Math. Struct. in Comp. Science (2016), *vol.* 26, *pp.* 908–932. © Cambridge University Press 2014
doi:10.1017/S0960129514000292 First published online 24 November 2014

Compositional methods for information-hiding[†]

KONSTANTINOS CHATZIKOKOLAKIS,
CATUSCIA PALAMIDESSI and CHRISTELLE BRAUN

LIX, Bâtiment Alan Turing, 1 rue Honoré d'Estienne d'Orves, Campus de l'École Polytechnique, 91120 Palaiseau, France
Email: kostas@lix.polytechnique.fr;catuscia@lix.polytechnique.fr; braun@lix.polytechnique.fr

Received December 2011; revised August 2013

Systems concerned with information hiding often use randomization to obfuscate the link between the observables and the information to be protected. The degree of protection provided by a system can be expressed in terms of the probability of error associated with the inference of the secret information. We consider a probabilistic process calculus to specify such systems, and we study how the operators affect the probability of error. In particular, we characterize constructs that have the property of not decreasing the degree of protection, and that can therefore be considered safe in the modular construction of these systems. As a case study, we apply these techniques to the dining cryptographers, and we derive a generalization of Chaum's strong anonymity result.

1. Introduction

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 which 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.

A recent example of such privacy concerns are the so-called 'biometric' passports. These passports, used by many countries and required by all visa waiver travellers to the United States, include an RFID chip containing information about the passport's owner. These chips can be read wirelessly without any contact with the passport and without the owner even knowing that his passport is being read. It is clear that such devices need protection mechanisms to ensure that the contained information will not be revealed to any non-authorized person.

In general, privacy can be defined as the ability of users to prevent information about themselves from becoming known to people other than those they choose to give the information to. We can further categorize privacy properties based on the nature of the

[†] This work has been partially supported by the project ANR-12-IS02-001 (PACE), by the project ANR-11-IS02-0002 (LOCALI), by the INRIA Large Scale Initiative CAPPRIS, and by the EU FP7 grant no. 295261 (MEALS). A preliminary version of this work appeared in the proceedings of FOSSACS 2008.

https://doi.org/10.1017/S0960129514000292 Published online by Cambridge University Press

hidden information. *Data protection* usually refers to confidential data like the credit card number. *Anonymity*, on the other hand, concerns the identity of the user who performed a certain action. *Unlinkability* refers to the link between the information and the user, and *unobservability* regards the actions of a user.

Information-hiding protocols aim at ensuring a privacy property during an electronic transaction. For example, the voting protocol Foo 92 (Fujioka *et al.* 1993) allows a user to cast a vote without revealing the link between the voter and the vote. The anonymity protocol Crowds (Reiter and Rubin 1998) allows a user to send a message on a public network without revealing the identity of the sender. These kinds of protocols often use *randomization* to introduce *noise*, thus limiting the inference power of a malicious observer.

1.1. *The information theory*

At an abstract level, information-hiding protocols can be viewed as *information-theoretic channels*. 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 the information theory to reason about the knowledge that the attacker can gain about the input by observing the output of the protocol (*information leakage*). This leakage is usually expressed in terms of *mutual information*, that is the difference between the *a priori* entropy (the initial uncertainty of the attacker) and the *a posteriori* entropy (the uncertainty of the attacker after the observation). The channel *capacity*, that is defined as the maximum mutual information under all possible *a priori* distributions, represents the worst case of leakage.

1.2. *Hypothesis testing*

The information theory is parametric on the notion of entropy. The most popular one is the Shannon entropy, because of its relation with the channel's transmission rate. With respect to the problem of information-hiding, however, one of the most natural notions is arguably the Rényi min entropy (Rényi 1961). As discussed by Smith (2009), this notion represents well the *one-try attacks*, and it is strictly related to the problem of *hypothesis testing* and to the *Bayes risk*.

In information-hiding systems 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 as known the *a priori* probability of the distribution of the hypotheses, and in deriving from that (and from the matrix of the conditional probabilities) the *a posteriori*

https://doi.org/10.1017/S0960129514000292 Published online by Cambridge University Press

distribution after a certain fact has been observed. It is well known that the best strategy for the adversary is to apply the maximum *a posteriori* probability (MAP) 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 called the *Bayes risk*. The *a posteriori* Rényi min entropy is the logarithm of the converse of the Bayes risk[†]. In Chatzikokolakis *et al.* (2008b), we proposed to define the *degree of protection* provided by a protocol as the Bayes risk associated to the matrix. McIver *et al.* (2010) have shown that the Bayes risk is the maximally discriminating among various notions of entropy, when compositionality is taken into account.

A major problem with the Bayesian method is that the *a priori* distribution is not always known. This is particularly true in security applications. In some cases, it may be possible to approximate the *a priori* distribution by statistical inference, but in most cases, especially when the input information changes over time, it may not. Thus other methods need to be considered, which do not depend on the *a priori* distribution. One such method is the one based on the so-called *maximum likelihood* criterion.

1.3. *Contribution*

In this paper, we consider the Bayesian hypothesis-testing approach to the one-try attacks, under the assumption that the input distribution is known. We consider as degree of protection the Bayes risk, i.e. the probability of error of an adversary trying to discover the secret using the MAP rule.

Next, we consider a probabilistic process algebra for the specification 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 specifications in a compositional way, while preserving the degree of protection.

We apply these compositional methods to the example of dining cryptographers (DC), 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 specification and using the fact that this can be done with a safe construct.

1.4. *Plan of the paper*

In the next section, we recall some basic notions. Section 3 introduces the language CCS_p. Section 4 shows how to model protocols and process terms as channels. Section 5 discusses hypothesis testing and presents some properties of the probability of error. Section 6 characterizes the constructs of CCS_p which are safe. Section 7 applies previous

[†] There are other possible definitions of the *a posteriori* Rényi min entropy. Smith proposed to use this one because of its suitability for the information-hiding problem.

https://doi.org/10.1017/S0960129514000292 Published online by Cambridge University Press

results to find a new property of the DC. Section 8 discusses related work. Section 9 concludes the paper.

2. Preliminaries

In this section, we give a brief overview of the technical concepts from the literature that will be used through the paper. More precisely, we recall some basic notions of metric spaces (see for instance Munkres (2000)), probability theory (see for instance Chung (2000)) and probabilistic automata (Segala 1995; Segala and Lynch 1995).

2.1. *Metric spaces*

A *metric space* is a pair (X, d), where X is a set and $d : X \times X \to [0, \infty)$ is a function (called *distance* or *metric*) which satisfies the following properties:

— for all $x, y \in X$, $d(x, y) = 0$ if and only if $x = y$,
— for all $x, y \in X$, $d(x, y) = d(y, x)$,
— for all $x, y, z \in X$, $d(x, y) \leqslant d(y, z) + d(z, y)$.

Let (X, d) be a metric space. A sequence of elements in X is converging if it has a limit in X with respect to the distance d. The metric space (X, d) is called *compact* if every sequence of elements in X has a converging subsequence.

2.2. *Probability spaces*

Let Ω be a set. A *σ-field* 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 U (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 *measure* on $\mathcal{F}$ is a function $\mu : \mathcal{F} \to [0, \infty]$ such that

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

A *probability measure* on $\mathcal{F}$ is a measure μ on $\mathcal{F}$ such that $\mu(\Omega) = 1$. A *probability space* is a tuple $(\Omega, \mathcal{F}, \mu)$, where Ω is a 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*.

We will denote by $\delta(x)$ (called the *Dirac measure* on x) the probability measure s.t. $\delta(x)(\{y\}) = 1$ if $y = x$, and $\delta(x)(\{y\}) = 0$ otherwise. If $\{c_i\}_i$ are convex coefficients, and $\{\mu_i\}_i$ are probability measures, we will denote by $\sum_i c_i \mu_i$ the probability measure defined as $(\sum_i c_i \mu_i)(A) = \sum_i c_i \mu_i(A)$.

If A, B are events then $A \cap B$ is also an event. If $\mu(A) > 0$ then we can define the *conditional probability* $p(B|A)$, meaning "the probability of B given that A holds", as

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

https://doi.org/10.1017/S0960129514000292 Published online by Cambridge University Press

Note that $p(\cdot|A)$ 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 needs of this paper. 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 $p : \Omega \to [0, 1]$ 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 $Disc(\Omega)$.

2.3. *Probabilistic automata*

A *probabilistic automaton* $\mathcal{M}$ is a tuple $(St, T_{init}, Act, \mathcal{T})$, where St is a set of states, $T_{init} \in St$ 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 $(T, a, \mu) \in \mathcal{T}$ then there is a transition from the state T performing the action a and leading to a distribution μ over the states of the automaton. (We use T for states instead of s because later in the paper states will be (process) terms, and s will be used for sequences of actions.) We also write $T \xrightarrow{a} \mu$ if $(T, a, \mu) \in \mathcal{T}$. 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 T' such that $\mu(T') > 0$) is performed probabilistically. A probabilistic automaton $\mathcal{M}$ is *fully probabilistic* if from each state of $\mathcal{M}$ there is at most one transition available.

An *execution fragment* α of a probabilistic automaton is a (possibly infinite) sequence $T_0 a_1 T_1 a_2 T_2 \ldots$ of alternating states and actions, such that for each i there is a transition $(T_i, a_{i+1}, \mu_i) \in \mathcal{T}$ and $\mu_i(T_{i+1}) > 0$. We will use $fst(\alpha), lst(\alpha)$ to denote the first and last state of a finite execution fragment α, respectively. An *execution* (or *history*) is an execution fragment such that $fst(\alpha) = T_{init}$. An execution α is maximal if it is infinite or there is no transition from $lst(\alpha)$ in $\mathcal{T}$. We denote by $exec^*(\mathcal{M})$, $exec^{\perp}(\mathcal{M})$ and $exec(\mathcal{M})$ the set of all the finite, all the non-maximal and all executions of $\mathcal{M}$, respectively.

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

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

such that $\zeta(\alpha) = (T, a, \mu) \in \mathcal{T}$ implies that $T = lst(\alpha)$.

The idea is that a scheduler selects a transition among the ones available in $\mathcal{T}$ and he can base his decision on the history of the execution. The *execution tree* of $\mathcal{M}$ relative to the scheduler ζ, denoted by $etree(\mathcal{M}, \zeta)$, is a fully probabilistic automaton $\mathcal{M}' = (St', T_{init}, Act, \mathcal{T}')$ such that $St' \subseteq exec^*(\mathcal{M})$, and $(\alpha, a, \mu') \in \mathcal{T}'$ if and only if $\zeta(\alpha) = (lst(\alpha), a, \mu)$ for some μ, and $\mu'(\alpha a T) = \mu(T)$. Intuitively, $etree(M, \zeta)$ is produced by unfolding the executions of $\mathcal{M}$ and resolving the non-determinism using ζ.

In the following, given two executions α, α', we write $\alpha \leqslant \alpha'$ to indicate that α is a prefix of α'. Given, a fully probabilistic automaton $\mathcal{M} = (St, T_{init}, Act, \mathcal{T})$ we can define a probability space $(\Omega_{\mathcal{M}}, \mathcal{F}_{\mathcal{M}}, p_{\mathcal{M}})$ on the space of executions of $\mathcal{M}$ as follows:

— $\Omega_{\mathcal{M}} \subseteq exec(\mathcal{M})$ is the set of maximal executions of $\mathcal{M}$.

https://doi.org/10.1017/S0960129514000292 Published online by Cambridge University Press

— If α is a finite execution of $\mathcal{M}$ we define the cone with prefix α as $C_\alpha = \{\alpha' \in \Omega_\mathcal{M} | \alpha \leqslant \alpha'\}$. Let $\mathcal{C}_\mathcal{M}$ be the collection of all cones of $\mathcal{M}$. Then $\mathcal{F}$ is the σ-field generated by $\mathcal{C}_\mathcal{M}$ (by closing under complement and countable union).
— We define the probability of a cone C_α where $\alpha = T_0 a_1 T_1 \ldots a_n T_n$ as

$$p(C_\alpha) = \prod_{i=1}^{n} \mu_i(T_i),$$

where μ_i is the (unique because the automaton is fully probabilistic) measure such that $(T_{i-1}, a_i, \mu_i) \in \mathcal{T}$. We define $p_\mathcal{M}$ as the measure extending p to $\mathcal{F}$ (see (Segala 1995) for more details about this construction).

3. CCS with internal probabilistic choice

We consider an extension of standard CCS ((Milner 1989)) obtained by adding internal probabilistic choice. The resulting calculus CCS_p can be seen as a simplified version of the probabilistic π-calculus presented in Herescu and Palamidessi (2000); Palamidessi and Herescu (2005) and it is similar to the one considered in Deng *et al.* (2005). Like in those calculi, computations have both a probabilistic and a non-deterministic nature. The main conceptual novelty is a distinction between *observable* and *secret* actions, introduced for the purpose of specifying information-hiding protocols.

We assume a countable set *Act* of actions a, and we assume that it is partitioned into a set *Sec* of *secret actions* s, a set *Obs* of *observable actions* o and the silent action τ. For each $s \in Sec$, we assume a complementary action $\bar{s} \in Sec$ such that $\bar{\bar{s}} = s$, and the same for *Obs*. The silent action τ does not have a complementary action, so the notation $\bar{a}$ will imply that $a \in Sec$ or $a \in Obs$.

The syntax of CCS_p is the following:

T	$::=$		*process term*
		$\sum_i p_i T_i$	*probabilistic choice*
	$\mid$	$\lfloor + \rfloor_i s_i.T_i$	*secret choice* $(s_i \in Sec)$
	$\mid$	$\lfloor + \rfloor_i r_i.T_i$	*nondeterministic choice* $(r_i \in Obs \cup \{\tau\})$
	$\mid$	$T \parallel T$	*parallel composition*
	$\mid$	$(\nu a)\, T$	*restriction*
	$\mid$	$!\, T$	*replication.*

All the summations in the syntax are finite. We will use the notation $T_1 \oplus_p T_2$ to represent a binary probabilistic choice $\sum_i p_i T_i$ with $p_1 = p$ and $p_2 = 1 - p$. Similarly, we will use $a_1.T_1 \lfloor + \rfloor a_2.T_2$ to represent a binary secret or non-deterministic choice. For the parallel operator, we use the symbol ' $\parallel$ ' instead of the more standard ' $\mid$ ' to avoid confusion with the notation for conditional probability.

https://doi.org/10.1017/S0960129514000292 Published online by Cambridge University Press

Table 1. *The semantics of CCS$_p$.*

$$\text{PROB}\quad \frac{}{\sum_i p_i\, T_i \xrightarrow{\tau} \sum_i p_i\, \delta(T_i)} \qquad \text{ACT}\quad \frac{j \in I}{\lfloor + \rfloor_I a_i.T_i \xrightarrow{a_j} \delta(T_j)}$$

$$\text{PAR1}\quad \frac{T_1 \xrightarrow{a} \mu}{T_1 \parallel T_2 \xrightarrow{a} \mu \parallel T_2} \qquad \text{PAR2}\quad \frac{T_2 \xrightarrow{a} \mu}{T_1 \parallel T_2 \xrightarrow{a} T_1 \parallel \mu} \qquad \text{REP}\quad \frac{T \parallel !\,T \xrightarrow{a} \mu}{!\,T \xrightarrow{a} \mu \parallel !\,T}$$

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

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 $\mathcal{T}$ are those derivable from the rules in Table 1. We recall that we use the notation $T \xrightarrow{a} \mu$ stands for $(T, a, \mu) \in \mathcal{T}$. We denote by $\mu \parallel T$ the measure μ' such that $\mu'(T' \parallel T) = \mu(T')$ for all processes T' and $\mu'(T'') = 0$ if T'' is not of the form $T' \parallel T$, and similarly for $T \parallel \mu$. Furthermore, we denote by $(va)\mu$ the measure μ' such that $\mu'((va)T) = \mu(T)$, and $\mu'(T') = 0$ if T' is not of the form $(va)T$.

Note that in the produced probabilistic automaton, all transitions to non-Dirac measures are silent. Note also that a probabilistic term generates exactly one (probabilistic) transition.

A transition of the form $T \xrightarrow{a} \delta(T')$, i.e. a transition having for target a Dirac measure, corresponds to a transition of a non-probabilistic automaton (a standard labelled transition system). Thus, all the rules of CCS$_p$ specialize to the ones of CCS except from PROB. The latter models the internal probabilistic choice: a silent τ transition is available from the sum to a measure containing all of its operands, with the corresponding probabilities.

A secret choice $\lfloor + \rfloor_i s_i.T_i$ produces the same transitions as the non-deterministic term $\lfloor + \rfloor_i r_i.T_i$, except for the labels.

The distinction between the two kinds of labels influences the notion of scheduler for CCS$_p$: the secret actions are assumed to be *inputs* of the system. Namely, we assume that the process receives in input a sequence of secret actions, and a secret action in the process can only be performed if it matches the next secret action in the input (this match 'consumes' the input). 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 to resolve only the residual non-determinism which derives from the non-deterministic choices and from the interleavings.

In the following, we use the notation $X \rightharpoonup Y$ to represent the partial functions from X to Y, and $\alpha_{|Sec}$ represents the projection of α on Sec.

Definition 3.1. Let T be a process in CCS$_p$ and $\mathcal{M}$ be the probabilistic automaton generated by T. A scheduler is a function

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

https://doi.org/10.1017/S0960129514000292 Published online by Cambridge University Press

such that,

if (i) $s = s_1 s_2 \ldots s_n$ and $\alpha_{|Sec} = s_1 s_2 \ldots s_m$ with $m \leqslant n$,

and (ii) there exists a transition $(lst(\alpha), a, \mu)$ such that either $a \notin Sec$,

or ($a \in Sec$ and $a = s_{m+1}$),

then $\zeta(s)(\alpha)$ is defined, and it is one of the transitions that satisfy (ii).

We will write $\zeta_s(\alpha)$ for $\zeta(s)(\alpha)$.

Note that this definition of scheduler is different from the one used in probabilistic automaton, where the scheduler can decide to stop, even if a transition is allowed. Here, the scheduler must proceed whenever a transition is allowed (provided that if it is labelled by a secret, that secret is the next one in the input string s). This view is **in-line with the standard operational semantics of CCS, where all enabled transitions are 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.

Definition 3.2. Let $\mathcal{M} = (St, T, Act, \mathcal{T})$ 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 $\mathcal{M}' = (St', T, Act, \mathcal{T}')$ such that $St' \subseteq exec(\mathcal{M})$, and $(\alpha, a, \mu') \in \mathcal{T}'$ if and only if $\zeta_s(\alpha) = (lst(\alpha), a, \mu)$ for some μ, and $\mu'(\alpha a T) = \mu(T)$.

Note that we do not require the input s to be completely consumed during the execution. Typically, the execution consumes only the prefix of s which is to proceed at each step in which a secret action is involved.

4. Modelling 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. An extended example is presented in Section 7.

4.1. *Protocols as channels*

We view protocols as *channels* in the information-theoretic sense (Cover and Thomas 1991). 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 $\vec{\pi} = (\pi_1, \pi_2, \ldots, \pi_m)$, where π_i is the probability of the ith element of $\mathcal{S}$.

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 non-deterministic

https://doi.org/10.1017/S0960129514000292 Published online by Cambridge University Press

behaviour, 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 output can be considered *random variables*. We will denote these random variables by S and O.

If the protocol contains some form of non-determinism, like internal components giving rise to different interleaving and interactions, then the behaviour 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 non-determinism 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.

4.2. *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 consider sequences as inputs instead of single actions, in order to be more general, and to account for the interactive nature of a process. Furthermore, when the operational semantics of a process calculus is expressed in terms of a labelled transition system (and this is necessary in our context in order to achieve the compositionally results), it is customary to use sequences of labels to represent the input/output behaviour. On the other hand, the standard notion of information-theoretic channel has only one input and one output, hence we need to regard the whole sequence of secret actions as one secret (the input of the channel), and analogously for the output.

We assume that only a finite set of finite 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. For the sake of generality, we allow the element of $\mathcal{S}$ (resp. $\mathcal{O}$) to have different lengths. Thus, $\mathcal{S} \subseteq_{fin} Sec^*$ and $\mathcal{O} \subseteq_{fin} Obs^*$.

Definition 4.1. Given a term T and a scheduler $\zeta : \mathcal{S} \to exec^*(\mathcal{M}) \to \mathcal{T}$, the matrix $M_\zeta(T)$ associated with 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 on Obs is o.

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

https://doi.org/10.1017/S0960129514000292 Published online by Cambridge University Press

Remark 4.2. Given a sequence $s = s_1 s_2 \dots s_h$, consider the term

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

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

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

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

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

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 on Obs gives o.

5. 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 5.1. 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 (Cover and Thomas 1991)). 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 5.2 (Cover and Thomas 1991). The probability of error with respect to a decision function f 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 with the so-called MAP rule, which prescribes choosing the hypothesis s which has MAP (for a given $o \in \mathcal{O}$),

https://doi.org/10.1017/S0960129514000292 Published online by Cambridge University Press

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 Cover and Thomas (1991) 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 the *Bayes risk*. We will denote it by $\mathcal{P}_{MAP}(M, \vec{\pi})$. The following characterization is an immediate consequence of Definition 5.2 and of the Bayes theorem $p(s|o) = p(o|s)\pi_s/p(o)$.

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

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

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

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

Definition 5.3. Consider a CCS_p process T, and let $\mathcal{M}$ be the probabilistic automaton generated by T. We define a distance d between schedulers in $\mathcal{A}$ as follows:

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

where $|\alpha|$ represents the length of α.

Note that $\mathcal{M}$ is finitely branching, both in the non-deterministic 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 5.4. $(\mathcal{A}, d)$ is a *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 5.5. For every CCS_p process T, we have that there exists a minimizing $\zeta_{\mathrm{m}} \in \mathcal{A}$ such that

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

Proof. By Proposition 5.4, $(\mathcal{A}, d)$ is compact. Furthermore, $\mathcal{P}_{MAP}(M_\zeta(T), \vec{\pi})$ is a continuous function from $(\mathcal{A}, d)$ to $([0,1], d')$, where d' is the standard distance on real numbers. In fact, since $\mathcal{S}$ and $\mathcal{O}$ are finite sets of finite sequences, we have that for every T there exists a δ such that $d(\zeta, \zeta') < \delta$ implies $M_\zeta(T) = M_{\zeta'}(T)$. Consequently, $(\{\mathcal{P}_{MAP}(M_\zeta(T), \vec{\pi}) \mid \zeta \in \mathcal{A}\}, d')$ is also compact, and, since it represents a set of probabilities,

https://doi.org/10.1017/S0960129514000292 Published online by Cambridge University Press

it is bounded, too. The rest follows from the fact that compact and bounded subsets of reals are closed, and hence they contain their infima. □

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

Definition 5.6. Given a CCS_p process T, the protection $Pt_{MAP}(T)$ provided by T, in the Bayesian approach, is given by

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

We conclude this section with some properties of $\mathcal{P}_{MAP}$. The next proposition shows that the probabilities of error are *concave* functions with respect to the space of matrices.

Proposition 5.7. 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 $0 \leqslant c_i \leqslant 1$ for all $i \in I$, and $\sum_{i \in I} c_i \;=\; 1$. Then,

$$\mathcal{P}_{MAP}\left(\sum_{i \in I} c_i\, M_i, \vec{\pi}\right) \;\geqslant\; \sum_{i \in I} c_i\, \mathcal{P}_{MAP}(M_i, \vec{\pi}).$$

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

$$\begin{aligned}
\mathcal{P}_{MAP}(\textstyle\sum_i c_i\, M_i, \vec{\pi}) \;&=\; 1 - \textstyle\sum_o \max_s(\sum_i c_i\, p_i(o|s)\pi_s) \\
&\geqslant\; 1 - \textstyle\sum_o \sum_i c_i\, \max_s(p_i(o|s)\pi_s) \\
&=\; 1 - \textstyle\sum_i \sum_o c_i\, \max_s(p_i(o|s)\pi_s) \;\text{ (since the summands are positive)} \\
&=\; 1 - \textstyle\sum_i c_i\, \sum_o \max_s(p_i(o|s)\pi_s) \\
&=\; \textstyle\sum_{i \in I} c_i - \sum_{i \in I} c_i\, \sum_{o \in \mathcal{O}} \max_s(p_i(o|s)\pi_s) \;\text{ (since } \textstyle\sum_{i \in I} c_i \;=\; 1) \\
&=\; \textstyle\sum_{i \in I} c_i(1 - \sum_{o \in \mathcal{O}} \max_s(p_i(o|s)\pi_s)) \\
&=\; \textstyle\sum_{i \in I} c_i \mathcal{P}_{MAP}(M_i, \vec{\pi}).
\end{aligned}$$

□

Corollary 5.8. 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,:

$$\mathcal{P}_{MAP}\left(\sum_{i \in I} c_i\, M_i, \vec{\pi}\right) \;\geqslant\; \min_{i \in I}\; \mathcal{P}_{MAP}(M_i, \vec{\pi}).$$

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 5.9. 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

https://doi.org/10.1017/S0960129514000292 Published online by Cambridge University Press

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,

$$\mathcal{P}_{MAP}(M', \vec{\pi}) \geqslant \mathcal{P}_{MAP}(M, \vec{\pi}).$$

Proof. The result derives from:

$$\begin{aligned} \sum_{o' \in \mathcal{O}'} \max_s (p'(o'|s)\pi_s) &= \sum_{o' \in \mathcal{O}'} \max_s (\sum_{f(o)=o'} p(o|s)\pi_s) \\ &\leqslant \sum_{o' \in \mathcal{O}'} \sum_{f(o)=o'} \max_s (p(o|s)\pi_s) \\ &= \sum_{o \in \mathcal{O}} \max_s (p(o|s)\pi_s). \end{aligned}$$

□

The following propositions are from the literature.

Proposition 5.10 (Chatzikokolakis *et al.* (2008a)). 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:

$$\mathcal{P}_{MAP}(M, \vec{\pi}) \geqslant \mathcal{P}_{MAP}(M', \vec{\pi}).$$

Proposition 5.11 (Bhargava and Palamidessi 2005). 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 (Chaum 1988) and was called *conditional anonymity* by Halpern and O'Neill (2005).

6. 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 6.1. Consider 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 outside $o_1, o_2, \ldots, o_k$.

https://doi.org/10.1017/S0960129514000292 Published online by Cambridge University Press

The following theorem states our main results for Pt_{MAP}. Note that they are also valid for Pt_{ML}, because $Pt_{ML}(T) = Pt_{MAP}(T, \vec{\pi}_u)$. In the following, given a function $f : X \to \mathbb{R}$ for which an element x_{m} exists such that $f(x_{\mathrm{m}}) = \min_{x \in X} f(x)$, we will denote such x_{m} by $\mathrm{minarg}_{x \in X} f(x)$ or alternatively by $\mathrm{minarg}_X f(x)$.

Theorem 6.2. The probabilistic choice, the non-deterministic 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) $$Pt_{MAP}\left(\sum_i p_i\, T_i, \vec{\pi}\right) \geqslant \sum_i p_i\, Pt_{MAP}(T_i, \vec{\pi}) \geqslant \min_i Pt_{MAP}(T_i, \vec{\pi})$$

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

(3) $$Pt_{MAP}((\nu\ o_1, o_2, \ldots, o_k)(T_1 \parallel T_2)) \geqslant Pt_{MAP}(T_2, \vec{\pi})$$
if T_1 is safe outside $o_1, o_2, \ldots, o_k$.

Proof.

1. By definition $Pt_{MAP}(\sum_i p_i\, T_i, \vec{\pi}) = \min_{\zeta \in \mathcal{A}} \mathcal{P}_{MAP}(M_\zeta(\sum_i p_i\, T_i), \vec{\pi})$.
   Let $\zeta_{\mathrm{m}} = \mathrm{minarg}_{\mathcal{A}} \mathcal{P}_{MAP}(M_\zeta(\sum_i p_i\, T_i), \vec{\pi})$. Hence,
   $$Pt_{MAP}\left(\sum_i p_i\, T_i, \vec{\pi}\right) = \mathcal{P}_{MAP}\left(M_{\zeta_{\mathrm{m}}}\left(\sum_i p_i\, T_i\right), \vec{\pi}\right).$$
   Consider, for each i, the scheduler $\zeta_{\mathrm{m}i}$ defined as ζ_{m} on the ith branch, except for the removal of the first state and the first τ-step from the execution fragments in the domain. It is easy to see that
   $$M_{\zeta_{\mathrm{m}}}\left(\sum_i p_i\, T_i\right) = \sum_i p_i M_{\zeta_{\mathrm{m}i}}(T_i).$$
   From Proposition 5.7 we derive
   $$\mathcal{P}_{MAP}\left(M_{\zeta_{\mathrm{m}}}\left(\sum_i p_i\, T_i\right), \vec{\pi}\right) \geqslant \sum_i p_i \mathcal{P}_{MAP}(M_{\zeta_{\mathrm{m}i}}(T_i), \vec{\pi}).$$
   Finally, observe that $\zeta_{\mathrm{m}i}$ is still compatible with $\mathcal{S}$, hence we have
   $$\mathcal{P}_{MAP}(M_{\zeta_{\mathrm{m}i}}(T_i), \vec{\pi}) \geqslant Pt_{MAP}(T_i, \vec{\pi}) \quad \text{for all } i,$$
   which concludes the proof in this case.
2. Let $\zeta_{\mathrm{m}} = \mathrm{minarg}_{\mathcal{A}} \mathcal{P}_{MAP}(M_\zeta(\lfloor + \rfloor_i o_i.T_i), \vec{\pi})$. Let $\mathcal{A}_i$ be the class of schedulers that choose the ith branch at the beginning of the execution, and define
   $$\zeta_{\mathrm{n}i} = \min_{\mathcal{A}_i} \mathrm{arg}\, \mathcal{P}_{MAP}\left(M_\zeta(\Big\lfloor + \Big\rfloor_i o_i.T_i), \vec{\pi}\right).$$
   Obviously, we have
   $$Pt_{MAP}\left(\Big\lfloor + \Big\rfloor_i o_i.T_i, \vec{\pi}\right) = \min_i \mathcal{P}_{MAP}(M_{\zeta_{\mathrm{n}i}}\left(\Big\lfloor + \Big\rfloor_i o_i.T_i\right), \vec{\pi}).$$

https://doi.org/10.1017/S0960129514000292 Published online by Cambridge University Press

Consider now, for each i, the scheduler $\zeta_{\mathrm{m}i}$ defined as $\zeta_{\mathrm{n}i}$, except for the removal of the first state and the first step from the execution fragments in the domain. Obviously, $\zeta_{\mathrm{m}i}$ is still compatible with $\mathcal{S}$, and the observables of T_i are i one-to-one correspondence with those of $\lfloor + \rfloor_i\, o_i.T_i$ via the bijective function $f_i(o_i o_{j_1} \dots o_{j_k}) = o_{j_1} \dots o_{j_k}$. Furthermore, all the probabilities of the channel $M_{\zeta_{ni}}(\lfloor + \rfloor_i\, o_i.T_i)$ are the same as those of $M_{\zeta_{mi}}(T_i)$ modulo the renaming of o into $f(o)$. Hence, we have

$$\mathcal{P}_{MAP}\left(M_{\zeta_{\mathrm{n}i}}\left(\left\lfloor + \right\rfloor_i o_i.T_i\right), \vec{\pi}\right) \;=\; \mathcal{P}_{MAP}(M_{\zeta_{\mathrm{m}i}}(T_i), \vec{\pi}) \;=\; Pt_{MAP}(T_i, \vec{\pi}),$$

which concludes the proof of this case.

3. Let $\zeta_{\mathrm{m}} = \mathrm{minarg}_{\mathcal{A}}\, \mathcal{P}_{MAP}(M_{\zeta}((v o_1, o_2, \dots, o_k)(T_1 \parallel T_2)), \vec{\pi})$. Hence,

$$Pt_{MAP}((v o_1, o_2, \dots, o_k)(T_1 \parallel T_2), \vec{\pi}) \;=\; \mathcal{P}_{MAP}(M_{\zeta_{\mathrm{m}}}((v o_1, o_2, \dots, o_k)(T_1 \parallel T_2)), \vec{\pi}).$$

The proof proceeds by constructing a set of series of schedulers whose limit with respect to the metric d in Definition 5.3 correspond to schedulers on the execution tree of T_2. Consider a generic node in the execution tree of $(v o_1, o_2, \dots, o_k)(T_1 \parallel T_2)$ under ζ_{m}, and let $(v o_1, o_2, \dots, o_k)(T_1' \parallel T_2')$ be the new term in that node. Assume α to be the execution history up to that node. Let us consider separately the three possible kinds of transitions derivable from the operational semantics:

a. $(v o_1, o_2, \dots, o_k)(T_1' \parallel T_2') \xrightarrow{a} (v o_1, o_2, \dots, o_k)(\mu \parallel T_2')$ due to a transition $T_1' \xrightarrow{a} \mu$. In this case, a must be τ because of the assumption that T_1 does not contain secret actions and all its observable actions are included in $\{o_1, o_2, \dots, o_k\}$. Assume that $\mu = \sum_i p_i\, \delta(T_{1i}')$. Then, we have

$$(v o_1, o_2, \dots, o_k)(\mu \parallel T_2') = \sum_i p_i\, \delta((v o_1, o_2, \dots, o_k)(T_{1i}' \parallel T_2')).$$

Let us now consider the tree obtained by replacing the above distribution by $\delta((v o_1, o_2, \dots, o_k)(T_{1i}' \parallel T_2'))$ (i.e., the tree obtained by pruning all alternatives except $(v o_1, o_2, \dots, o_k)(T_{1i}' \parallel T_2')$, and assigning to it probability 1). Let $\zeta_{\mathrm{m}i}$ be the projection of ζ_{m} on the new tree (i.e., $\zeta_{\mathrm{m}i}$ is defined as the projection of ζ_{m} on the histories α' such that if α is a proper prefix of α' then $\alpha\tau(v o_1, o_2, \dots, o_k)(T_{1i}' \parallel T_2')$ is a prefix of α'). We have

$$\begin{aligned}
&\mathcal{P}_{MAP}(M_{\zeta_{\mathrm{m}}}((v o_1, o_2, \dots, o_k)(T_1 \parallel T_2)), \vec{\pi}) \\
&\quad= \mathcal{P}_{MAP}(\textstyle\sum_i p_i\, M_{\zeta_{\mathrm{m}i}}((v o_1, o_2, \dots, o_k)(T_1 \parallel T_2)), \vec{\pi}) \\
&\quad\geqslant \text{(by Proposition 5.7)} \\
&\qquad\quad \times \textstyle\sum_i p_i\, \mathcal{P}_{MAP}(M_{\zeta_{\mathrm{m}i}}((v o_1, o_2, \dots, o_k)(T_1 \parallel T_2)), \vec{\pi}).
\end{aligned}$$

In the execution tree of T_2, the above transition does not have a correspondent, but it obliges us to consider all different schedulers that are associated to the various $\zeta_{\mathrm{m}i}$'s for different i's.

https://doi.org/10.1017/S0960129514000292 Published online by Cambridge University Press

b. $(vo_1, o_2, \ldots, o_k)(T_1' \parallel T_2') \xrightarrow{a} (vo_1, o_2, \ldots, o_k)(T_1' \parallel \mu)$ 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 must choose the same transition, i.e., $T_2' \xrightarrow{a} \mu$.

c. $(vo_1, o_2, \ldots, o_k)(T_1' \parallel T_2') \xrightarrow{\tau} (vo_1, o_2, \ldots, o_k)\delta(T_1'' \parallel T_2'')$ 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 sequences of schedulers of the form $\{\zeta_{\mathrm{m}}, \zeta_{\mathrm{m}i}, \zeta_{\mathrm{m}ij}, \zeta_{\mathrm{m}ijh}, \ldots\}$, which behave like ζ_{m} until the node $(vo_1, o_2, \ldots, o_k)(T_1' \parallel T_2')$, and correspond to paths of increasing depth in the subtree of that node. Let $p_i, q_j, r_h \ldots$ the probabilities of the branches selected by $\zeta_{\mathrm{m}i}, \zeta_{\mathrm{m}ij}, \zeta_{\mathrm{m}ijh}, \ldots$ at each next choice point (which corresponds to a choice point with the same probability in the execution of T_2). We have:

$$\begin{aligned}
&\mathcal{P}_{MAP}(M_{\zeta_{\mathrm{m}}}((vo_1, o_2, \ldots, o_k)(T_1 \parallel T_2)), \vec{\pi}) \\
&\quad \geqslant \textstyle\sum_i p_i \, \mathcal{P}_{MAP}(M_{\zeta_{\mathrm{m}i}}((vo_1, o_2, \ldots, o_k)(T_1 \parallel T_2)), \vec{\pi}) \\
&\quad \geqslant \textstyle\sum_i p_i \sum_j q_j \, \mathcal{P}_{MAP}(M_{\zeta_{\mathrm{m}ij}}((vo_1, o_2, \ldots, o_k)(T_1 \parallel T_2)), \vec{\pi}) \\
&\quad \geqslant \textstyle\sum_i p_i \sum_j q_j \sum_h r_h \, \mathcal{P}_{MAP}(M_{\zeta_{\mathrm{m}ijh}}((vo_1, o_2, \ldots, o_k)(T_1 \parallel T_2)), \vec{\pi}) \\
&\quad \geqslant \ldots
\end{aligned}$$

Observe now that for every $i, j, h, \ldots$ $\{\zeta_{\mathrm{m}}, \zeta_{\mathrm{m}i}, \zeta_{\mathrm{m}ij}, \zeta_{\mathrm{m}ijh}, \ldots\}$ is a converging series of schedulers whose limit $\zeta_{\mathrm{m}ijh\ldots}$ 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 the observables. Hence, Proposition 5.9 applies, and we have:

$$\begin{aligned}
&\mathcal{P}_{MAP}(M_{\zeta_{\mathrm{m}}}((vo_1, o_2, \ldots, o_k)(T_1 \parallel T_2)), \vec{\pi}) \\
&\quad \geqslant \textstyle\sum_i p_i \sum_j q_j \sum_h r_h \ldots \mathcal{P}_{MAP}(M_{\zeta_{\mathrm{m}ijh\ldots}}((vo_1, o_2, \ldots, o_k)(T_1 \parallel T_2)), \vec{\pi}) \\
&\quad \geqslant \text{(by Proposition 5.9)} \\
&\qquad \times \textstyle\sum_i p_i \sum_j q_j \sum_h r_h \ldots \mathcal{P}_{MAP}(M_{\zeta_{\mathrm{m}ijh\ldots}}(T_2), \vec{\pi}) \\
&\quad \geqslant \textstyle\sum_i p_i \sum_j q_j \sum_h r_h \ldots \min_{\zeta \in \mathcal{A}} \mathcal{P}_{MAP}(M_{\zeta}(T_2), \vec{\pi}).
\end{aligned}$$

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

$$\mathcal{P}_{MAP}(M_{\zeta_{\mathrm{m}}}((vo_1, o_2, \ldots, o_k)(T_1 \parallel T_2)), \vec{\pi}) \geqslant \min_{\zeta \in \mathcal{A}} \mathcal{P}_{MAP}(M_{\zeta}(T_2), \vec{\pi}),$$

which concludes the proof. □

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

https://doi.org/10.1017/S0960129514000292 Published online by Cambridge University Press

Example 6.3. Let $Sec = \{s_1, s_2\}$ and assume that $\mathcal{S}$ does not contain the empty sequence. Let $T = o_1.0 \boxplus o_2.0$. For all sequences $s, s' \in \mathcal{S}$ we have $p(o_1|s) = p(o_1|s')$ and $p(o_2|s) = p(o_1|s')$. Hence, by Proposition 5.10 we have $Pt_{MAP}(T, \vec{\pi}) = 1 - \max_s \pi_s$. Consider now $T' = s_1.T \boxplus s_2.T$. Let us define a scheduler that, if the secret starts with s_1 selects o_1, and if the secret starts with s_2 selects o_2. We note that, 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, $Pt_{MAP}(T', \vec{\pi}) = 1 - p_1 - p_2$, 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 $\max_s \pi_s = p_1$ or $\max_s \pi_s = p_2$ because of the 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 $1 - p_1 - p_2 < 1 - \max_s \pi_s$, hence $Pt_{MAP}(T', \vec{\pi}) < Pt_{MAP}(T, \vec{\pi})$.

The reason why we need the condition *(i)* in Definition 6.1 for the parallel operator is analogous to the case of secret choice. The following is a counterexample.

Example 6.4. Let *Sec* and $\mathcal{S}$ be as in Example 6.3. Define $T_1 = s_1.0 \boxplus s_2.0$ and $T_2 = o_1.0 \boxplus o_2.0$. Again, we have $Pt_{MAP}(T_2, \vec{\pi}) = 1 - \max_s \pi_s$. Consider now the term $T_1 \parallel T_2$ 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 6.3, where $T' = T_1 \parallel T_2$ and $T = T2$.

The reason why we need the condition *(ii)* in Definition 6.1 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.

Example 6.5. Let *Sec* and $\mathcal{S}$ be as in Example 6.3. 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 $Pt_{MAP}(T_2, \vec{\pi}) = 1 - \max_s \pi_s$. Consider the term $T_1 \parallel T_2$ 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, $p(oo_1|s_1 s) = p(oo_2|s_1 s) = .5$ and also $p(o_1 o|s_2 s) = p(o_2 o|s_2 s) = .5$ while $p(oo_1|s_2 s) = p(oo_2|s_2 s) = 0$ and $p(o_1 o|s_1 s) = p(o_2 o|s_1 s) = 0$. Therefore, $Pt_{MAP}(T, \vec{\pi}) = 1 - p_1 - p_2$, where p_1 and p_2 are the maximum probabilities of the secrets of the form $s_1 s$ and $s_2 s$, respectively. Following the same reasoning as in Example 6.3, we have that $Pt_{MAP}(T', \vec{\pi}) < Pt_{MAP}(T, \vec{\pi})$.

7. A case study: the dining cryptographers

In this section, we consider the DC protocol proposed by Chaum (1988), we show how to describe it in CCS_p, and we apply the results of the previous section to obtain a generalization of Chaum's strong anonymity result.

In its most general formulation, the DC consists of a multigraph where one of the nodes (cryptographers) may be secretly designated to pay for the dinner. The cryptographers would like to find out whether there is a payer or not, but without either discovering the identity of the payer, nor revealing it to an external observer. The problem can be solved as follows: we put on each edge a probabilistic coin, with two possible, mutually exclusive results, 0 or 1. The coins are tossed, and each cryptographer computes the binary sum of

https://doi.org/10.1017/S0960129514000292 Published online by Cambridge University Press

all (the results of) the adjacent coins. Furthermore, it adds 1 if it is designated to be the payer. Finally, all the cryptographers declare their result.

It is easy to see that this protocol solves the problem of figuring out the existence of a payer: the binary sum of all declarations is 1 if and only if there is a payer, because all the coins are counted twice, so their contribution to the total sum is 0.

The property we are interested in, however, is the anonymity of the system. Chaum proved that the DC is strongly anonymous if all the coins are fair, i.e., they give 0 and 1 with equal probability, and the 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 7.1 (Chaum 1988). If the 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 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. The new notation is just syntactic sugar, in the sense that it is expressed in terms of the constructs of CCS_p.

$$\begin{aligned} \mathit{Input} \quad c(x).T &= \Big\lfloor + \Big\rfloor_{v} c_v.T[v/x] \\ \mathit{Output} \quad \bar{c}\langle v\rangle &= \bar{c}_v. \end{aligned}$$

The protocol, represented in Table 2, is defined as the parallel composition of the cryptographers processes $Crypt_i$, of the coin processes $Coin_h$, and of a process *Collect* whose purpose is to collect all the declarations of the cryptographers, and output them in the form of a tuple. The reason for using this process *Collect* is that in the original Chaum protocol all cryptographers are supposed to make their declaration at the same time. This is important because otherwise the schedule could reveal the identity of the payer by scheduling his declaration first, for example. However in CCS_p, concurrency is resolved by interleaving, and it is not possible to enforce the simultaneity of a set of actions (with the exception of a pair of complementary actions in a synchronisation step). Hence, we need to simulate the simultaneous declarations by means of a single process (*Collect*) that collects the results into a tuple and then declares the tuple in a single step, and by internalizing the interleavings corresponding to the possible orders in which the results are transmitted by the cryptographers to *Collect*. Since these interleavings are internalized (i.e. they do not produce any observable action), their order does not matter: they are all equivalent (namely, invisible) from the point of view of the observer. In this protocol, the secret actions are pay_i. All the others are observable actions.

Each coin communicates with two cryptographers. $c_{i,h}$ represents the communication channel between $Coin_h$ and $Crypt_i$ if h is indeed the index of a coin, otherwise it represents a communication channel 'with the environment'. We will call the latter *external*. 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 test the system, for the purpose of discovering the secret.

https://doi.org/10.1017/S0960129514000292 Published online by Cambridge University Press

Table 2. *The dining cryptographers protocol expressed in CCS_p.*

$$
\begin{aligned}
Crypt_i &= c_{i,i_1}(x_1).\ldots.c_{i,i_k}(x_k).pay_i(x).\bar{d}_i\langle x_1+\ldots+x_k+x\rangle \\
Coin_h &= \bar{c}_{\ell,h}\langle 0\rangle.\bar{c}_{r,h}\langle 0\rangle.0 \ \oplus_{p_h}\ \bar{c}_{\ell,h}\langle 1\rangle.\bar{c}_{r,h}\langle 1\rangle.0 \\
Collect &= d_1(y_1).d_2(y_2).\ldots.d_n(y_n).\overline{out}\langle y_1,y_2,\ldots,y_n\rangle \\
DC &= (v\vec{c})(v\vec{d})(\textstyle\prod_i Crypt_i \parallel \prod_h Coin_h \parallel Collect)
\end{aligned}
$$

We are now ready to state our generalization of Chaum's result:

Theorem 7.2. A DC is strongly anonymous if it has a spanning tree consisting of fair coins only.

Proof. Consider the term DC in Table 2. 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. Since all the remaining coins are fair, this situation corresponds to the original formulation of Chaum and so we can apply Chaum's result (Theorem 7.1) and Proposition 5.11 to conclude that all the rows of the matrix M are the same and hence, by Proposition 5.10, $\mathcal{P}_{MAP}(M,\vec{\pi}) = 1 - \max_i \pi_i$.

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 channels. 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 5.9 in both directions and derive that $\mathcal{P}_{MAP}(M',\vec{\pi}) = 1 - \max_i \pi_i$.

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:

$$T' = (vc_{\ell,h}c_{r,h})(Coin_h \parallel T).$$

By applying Theorem 6.2 we can deduce that

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

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

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

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

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

which concludes the proof. □

https://doi.org/10.1017/S0960129514000292 Published online by Cambridge University Press

Interestingly, also the other direction of Theorem 7.2 holds. We report this result for completeness, however we have proved it by using traditional methods, not by applying the compositional methods of Section 6.

Theorem 7.3. A DC is strongly anonymous only if it 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 $m \geqslant 1$.

Let a_1 be a vertex in G_1 and a_2 be a vertex in G_2. By our assumption of strong anonymity, for every observable o (which, we recall, is a vector of bits representing the declarations of the cryptographers) we have

$$p(o \mid a_1) = p(o \mid a_2). \tag{1}$$

Here, a_1, resp. a_2, represents the event that the cryptographer a_1, resp. a_2, is the payer.

Observe now that, if w is a binary vector of dimension n containing 1 exactly twice, in correspondence of a_1 and a_2, then $p(o \mid a_1) = p(o + w \mid a_2)$, where $o + w$ is the vector resulting from the **component wise binary sum of o and w. This is because by adding 1 to their declarations, we invert the payer/non-payer behaviour of a_1 and a_2. Thus, by adding w to the declarations in the case in which a_2 is the payer (and a_1 is a non-payer) we obtain the same result as the case in which a_1 is the payer (and a_2 is a non-payer). Hence, (1) becomes

$$p(o + w \mid a_2) = p(o \mid a_2). \tag{2}$$

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 (2), being valid for all o's, implies

$$p(d + 1 \mid a_2) = p(d \mid a_2) \tag{3}$$

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 (3), and the fact that the coins are independent from the choice of the payer, we derive

$$p\Big(\sum_{i=1}^{m} c_i = 0\Big) = p\Big(\sum_{i=1}^{m} c_i = 1\Big) = 1/2. \tag{4}$$

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 $(c_1, c_2, \ldots, c_{m-1})$.

https://doi.org/10.1017/S0960129514000292 Published online by Cambridge University Press

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 (4) we derive

$$p\left(\sum_{i=1}^{m-1} c_i = 0\right) p(c_m = 0) + p\left(\sum_{i=1}^{m-1} c_i = 1\right) p(c_m = 1) = 1/2. \tag{5}$$

Now, it is easy to see that (5) 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. □

8. Related work

In the field of information flow there have been various works (Alvim *et al.* 2010; Andrés *et al.* 2010a; Boreale *et al.* 2011; Clark *et al.* 2001, 2005; Heusser and Malacaria 2009; Gray III 1991; Lowe 2002; Malacaria 2007; Malacaria and Chen 2008; McLean 1990; Smith 2009) in which the *high information* and the *low information* are seen as the input and output, respectively of a (noisy) channel. Information leakage is formalized in this setting as the channel mutual information or the channel capacity. The idea is that the leakage represents the difference between the *a priori* uncertainty about the (secret) high information, and the *a posteriori* uncertainty, after the low information has become publically known. The uncertainty is expressed in terms of entropy, and there are various alternative notions depending on the notion of attack that one wishes to model, as discussed in Köpf and Basin (2007). Most of the above approaches are based either on the Shannon or Rényi min entropy.

Channel capacity has been also used in relation to anonymity in (Moskowitz *et al.* 2003b,a). These works propose a method to create covert communication by means of non-perfect anonymity.

A related line of work is (Díaz *et al.* 2002; Serjantov and Danezis 2002), where the main idea is to express the lack of (probabilistic) information in terms of entropy.

A different information-theoretic approach is taken in Clarkson *et al.* (2009). In this paper, the authors define information leakage as the difference between the *a priori* accuracy of the guess of the attacker, and the *a posteriori* one, after the attacker has made his observation. The accuracy of the guess is defined as the Kullback–Leibler distance between the *belief* (which is a weight attributed by the attacker to each input hypothesis) and the true distribution on the hypotheses. This approach, which was Shannon-based in Clarkson *et al.* (2009), was later applied by Hamadou *et al.* (2010) also to the case of the Rényi min entropy.

The problem of preservation of information protection under program composition was considered also by McIver *et al.* (2010). In that paper, the author define an order $\preceq$ on specifications based on the Bayes risk, and they identify a compositional subset of it: a refinement order $\sqsubseteq$ such that $S \sqsubseteq I$ implies $C[S] \preceq C[I]$ for all contexts C. They also show that $\sqsubseteq$ is the *compositional closure* of $\preceq$, in the sense that $S \not\sqsubseteq I$ only when $C[S] \not\preceq C[I]$ for some C. Finally, they prove that $\sqsubseteq$ is sound for other three, competing notions of elementary test and that therefore Bayes-risk testing, with context, is maximally discriminating among them.

https://doi.org/10.1017/S0960129514000292 Published online by Cambridge University Press

Desharnais *et al.* (2002) defined a notion of metric between probabilistic processes and proved that the Shannon capacity associated to a protocol associated to a process is continuous with respect to this metric.

Deng *et al.* (2006) consider a probabilistic calculus similar to CCS_p, and define a relation between traces based on the Kullback–Leibler divergence. They show that certain constructs of their calculus preserve the relation, in the sense that they do not increase the divergence.

The analysis of quantitative information flow in concurrent programs has also been investigated in purely probabilistic (i.e. non-information-theoretic) frameworks. For instance, Garcia *et al.* (2007) proposed to verify strong probabilistic anonymity by checking the sufficient condition that if two states differ only for the choice of the secret value, then the probabilistic automata rooted in those two states are isomorphic. This idea was further developed by Andrés *et al.* (2010b).

9. Conclusion, discussion and future work

In this paper, we have proposed to use the notion of the Bayes risk to measure the degree of protection offered by an information-hiding protocol. We have investigated CCS_pconstructs that are safe, i.e., that are guaranteed not to decrease the protection. Then, we have applied these results to strengthen a result of Chaum: the DC are strongly anonymous if and only if they have a spanning tree of fair coins.

We recall that the Bayes risk represents the probability of error of an adversary who knows *the exact prior probability distribution*, and therefore he applies the MAP rule exactly. In case the adversary does not know the exact prior, he can still apply (an approximation of) the MAP rule of course, using for instance the uniform distribution, or his best bet, as the prior. Since the MAP rule is optimal, the probability of error in the approximate case is always greater than or equal to the Bayes risk. Hence, the compositional method developed in this paper is still useful to compute a lower bound on the degree of protection.

One natural question is whether we could obtain a better (more precise) bound in the case we know for sure that the adversary is going to use a certain distribution as prior. Typically, if the adversary does not know anything about the prior distribution, he will use the uniform prior. This is equivalent to applying 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 corresponding probability of error $\mathcal{P}_{ML}(M, \vec{\pi})$ can be characterized as follows (as an immediate consequence of Definition 5.2 and od the Bayes theorem).

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

where $p(o|s_o) = \max_s(p(o|s))$.

[†] The name comes from the fact that $p(o|s)$ is called the *likelihood* of s given o.

https://doi.org/10.1017/S0960129514000292 Published online by Cambridge University Press

However, the methods developed in this paper cannot be applied to compute a better lower bound for $\mathcal{P}_{ML}(M, \vec{\pi})$, because Proposition 5.9, which is crucial for the compositionally results, does not hold for $\mathcal{P}_{ML}(M, \vec{\pi})$. We leave the problem of developing a compositional method for computing the risk of ML-based attacks as a topic for future work.

Another problem that we would like to investigate in the future is the extension our results to other constructs of the language, and in particular to a more liberal type of parallel composition. This is not possible in the present setting, as the examples after Theorem 6.2 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. All the examples after Theorem 6.2 are based on this kind of problem. In Chatzikokolakis and Palamidessi (2010), we have studied the problem and we came out with a language-based solution 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.

References

Alvim, M. S., Andrés, M. E. and Palamidessi, C. (2010) Information flow in interactive systems In: Gastin, P. and Laroussinie, F. (eds.) Proceedings of the 21th International Conference on Concurrency Theory (CONCUR 2010). *Springer Lecture Notes in Computer Science* **6269** 102–116.

Andrés, M. E., Palamidessi, C., van Rossum, P. and Smith, G. (2010a) Computing the leakage of information-hiding systems. In: J. Esparza and R. Majumdar, (eds.) Proceedings of the 16th International Conference on Tools and Algorithms for the Construction and Analysis of Systems (TACAS 2010). *Springer Lecture Notes in Computer Science* **6015** 373–389.

Andrés, M. E., Palamidessi, C., van Rossum, P. and Sokolova, A. (2010*b*) Information hiding in probabilistic concurrent systems. In: Proceedings of the 7th IEEE International Conference on Quantitative Evaluation of SysTems (QEST 2010), IEEE Computer Society 17–26.

Bhargava, M. and Palamidessi, C. (2005) Probabilistic anonymity. In: Abadi, M. and de Alfaro, L. (eds.) Proceedings of the 16th International Conference on Concurrency Theory (CONCUR 2005). *Springer Lecture Notes in Computer Science* **3653** 171–185.

Boreale, M., Pampaloni, F. and Paolini, M. (2011) Asymptotic information leakage under one-try attacks. In: Hofmann, M. (ed.) Proceedings of the 14th International Conference on the Foundations of Software Science and Computational Structures (FOSSACS'11). *Springer Lecture Notes in Computer Science* **6604** 396–410.

Chatzikokolakis, K. and Palamidessi, C. (2010) 'Making random choices invisible to the scheduler. *Information and Computation* **208** (6) 694–715.

Chatzikokolakis, K., Palamidessi, C. and Panangaden, P. (2008a) Anonymity protocols as noisy channel. *Information and Computation* **206** (2–4) 378–401.

Chatzikokolakis, K., Palamidessi, C. and Panangaden, P. (2008b) On the Bayes risk in information-hiding protocols. *Journal of Computer Security* **16** (5) 531–571.

Chaum, D. (1988) The dining cryptographers problem: Unconditional sender and recipient untraceability. *Journal of Cryptology* **1** 65–75.

Chung, K. L. (2000) *A Course in Probability Theory*, 2nd edition, Academic Press, New York.

Clark, D., Hunt, S. and Malacaria, P. (2001) Quantitative analysis of the leakage of confidential data. In: Proceedings of the Workshop on Quantitative Aspects of Programming Languages (QAPL 2001). *Electronic Notes in Theoretical Computer Science* **59** (3) 238–251.

https://doi.org/10.1017/S0960129514000292 Published online by Cambridge University Press

Clark, D., Hunt, S. and Malacaria, P. (2005) Quantified interference for a while language. In: Proceedings of the 2nd Workshop on Quantitative Aspects of Programming Languages (QAPL 2004). *Electronic Notes in Theoretical Computer Science* **112** 149–166.

Clarkson, M. R., Myers, A. C. and Schneider, F. B. (2009) Belief in information flow. *Journal of Computer Security* **17** (5) 655–701.

Cover, T. M. and Thomas, J. A. (1991) *Elements of Information Theory*, John Wiley and Sons, Inc.

Deng, Y., Palamidessi, C. and Pang, J. (2005) Compositional reasoning for probabilistic finite-state behaviors. In: Middeldorp, A., van Oostrom, V., van Raamsdonk, F. and de Vrijer, R. C. (eds.) Processes, Terms and Cycles: Steps on the Road to Infinity. *Springer Lecture Notes in Computer Science* **3838** 309–337.

Deng, Y., Pang, J. and Wu, P. (2006) Measuring anonymity with relative entropy. In: Dimitrakos, T., Martinelli, F., Ryan, P. Y. A. and Schneider, S. A. (eds.) Postproceedings of the 4th International Workshop on Formal Aspects in Security and Trust (FAST 2006). *Lecture Notes in Computer Science* **4691** 65–79.

Desharnais, J., Jagadeesan, R., Gupta, V. and Panangaden, P. (2002) The metric analogue of weak bisimulation for probabilistic processes. In: Proceedings of the 17th Annual IEEE Symposium on Logic in Computer Science, IEEE Computer Society 413–422.

Díaz, C., Seys, S., Claessens, J. and Preneel, B. (2002) Towards measuring anonymity. In: Dingledine, R. and Syverson, P. F. (eds.) Proceedings of the workshop on Privacy Enhancing Technologies (PET 2002). *Springer Lecture Notes in Computer Science* **2482** 54–68.

Fujioka, A., Okamoto, T. and Ohta, K. (1993) A practical secret voting scheme for large scale elections. In: Seberry, J. and Zheng, Y. (eds.) *Proceedings of the Workshop on the Theory and Application of Cryptographic Techniques*. Advancaes in Cryptology (AUSCRYPT '92), Springer-Verlag, London, UK 244–251.

Garcia, F. D., van Rossum, P. and Sokolova, A. (2007) Probabilistic anonymity and admissible schedulers arXiv:0706.1019v1.

Halpern, J. Y. and O'Neill, K. R. (2005) Anonymity and information hiding in multiagent systems. *Journal of Computer Security* **13** (3) 483–512.

Hamadou, S., Palamidessi, C. and Sassone, V. (2010) Reconciling belief and vulnerability in information flow. In: *Proceedings of the 31st IEEE Symposium on Security and Privacy*, IEEE Computer Society 79–92.

Herescu, O. M. and Palamidessi, C. (2000) Probabilistic asynchronous π-calculus. In: Tiuryn, J. (ed.) Proceedings of the 3rd International Conference on Foundations of Software Science and Computation Structures (FOSSACS 2000). *Springer Lecture Notes in Computer Science* **1784** 146–160.

Heusser, J. and Malacaria, P. (2009) Applied quantitative information flow and statistical databases. In: Degano, P. and Guttman, J. D. (eds.) Proceedings of the International Workshop on Formal Aspects in Security and Trust (FAST 2009). *Springer Lecture Notes in Computer Science* **5983** 96–110.

Gray, J. W. III (1991) Toward a mathematical foundation for information flow security. In: *Proceedings of the 1991 IEEE Computer Society Symposium on Research in Security and Privacy (SSP '91)*, IEEE Computer Society Press 21–35.

Köpf, B. and Basin, D. A. (2007) An information-theoretic model for adaptive side-channel attacks. In: Ning, P., di Vimercati, S. D. C. and Syverson, P. F. (eds.) *Proceedings of the 2007 ACM Conference on Computer and Communications Security (CCS 2007)*, ACM 286–296.

Lowe, G. (2002) Quantifying information flow. In: *Proceedings of the 15th IEEE Computer Security Foundations Workshop (CSFW 2002)*, IEEE Computer Society Press 18–31.

https://doi.org/10.1017/S0960129514000292 Published online by Cambridge University Press

Malacaria, P. (2007) Assessing security threats of looping constructs. In: Hofmann, M. and Felleisen, M. (eds.) *Proceedings of the 34th ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages (POPL 2007)*, ACM 225–235.

Malacaria, P. and Chen, H. (2008) Lagrange multipliers and maximum information leakage in different observational models. In: Erlingsson, Ú. and Pistoia, M. (eds.) *Proceedings of the 2008 Workshop on Programming Languages and Analysis for Security (PLAS 2008)*, ACM 135–146.

McIver, A., Meinicke, L. and Morgan, C. (2010) Compositional closure for bayes risk in probabilistic noninterference. In: Abramsky, S., Gavoille, C., Kirchner, C., auf der Heide, F. M. and Spirakis, P. G. (eds.) Proceedings of the 37th International Colloquium on Automata, Languages and Programming (ICALP 2010). *Springer Lecture Notes in Computer Science* **6199** 223–235.

McLean, J. (1990) Security models and information flow. In: Proceedings of the 1990 IEEE Symposium on Security and Privacy (SSP'90), IEEE 180–189.

Milner, R. (1989) *Communication and Concurrency*, International Series in Computer Science, Prentice-Hall.

Moskowitz, I. S., Newman, R. E., Crepeau, D. P. and Miller, A. R. (2003b) Covert channels and anonymizing networks.. In: Jajodia, S., Samarati, P. and Syverson, P. F. (eds.) *Workshop on Privacy in the Electronic Society 2003*, ACM 79–88.

Moskowitz, I. S., Newman, R. E. and Syverson, P. F. (2003a) Quasi-anonymous channels. In: Proceedings of the IASTED International Conference on Communication, Network, and Information Security (CNIS 2003). *IASTED* 126–131.

Munkres, J. R. (2000) *Topology: A First Course*, 2nd edition, Prentice-Hall.

Palamidessi, C. and Herescu, O. M. (2005) A randomized encoding of the π-calculus with mixed choice. *Theoretical Computer Science* **335** (2–3) 373–404.

Reiter, M. K. and Rubin, A. D. (1998) Crowds: Anonymity for web transactions. *ACM Transactions on Information and System Security* **1** (1) 66–92.

Rényi, A. (1961) On measures of entropy and information. In: *Proceedings of the 4th Berkeley Symposium on Mathematics, Statistics, and Probability* 547–561.

Segala, R. (1995) Modeling and verification of randomized distributed real-time systems, Ph.D. thesis. *Technical Report MIT/LCS/TR-676.*

Segala, R. and Lynch, N. (1995) Probabilistic simulations for probabilistic processes. *Nordic Journal of Computing* **2** (2) 250–273.

Serjantov, A. and Danezis, G. (2002) Towards an information theoretic metric for anonymity.. In: Dingledine, R. and Syverson, P. F. (eds.) Proceedings of the Workshop on Privacy Enhancing Technologies (PET 2002). *Springer Lecture Notes in Computer Science* **2482** 41–53.

Smith, G. (2009) On the foundations of quantitative information flow. In: de Alfaro, L. (ed.) Proceedings of the 12th International Conference on Foundations of Software Science and Computation Structures (FOSSACS 2009). *Springer Lecture Notes in Computer Science* **5504** 288–302.

https://doi.org/10.1017/S0960129514000292 Published online by Cambridge University Press