Modelling opacity using petri nets

Bryans, JW, Koutny, M & Ryan, PYA

DOI 10.1016/j.entcs.2004.10.010
ISSN 1571-0661

Publisher: Elsevier

Available online at www.sciencedirect.com
SCIENCE DIRECT®

Electronic Notes in Theoretical Computer Science

ELSEVIER Electronic Notes in Theoretical Computer Science 121 (2005) 101–115
www.elsevier.com/locate/entcs

Modelling Opacity Using Petri Nets

Jeremy W. Bryans, Maciej Koutny and Peter Y. A. Ryan[1],[2]

School of Computing Science
University of Newcastle
Newcastle upon Tyne, NE1 7RU, U.K.

Abstract

We consider opacity as a property of the local states of the secure (or high-level) part of the system, based on the observation of the local states of a low-level part of the system as well as actions. We propose a Petri net modelling technique which allows one to specify different information flow properties, using suitably defined observations of system behaviour. We then discuss expressiveness of the resulting framework and the decidability of the associated verification problems.

Keywords: opacity, non-deducibility, Petri nets, observable behaviour

1 Introduction

The notion of secrecy has been formulated in various ways in the computer security literature. One well known formulation is that of non-interference, generally attributed to Goguen and Meseguer in [5] and [6], but the idea can be traced back to earlier work of Feiertag [3] and Cohen [2]. This seeks to formalise the absence of any information flow or more precisely, the absence of any causal flow from one process to another. A reformulation of this notion in terms of structural characteristics of Petri nets has recently been given by Busi and Gorrieri [1].

In this paper we examine a different approach to formulating information flow policies, following an approach that can be traced back to the notion of non-deducibility due to Sutherland [11]. The essential idea is to stipulate

[1] This research was supported by the EPSRC GOLD and SCREEN projects and DSTL
[2] {jeremy.bryans, peter.ryan, maciej.koutny}@newcastle.ac.uk

doi:10.1016/j.entcs.2004.10.010

that whatever observations an uncleared user may make of the system, the space of possible high level (secret) inputs consistent with those observations is unchanged. In other words, the uncleared user should be unable to make any useful deductions about the interactions of a secret user with the system. A number of problems and limitations of the original formulation have been noted over the years and several variants proposed. An overview can be found in [9].

A variant of the idea has appeared recently in the context of the analysis of security protocols as the notion of *opacity* [4]. Whereas non-interference seeks to capture the complete absence of information flow, opacity is specific to a particular item of information. Thus, for example, the value of a variable v say, is deemed to be opaque for a particular run of a protocol if the adversary is unable to deduce its value from the observations and deductions available to him during the run. For the protocol to satisfy such a requirement it must be the case that, for any alternative value of v, there is another possible run of the protocol that gives rise to observations by the adversary that are indistinguishable from the original observations. Of course care has to be taken over the definition of 'indistinguishability' here, especially in the presence of cryptography and non-determinism. A specialisation of this notion is to require that the adversary be unable to determine the satisfaction of a particular property.

A number of important information flow requirements are naturally captured by such a formulation. Anonymity is a prime example of a situation where some information flow is permitted whilst particular items of information must be kept secret. Thus, for ballot secrecy, it will typically be regarded as acceptable for an observer to know that a particular member of the electorate cast a vote, but it is essential that the actual vote is kept secret.

Similarly, for encrypted channels and cryptographic protocols, it is often regarded as acceptable or unavoidable that an adversary can perform traffic analysis, observe sources, destinations and lengths of messages, but essential that the contents of the messages remain secret.

Such scenarios fit awkwardly into traditional, strict formulations of non-interference which regard information flow as a binary property.

Note that, as is standard in the modelling of information flows, we assume that the adversary has full knowledge of the construction of the system. This is an analogue of Kerckhoffs' principle in cryptography of not seeking security in obscurity and is in effect a worst case assumption.

In this paper we extend the notion of opacity to systems in general, rather than just to cryptographic protocols and cast it in the framework of Petri nets. The flexibility in defining the adversary's visibility of transitions and places as well as flexibility of predicates over states allows us to express very rich information flow requirements. The ability to define arbitrary adversary views of the system state gives a convenient way to model certain threat scenarios, e.g., various forms of probing using side channel analysis.

Casting our models in Petri nets gives us access to a raft of existing results and tools that have been developed in the Petri net community.

2 Petri nets

In this section, we introduce Petri nets with weighted arcs [7], and give their operational semantics in terms of step sequences.

A (weighted) *net* is a triple $N = (P, T, W)$ such that P and T are disjoint finite sets, and $W : (T \times P) \cup (P \times T) \to \mathbb{N}$. The elements of P and T are respectively the *places* and *transitions*, and W is the *weight function* of N. In diagrams, places are drawn as circles, and transitions as rectangles. If $W(x, y) \geq 1$ for some $(x, y) \in (T \times P) \cup (P \times T)$, then (x, y) is an *arc* leading from x to y. As usual, arcs are annotated with their weight if this is 2 or more. We assume that, for every $t \in T$, there is a place p such that $W(p, t) \geq 1$.

The *pre-* and *post-multiset* of a transition $t \in T$ are multisets of places, $pre_N(t)$ and $post_N(t)$, respectively given by

$$pre_N(t)(p) = W(p, t) \quad \text{and} \quad post_N(t)(p) = W(t, p),$$

for all $p \in P$. Both notations extend to finite multisets of transitions U:

$$pre_N(U) = \sum_{t \in U} U(t) \cdot pre_N(t) \qquad \text{and} \qquad post_N(U) = \sum_{t \in U} U(t) \cdot post_N(t)\ .$$

A *marking* of a net N is a multiset of places. Following the standard terminology, given a marking M of N and a place $p \in P$, we say that p is marked if $M(p) \geq 1$ and that $M(p)$ is the number of tokens in p. In diagrams, M will be represented by drawing in each place p exactly $M(p)$ tokens (black dots).

Transitions represent actions which may occur at a given marking and then lead to a new marking. Here we define this dynamics in terms of multisets of (simultaneously occurring) transitions.

A *step* is a non-empty finite multiset of transitions, $U : T \to \mathbb{N}$. It is *enabled* at a marking M if $M \geq pre_N(U)$. Thus, in order for U to be enabled at M, for each place p, the number of tokens in p under M should at least be equal to the total number of tokens that are needed as an input to U, respecting the weights of the input arcs.

If U is enabled at M, then it can be *executed* leading to the marking $M' = M - pre_N(U) + post_N(U)$. This means that the execution of U 'consumes' from each place p exactly $W(p,t)$ tokens for each occurrence of a transition $t \in U$ that has p as an input place, and 'produces' in each place p exactly $W(t,p)$ tokens for each occurrence of a transition $t \in U$ with p as an output place. If the execution of U leads from M to M' we write $M[U\rangle M'$.

An *execution* from a marking M to a marking M' is a sequence

$$\mu = MU_1M_1 \dots M_{n-1}U_nM'$$

such that

$$M\,[U_1\rangle\, M_1 \,\cdots\, M_{n-1}\,[U_n\rangle\, M' \;.$$

We also say that M' is *reachable* from M.

3 Observing Petri net behaviour

In this section, we introduce a specific device aimed at modelling various observation capabilities based on the executed behaviours of a Petri net. Our framework is deliberately general to allow one to deal with a wider range of observation scenarios.

We start by making a small (but important from the point of view of applications) adjustment of the standard notion of a marked net, by assuming that the system specification we are given at the outset is a pair $\Sigma = (N, \mathcal{M}_0)$, where N is a net as defined in the previous section and $\mathcal{M}_0$ is a non-empty finite set of initial markings. This allows us to easily model situations where only partial information of the initial state of the system is available to an observer.

We will denote by $[\mathcal{M}_0\rangle$ the set of all markings reachable from any of the markings in $\mathcal{M}_0$, and by $RG(\Sigma)$ the reachability graph of Σ defined as the labelled directed graph whose nodes are the markings in $[\mathcal{M}_0\rangle$, and the labelled arcs represent all steps executed at these markings according to the rules from the previous section (see, e.g., figure 2(a,b) for an example of a net with a single initial marking and its reachability graph). We will denote by $RG_{steps}(\Sigma)$ the set of all the steps labelling the arcs of $RG(\Sigma)$.

3.1 Visibility of reachable markings and executed steps

In our approach, we assume that there is a mapping *obs* which for each reachable marking in $[\mathcal{M}_0\rangle$ and every step of executed transitions in $RG_{steps}(\Sigma)$ returns some label. This label is meant to capture the observable or visible aspects of system behaviour, in this case global states (markings) and executed actions (steps of transitions). We do not place any restrictions on the nature of the *obs* mapping at this point; indeed, it is left under-specified deliberately to accommodate a wide range of observation scenarios.

Suitable choices of *obs* mapping can be used to encode the various levels of visibility of system behaviour that we attribute to the environment or adversary. Thus transitions visible only to a secret user might be mapped to a τ label. Such events would be completely invisible to the environment, i.e., the environment would not be aware that any transition had occurred. Transitions corresponding to the transmission of encrypted values could be mapped to a single label. We discuss flavours of invisibility further in section 3.3. Transitions deemed visible to the adversary may be left unchanged.

Note that, in particular, *obs* allows us to 'detect' properties like deadlock-freeness or acceptance sets. The theory is rich enough to incorporate and reason about them. It is another matter, of course, how deadlocks would be detected or observed in the real life system, but these issues are beyond the scope of the current paper.

We assume that markings and steps are visible using different sets of observed labels (i.e., $obs(M) \neq obs(U)$, for all $M \in [\mathcal{M}_0\rangle$ and $U \in RG_{steps}(\Sigma)$); in other words, observers can always distinguish between a state and a transition.

The two basic forms of defining the *obs* mapping are *transition labelling* and *marking projection*. In the first case, we assume that each transition t has its own (not necessarily unique) label $\ell(t)$ and then the visibility of a step $U = \{t_1, \ldots, t_k\}$ is defined as the multiset $\ell(U) = \{\ell(t_1), \ldots, \ell(t_k)\}$. In the case of marking projection, we assume that $Vis \subseteq P$ is a set of places on which we can always see the tokens, and all places in $P \setminus Vis$ are hidden from us (in the extreme case, $Vis = \varnothing$ which effectively means that no information about the tokens is available). Then, for every marking M, we define $M|_{Vis}$ as a multiset over Vis such that $M|_{Vis}(p) = M(p)$ for every place $p \in Vis$.

Having defined the observable aspects of individual markings and steps of transitions, it is straightforward to define the effect of the observation mapping on the executions of the marked net Σ: given an execution

$$\mu = M_0 U_1 M_1 \ldots M_{n-1} U_n M_n,$$

where $M_0 \in \mathcal{M}_0$, we observe it as the sequence

$$obs(\mu) = obs(M_0) obs(U_1) obs(M_1) \ldots obs(M_{n-1}) obs(U_n) obs(M_n).$$

The whole behaviour of Σ can now be viewed as the labelled directed graph $obs(RG(\Sigma))$ obtained from $RG(\Sigma)$ by replacing each arc label U by $obs(U)$, and by labelling each node M by $obs(M)$. Moreover, any actual observation of the system behaviour is simply a sequence of node and arc labels along any directed path originating from one of the initial nodes.

3.2 Opacity

In the present framework, we are interested in whether an observer can establish a property $\mathcal{P}$ at some specific state(s) of the execution of the system solely on the basis of its visible version. We consider here any state property, i.e., one which can be evaluated at any reachable marking in $[\mathcal{M}_0\rangle$. Clearly, any such property can simply be represented as the set of those reachable markings where it holds, and so we will take $\mathcal{P}$ to be any subset of $[\mathcal{M}_0\rangle$.

Now, given an observable execution of the system, we will be interested in finding out whether the fact that an underlying marking belongs to $\mathcal{P}$ can be deduced by the observer. Note, however, that we are not interested in establishing whether the underlying marking does not belong to $\mathcal{P}$. To do this, we would rather consider the property $\overline{\mathcal{P}} = [\mathcal{M}_0\rangle \setminus \mathcal{P}$.

What it means to deduce a property can mean different things depending on what is relevant or important from the point of view of real application. Below, we formalise three possible ways of defining variants of opacity:

- $\mathcal{P}$ is *initial-opaque* if for every execution μ from any marking $M_0 \in \mathcal{M}_0 \cap \mathcal{P}$, there exists an execution μ' from a marking $M'_0 \in \mathcal{M}_0 \setminus \mathcal{P}$ such that $obs(\mu) = obs(\mu')$.
  I.e., we are only interested in the holding of our property in the initial state.
- $\mathcal{P}$ is *final-opaque* if for every execution μ from any marking $M_0 \in \mathcal{M}_0$ to a marking $\widehat{M} \in \mathcal{P}$, there exists an execution μ' from a marking $M'_0 \in \mathcal{M}_0$ to a marking $\widehat{M'} \notin \mathcal{P}$ such that $obs(\mu) = obs(\mu')$.
  I.e., we are only interested in the holding of our property in the current state.
- $\mathcal{P}$ is *always-opaque* if for every execution
  $$\mu = M_0 U_1 M_1 \ldots M_{n-1} U_n M_n$$
  from any marking $M_0 \in \mathcal{M}_0$ and every $i \leq n$ such that $M_i \in \mathcal{P}$, there exists an execution
  $$\mu' = M'_0 U'_1 M'_1 \ldots M'_{n-1} U'_n M'_n$$
  from a marking $M'_0 \in \mathcal{M}_0$ such that $obs(\mu) = obs(\mu')$ and $M'_i \notin \mathcal{P}$.
  I.e., we are interested in the holding of our property at all states of the observed execution.

Later on, *initial-opacity* is illustrated by the dining cryptographers example. It would appear that it is suited to modelling situations in which initialisation information such as crypto keys, etc., needs to be kept secret. More generally, situations in which confidential information can be modelled in terms of initially resolved non-determinism can be captured in this way. On the other hand, *always-opacity* would seem more appropriate to capture situation in which secret information is input at run time, for example due to high level interactions. We will be exploring the application of these formal properties in a future paper.

Proposition 3.1 *If $\mathcal{P}$ is always-opaque then it is both initial-opaque and final-opaque; no other implication of this kind in general holds.*

Proof. The first part follows from the definitions. Moreover, together with the two counterexamples below, it implies that the second part holds as well.

The first example shows that in general initial-opacity does not imply final-opacity. Take the net in figure 1(a) with the initial markings $\mathcal{M}_0 = \{\{s_1, s_2\}, \{s_1\}\}$ (the reachability graph is shown in figure 1(b)). Assume further that *obs* is given by the transition labelling $\ell(t) = a$, and marking projection on the place s_1. The resulting observation graph is shown in figure 1(c). Suppose now that we are interested in establishing that the invisible place s_2 is non-empty, which is captured by $\mathcal{P} = [\mathcal{M}_0\rangle \setminus \{\{s_1\}\}$. It is easily seen that $\mathcal{P}$ is initial-opaque, but it is not final-opaque (basically, after executing the only transition, s_2 is bound to contain at least one token).

The second example shows that final-opacity does not imply initial-opacity. Take the net in figure 1(d) with the initial markings $\mathcal{M}_0 = \{\{s\}, \varnothing\}$ (the reachability graph is shown in figure 1(e)). Assume further that *obs* is given by the transition labelling $\ell(t) = \ell(u) = a$, and marking projection on the empty set of places. The resulting observation graph is shown in figure 1(f). Suppose now that we are interested in establishing that the invisible place s contains a token, which is captured by $\mathcal{P} = \{\{s\}\}$. It is easily seen that $\mathcal{P}$ is final-opaque, but it is not initial-opaque (basically, after executing any sequence of the two transitions, we know for sure that s must have contained initially a token, but since we have no idea which of the two transitions was executed at the end of the sequence, the current marking of s is undetermined). □

Proposition 3.2 *For $x \in \{initial, final, always\}$, it is the case that $\mathcal{P} = \varnothing$ is x-opaque, $\mathcal{P} = [\mathcal{M}_0\rangle$ is not x-opaque, and if $\mathcal{P} \subseteq \mathcal{P}'$ and $\mathcal{P}'$ is x-opaque then $\mathcal{P}$ is x-opaque.*

Proof. Follows from definitions. □

What now follows are crucial results stating that the three notions of opacity are decidable provided that the system has finitely many states.

Theorem 3.3 *If $[\mathcal{M}_0\rangle$ is finite*[3] *then it is decidable whether $\mathcal{P}$ is initial-opaque.*

[3] Note that the finiteness of $[\mathcal{M}_0\rangle$ is decidable, and can be checked using the standard coverability tree construction [7].

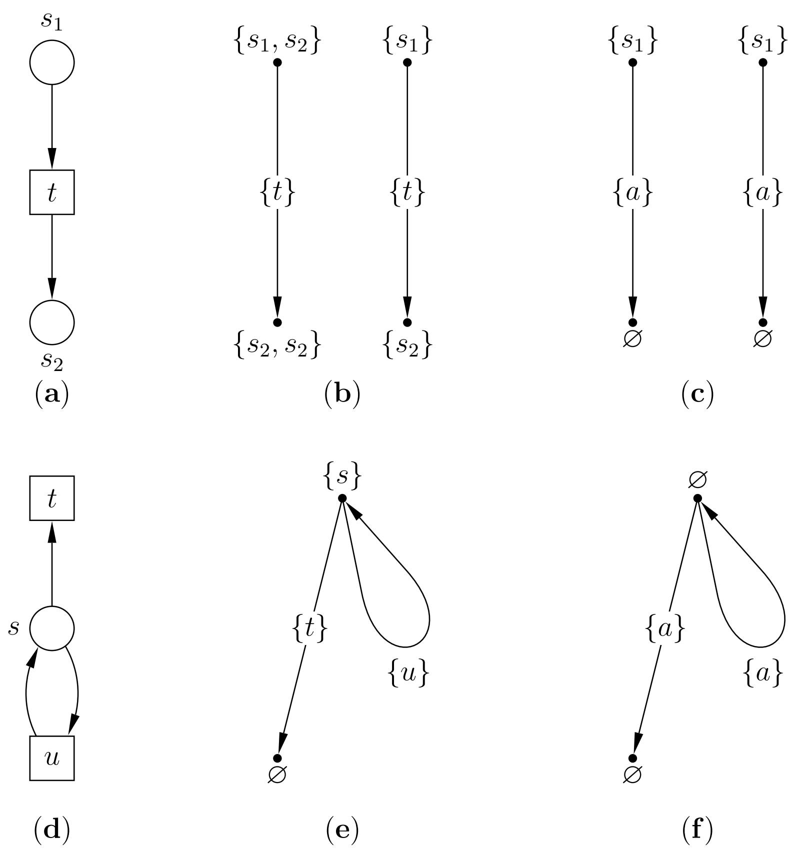

Fig. 1. Two counterexamples for the proof of proposition 3.1.

Proof. The proof of this and the next two results are based on a language-theoretic argument centered around a directed graph G obtained from $RG(\Sigma)$ in the following way: for each arc (M, U, M'), the label U is changed to the two-label sequence $obs(U)obs(M')$. Note that G is finite since $[\mathcal{M}_0\rangle$ is finite and so $RG(\Sigma)$ is a finite reachability graph.

To decide initial-opacity we proceed as follows. First, we construct a finite state machine F_1 by taking G with a fresh initial node connected by an $obs(M)$-arc with every node $M \in \mathcal{M}_0 \cap \mathcal{P}$, and all the states being treated as final. After that we construct in a similar way a finite state machine F_2, except that now the initial node is connected by an $obs(M)$-arc with every node $M \in \mathcal{M}_0 \setminus \mathcal{P}$.

It is easy to see that $\mathcal{P}$ is initial-opaque iff $L(F_1) \subseteq L(F_2)$, where $L(F_i)$ is the language accepted by F_i. $\square$

Theorem 3.4 *If $[\mathcal{M}_0\rangle$ is finite then it is decidable whether $\mathcal{P}$ is final-opaque.*

Proof. First, we construct a finite state machine F_1 by taking G with a fresh initial node connected by an $obs(M)$-arc with every node $M \in \mathcal{M}_0$, and all the states in $[\mathcal{M}_0\rangle \cap \mathcal{P}$ being treated as final. After that we construct in a similar way a finite state machine F_2, except that now all the states in $[\mathcal{M}_0\rangle \setminus \mathcal{P}$ are treated as final.

It is easy to see that $\mathcal{P}$ is final-opaque iff $L(F_1) \subseteq L(F_2)$. $\square$

Theorem 3.5 *If $[\mathcal{M}_0\rangle$ is finite then it is decidable whether $\mathcal{P}$ is always-opaque.*

Proof. Let $\xi = (M, ll', M')$ be any arc in G such that $M' \in \mathcal{P}$. First, we construct a finite state machine F_1^ξ by taking G with a fresh initial node connected by an $obs(\widehat{M})$-arc with every node $\widehat{M} \in \mathcal{M}_0$, an extra arc (M, a, M') where a is a fresh label, and all the states in $[\mathcal{M}_0\rangle$ being treated as final. After that we construct a finite state machine F_2^ξ by taking G with a fresh initial node connected by an $obs(\widehat{M})$-arc with every node $\widehat{M} \in \mathcal{M}_0$, an extra arc $(\widehat{M}, a, \widehat{M'})$ for each existing arc $(\widehat{M}, ll', \widehat{M'})$ in G such that $\widehat{M'} \notin \mathcal{P}$, and all the states in $[\mathcal{M}_0\rangle$ being treated as final.

It is easy to see that $\mathcal{P}$ is always-opaque iff $\mathcal{P}$ is initial-opaque and, for every arc ξ as above, $L(F_1^\xi) \cap \mathcal{L} \subseteq L(F_2^\xi)$, where $\mathcal{L}$ is the language generated by the regular expression A^*aA^* and A is the set of all arc labels used in F_1^ξ except for the label a. $\square$

3.3 Invisible transitions

So far we have tacitly assumed that it is always possible to observe in some way every step of executed transitions (i.e., some label is always generated). However, one might also wish to deal with totally invisible transitions and steps. We now will outline how such a feature could be incorporated within our present framework.

To start with, we assume that there is a special label τ which can be used to label steps of transitions which are completely invisible to an observer. Moreover, we assume that executing an invisible step does not change the visibility of the marking (any such change would indicate that something must have been executed even if we do not know what). Then, given an execution $\mu = M_0U_1M_1 \dots M_{n-1}U_nM_n$, where $M_0 \in \mathcal{M}_0$, we denote by $obs_\tau(\mu)$ the sequence obtained from

$$obs(M_0)obs(U_1)obs(M_1) \dots obs(M_{n-1})obs(U_n)obs(M_n)$$

by deleting all pairs $obs(U_i)obs(M_i)$ such that $obs(U_i) = \tau$. With this modified definition of observability, our notions of opacity are re-stated:

- $\mathcal{P}$ is *τ-initial-opaque* if for every execution μ from any marking $M \in \mathcal{M}_0 \cap \mathcal{P}$, there exists an execution μ' from a marking $M' \in \mathcal{M}_0 \setminus \mathcal{P}$ such that $obs_\tau(\mu) = obs_\tau(\mu')$.
- $\mathcal{P}$ is *τ-final-opaque* if for every execution μ from any marking $M \in \mathcal{M}_0$ to a marking $\widehat{M} \in \mathcal{P}$, there exists an execution μ' from a marking $M' \in \mathcal{M}_0$ to a marking $\widehat{M}' \notin \mathcal{P}$ such that $obs_\tau(\mu) = obs_\tau(\mu')$.

The case of always-opacity is more complicated and will be discussed in a forthcoming paper. What is important, however, is that the properties already established in this paper for different forms of opacity carry over to their τ-versions.

3.4 Step vs. interleaving semantics

In this paper we adopted the step semantics of Petri nets. But the whole discussion could just as well be carried out in terms of the interleaving semantics. No change to the notions developed nor the results would be necessary (other than assuming that all executions are based on singleton steps of transitions). The resulting model, however, would not be equivalent to the current one.

Take, for example, the net in figure 2(a) (with exactly one initial marking $\{s_1, s_3\}$, as shown in the diagram), whose reachability graph $RG(\Sigma)$ is given in figure 2(b). Assume further that all steps are 'visible' as the same label a, and that each reachable marking M is visible as the projection $M|_{\{s_4,s_5\}}$. Applying obs to the reachability graph results in the graph $obs(RG(\Sigma))$ in figure 2(c). Moreover, under the interleaving semantics, the graph $obs(RG(\Sigma))$ is as in figure 2(d).

Suppose now that we are interested in establishing whether the system is final-opaque w.r.t. the presence of a token in place s_2; in other words, $\mathcal{P} = \{\{s_2, s_4, s_5\}\}$. Then, using the interleaving version we can find that after the observed execution $\varnothing a \{s_4, s_5\}$ the place s_2 is marked, and so $\mathcal{P}$ is not final-opaque. On the other hand, the same cannot be established using the graph in figure 2(c) based on steps of transitions; in fact, in this case $\mathcal{P}$ is final-opaque.

4 Dining cryptographers

To illustrate our approach, we use a simplified (and more intimate) version of the dining cryptographers with just two dining companions. This version is also used in [4]. The standard dining cryptographers involves three diners and admits some further anonymity properties, e.g., a paying cryptographer

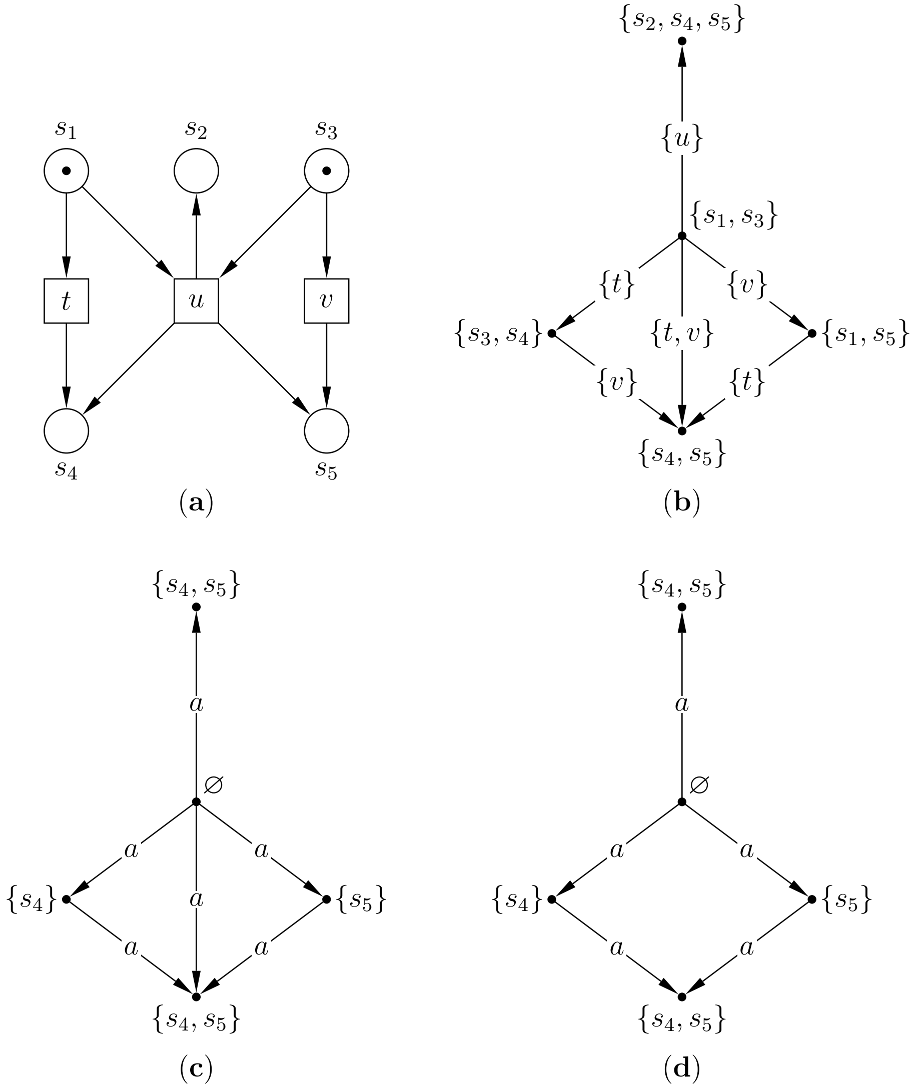

Fig. 2. Step vs. interleaving semantics.

can remain anonymous w.r.t. his or her companions. Our construction is straightforwardly extended to three or more diners.

Two cryptographers, Anne and Bob, enjoy a meal in a restaurant. When they call for the bill, the waiter tells them that it has already been paid. Anne and Bob each wish to know whether the bill was paid by the NSA, or if it was one of them. However, if one of them paid, they do not want an eavesdropper, Yves, on the neighbouring table to know which of them paid. The protocol they choose to solve this problem is as follows:

They toss two coins, visible to both of them, ensuring that Yves cannot see either of them. If Anne paid, she lies about the parity of the two coins (she calls 'agree' if she sees a head and a tail, and 'disagree' otherwise). If Anne did not pay, she tells the truth about the parity of the coins. Similarly for Bob. Now Anne and Bob both know if one of them paid: if their calls are the same they know that the NSA paid, otherwise it must have been one of them (and in this example they actually both know which). Yves, on the other hand, can only tell whether or not one of Anne and Bob paid, but not which one.

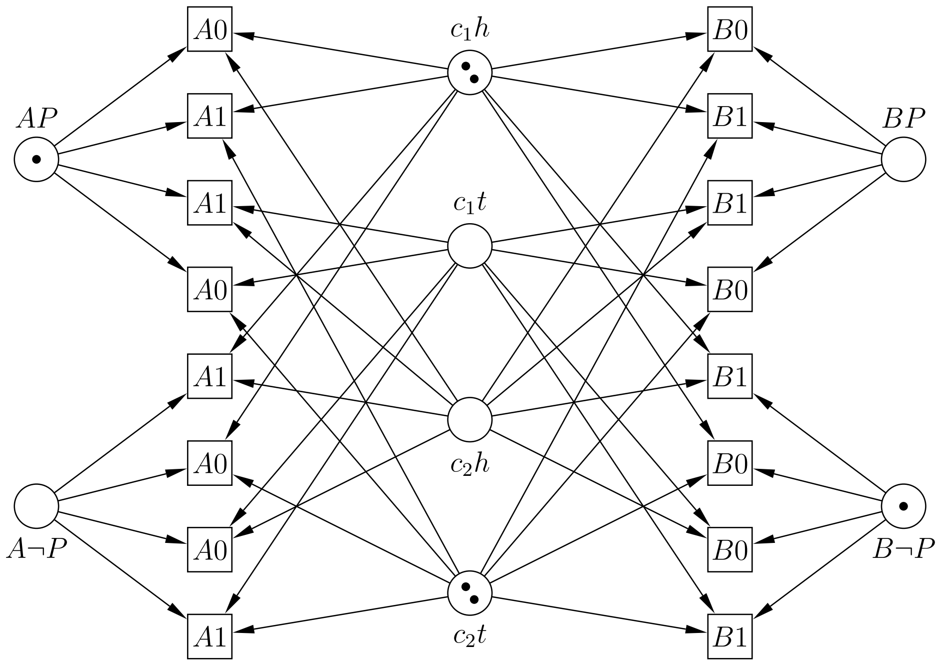

Fig. 3. Net for the dining cryptographers example with one of the 12 initial markings.

Figure 3 presents a possible encoding of the protocol. The two places at the left of the diagram represent Anne's initial state (having paid is represented by placing a single token in place AP, and having not paid is represented by placing a single token in place $A\neg P$), and the two places at the right represent Bob's initial state. The possible initial markings for these places are

$$\{AP, B\neg P\}, \{A\neg P, BP\}, \{A\neg P, B\neg P\}.$$

The top two places in the centre of the diagram represent one coin (heads is represented by placing tokens in place c_1h, and tails is represented by placing tokens in place c_1t), and the bottom two places represent the second coin. For each pair, the marked place must contain two tokens. This is because both Anne and Bob must see each coin. The possible initial markings for the coins are therefore

$$\{c_1h, c_1h, c_2h, c_2h\}\ \{c_1h, c_1h, c_2t, c_2t\}$$

$$\{c_1t, c_1t, c_2h, c_2h\}\ \ \{c_1t, c_1t, c_2t, c_2t\}$$

The set of possible initial markings, $\mathcal{M}_0$, is the cross product of the cryptographer markings and the coin markings.

The eight transitions on the left represent the eight possible scenarios for Anne, given by two possibilities for each coin multiplied by the two possibilities for her own initial state. Each transition on the left is labelled with '$A0$' (if Anne says the coins 'disagree') or '$A1$' (if Anne says the coins 'agree'). Similarly for Bob on the right. This gives the transition labelling ℓ which will be used for defining the visibility of steps.

Yves' observation function is simple. He can see none of the places (he does not know the initial state of Anne and Bob, nor the state of the coins), but he can see all of the labels of the executed transitions (he can hear all that Anne and Bob say). In other words, for every reachable marking M and executed step U, we have the following (see section 3.1):

$$obs_Y(M) = M|_{\varnothing} = \varnothing$$

$$obs_Y(U) = \ell(U).$$

Note that Yves also knows the structure of the original net, i.e., the protocol.

We wish to demonstrate that after observing the execution of transitions, although Yves may be able to determine whether the meal was paid for by one of the cryptographers, he can never know which one. The two (symmetric) properties we wish to be initial-opaque are therefore $\mathcal{P}_1 = \{M \in \mathcal{M}_0 \mid M(AP) = 1\}$ and $\mathcal{P}_2 = \{M \in \mathcal{M}_0 \mid M(BP) = 1\}$.

If, for example, Yves observes $\{A0, B1\}$ he knows that the initial marking was either $\{AP, B\neg P\}$ with either two heads or two tails, or $\{A\neg P, BP\}$ with the two coins distinct. Yves cannot determine the satisfaction of either of the two properties. Similarly if he observes $\{A1, B0\}$. Note, however, that Yves can in either case determine the satisfaction of the property $\mathcal{P} = \{M \in \mathcal{M}_0 \mid M(AP)+M(BP) = 1\}$, i.e., he knows when one or other of the cryptographers paid the bill. In terms of our framework, both $\mathcal{P}_1$ and $\mathcal{P}_2$ are initial-opaque, but $\mathcal{P}$ is not. If Yves hears $\{A0, B0\}$ or $\{A1, B1\}$ he of course knows that neither of the cryptographers paid.

This example can easily be altered to model the point of view of one of the two cryptographers. We simply change the *obs* function to model the increased level of knowledge. For example, the observation function of Anne is such that, for every reachable marking M and executed step U,

$$obs_{Anne}(M) = M|_{\{AP, A\neg P, c_1h, c_1t, c_2h, c_2t\}}$$

$$obs_{Anne}(U) = \ell(U).$$

Anne knows her own initial state, and can see the state of both of the coins. Given this observation function, she learns what she wants to know — whether or not Bob paid the bill.

5 Conclusions and future work

We have presented a Petri net framework in which a rich class of information flow requirements can be conveniently expressed and analysed. The example of the dining cryptographers illustrates how an anonymity property can be captured. This kind of property is problematic to capture in the traditional, strict formulations of non-interference.

We have further presented a number of decidability results for the opacity properties presented here.

In future work we intend to explore the formulation of richer information flow requirements, e.g., partial, conditional, intransitive flows [8]. We also intend to explore the relationship of the approach presented here to process algebraic formulations of generalised non-interference [9] and anonymity [10]. Here, richer notions of information flow are formalised as the invariance of an appropriate abstraction of the system under certain transformations. Thus anonymity can be expressed as invariance under permutations over a set of identities. Message secrecy can similarly be expressed as invariance under (length preserving) transformations of plaintext and so on.

A major challenge in such work is the choice of appropriate abstractions to encode the adversary's observational capabilities. This is particularly delicate where cryptographic mechanisms are involved. Adversary deductions, algebraic manipulations, and key compromise complicate the modelling. It may be that incorporating the possibility of dynamic *obs* mappings may help address such issues.

We will also investigate the problem of preservation of opacity properties under refinement and composition of Petri nets.

A further line of research is to explore analogues in this framework of the notion of *non-deducibility on strategies*, due to Johnson and Wittbold [12].

This seeks to capture the possibility of a secret user and an uncleared user colluding and using adaptive strategies to cause information flows in violation of the policy. This is likely to require more precise modelling of various flavours of non-determinism within the Petri net framework.

References

[1] Busi, N., and R. Gorrieri, *Structural Non-interference with Petri Nets*, Proc. of WITS, 2004, 27-42.

[2] Cohen, E., *Information Transmission in Computational Systems*, Proc. of ACM Symposium on Operating System Principles, 1997, 133-139.

[3] Feiertag, R. J., *A Technique for Proving Specifications are Multi-level Secure*, SRI International, Techical Report, 1980.

[4] Mazaré, L., *Using Unification for Opacity Properties*, Proc. of WITS, 2004, 165-176.

[5] Goguen, J. A., and J. Meseguer, *Security Policies and Security Models*, Proc. of IEEE Symposium on Security and Privacy, 1982, 1-10.

[6] Goguen, J. A., and J. Meseguer, *Inference Control and Unwinding*, Proc. of IEEE Symposium on Research in Security and Privacy, 1984, 75-86.

[7] Reisig, W., and G. Rozenberg, "Lectures on Petri Nets," Springer Verlag, 1998.

[8] Rushby, J., *Noninterference, Transitivity and Channel-Control Security Policies*, SRI International, Techical Report, 1992.

[9] Ryan, P. Y. A., *Mathematical Models of Computer Security*, Proc. of Foundations of Security Analysis and Design, Springer-Verlag, Lecture Notes in Computer Science 2172, 2001, 1-62.

[10] Schneider, S. A., and A. Sidiropoulos, *CSP and Anonymity*, Proc. of ESORICS, 2000, 198-218.

[11] Sutherland, D., *A Model of Information*, Proc. of National Computer Security Conference, 1986, 175-183.

[12] Wittbold, J. T., and D. M. Johnson, *Information Flow in Nondeterministic Systems*, Proc. of Symposium on Research on Security and Privacy, 1990, 144-161.