Probabilistic Anonymity*

Mohit Bhargava[1]** and Catuscia Palamidessi[2]

[1] Indian Institute of Technology Delhi
[2] INRIA Futurs and LIX, École Polytechnique

Abstract. The concept of anonymity comes into play in a wide range of situations, varying from voting and anonymous donations to postings on bulletin boards and sending mails. The systems for ensuring anonymity often use random mechanisms which can be described probabilistically, while the agents' interest in performing the anonymous action may be totally unpredictable, irregular, and hence expressable only nondeterministically.
Formal definitions of the concept of anonymity have been investigated in the past either in a totally nondeterministic framework, or in a purely probabilistic one. In this paper, we investigate a notion of anonymity which combines both probability and nondeterminism, and which is suitable for describing the most general situation in which both the systems and the user can have both probabilistic and nondeterministic behavior. We also investigate the properties of the definition for the particular cases of purely nondeterministic users and purely probabilistic users.
We formulate our notions of anonymity in terms of observables for processes in the probabilistic π-calculus, whose semantics is based on Probabilistic Automata. We illustrate our ideas by using the example of the dining cryptographers.

1 Introduction

The concept of *anonymity* comes into play in those cases in which we want to keep secret the identity of the agent participating to a certain event. There is a wide range of situations in which this property may be needed or desirable; for instance: delation, voting, anonymous donations, and posting on bulletin boards.

An important characteristic of anonymity is that it is usually relative to a particular point of view. In general an event can be observed from various viewpoints - differing in the information they give access to, and therefore, the anonymity property depends on the view from which the event is being looked at (that is the exact information available to the observer). For example, in the situation of electronic bulletin boards, a posting by one member of the group is kept anonymous to the other members; however, it may be possible that the administrator of the board has access to some privileged information and can determine the member who posted the message(s), either directly or indirectly.

In general anonymity may be required for a subset of the agents only. In order to completely define anonymity for a system it is therefore necessary to specify which

* This work has been partially supported by the Project Rossignol of the ACI Sécurité Informatique (Ministère de la recherche et nouvelles technologies).
** This work has been carried out during the visit of Mohit Bhargava at INRIA, which has been supported by the INRIA/DREI programme for International Internship.

set(s) of members has to be kept anonymous. A further generalization is the concept of *group anonymity*: the members are divided into a number of sets, and it is revealed which of the groups is responsible for an event, but the information as to which particular member has performed the event must be hidden. In this paper, however, we do not consider the notion of group anonymity, we leave it for further work.

Various formal definitions and frameworks for analyzing anonymity have been developed in literature. They can be classified into approaches based on process-calculi ([24, 22]), epistemic logic ([26, 11]), and "function views" ([13]). In this paper, we focus on the approach based on process-calculi.

The framework and techniques of process calculi have been used extensively in the area of security, to formally define security properties, and to verify cryptographic protocols. See, for instance, [2, 15, 21, 23, 3]. The common denominator is that the various entities involved in the system to verify are specified as concurrent processes and present typically a nondeterministic behavior. In [24, 22], the nondeterminism plays a crucial role to define the concept of anonymity. More precisely, this approach to anonymity 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, i.e. if one such user can cause a certain observable trace in one possible computation, then there must be alternative computations in which each other anonymous user can give rise to the same observable trace (modulo the identity of the anonymous users).

The principle of anonymity described above is very elegant and general, however it has a limitation: Many systems for anonymity use random mechanisms. See, for example, Crowds ([20]) and Onion Routing ([27]). If the observer has the means to repeat the experiment and perform statistical analysis, he may be able to deduce certain quantitative information on the system. In particular, he may be able to compute the probability of certain observables and from that infer the probability of the relation between users and observables. Now, the situation of perfect anonymity can be only achieved when one cannot differenciate the agents by the observable. However this condition cannot be expressed in the nondeterministic approach, since the latter is based on set-theoretic notions, and it is therefore only able to detect the difference between possible and impossible (which in the finite case correspond to positive and zero probability respectively). Even the case in which one user has probability close to 1 will be considered acceptable by the definition of anonymity based on nondeterminism, provided that all the other users have positive probability.

Probabilistic information also allows to classify various notions of anonymity according to their strength. See for instance the hierarchy proposed by Reiter and Robin ([20]). In this paper we explore a notion of anonymity which corresponds to the strongest one (perfect anonymity: from the observables we deduce no information about the possible user).

A probabilistic notion of anonymity was developed (as a part of a general epistemological approach) in [11]. The approach there is purely probabilistic, in the sense that both the system and the users are assumed to act probabilistically. In particular the emphasis is on the probabilistic behavior of the users.

In this work, we take the opposite point of view, namely we assume that nothing may be known about the relative frequency by which the users perform the anonymous action. More precisely, the users can in principle be totally unpredictable and change intention every day, so that their behavior cannot be described probabilistically, not even by repeating statistical observations. The mechanisms of the systems, on the contrary, are like coin tossing, or random selection of a nearby node, are supposed to exhibit a certain regularity and obey a probabilistic distribution. Hence, we investigate a notion of anonymity which combines both probability and nondeterminism, and which is suitable for describing the most general situation in which both the systems and the user can have both probabilistic and nondeterministic behavior. We also investigate the properties of the definition for the particular cases of purely nondeterministic users and purely probabilistic users.

In order to define the notion of probability we need, of course, a model of computation able to express both probabilistic and nondeterministic choices. This kind of systems is by now well established in literature, see for instance the probabilistic automata of [25], and have been provided with solid mathematical foundations and sophisticated tools for verification. These models have practically replaced nowadays the purely probabilistic models since it was recognized that nondeterministic behavior is not "probabilistic behavior with unknown probabilities", but rather a phenomenon on its own, which is needed to describe situations in which an entity has a completely unpredictable and irregular behavior.

For reasons of space we omit the proofs and we only sketch the preliminary notions. The full details can be found on the report version of this paper, available on line ([4]).

2 The nondeterministic approach to anonymity

In this section we briefly recall the approach in [24, 22]. In these works, the actions of a system S are classified into three sets which determine the "point of view":

- A: the actions that are intended to be known anonymously by the observer,
- B: the actions that are intended to be known completely by the observer,
- C: the actions that are intended to be abstracted (hidden) to the observer.

Typically the set A consists of actions of the form $a.i$, where a is a fixed "abstract" action (the same for all the elements of A), and i represents the identity of an anonymous user. Hence $A = \{a.i \,|\, i \in I\}$., where I is the set of all the identities of the anonymous users.

Consider a dummy action d (different from all actions in S) and let f be the function on the actions of $A \bigcup B$ defined by $f(\alpha) = d$ if $\alpha \in A$, and $f(\alpha) = \alpha$ otherwise. Then S is said to be (strongly) anonymous on the actions in A if $f^{-1}(f(S \backslash C)) \sim_T S \backslash C$, where, following the CSP notation ([5]), $S \backslash C$ is the system resulting from hiding C in S, $f(S')$ is the system obtained from S' by applying the relabeling f to each (visible) action, f^{-1} is the relation inverse of f, and $\sim_T$ represents trace equivalence.

Intuitively, the above definition means that for any action sequence $\vec{\alpha} \in A^*$, if an observable trace t containing $\vec{\alpha}$ (not necessarily as a consecutive sequence) is a possible

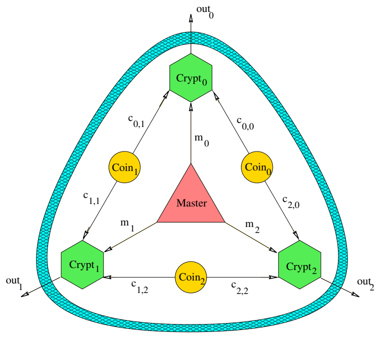

Fig. 1. Chaum's system for the Dining Cryptographers ([7, 22]).

outcome of $S \backslash C$, then, any trace t' obtained from t by replacing $\vec{\alpha}$ with an arbitrary $\vec{\alpha}\,' \in A^*$ must also be a possible outcome of $S \backslash C$.

We now illustrate the above definition on the example of the Dining Cryptographers.

3 The Dining Cryptographers' Problem

This problem, described by Chaum in [7], involves a situation in which three cryptographers are dining together. At the end of the dinner, each of them is secretly informed by a central agency (master) whether she should pay the bill, or not. So, either the master will pay, or one of the cryptographers will be asked to pay. The cryptographers (or some external observer) would like to find out whether the payer is one of them or the master. However, if the payer is one of them, they also wish to maintain anonymity over the identity of the payer.

A possible solution to this problem, described in [7], is that each cryptographer tosses a coin, which is visible to herself and her neighbor to the right. Each cryptographer observes the two coins that she can see and announces *agree* or *disagree*. If a cryptographer is not paying, she will announce *agree* if the two sides are the same and *disagree* if they are not. However, the paying cryptographer will say the opposite. It can be proved that if the number of *disagrees* is even, then the master is paying; otherwise, one of the cryptographers is paying. Furthermore, if one of the cryptographers is paying, then neither an external observer nor the other two cryptographers can identify, from their individual information, who exactly his paying.

The Dining Cryptographers (DC) will be a running example through the paper.

3.1 Nondeterministic Dining Cryptographers

In this approach the outcome of the coin tossing and the decision of the master regarding the payment of bill are considered to be nondeterministic ([24, 22]).

The specification of the solution can be given in a process calculus style as illustrated below. In the original works ([24, 22]) the authors used CSP. For the sake of

$$
\begin{aligned}
Master &= \textstyle\sum_{i=0}^{2} \tau \,.\, \overline{m}_i \mathsf{p} \,.\, \overline{m}_{i\oplus 1} \mathsf{n} \,.\, \overline{m}_{i\oplus 2} \mathsf{n} \,.\, 0 \\
&\qquad + \;\; \tau . \overline{m}_0 \mathsf{n} \,.\, \overline{m}_1 \mathsf{n} \,.\, \overline{m}_2 \mathsf{n} \,.\, 0 \\
Crypt_i &= m_i(x) \,.\, c_{i,i}(y) \,.\, c_{i,i\oplus 1}(z) \,. \\
&\qquad \text{if } x = \mathsf{p} \\
&\qquad\quad \text{then } \overline{pay}_i \;.\, \text{if } y = z \\
&\qquad\qquad \text{then } \overline{out}_i \, disagree \\
&\qquad\qquad \text{else } \overline{out}_i \, agree \\
&\qquad\quad \text{else if } y = z \\
&\qquad\qquad \text{then } \overline{out}_i \, agree \\
&\qquad\qquad \text{else } \overline{out}_i \, disagree \\
Coin_i &= \tau \,.\, Head_i + \tau \,.\, Tail_i \\
Head_i &= \overline{c}_{i,i} head \,.\, \overline{c}_{i\ominus 1,i} head \,.\, 0 \\
Tail_i &= \overline{c}_{i,i} tail \,.\, \overline{c}_{i\ominus 1,i} tail \,.\, 0 \\
DCP &= (\nu \vec{m})(Master \\
&\qquad \mid \;\; (\nu \vec{c})(\mathit{\Pi}_{i=0}^{2} Crypt_i \;\mid\; \mathit{\Pi}_{i=0}^{2} Coin_i)\,)
\end{aligned}
$$

Table 1. The dining cryptographer protocol specified in π-calculus.

uniformity we use here the π-calculus ([18]). We recall that $+$ ($\sum$) is the nondeterministic sum and $\mid$ ($\mathit{\Pi}$) is the parallel composition. 0 is the empty process. τ is the silent (or internal) action. $\overline{c}m$ and $c(x)$ are, respectively, send and receive actions on channel c, where m is the message being transmitted and x is the formal parameter. ν is an operator that, in the π-calculus, has multiple purposes: it provides abstraction (hiding), enforces synchronization, and generates new names. For more details on the π-calculus and its semantics, we refer to [18, 17].

In the code, given in Table 1, $\oplus$ and $\ominus$ represent the sum and the subtraction modulo 3. Messages p and n sent by the master are the requests to pay or to not pay, respectively. $\overline{pay}_i$ is the action of paying for cryptographer i.

We remark that we do not need all the expressive power of the π-calculus for this program. More precisely, we do not need guarded choice (all the choices are internal because they start with τ), and we do not need neither name-passing nor scope extrusion, thus ν is used just like the restriction operator of CCS ([16]).

Let us consider the point of view of an external observer. The actions that are to be hidden (set C) are the communications of the decision of the master and the results of the coins ($\vec{m}$, $\vec{c}$). These are already hidden in the definition of the system DCP. The anonymous users are of course the cryptographers, and the anonymous actions (set A) is constituted by the $\overline{pay}_i$ actions, for $i = 0, 1, 2$. The set B is constituted by the actions of the form $\overline{out}_i \, agree$ and $\overline{out}_i \, disagree$, for $i = 0, 1, 2$.

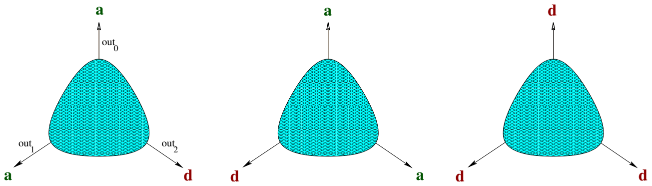

Fig. 2. Illustration of Example 1: the results that are observed with high frequency.

Let f be the function $f(\overline{pay}_i) = \overline{pay}$ and $f(\alpha) = \alpha$ for all the other actions. It is possible to check that $f^{-1}(f(DCP)) \sim_T DCP$, where we recall that $\sim_T$ stands for trace equivalence. Hence the nondeterministic notion of anonymity, as defined in Section 2, is verified.

3.2 Limitations of the nondeterministic approach

As a result of the nondeterminism, we cannot differentiate between a fair coin and an unfair one. However, it is evident that the fairness of the coins is essential to ensure the anonymity property in the system, as illustrated by the following example.

Example 1. Assume that, whenever a cryptographer pays, an external observer obtains *almost always* (i.e. with high frequency, say 99% of the times) one of the three outcomes represented in Figure 2, where a stands for *agree* and d for *disagree*. What can the observer deduce? By examining all possible cases, it is easy to see that the coins must be biased, and more precisely, $Coin_0$ and $Coin_1$ must produce almost always *head*, and $Coin_2$ must produce almost always *tail* (or vice-versa). From this estimation, it is immediate to conclude that, in the first case, the payer is *almost for sure* $Crypt_1$, in the second case $Crypt_2$, and in the third case $Crypt_0$.

In the situation illustrated in the above example, clearly, the system does not provide anonymity. However the nondeterministic definition of anonymity is still satisfied (as long as "almost always" is not "always", which in terms of observations means that the fourth configuration d, a, a must also appear, from time to time). The problem is that that definition can only express whether or not it is possible to have a particular trace, but cannot express whether one trace is more likely than the other.

3.3 Probabilistic Dining Cryptographers

The probabilistic version of the system can be obtained from the nondeterministic one by attaching probabilities to the the outcome of the coin tossing. We wish to remark that this is the essential change in perspective: the *random mechanisms internal to the system* which is designed to ensure anonymity are assumed to have a probabilistic behavior. As for the choices of the master, those are in a sense external to the system, and it is secondary whether they are nondeterministic or probabilistic.

We start by considering a nondeterministic master, which is in a sense the basic case: the fact that the master is nondeterministic means that we cannot assume any regularity in its behavior, nobody has any information about it, not even a probabilistic one. The anonymity system must then assure that this complete lack of knowledge be preserved through the observations of the possible outcomes (except, of course, for gaining the information on whether the payer is one of the cryptographers or not).

We use the probabilistic π-calculus (π_p) introduced in [12, 19] to represent the probabilistic system. The essential difference with respect to the π-calculus is the presence of a *probabilistic choice operator* of the form $\sum_i p_i \alpha_i.P_i$, where the p_i's represents probabilities, i.e. they satisfy $p_i \in [0,1]$ and $\sum_i p_i = 1$, and the α_i's are non-output prefixes, i.e. either input or silent prefixes. (Actually, for the purpose of this paper, only silent prefixes are used.) For the detailed presentation of this calculus we refer to [12, 19, 4].

The only difference with respect to the program presented in Section 3.1 is the definition of the $Coin_i$'s, which is as follows (p_h and p_t represent the probabilities of the outcome of the coin tossing):

$$Coin_i = p_h \tau \,.\, Head_i + p_t \tau \,.\, Tail_i$$

It is clear that the system obtained in this way combines probabilistic and nondeterministic behavior, not only because the master is nondeterministic, but also because the various components of the system and their internal interactions can follow different scheduling policies, selected nondeterministically (although it can be proved that this latter form of nondeterminism is not relevant for this particular problem).

This kind of systems (combining probabilistic and nondeterministic choices) is by now well established in literature, see for instance the probabilistic automata of [25], and have been provided with solid mathematical foundations and sophisticated tools for verification. In particular, we are interested here in the definition of the probability associated to a certain observable. The canonical way of defining such a probability is the following: First we *solve* the nondeterminism, i.e. we determine a function (*scheduler*) which, for each nondeterministic choice in the the computation tree, selects one alternative. After pruning the tree from all the non-selected alternatives, we obtain a *fully probabilistic automaton*, and we can define the probabilities of (measurable) sets of runs (and therefore of the intended observables) in the standard way. See [4] for the details.

One thing that should be clear, from the description above, is that in general the probability of an observable *depends on the given scheduler*.

4 Probabilistic anonymity for nondeterministic users

In this section we propose our notion of probabilistic anonymity for the case in which the anonymous user is selected nondeterministically.

The system in which the anonymous users live and operate is modeled as a probabilistic automaton M ([25], see [4]. Following [24, 22] we classify the actions of M

into the three sets A, B and C (cfr. Section 2). As before, these three sets are determined by the set of the anonymous users, the specific type of action on which we want anonymity, and the observer. We only change notation slightly:

- The set of the anonymous actions: $A = \{a(i) \mid i \in I\}$, where I is the set of the identities of the anonymous users and a is an injective functions from I to the set of actions which we call *abstract action*. We also call the pair (I, a) *anonymous action generator*.
- The set of the actions that we observe entirely, B. We will use b, b', ... to denote the elements of this set.
- The set of the hidden actions C.

It should be remarked the the term "observable" here is relative: we assume that the observer can observe only B and a, but, to the purpose of defining anonymity and checking whether a system is anonymous, we need the elements of A to be visible outcomes of the system.

Definition 1. *An anonymity system is a tuple (M, I, a, B, Z, p), where M is a probabilistic automaton which we assume already restricted (abstracted) on C, (I, a) is an anonymous action generator, B is a set of observable actions, Z is the set of all possible schedulers for M, and for every $\varsigma \in Z$, p_ς is a probability measure on the event space generated by the execution tree of M under ς (denoted by $etree(M, \varsigma)$), i.e. the σ–field generated by the cones in $etree(M, \varsigma)$ (cfr. [4]).*

Note that, as expressed by the above definition, given a scheduler ς, an *event* is a set of executions in $etree(M, \varsigma)$. We introduce the following notation to represent the events of interest:

- $a(i)$: all the executions in $etree(M, \varsigma)$ containing the action $a(i)$
- a : all the executions in $etree(M, \varsigma)$ containing an action $a(i)$ for an arbitrary i
- o : all the executions in $etree(M, \varsigma)$ containing as their maximal sequence of observable actions the sequence o (where o is of the form $\langle b_1, b_2, \ldots, b_n\rangle$ for some $b_1, b_2, \ldots, b_n \in B$). We denote by O the set of all such o's (*observables*).

We use the symbols $\cup$, $\cap$ and $\neg$ to represent the union, the intersection, and the complement of events, respectively.

We wish to keep the notion of observables as general as possible, but we still need to make some assumptions on them. First, we want the observables to be disjoint events. Second, they must cover all possible outcomes. Third, an observable o must indicate unambiguously whether a has taken place or not, i.e. it either implies a, or it implies $\neg a$. In set-theoretic terms it means that either o is contained in a or in the complement of a. Formally:

Assumption 1 (Assumptions on the observables) *1.* $\forall \varsigma \in Z.\ \forall o_1, o_2 \in O.\ \ o_1 \neq o_2 \ \Rightarrow\ p_\varsigma(o_1 \cup o_2) = p_\varsigma(o_1) + p_\varsigma(o_2)$

2. $\forall \varsigma \in Z.\ \ p_\varsigma(O) = 1$

3. $\forall \varsigma \in Z.\ \forall o \in O.\ \ p_\varsigma(o \cap a) = p_\varsigma(o) \vee p_\varsigma(o \cap \neg a) = p_\varsigma(o)$

Analogously, we need to make some assumption on the anonymous actions. We consider here conditions tailored for the nondeterministic users: Each scheduler determines completely whether an action of the form $a(i)$ takes place or not, and in the positive case, there is only one such i. Formally:

Assumption 2 (Assumption on the anonymous actions for nondeterministic users)

$$\forall \varsigma \in Z.\ \ p_\varsigma(a) = 0 \vee (\exists i \in I.\ (p_\varsigma(a(i)) = 1\ \wedge\ \forall j \in I.\ j \neq i \Rightarrow p_\varsigma(a(j)) = 0))$$

We are now ready to give the definition of anonymity for the case in which the anonymous user is selected nondeterministically:

Definition 2 (Probabilistic anonymity for nondeterministic users).
A system (M, I, a, B, Z, p) is anonymous if

$$\forall \varsigma, \vartheta \in Z.\, \forall o \in O.\ p_\varsigma(a) = p_\vartheta(a) = 1\ \Rightarrow\ p_\varsigma(o) = p_\vartheta(o)$$

Intuitively, the above definition expresses the fact that, for every two possible nondeterministic choices ς and ϑ which both select a, (say $a(i)$ and $a(j)$, with i and j possibly different) it should not be possible to detect from the probabilistic measure of the observables whether the choice was ς or ϑ (i.e. whether the user was i or j).

Example 2. Consider the DC with probabilistic coins and nondeterministic master. If the coins can give both *head* and *tail*, then, for each scheduler which chooses a (i.e. $\overline{m}_i\mathsf{p}$ for some i), the possible observable events are $\langle a, a, d\rangle$, $\langle a, d, a\rangle$, $\langle d, a, a\rangle$, and $\langle d, d, d\rangle$ ($\langle r_0, r_1, r_2\rangle$ here represents the collective result $\overline{out}_0 r_0$, $\overline{out}_1 r_1$, and $\overline{out}_2 r_2$). In principle different schedulers would affect also the order in which the outputs are emitted, but it is easy to see that the order, in this system, does not matter.

Consider the case in which the coins are totally fair, i.e. each of them gives *head* and *tail* with $1/2$ probability each. By considering all the 8 possible configurations of the coins, $\langle h, h, h\rangle$, $\langle h, h, t\rangle$, $\ldots \langle t, t, t\rangle$, it is easy to see that, for each possible payer i, each of the above observables is produced by exactly two configurations and hence has probability $1/4$. Hence Definition 2 is verified.

Consider now the case in which the coins are biased. Say, $Coin_0$ and $Coin_1$ give *head* with probability $9/10$ and *tail* with probability $1/10$, and vice-versa $Coin_2$ gives *head* with probability $1/10$ and *tail* with probability $9/10$. (This case is analogous to that illustrated in Example 1). Let us consider the observable $\langle a, a, d\rangle$. In case $Crypt_1$ is the payer, then the probability to get $\langle a, a, d\rangle$ is equal to the probability that the result of the coins is $\langle h, h, t\rangle$, plus the the probability that the result of the coins is $\langle t, t, h\rangle$, which is $9/10 * 9/10 * 9/10 + 1/10 * 1/10 * 1/10 = 730/1000$. In case $Crypt_2$ is the payer, then the probability to get $\langle a, a, d\rangle$ is equal to the probability that the result of the coins is $\langle h, h, h\rangle$, plus the the probability that the result of the coins is $\langle t, t, t\rangle$, which is $9/10 * 9/10 * 1/10 + 1/10 * 1/10 * 9/10 = 90/1000$.

Hence, in the biased case, Definition 2 is not verified. And this is what we expect, because the system, intuitively, is not anonymous: when we observe $\langle a, a, d\rangle$, $Crypt_1$ is much more likely to be the payer than $Crypt_2$.

As proved in the example above, the DC with fair coins are anonymous.

Proposition 1. *The DC with probabilistic fair coins and nondeterministic master are probabilistically anonymous.*

5 Probabilistic anonymity for users with combined probabilistic and nondeterministic behavior

In this section we develop a notion of anonymity for the more general case in which also the users may be selected according to some probabilistic distribution, possibly combined with a nondeterministic selection.

An example of such kind of behavior in the DC can be obtained by specifying the master as making first a nondeterministic choice on which probabilistic distribution to apply for selecting the payer, and then a probabilistic choice.

An example of such a master in π_p would be the following ($p_0, \ldots, p_3, q_0, \ldots, q_3$ represent the probabilities of the various decisions of the master)

$$
\begin{aligned}
Master \;&= \tau.Master_1 + \tau.Master_2 \\
Master_1 &= \textstyle\sum_{i=0}^{2} p_i \; \tau \,.\, \overline{m}_i \mathsf{p} \,.\, \overline{m}_{i\oplus 1} \mathsf{n} \,.\, \overline{m}_{i\oplus 2} \mathsf{n} \,.\, 0 \\
&\qquad + \;\; p_3 \tau.\overline{m}_0 \mathsf{n} \,.\, \overline{m}_1 \mathsf{n} \,.\, \overline{m}_2 \mathsf{n} \,.\, 0 \\
Master_2 &= \textstyle\sum_{i=0}^{2} q_i \; \tau \,.\, \overline{m}_i \mathsf{p} \,.\, \overline{m}_{i\oplus 1} \mathsf{n} \,.\, \overline{m}_{i\oplus 2} \mathsf{n} \,.\, 0 \\
&\qquad + \;\; q_3 \tau.\overline{m}_0 \mathsf{n} \,.\, \overline{m}_1 \mathsf{n} \,.\, \overline{m}_2 \mathsf{n} \,.\, 0
\end{aligned}
$$

Note that the choice in *Master* is nondeterministic while the choices in $Master_1$ and $Master_2$ are probabilistic.

While the assumptions on the observables remain the same, the assumption on the anonymous actions in this case is much weaker: the scheduler does not determine completely, in general, whether a is executed or not, and who is the user. However, we still require that there be at most an user which performs a for each computation, i.e. $a(i)$ and $a(j)$ must be disjoint for $i \neq j$. Formally:

Assumption 3 (Assumption on the anonymous actions for users with combined probabilistic and nondeterministic behavior)

$$\forall \varsigma \in Z.\, \forall i, j \in I.\;\; i \neq j \;\Rightarrow\; p_\varsigma(a(i) \cup a(j)) = p_\varsigma(a(i)) + p_\varsigma(a(j))$$

The notion of anonymity, in this case, must take into account the probabilities of the $a(i)$'s. When we observe a certain event o, the probability of o having been induced by $a(i)$ must be the same as the probability of o having been induced by $a(j)$ for any other $j \in I$. To formalize this notion, we need the concept of *conditional probability*. Given two events x and y with $p(y) > 0$, the conditional probability of x given y is defined as $p(x \,|\, y) = \frac{p(x \cap y)}{p(y)}$.

Definition 3 (Probabilistic anonymity for users with combined probabilistic and nondeterministic behavior). *A system* (M, I, a, B, Z, p) *is anonymous if*

$$
\begin{aligned}
&\forall \varsigma, \vartheta \in Z.\, \forall i, j \in I.\, \forall o \in O. \\
&\qquad\qquad (p_\varsigma(a(i)) > 0 \wedge p_\vartheta(a(j)) > 0) \Rightarrow p_\varsigma(o \,|\, a(i)) = p_\vartheta(o \,|\, a(j))
\end{aligned}
$$

Example 3. Consider the DC with probabilistic coins and the nondeterministic and probabilistic master illustrated above. Assume that the coins are totally fair. Consider a scheduler ς which selects $Master_1$ and assume that $Master_1$ selects $i \in I$ as the payer, with probability p_i. Consider now a scheduler ϑ which selects $Master_2$, and assume that $Master_2$ selects $j \in I$ as the payer, with probability q_j. Again, the possible observable events are $\langle a, a, d\rangle$, $\langle a, d, a\rangle$, $\langle d, a, a\rangle$, and $\langle d, d, d\rangle$. By considering all the 8 possible configurations of the coins, it is easy to see that if the scheduler is ς and the payer is i, each of the above observables is produced by exactly two configurations and hence the conditional probability of that observable, given that i is the payer, is $1/4$. The same holds for ϑ and j, hence Definition 3 is verified.

The behavior of a master which combines nondeterministic and probabilistic behavior can be much more complicated than the one illustrated above. However it is easy to see that as long as the master does not influence the behavior of the coins, and these are fair, the conditional probability of each observable for a given payer is $1/4$.

Proposition 2. *The DC with probabilistic fair coins and nondeterministic and probabilistic master are probabilistically anonymous.*

We terminate this section by giving an alternative characterization of anonymity.

Theorem 1. *A system (M, I, a, B, Z, p) is anonymous iff*

$$\forall \varsigma, \vartheta \in Z.\, \forall i \in I.\, \forall o \in O.\;\; (p_\varsigma(a(i)) > 0 \wedge p_\vartheta(a) > 0) \Rightarrow p_\varsigma(o \,|\, a(i)) = p_\vartheta(o \,|\, a)$$

6 Probabilistic Anonymity for fully probabilistic users

In this section we consider the case of a totally probabilistic system. For instance, in the case of the dining philosophers, the master would be of the form

$$\begin{aligned} Master = {} & \textstyle\sum_{i=0}^{2} p_i\; \tau\,.\,\overline{m}_i\mathsf{p}\,.\,\overline{m}_{i\oplus 1}\mathsf{n}\,.\,\overline{m}_{i\oplus 2}\mathsf{n}\,.\,0 \\ & +\;\; p_3\tau.\overline{m}_0\mathsf{n}\,.\,\overline{m}_1\mathsf{n}\,.\,\overline{m}_2\mathsf{n}\,.\,0 \end{aligned}$$

Furthermore, we would fix the scheduling of the parallel activities, so that each step in the computation would be either deterministic or probabilistic.

Since there is no nondeterminism, there is no choice of scheduler either, so we can eliminate Z from the tuple and we can write $p(x)$ instead of $p_\varsigma(x)$. The definition of probabilistic anonymity given in previous section simplifies into the following:

Definition 4 (Probabilistic anonymity for probabilistic users).
A system (M, I, a, B, p) is anonymous if

$$\forall i, j \in I.\, \forall o \in O.\;\; (p(a(i)) > 0 \wedge p(a(j)) > 0) \Rightarrow p(o \,|\, a(i)) = p(o \,|\, a(j))$$

Furthermore, the alternative characterization in Theorem 1 reduces to the following:

Corollary 1. *A system* (M, I, a, B, p) *is anonymous iff*

$$\forall i \in I.\, \forall o \in O.\;\; (p(a(i)) > 0 \wedge p(a) > 0) \Rightarrow p(o \,|\, a(i)) = p(o \,|\, a)$$

In the fully probabilistic case there are two other interesting characterizations: The first is based on the intuition that a system is anonymous if the observations do not change the probability of $a(i)$: we may know the probability of $a(i)$ by some means external to the system, but the system should not increase our knowledge about it (cfr. [11]). The second is based on the (similar) idea that observing o rather than o' should not change our knowledge of the probability of $a(i)$. Formally:

Proposition 3. *The following conditions are equivalent, and are equivalent to the condition of anonymity.*

1. $\forall i \in I.\, \forall o \in O.\;\; p(o \cap a) > 0 \;\Rightarrow\; p(a(i) \,|\, o) = p(a(i))/p(a)$
2. $\forall i \in I.\, \forall o, o' \in O.\;\; (p(o \cap a) > 0 \wedge p(o' \cap a) > 0) \;\Rightarrow\; p(a(i) \,|\, o) = p(a(i) \,|\, o')$.

Proposition 3 can be reformulated as a general property of probablistic spaces, independent from anonymity. Similar results have been presented in [10] and in [9].

6.1 Characterizations given by Proposition 3 and nondeterminism

It is not clear whether the characterizations expressed in Proposition 3 can be generalized to the case of the users with combined nondeterministic and probabilistic behavior. The "naive" extensions obtained by introducing the scheduler in the formulae would not work. Let us consider the first characterization in Proposition 3 (the other would be analogous). One possible way of reintroducing the notion of scheduler is

$$\begin{aligned} &\forall \varsigma, \vartheta \in Z.\, \forall i \in I.\, \forall o \in O. \\ &\qquad (p_\varsigma(o \cap a) > 0 \wedge p_\vartheta(a) > 0) \Rightarrow\; p_\varsigma(a(i) \,|\, o) = p_\vartheta(a(i))/p_\vartheta(a) \end{aligned}$$

However this condition is too strong because it implies that $p_\vartheta(a(i))/p_\vartheta(a)$ is the same for every ϑ, and this is clearly not the case for instance for the nondeterministic and probabilistic master specified in Section 5.

On the other hand, if we weaken the condition by identifying ς and ϑ:

$$\forall \varsigma \in Z.\, \forall i \in I.\, \forall o \in O.\;\; p_\varsigma(o \cap a) > 0 \Rightarrow\; p_\varsigma(a(i) \,|\, o) = p_\varsigma(a(i))/p_\varsigma(a)$$

then the condition would be too weak to ensure anonymity, as shown by next example:

Example 4. Consider a system in which the master influences the behavior of the coins somehow, in such a way that when $Crypt_i$ is chosen as the payer (say, purely nondeterministically, by ς_i) the result is always $o_0 = \langle d, a, a\rangle$ for $i = 0$, $o_1 = \langle a, d, a\rangle$ for $i = 1$, and $o_2 = \langle a, a, d\rangle$ for $i = 2$. Then we would have $p_{\varsigma_i}(o_j \cap a) > 0$ only if $j = i$, and $p_{\varsigma_i}(a(i) \,|\, o_i) = 1 = p_{\varsigma_i}(a(i))/p_{\varsigma_i}(a)$. Hence the above condition would be verified, but the system is not be anonymous at all: whenever we observe $\langle d, a, a\rangle$, for instance, we are sure that $Crypt_0$ is the payer.

6.2 Independence from the probability distribution of the users

One important property of Definition 4 is that it is independent from the probability distribution of the users. Intuitively, this is due to the fact that the condition of anonymity implies that $p(o \mid a(i)) = p(o)/p(a)$, hence it is independent from $p(a(i))$.

Theorem 2. *If (M, I, a, B, p) is anonymous (according to Definition 4) then for any p' which differs from p only on the $a(i)$'s, (M, I, a, B, p') is anonymous.*

Also the characterizations of anonymity given in Proposition 3 are (obviously) independent from the probability distribution of the users. It should be remarked, however, that their correspondence with Definition 4, and the property of independence from the probability of the users, only holds under the hypothesis that there is at most one agent performing a. (Assumption 3.)

7 Related work

The work [13] presents a modular framework to formalize a range of properties (including numerous flavors anonymity and privacy) of computer systems in which an observer has only partial information about system behavior, thereby combining the benefits of the knowledge-based approach (natural specification of information-hiding) and the algebra-based approach (natural specification of system behavior). It proposes the notion of *function view* to represent a mathematical abstraction of partial knowledge of a function. The logical formulas describing a property are characterized as *opaqueness* of certain function views, converted into predicates over observational equivalence classes, and verified, when possible, using the proof techniques of the chosen process formalism.

In [11, 26] epistemic logic is used to characterize a number of information-hiding requirements (including anonymity). In particular, [26] introduces the notion of a *group principal* and an associated model, language and logic to axiomatize anonymity. The main advantage of modal logic is that even fairly complex properties can be stated directly as formulas in the logic. On the other hand, [11] uses a completely semantic approach and provides an appropriate semantic framework in which to consider anonymity. It also propose notions of probabilistic anonymity in a purely probabilistic framework. In particular, it propose a notion of conditional probability (cfr. Definition 4.4 in [11]) which is similar to the first characterization in Proposition 3, if we interpret the formula φ in [11] as the occurrence of the event a.

The first characterization in Proposition 3 was also implicitly used by Chaum in [7] (in which he considered a purely probabilistic system) as definition of anonymity. The factor $p(a)$ is not present in the formula of Chaum, but that's probably a typo, because in the informal explanation he gives, that factor is taken into account.

In literature there have been other works involving the use of variants of the π-calculus for formalizing protocols providing anonymity or similar properties. See for example [1, 14].

8 Conclusion and future work

We have proposed a new notion of anonymity based on a model which combines probability and nondeterminism, and we have shown that it can be regarded as a generalization of the probabilistic one given in [11].

We have formulated the notion of anonymity in terms of observables for processes in the probabilistic π-calculus, whose semantics is based on the probabilistic automata of [25]. This opens the way to the automatic verification of the property. We are currently developing a model checker for the probabilistic π-calculus.

We are currently investigating weaker versions of our notion of anonymity, and studying their application to protocols like Crowds [8, 6].

References

1. Martín Abadi and Cédric Fournet. Private authentication. *Theoretical Computer Science*, 322(3):427–476, 2004.
2. Martín Abadi and Andrew D. Gordon. A calculus for cryptographic protocols: The spi calculus. *Information and Computation*, 148(1):1–70, 10 January 1999.
3. 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.
4. 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`.
5. S.D. Brookes, C.A.R. Hoare, and A.W. Roscoe. A theory of communicating sequential processes. *Journal of the ACM*, 31(3):560–599, 1984.
6. Kostantinos Chatzikokolakis and Catuscia Palamidessi. Probable innocence revisited. Technical report, INRIA Futurs and LIX, 2005. `http://www.lix.polytechnique.fr/~catuscia/papers/Anonymity/reportPI.pdf`.
7. David Chaum. The dining cryptographers problem: Unconditional sender and recipient untraceability. *Journal of Cryptology*, 1:65–75, 1988.
8. Yuxin Deng, Catuscia Palamidessi, and Jun Pang. Weak probabilistic anonymity. Technical report, INRIA Futurs and LIX, 2005. Submitted for publication. `http://www.lix.polytechnique.fr/~catuscia/papers/Anonymity/reportWA.pdf`.
9. R.D. Gill, M. van der Laan, and J. Robins. Coarsening at random: Characterizations, conjectures and counterexamples. In D.Y. Lin and T.R. Fleming, editors, *Proceedings of the First Seattle Symposium in Biostatistics*, Lecture Notes in Statistics, pages 255–294. Springer-Verlag, 1997.
10. P. D. Grunwald and J. Y. Halpern. Updating probabilities. *Journal of Artificial Intelligence Research*, 19:243–278, 2003.
11. 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.
12. 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-Verlag, 2000.
13. Dominic Hughes and Vitaly Shmatikov. Information hiding, anonymity and privacy: a modular approach. *Journal of Computer Security*, 12(1):3–36, 2004.

14. Steve Kremer and Mark D. Ryan. Analysis of an electronic voting protocol in the applied pi-calculus. In Mooly Sagiv, editor, *Programming Languages and Systems – Proceedings of the 14th European Symposium on Programming (ESOP'05)*, volume 3444 of *Lecture Notes in Computer Science*, pages 186–200, Edinburgh, U.K., April 2005. Springer.
15. 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.
16. R. Milner. *Communication and Concurrency*. International Series in Computer Science. Prentice Hall, 1989.
17. Robin Milner. *Communicating and mobile systems: the π-calculus*. Cambridge University Press, 1999.
18. Robin Milner, Joachim Parrow, and David Walker. A calculus of mobile processes, I and II. *Information and Computation*, 100(1):1–40 & 41–77, 1992. A preliminary version appeared as Technical Reports ECF-LFCS-89-85 and -86, University of Edinburgh, 1989.
19. Catuscia Palamidessi and Oltea M. Herescu. A randomized encoding of the π-calculus with mixed choice. *Theoretical Computer Science*, 335(2-3):73–404, 2005.
20. Michael K. Reiter and Aviel D. Rubin. Crowds: anonymity for Web transactions. *ACM Transactions on Information and System Security*, 1(1):66–92, 1998.
21. 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.
22. Peter Y. Ryan and Steve Schneider. *Modelling and Analysis of Security Protocols*. Addison-Wesley, 2001.
23. S. Schneider. Security properties and csp. In *Proceedings of the IEEE Symposium Security and Privacy*, 1996.
24. 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-Verlag, 1996.
25. 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.
26. 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.
27. P.F. Syverson, D.M. Goldschlag, and M.G. Reed. Anonymous connections and onion routing. In *IEEE Symposium on Security and Privacy*, pages 44–54, Oakland, California, 1997.