twi] Twitter. http://twitter.com.

[VS00] Dennis Volpano and Geoffrey Smith. Verifying secrets and relative secrecy. In *In Proc. 27th ACM Symp. on Principles of Programming Languages (POPL*, pages 268–276. ACM Press, 2000.

[VSC07] J. A. Vaccaro, Joseph Spring, and Anthony Chefles. Quantum protocols for anonymous voting and surveying. *Physical Review A*, 75:012333, 2007.

[VTP] Caltech/mit voting technology project. http://vote.caltech.edu.

[WJ90] J. Todd Wittbold and Dale M. Johnson. Information flow in nondeterministic systems. *Security and Privacy, IEEE Symposium on*, 0:144, 1990.

[YLZC09] Shuiming Ye, Ying Luo, Jian Zhao, and Sen-Ching S. Cheung. Anonymous biometric access control. 2009.

[YSE+07] Heng Yin, Dawn Song, Manuel Egele, Christopher Kruegel, and Engin Kirda. Panorama: capturing system-wide information flow for malware detection and analysis. In *CCS '07: Proceedings of the 14th ACM conference on Computer and communications security*, pages 116–127, New York, NY, USA, 2007. ACM.

[ZB05] Ye Zhu and Riccardo Bettati. Anonymity vs. information leakage in anonymity systems. In *Proc. of ICDCS*, pages 514–524. IEEE Computer Society, 2005.

[ZL95] A. Zakinthinos and E. S. Lee. The composability of non-interference [system security]. In *CSFW '95: Proceedings of the 8th IEEE workshop on Computer Security Foundations*, page 2, Washington, DC, USA, 1995. IEEE Computer Society.

[ZL96] A. Zakinthinos and E. S. Lee. How and why feedback composition fails [secure systems]. In *CSFW '96: Proceedings of the 9th IEEE workshop on Computer Security Foundations*, page 95, Washington, DC, USA, 1996. IEEE Computer Society.

[ZL98] A. Zakinthinos and E. Lee. Composing secure systems that have emergent properties. In *CSFW '98: Proceedings of the 11th IEEE workshop on Computer Security Foundations*, page 117, Washington, DC, USA, 1998. IEEE Computer Society.