On communication models when verifying equivalence properties

Kushal Babel, Vincent Cheval, Steve Kremer

On communication models when verifying equivalence properties

Kushal Babel[1], Vincent Cheval[2], Steve Kremer[2]

[1] IIT Bombay, India
[2] LORIA, Inria Nancy & CNRS & Université de Lorraine, France

Abstract. Symbolic models for security protocol verification, following the seminal ideas of Dolev and Yao, come in many flavors, even though they share the same ideas. A common assumption is that the attacker has complete control over the network: he can therefore intercept any message. Depending on the precise model this may be reflected either by the fact that any protocol output is directly routed to the adversary, or communications may be among any two participants, including the attacker — the scheduling between which exact parties the communication happens is left to the attacker. These two models may seem equivalent at first glance and, depending on the verification tools, either one or the other semantics is implemented. We show that, unsurprisingly, they indeed coincide for reachability properties. However, when we consider indistinguishability properties, we prove that these two semantics are incomparable. We also introduce a new semantics, where internal communications are allowed but messages are always eavesdropped by the attacker. We show that this new semantics yields strictly stronger equivalence relations. We also identify two subclasses of protocols for which the three semantics coincide. Finally, we implemented verification of trace equivalence for each of these semantics in the APTE tool and compare their performances on several classical examples.

1 Introduction

Automated, symbolic analysis of security protocols, based on the seminal ideas of Dolev and Yao, comes is many variants. All of these models however share a few fundamental ideas:

- messages are represented as abstract terms,
- adversaries are computationally unbounded, but may manipulate messages only according to pre-defined rules (this is sometimes referred to as the perfect cryptography assumption), and
- the adversary completely controls the network.

In this paper we will revisit this last assumption. Looking more precisely at different models we observe that this assumption may actually slightly differ among the models. The fact that the adversary controls the network is supposed to represent a *worst case* assumption.

In some models this assumption translates to the fact that every protocol output is sent to the adversary, and every protocol input is provided by the adversary. This is the

case in the original Dolev Yao model and also in the models underlying several tools, such as AVISPA [6], Scyther [13], Tamarin [20], Millen and Shmatikov's constraint solver [17], and the model used in Paulson's inductive approach [18].

Some other models, such as those based on process algebras, e.g. work based on CSP [19], the Spi [3] and applied pi calculus [1], but also the strand space model [21], consider a slightly different communication model: any two agents may communicate. Scheduling whether communication happens among two honest participants, or a honest participant and the attacker is under the attacker's control.

When considering *reachability properties*, these two communication models indeed coincide: intuitively, any internal communication could go through the adversary who acts as a relay and increases his knowledge by the transmitted message. However, when considering *indistinguishability properties*, typically modelled as process equivalences, these communication models diverge. Interestingly, when forbidding internal communication, i.e., forcing all communication to be relayed by the attacker, we may weaken the attacker's distinguishing power.

In many recent work privacy properties have been modelled using process equivalences, see for instance [14, 5, 15]. The number of tools able to verify such properties is also increasing [9, 22, 11, 10]. We have noted that for instance the AKISS tool [10] does not allow any direct communication on public channels, while the APTE tool [11] allows the user to choose among the two semantics. One motivation for disallowing direct communication is that it allows for more efficient verification (as less actions need to be considered and the number of interleavings to be considered is smaller).

Our contributions. We have formalised three semantics in the applied pi calculus which differ by the way communication is handled:

- the *classical* semantics (as in the original applied pi calculus) allows both internal communication among honest participants and communication with the adversary;
- a *private* semantics allows internal communication only on private channels while all communication on public channels is routed through the adversary;
- an *eavesdropping* semantics which allows internal communication, but as a side-effect adds the transmitted message to the adversary's knowledge.

For each of the new semantics we define may-testing and observational equivalences. We also define corresponding labelled semantics and trace equivalence and bisimulation relations (which may serve as proof techniques).

We show that, as expected, the three semantics coincide for reachability properties. For equivalence properties we show that the classical and private semantics yield incomparable equivalences, while the eavesdropping semantics yields strictly stronger equivalence relations than both other semantics. The results are summarized in Figure 7.

An interesting question is whether these semantics coincide for specific subclasses of processes. We first note that the processes that witness the differences in the semantics do not use replication, private channels, nor terms other than names, and no equational theory. Moreover, all except one of these examples only use trivial *else* branches (of the form $\mathsf{else}\ 0$); the use of a non-trivial else branch can however be avoided by allowing a single free symbol.

However conditions on the channel names may yield such a subclass. We first observe that the class of *simple processes* [12], for which already observational, testing, trace equivalence and labelled bisimulation coincide, do have this property. Simple processes may however be too restrictive for modelling some protocols that should guarantee anonymity (as no parallel processes may share channel names). We therefore identify a syntactic class of processes, that we call *I/O-unambiguous*. For this class we forbid communication on private channels, communication of channel names and an output may not be sequentially followed by an input on the same channel directly, or with only conditionals in between. Note that I/O-unambiguous processes do however allow outputs and inputs on the same channel in parallel. We show that for this class the eavesdropping semantics (which is the most strict relation) coincides with the private one (which is the most efficient for verification).

Finally, we have extended the APTE tool to support verification of trace equivalence for the three semantics. Verifying existing protocols in the APTE example repository we verified that the results, fortunately, coincided for each of the semantics. We also made slight changes to the encodings, renaming some channels, to make them I/O-unambiguous. Interestingly, using different channels, significantly increased the performance of the tool. Finally, we also observed that, as expected, the private semantics yields more efficient verification. The results of our experiments are summarized in Figure **??**.

Outline. In Section 2 we define the three semantics we consider. In Section 3 we present our main results on comparing these semantics. We present subclasses for which (some) semantics coincide in Section 4 and compare the performances when verifying protocols for different semantics using APTE in Section 5, before concluding in Section 6.

Because of lack of space we did not include all proofs. Missing proofs are available in an extended [7].

2 Model

The *applied pi calculus* [1] is a variant of the pi calculus that is specialised for modelling cryptographic protocols. Participants in a protocol are modelled as processes and the communication between them is modelled by message passing on channels. In this section, we describe the syntax and semantics of the applied pi calculus as well as the two new variants that we study in this paper.

2.1 Syntax

We consider an infinite set $\mathcal{N}$ of names of *base type* and an infinite set $\mathcal{C}h$ of names of *channel type*. We also consider an infinite set of variables $\mathcal{X}$ of base type and channel type and a signature $\mathcal{F}$ consisting of a finite set of *function symbols*. We rely on a sort system for terms. In particular, the sort base type differs from the sort channel type. Moreover, any function symbol can only be applied and returns base type terms. We define *terms* as names, variables and function symbols applied to other terms. Given $N \subseteq \mathcal{N}$, $X \subseteq \mathcal{X}$ and $F \subseteq \mathcal{F}$, we denote by $\mathcal{T}(F, X, N)$ the sets of terms built from

X and N by applying function symbols from F. We denote $fv(t)$ the sets of variables occurring in t. We say that t is *ground* if $fv(t) = \emptyset$. We describe the behaviour of cryptographic primitives by the means of an *equational theory* E that is a relation on terms closed under substitutions of terms for variables and closed under one-to-one renaming. Given two terms u and v, we write $u =_{\mathsf{E}} v$ when u and v are equal modulo the equational theory.

In the original syntax of the applied pi calculus, there is no distinction between an output (resp. input) from a protocol participant and from the environment, also called the attacker. In this paper however, we will make this distinction in order to concisely present our new variants of the semantics. Therefore, we consider two *process tags* ho and at that respectively represent honest and attacker actions. The syntax of *plain processes* and *extended processes* is given in Figure 1.

$P, Q :=$		plain processes	
	0		
	$P \mid Q$		
	$!P$		
	$\nu n.P$		
	$\mathsf{if}\ u = v\ \mathsf{then}\ P\ \mathsf{else}\ Q$		
	$\mathsf{in}^{\theta}(c, x).P$		
	$\mathsf{out}^{\theta}(c, u).P$		
	$\mathsf{eav}(c, x).P$		
$A, B :=$		extended processes	
	P		
	$A \mid B$		
	$\nu n.A$		
	$\nu x.A$		
	$\{^{u}/_{x}\}$		
	ωc		

where u and v are base type terms, n is a name, x is a variable and c is a name or variable of channel type, θ is a tag, *i.e.* $\theta \in \{\mathsf{ho}, \mathsf{at}\}$.

Fig. 1. Syntax of processes

The process $\mathsf{out}^{\theta}(c, u)$ represents the output by θ of the message u on the channel c. The process $\mathsf{in}^{\theta}(c, x)$ represents an input by θ on the channel c. The input message will instantiate the variable x. The process $\mathsf{eav}(c, x)$ models the capability of the attacker to eavesdrop a communication on channel c. The process $!P$ represents the replication of the process P, *i.e.* unbounded number of copies of P. The process $P \mid Q$ represents the parallel composition of P and Q. The process $\nu n.P$ (resp. $\nu x.A$) is the restriction of the name n in P (resp. variable x in A). The process $\mathsf{if}\ u = v\ \mathsf{then}\ P\ \mathsf{else}\ Q$ is the conditional branching under the equality test $u = v$. The process ωc records that a private channel c has been opened, i.e., it has been sent on a public or previously opened channel. Finally, the substitution $\{^{u}/_{x}\}$ is an active substitution that replaces the variable x with the term u of base type.

We say that a process P (resp. extended process A) is an *honest process* (resp. *honest extended process*) when all inputs and outputs in P (resp. A) are tagged with ho and when P (resp. A) does not contain eavesdropping processes and ωc. We say that a process P (resp. extended process A) is an *attacker process* (resp. *attacker extended process*) when all inputs and outputs in P (resp. A) are tagged with at.

As usual, names and variables have scopes which are delimited by restrictions, inputs and eavesdrops. We denote $fv(A), bv(A), fn(A), bn(A)$ the sets of free variables, bound variables, free names and bound names respectively in A. Moreover, we denote

by $oc(A)$ the sets of terms c of channel type opened in A, *i.e.* that occurs in a process ωc. We say that an extended process A is closed when all variables in A are either bound or defined by an active substitution in A. We define an *evaluation context* $C[\_]$ as an extended process with a hole instead of an extended process. As for processes, we define an *attacker evaluation context* as an evaluation context where all outputs and inputs in the context are tagged with at.

Note that our syntax without the eavesdropping process, opened channels and tags correspond exactly to the syntax of the original applied pi calculus.

Lastly, we consider the notion of *frame* that are extended processes built from 0, parallel composition, name and variable restrictions and active substitution. Given a frame φ, we consider the domain of φ, denoted $dom(\varphi)$, as the set of free variables in φ that are defined by an active substitution in φ. Given an extended process A, we define the frame of A, denoted $\phi(A)$, as the process A where we replace all plain processes by 0. Finally, we write $dom(A)$ as syntactic sugar for $dom(\phi(A))$.

2.2 Operational semantics

In this section, we define the three semantics that we study in this paper, namely:

- the *classical semantics* from the applied pi calculus, where internal communication can occur on both public and private channels;
- the *private semantics* where internal communication can only occur on private channels; and
- the *eavesdropping semantics* where the attacker is able to eavesdrop on a public channel.

We first define the *structural equivalence* between extended processes, denoted $\equiv$, as the smallest equivalence relation on extended processes that is closed under renaming of names and variables, closed by application of evaluation contexts, that is associative and commutative w.r.t. $|$, and such that:

$$A \equiv A \mid 0 \qquad !P \equiv !P \mid P \qquad \nu n.0 \equiv 0$$
$$\nu i.\nu j.A \equiv \nu j.\nu i.A \qquad \nu x.\{^u/_x\} \equiv 0 \qquad \{^u/_x\} \mid A \equiv \{^u/_x\} \mid A\{^u/_x\}$$
$$A \mid \nu i.B \equiv \nu i.(A \mid B) \quad \text{when } i \notin fv(A) \cup fn(A) \qquad \omega c \equiv \omega c \mid \omega c$$
$$\{^u/_x\} \equiv \{^v/_x\} \quad \text{when } u =_{\mathsf{E}} v$$

The three operational semantics of extended processes are defined by the structural equivalence and by three respective *internal reductions*, denoted $\rightarrow_{\mathsf{c}}$, $\rightarrow_{\mathsf{p}}$ and $\rightarrow_{\mathsf{e}}$. These three reductions are the smallest relations on extended processes that are closed under application of evaluation context, structural equivalence and such that:

$$\mathsf{if}\ u = v\ \mathsf{then}\ P\ \mathsf{else}\ Q \xrightarrow{\tau}_s P \quad \text{where } u =_{\mathsf{E}} v \text{ and } s \in \{\mathsf{c}, \mathsf{p}, \mathsf{e}\} \qquad \textsc{Then}$$
$$\mathsf{if}\ u = v\ \mathsf{then}\ P\ \mathsf{else}\ Q \xrightarrow{\tau}_s Q \qquad \textsc{Else}$$
$$\text{where } u, v \text{ ground, } u \neq_{\mathsf{E}} v \text{ and } s \in \{\mathsf{c}, \mathsf{p}, \mathsf{e}\}$$

$$\mathsf{out}^{\theta}(c, u).P \mid \mathsf{in}^{\theta'}(c, x).Q \xrightarrow{\tau}_{\mathsf{c}} P \mid Q\{^u/_x\} \qquad \textsc{Comm}$$

$$\nu c.(\mathsf{out}^{\theta}(c,u).P \mid \mathsf{in}^{\theta'}(c,x).Q \mid R) \xrightarrow{\tau}_s \nu c.(P \mid Q\{^u/_x\} \mid R) \quad \text{C-PRIV}$$
where $c \notin oc(R)$ and $s \in \{\mathsf{p},\mathsf{e}\}$

$$\mathsf{out}^{\theta}(c,u).P \mid \mathsf{in}^{\theta'}(c,x).Q \xrightarrow{\tau}_s P \mid Q\{^u/_x\} \quad \text{C-ENV}$$
$\mathsf{at} \in \{\theta,\theta'\}$, u is of base type and $s \in \{\mathsf{p},\mathsf{e}\}$

$$\mathsf{out}^{\theta}(c,d).P \mid \mathsf{in}^{\theta'}(c,x).Q \xrightarrow{\tau}_s P \mid Q\{^d/_x\} \mid \omega d \quad \text{C-OPEN}$$
$\mathsf{at} \in \{\theta,\theta'\}$, d is of channel type and $s \in \{\mathsf{p},\mathsf{e}\}$

$$\mathsf{out}^{\mathsf{ho}}(c,u).P \mid \mathsf{in}^{\mathsf{ho}}(c,x).Q \mid \mathsf{eav}(c,y).R \xrightarrow{\tau}_{\mathsf{e}} P \mid Q\{^u/_x\} \mid R\{^u/_y\} \quad \text{C-EAV}$$
where u is of base type

$$\mathsf{out}^{\mathsf{ho}}(c,d).P \mid \mathsf{in}^{\mathsf{ho}}(c,x).Q \mid \mathsf{eav}(c,y).R \xrightarrow{\tau}_{\mathsf{e}} P \mid Q\{^d/_x\} \mid R\{^d/_y\} \mid \omega d \quad \text{C-OEAV}$$
where d is of channel type

We emphasise that the application of the rule is closed under application of arbitrary evaluation contexts. In particular the context may restrict channels, *e.g.* the rule C-OPEN may be used under the context $\nu c._-$ resulting in a private channel c, but with the attacker input/output being in the scope of this restriction. It follows from the definition of evaluation contexts that the resulting processes are always well defined. We denote by $\Rightarrow_s$ the reflexive, transitive closure of $\xrightarrow{\tau}_s$ for $s \in \{\mathsf{c},\mathsf{p},\mathsf{e}\}$. We note that the classical semantics $\xrightarrow{\tau}_{\mathsf{c}}$ is independent of the tags θ,θ', the eavesdrop actions and the ωc processes.

Example 1. Consider the process

$$A = (\nu d.\mathsf{out}^{\theta}(c,d).\mathsf{in}^{\theta}(d,x).P) \mid (\mathsf{in}^{\theta'}(c,y).\mathsf{out}^{\theta'}(y,t).Q)$$

where d is a channel name and t a term of base type. Suppose $\theta = \theta' = \mathsf{ho}$ then we have that communication is only possible in the classical semantics (using twice the COMM rule):

$$\begin{aligned} A &\xrightarrow{\tau}_{\mathsf{c}} \nu d.(\mathsf{in}^{\theta}(d,x).P \mid \mathsf{out}^{\theta'}(d,t).Q\{^d/_y\}) \\ &\xrightarrow{\tau}_{\mathsf{c}} \nu d.(P\{^t/_x\} \mid Q\{^d/_y\}) \end{aligned}$$

while no transitions are available in the two other semantics. To enable communication in the eavesdropping semantics we need to explicitly add eavesdrop actions. Applying the rules C-OEAV and C-EAV we have that

$$\begin{aligned} A \mid \mathsf{eav}(c,z_1).\mathsf{eav}(z_1,z_2).R \xrightarrow{\tau}_{\mathsf{e}} &\nu d.(\mathsf{in}^{\theta}(d,x).P \mid \mathsf{out}^{\theta'}(d,t).Q\{^d/_y\} \\ &\mid \mathsf{eav}(d,z_2).R\{^d/_{z_1}\} \mid \omega d) \\ \xrightarrow{\tau}_{\mathsf{e}} &\nu d.(P\{^t/_x\} \mid Q\{^d/_y\} \mid R\{^d/_{z_1}\}\{^t/_{z_2}\} \mid \omega d) \end{aligned}$$

We note that the first transition adds the information ωd to indicate that d is now available to the environment.

Finally, if we consider that $\mathsf{at} \in \theta,\theta'$ then internal communication on a public channel is possible and, using rules C-OPEN and C-ENV we obtain for $s \in \{\mathsf{p},\mathsf{e}\}$ that

$$\begin{aligned} A &\xrightarrow{\tau}_s \nu d.(\mathsf{in}^{\theta}(d,x).P \mid \mathsf{out}^{\theta'}(d,t).Q\{^d/_y\} \mid \omega d) \\ &\xrightarrow{\tau}_s \nu d.(P\{^t/_x\} \mid Q\{^d/_y\} \mid \omega d) \end{aligned}$$

2.3 Reachability and behavioural equivalences

We are going to compare the relation between the three semantics for the two general kind of security properties, namely *reachability properties* encoding security properties such as secrecy, authentication, and *equivalence properties* encoding anonymity, unlinkability, strong secrecy, receipt freeness, ... Intuitively, reachability properties encode that a process cannot reach some bad state. Equivalences define the fact that no attacker can distinguish two processes. This was originally defined by the *(may)-testing equivalence* [3] in the spi-calculus. An alternate equivalence, which was considered in the applied pi calculus [1], is observational equivalence.

Reachability properties can simply be encoded by verifying the capability of a process to perform an output on a given cannel. We define $A \Downarrow_c^{s,\theta}$ to hold when $A \Rightarrow_s C[\mathsf{out}^\theta(c,t).P]$ for some evaluation context C that does not bind c, some term t and some plain process P, and $A \Downarrow_c^s$ to hold when $A \Downarrow_c^{s,\theta}$ for some $\theta \in \{\mathsf{at}, \mathsf{ho}\}$. For example the secrecy of s in the process $\nu s.A$ can be encoded by checking whether for all attacker plain process I, we have that

$$I \mid \nu s.(A \mid \mathsf{in}^{\mathsf{ho}}(c,x).\mathsf{if}\ x = s\ \mathsf{then}\ \mathsf{out}^{\mathsf{ho}}(\mathsf{bad}, s)) \not\Downarrow_{\mathsf{bad}}^{s,\mathsf{ho}}$$

where $\mathsf{bad} \notin fn(A)$.

Authentication properties are generally expressed as correspondence properties between events annotating processes, see e.g. [8]. A correspondence property between two events begin and end, denoted $\mathsf{begin} \Leftarrow \mathsf{end}$, requires that the event end is preceded by the event begin on every trace. A possible encoding of this correspondence property consists in first replacing all instances of the events in A by outputs $\mathsf{out}^{\mathsf{ho}}(ev, \mathsf{begin})$ and $\mathsf{out}^{\mathsf{ho}}(ev, \mathsf{end})$ where $ev \notin fn(A) \cup bn(A)$. This new process A' can then be put in parallel with a cell $Cell$ that reads on the channel ev and stores any new value unless the value is end and the current stored value in the cell is not begin. In such a case, the cell will output on the channel bad. The correspondance property can therefore be encoded by checking whether for all attacker plain process I, we have that $I \mid \nu ev.(A' \mid Cell) \not\Downarrow_{\mathsf{bad}}^{s,\mathsf{ho}}$.

We say that an attacker evaluation context $C[\_]$ is c-closing for an extended process A if $fv(C[A]) = \emptyset$. For $s \in \{\mathsf{p}, \mathsf{e}\}$, we say that $C[\_]$ is s-closing for A if it is c-closing for A, variables and names are bound only once in $C[\_]$ and for all channels $c \in bn(C[\_]) \cap fn(A)$, if the scope of c includes $\_$ then the scope of c also includes ωc.

We next introduce the two main notions of behavioural equivalences: may testing and observational equivalence.

Definition 1 ((May-)Testing equivalences $\approx_m^{\mathsf{c}}$, $\approx_m^{\mathsf{p}}$, $\approx_m^{\mathsf{e}}$). *Let $s \in \{\mathsf{c}, \mathsf{p}, \mathsf{e}\}$. Let A and B two closed honest extended processes such that $dom(A) = dom(B)$. We say that $A \approx_m^s B$ if for all attacker evaluation contexts $C[\_]$ s-closing for A and B, for all channels c, we have that $C[A] \Downarrow_c^s$ if and only if $C[B] \Downarrow_c^s$.*

Definition 2 (Observational equivalences $\approx_o^{\mathsf{c}}$, $\approx_o^{\mathsf{p}}$, $\approx_o^{\mathsf{e}}$). *Let $s \in \{\mathsf{c}, \mathsf{p}, \mathsf{e}\}$. Let A and B two closed extended processes such that $dom(A) = dom(B)$. We say that $A \approx_m^s B$ if $\approx_m^s$ is the largest equivalence relation such that:*

- *$A \Downarrow_c^s$ implies $B \Downarrow_c^s$;*

- $A \xrightarrow{\tau}_s A'$ *implies* $B \stackrel{\epsilon}{\Rightarrow}_s B'$ *and* $A' \approx^s_m B'$ *for some* B'*;*
- $C[A] \approx^s_m C[B]$ *for all attacker evaluation contexts* $C[\_]$ *s-closing for* A *and* B*.*

For each of the semantics we have the usual relation between these two notions: observational equivalence implies testing equivalence.

Proposition 1. $\approx^s_o \subsetneq \approx^s_m$ *for* $s \in \{\mathsf{c}, \mathsf{e}, \mathsf{p}\}$.

Example 2. Consider processes A and B of Figure 2. Process A computes a value $h^n(a)$ to be output on channel c, where $h^n(a)$ denotes n applications of h and $h^0(a) = a$. The value is initially a and A may choose to either output the current value, or update the current value by applying the free symbol h. B may choose non-deterministically to either behave as A or output the fresh name s. (The non-deterministic choice is encoded by a communication on the private channel e which may be received by either the process behaving as A or the process outputting s.)

We have that $A \not\approx^s_o B$. The two processes can indeed be distinguished by the context

$$\begin{aligned} C[\_] \mathrel{\hat{=}} \_ \mid \mathsf{out}^{\mathsf{at}}(c_a, a) \mid !(\mathsf{in}^{\mathsf{at}}(c_a, x).\mathsf{out}^{\mathsf{at}}(c_a, h(x)) \\ \mid \mathsf{in}^{\mathsf{at}}(c_a, y).\mathsf{in}^{\mathsf{at}}(c, z).\mathsf{if}\ y = z\ \mathsf{then}\ \mathsf{out}^{\mathsf{at}}(c_t, h(x)) \end{aligned}$$

Intuitively, when B outputs s the attacker context $C[\_]$ can iterate the application of h the same number of times as would have done process A. Comparing the value computed by the adversary ($h^n(a)$) and the honestly computed value (either $h^n(a)$ or s) the adversary distinguishes the two processes by outputting on the test channel c_t.

However, we have that $A \approx^s_m B$. Indeed, for any s-closing context $D[\_]$ and all public channel ch we have that $D[A] \Downarrow^s_{ch}$ if and only if $D[B] \Downarrow^s_{ch}$. In particular for context $C[\_]$ defined above we have that both $C[A] \Downarrow^s_{ch}$ and $C[B] \Downarrow^s_{ch}$ for $ch \in \{c_a, c_t, c\}$. Unlike observational equivalence, may testing does not require to "mimick" the other process stepwise and we cannot force a process into a particular branch.

$$\begin{aligned} A &\mathrel{\hat{=}} \nu d.\mathsf{out}^{\mathsf{ho}}(d, a) \mid !\mathsf{in}^{\mathsf{ho}}(d, x).\mathsf{out}^{\mathsf{ho}}(d, h(x)) \mid \mathsf{in}^{\mathsf{ho}}(d, y).\mathsf{out}^{\mathsf{ho}}(c, y) \\ B &\mathrel{\hat{=}} \nu e.\mathsf{out}^{\mathsf{ho}}(e, a) \mid \mathsf{in}^{\mathsf{ho}}(e, z).A \mid \mathsf{in}^{\mathsf{ho}}(e, z).\nu s.\mathsf{out}^{\mathsf{ho}}(c, s) \end{aligned}$$

Fig. 2. Processes A and B such that $A \approx^s_m B$, but $A \not\approx^s_o B$ and $A \not\approx^s_t B$ for $s \in \{\mathsf{c}, \mathsf{e}, \mathsf{p}\}$.

2.4 Labelled semantics

The internal reduction semantics introduced in the previous section requires to reason about arbitrary contexts. Similar to the original applied pi calculus, we extend the three operational semantics by a *labeled operational semantics* which allows processes to directly interact with the (adversarial) environment: we define the relation $\xrightarrow{\ell}_{\mathsf{c}}$, $\xrightarrow{\ell}_{\mathsf{p}}$ and $\xrightarrow{\ell}_{\mathsf{e}}$ where ℓ is part of the alphabet $\mathcal{A} = \{\tau, out(c, d), eav(c, d), in(c, w), \nu k.out(c, k), \nu k.eav(c, k) \mid c, d \in \mathcal{C}h, k \in \mathcal{X} \cup \mathcal{C}h$ and w is a term of any sort$\}$. The labeled rules are given in Figure 3.

$$\textsc{In}\quad \mathsf{in}^{\mathsf{ho}}(c,y).P \xrightarrow{in(c,t)}_s P\{^t/_y\} \qquad \textsc{Scope}\quad \frac{A \xrightarrow{\ell}_s A' \quad u \text{ does not occur in } \ell}{\nu u.A \xrightarrow{\ell}_s \nu u.A'}$$

$$\textsc{Out-Ch}\quad \mathsf{out}^{\mathsf{ho}}(c,d).P \xrightarrow{out(c,d)}_s P$$

$$\textsc{Open-Ch}\quad \frac{A \xrightarrow{out(c,d)}_s A' \qquad d \neq c}{\nu d.A \xrightarrow{\nu d.out(c,d)}_s A'} \qquad \textsc{Par}\quad \frac{A \xrightarrow{\ell}_s A' \quad bn(\ell) \cap fn(B) = \emptyset \quad bv(\ell) \cap fv(B) = \emptyset}{A \mid B \xrightarrow{\ell}_s A' \mid B}$$

$$\textsc{Eav-Och}\quad \frac{A \xrightarrow{eav(c,d)}_{\mathsf{e}} A' \qquad d \neq c}{\nu d.A \xrightarrow{\nu d.eav(c,d)}_{\mathsf{e}} A'} \qquad \textsc{Struct}\quad \frac{A \equiv B \quad B \xrightarrow{\ell}_s B' \quad B' \equiv A'}{A \xrightarrow{\ell}_s A'}$$

$$\textsc{Eav-Ch}\quad \mathsf{out}^{\mathsf{ho}}(c,d).P \mid \mathsf{in}^{\mathsf{ho}}(c,x).Q \xrightarrow{eav(c,d)}_{\mathsf{e}} P \mid Q\{^d/_x\}$$

$$\textsc{Eav-T}\quad \mathsf{out}^{\mathsf{ho}}(c,t).P \mid \mathsf{in}^{\mathsf{ho}}(c,x).Q \xrightarrow{\nu y.eav(c,y)}_{\mathsf{e}} P \mid Q\{^t/_x\} \mid \{^t/_y\}$$

$$\textsc{Out-T}\quad \mathsf{out}^{\mathsf{ho}}(c,t).P \xrightarrow{\nu x.out(c,x)}_s P \mid \{^t/_x\} \quad x \notin fv(P) \cup fv(t)$$

where $s \in \{\mathsf{c}, \mathsf{p}, \mathsf{e}\}$.

Fig. 3. Labeled semantics

Consider our alphabet of actions $\mathcal{A}$ defined above. Given $w \in \mathcal{A}^*$, $s \in \{\mathsf{c}, \mathsf{p}, \mathsf{e}\}$ and an extended process A, we say that $A \xrightarrow{w}_s A_n$ when $A \xrightarrow{\ell_1}_s A_1 \xrightarrow{\ell_2}_s A_2 \xrightarrow{\ell_3}_s \ldots \xrightarrow{\ell_n}_s A_n$ for some extended processes $A_1, \ldots, A_n$ and $w = \ell_1 \cdot \ldots \cdot \ell_n$. By convention, we say that $A \xrightarrow{\epsilon}_s A$ where ϵ is the empty word. Given $\mathsf{tr} \in (\mathcal{A} \setminus \{\tau\})^*$, we say that $A \overset{\mathsf{tr}}{\Longrightarrow}_s A'$ when there exists $w \in \mathcal{A}^*$ such that tr is the word w where we remove all τ actions and $A \xrightarrow{w}_s A'$.

Example 3. Coming back to Example 1, we saw that $A \xrightarrow{\tau}_{\mathsf{c}} \xrightarrow{\tau}_{\mathsf{c}} \nu d.(P\{^t/_x\} \mid Q\{^d/_y\})$ and no τ-actions in the other two semantics were available. Instead of explicitly adding eavesdrop actions, we can apply the rules Eav-Och and Eav-T and obtain that

$$A \xrightarrow{\nu d.eav(c,d)}_{\mathsf{e}} \mathsf{in}^{\mathsf{ho}}(d,x).P \mid \mathsf{out}^{\mathsf{ho}}(d,t).Q\{^d/_y\}) \xrightarrow{\nu z.eav(d,z)}_{\mathsf{e}} P\{^t/_x\} \mid Q\{^d/_y\} \mid \{^t/_z\}$$

We can now define both reachability and different equivalence properties in terms of these labelled semantics and relate them to the internal reduction. To define reachability properties in the labelled semantics, we define $A \downarrow\downarrow_c^s$ to hold when $A \overset{\mathsf{tr}}{\Rightarrow} A'$, $\mathsf{tr} = \mathsf{tr}_1 out(c,t) \mathsf{tr}_2$ and tr_1 does not bind c for some $\mathsf{tr}, \mathsf{tr}_1, \mathsf{tr}_2 \in (\mathcal{A} \setminus \{\tau\})^*$, term t and extended process A'.

The following proposition states that any reachability property modelled in terms of $A \Downarrow_c^{s,\theta}$ and universal quantification over processes, can also be expressed using $A \downarrow\downarrow_c^s$ without the need to quantify over processes.

Proposition 2. *For all closed honest plain processes A, for all $s \in \{\mathsf{c}, \mathsf{e}, \mathsf{p}\}$, $A \downarrow\downarrow_c^s$ iff there exists an attacker plain process I^s such that $I^s \mid A \Downarrow_c^{s,\mathsf{ho}}$.*

Next, we define equivalence relations using our labelled semantics that may serve as proof techniques for the may testing relation. First we need to define an indistinguishability relation on frames, called static equivalence.

Definition 3 (Static equivalence $\sim$). *Two terms u and v are* equal in the frame ϕ, *written $(u =_{\mathsf{E}} v)\phi$, if there exists $\tilde{n}$ and a substitution σ such that $\phi \equiv \nu\tilde{n}.\sigma$, $\tilde{n} \cap (fn(u) \cup fn(v)) = \emptyset$, and $u\sigma =_{\mathsf{E}} v\sigma$.*

Two closed frames ϕ_1 and ϕ_2 are statically equivalent*, written $\phi_1 \sim \phi_2$, when:*

- *$dom(\phi_1) = dom(\phi_2)$, and*
- *for all terms u, v we have that: $(u =_{\mathsf{E}} v)\phi_1$ if and only if $(u =_{\mathsf{E}} v)\phi_2$.*

Example 4. Consider the equational theory generated by the equation $\mathsf{dec}(\mathsf{enc}(x, y), y) = x$. Then we have that

$$\begin{array}{c} \nu k.\ \{^{\mathsf{enc}(a,k)}/_{x_1}\} \sim \nu k.\ \{^{\mathsf{enc}(b,k)}/_{x_1}\} \\ \nu k.\ \{^{\mathsf{enc}(a,k)}/_{x_1}, {}^{k}/_{x_2}\} \not\sim \nu k.\ \{^{\mathsf{enc}(b,k)}/_{x_1}, {}^{k}/_{x_2}\} \\ \nu k, a.\ \{^{\mathsf{enc}(a,k)}/_{x_1}, {}^{k}/_{x_2}\} \sim \nu k, b.\ \{^{\mathsf{enc}(b,k)}/_{x_1}, {}^{k}/_{x_2}\} \end{array}$$

Intutively, the first equivalence confirms that encryption hides the plaintext when the decryption key is unknown. The second equivalence does not hold as the test $(\mathsf{dec}(x_1, x_2) =_{\mathsf{E}} a)$ holds on the left hand side, but not on the right hand side. Finally, the third equivalence again holds as two restricted names are indistinguishable.

Now we are ready to define two classical equivalences on processes, based on the labelled semantics: trace equivalence and labelled bisimulation.

Definition 4 (Trace equivalences $\approx_t^{\mathsf{c}}$, $\approx_t^{\mathsf{p}}$, $\approx_t^{\mathsf{e}}$). *Let $s \in \{\mathsf{c}, \mathsf{p}, \mathsf{e}\}$. Let A and B be two closed honest extended processes. We say that $A \sqsubseteq_t^s B$ if for all $A \stackrel{\mathsf{tr}}{\Rightarrow}_s A'$ such that $bn(\mathsf{tr}) \cap fn(B) = \emptyset$, there exists B' such that $B \stackrel{\mathsf{tr}}{\Rightarrow}_s B'$ and $\phi(A') \sim \phi(B')$. We say that $A \approx_t^s B$ when $A \sqsubseteq_t^s B$ and $B \sqsubseteq_t^s A$.*

Definition 5 (Labeled bisimulations $\approx_\ell^{\mathsf{c}}$, $\approx_\ell^{\mathsf{p}}$, $\approx_\ell^{\mathsf{e}}$). *Let $s \in \{\mathsf{c}, \mathsf{p}, \mathsf{e}\}$. Let A and B two closed honest extended processes such that $dom(A) = dom(B)$. We say that $A \approx_\ell^s B$ if $\approx_\ell^s$ is the largest equivalence relation such that:*

- *$\phi(A) \sim \phi(B)$*
- *$A \stackrel{\tau}{\rightarrow}_s A'$ implies $B \stackrel{\epsilon}{\Rightarrow}_s B'$ and $A' \approx_\ell^s B'$ for some B',*
- *$A \stackrel{\ell}{\rightarrow}_s A'$ and $bn(\ell) \cap fn(B) = \emptyset$ implies $B \stackrel{\ell}{\Rightarrow}_s B'$ and $A' \approx_\ell^s B'$ for some B'.*

We again have, as usual that labelled bisimulation implies trace equivalence.

Proposition 3. *$\approx_\ell^s \subsetneq \approx_t^s$ for $s \in \{\mathsf{c}, \mathsf{e}, \mathsf{p}\}$.*

In [1] it is shown that $\approx_o^{\mathsf{c}} = \approx_\ell^{\mathsf{c}}$. We conjecture that for the new semantics p and e this same equivalence holds as well. Re-showing these results is beyond the scope of this paper, and we will mainly focus on testing/trace equivalence. As shown in [12], for the classical semantics trace equivalence implies may testing, while the converse does not hold in general. The two relations do however coincide on image-finite processes.

Definition 6. *Let A be a closed extended process. A is* image-finite *for the semantics $s \in \{\mathsf{c}, \mathsf{e}, \mathsf{p}\}$ if for each trace tr the set of equivalence classes $\{\phi(B) \mid A \stackrel{\mathsf{tr}}{\Rightarrow}_s B\}/\sim$ is finite.*

Note that any replication-free process is necessarily image-finite as there are only a finite number of possible traces for any given sequence of labels tr. The same relations among trace equivalence and may testing shown for the classical semantics hold also for the other semantics.

Theorem 1. *$\approx_t^s \subsetneq \approx_m^s$ and $\approx_t^s = \approx_m^s$ on image-finite processes for $s \in \{\mathsf{c}, \mathsf{e}, \mathsf{p}\}$.*

The proof of this result (for the classical semantics) is given in [12] and is easily adapted to the other semantics. To see that the implication is strict, we continue Example 2 on processes A and B defined in Figure 2. We already noted that $A \approx_m^s B$, but will now show that $A \not\approx_t^s B$ (for $s \in \{\mathsf{c}, \mathsf{e}, \mathsf{p}\}$). All possible traces of A are of the form $A \stackrel{\nu x.out(c,x)}{\Longrightarrow}_s A'$ where $\phi(A') = \{{}^{h^n(a)}/_x\}$ for $n \in \mathbb{N}$. We easily see that $A \not\approx_t^s B$ as for any n we have that $\{{}^{h^n(a)}/_x\} \not\sim \{{}^{s}/_x\}$, by testing $x = h^n(a)$. On the other hand, given an image-finite process, we can only have a finite number of different frames for a given trace, and therefore we can bound the context size that is necessary for distinguishing the processes.

3 Comparing the different semantics

In this section we state our results on comparing these semantics. We first show that, as expected, all the semantics coincide for reachability properties.

Theorem 2. *For all ground, closed honest extended processes A, for all channels d, we have that $A \downarrow\!\downarrow_d^{\mathsf{p}}$ iff $A \downarrow\!\downarrow_d^{\mathsf{c}}$ iff $A \downarrow\!\downarrow_d^{\mathsf{e}}$.*

The next result is, in our opinion, more surprising. As the private semantics force the adversary to observe all information, one might expect that his distinguishing power increases over the classical one. This intuition is however wrong: the classical and private trace equivalences, testing equivalence and labelled bisimulations appear to be incomparable.

Theorem 3. *$\approx_r^{\mathsf{p}} \not\subseteq \approx_r^{\mathsf{c}}$ and $\approx_r^{\mathsf{c}} \not\subseteq \approx_r^{\mathsf{p}}$ for $r \in \{\ell, t, m\}$.*

Proof. We first show that there exist A and B such that $A \approx_\ell^{\mathsf{p}} B$, but $A \not\approx_m^{\mathsf{c}} B$. Note that, as $\approx_\ell^s \subset \approx_t^s \subseteq \approx_m^s$ for $s \in \{\mathsf{c}, \mathsf{p}\}$ these processes demonstrate both that $\approx_\ell^{\mathsf{p}} \not\subseteq \approx_\ell^{\mathsf{c}}$, $\approx_t^{\mathsf{p}} \not\subseteq \approx_t^{\mathsf{c}}$ and $\approx_m^{\mathsf{p}} \not\subseteq \approx_m^{\mathsf{c}}$.

Consider processes A and B defined in Figure 4. In short, the result follows from the fact that if A performs an internal communication on channel c followed by an output on d (from P_1), B has no choice other then performing the output on d in P_2. In the private semantics, however, the internal communication will be split in an output followed by an input: after the output on c, the input $\mathsf{in}^{\mathsf{ho}}(c, x).P_2(x)$ following the output becomes available. More precisely, to see that $A \approx_\ell^{\mathsf{p}} B$ we first observe that if

$$A \mathrel{\hat{=}} \nu s_1.\nu s_2.((\mathsf{out}^{\mathsf{ho}}(c,s_1).\mathsf{in}^{\mathsf{ho}}(c,x).P_1(x)) \mid (\mathsf{in}^{\mathsf{ho}}(c,y).P_2(y)))$$
$$B \mathrel{\hat{=}} \nu s_1.\nu s_2.((\mathsf{out}^{\mathsf{ho}}(c,s_1).\mathsf{in}^{\mathsf{ho}}(c,x).P_2(x)) \mid (\mathsf{in}^{\mathsf{ho}}(c,y).P_1(y)))$$

where

$$P_1(x) \mathrel{\hat{=}} (\mathsf{if}\ x = s_1\ \mathsf{then}\ \mathsf{out}^{\mathsf{ho}}(d,s_2)) \mid (\mathsf{if}\ x = s_2\ \mathsf{then}\ \mathsf{out}^{\mathsf{ho}}(e,x))$$
$$P_2(x) \mathrel{\hat{=}} (\mathsf{if}\ x = s_1\ \mathsf{then}\ \mathsf{out}^{\mathsf{ho}}(d,s_2))$$

To emit on channel e, processes A and B must execute $P_2(s_1)$ followed by $P_1(s_2)$. In the classical semantics, a trace of A emitting on e through an internal communication between $\mathsf{out}^{\mathsf{ho}}(c,s_1)$ and $\mathsf{in}^{\mathsf{ho}}(c,y)$ forces B to execute $P_1(s_1)$ thus preventing it to emit on e.

Fig. 4. Processes A and B such that $A \approx^{\mathsf{p}}_{\ell} B$ and $A \not\approx^{\mathsf{c}}_{m} B$.

$A \xrightarrow{\nu z.out(c,z)}_{\mathsf{p}} A'$ then $B \xrightarrow{\nu z.out(c,z)}_{\mathsf{p}} B'$ and $A' \equiv B'$, and vice-versa. If $A \xrightarrow{in(c,t)}_{\mathsf{p}} A'$ then $B \xrightarrow{in(c,t)}_{\mathsf{p}} B'$. As $t \notin \{s_1, s_2\}$ we have that $P_1(t) \approx^{\mathsf{p}}_{\ell} 0 \approx^{\mathsf{p}}_{\ell} P_2(t)$. Finally, if $t \neq s_2$ we also have that $P_1(t) \approx^{\mathsf{p}}_{\ell} P_2(t)$ as in particular $P_1(s_1) \approx^{\mathsf{p}}_{\ell} P_2(s_1)$. Therefore,

$$\nu s_1.\nu s_2.(\mathsf{out}^{\mathsf{ho}}(c,s_1).\mathsf{in}^{\mathsf{ho}}(c,x).P_1(x)) \;\approx^{\mathsf{p}}_{\ell}\; \nu s_1.\nu s_2.(\mathsf{out}^{\mathsf{ho}}(c,s_1).\mathsf{in}^{\mathsf{ho}}(c,x).P_2(x))$$

which allows us to conclude.

As A and B are image-finite, we have that $A \approx^{\mathsf{c}}_{m} B$ if and only if $A \approx^{\mathsf{c}}_{t} B$. To see that $A \not\approx^{\mathsf{c}}_{t} B$ we observe that A may perform the following transition sequence, starting with an internal communication on a public channel:

$$\begin{aligned} A &\xrightarrow{\tau}_{\mathsf{c}} \nu s_1.\nu s_2.((\mathsf{in}^{\mathsf{ho}}(c,x).P_1(x)) \mid (P_2(s_1))) \\ &\xrightarrow{\nu z.out(d,z)}_{\mathsf{c}} \nu s_1.\nu s_2.((\mathsf{in}^{\mathsf{ho}}(c,x).P_1(x)) \mid \{^{s_2}/_{z}\}) \\ &\xrightarrow{in(c,z)}_{\mathsf{c}} \nu s_1.\nu s_2.(P_1(s_2) \mid \{^{s_2}/_{z}\}) \end{aligned}$$

In order to mimic the behaviour of A, B must perform the same sequence of observable transitions:

$$B \xrightarrow{\nu z.out(d,z)\ in(c,z)}_{\mathsf{c}} \nu s_1.\nu s_2.(P_2(s_2) \mid \{^{s_2}/_{z}\})$$

We conclude as $\nu s_1.\nu s_2.(P_1(s_2) \mid \{^{s_2}/_{z}\}) \xrightarrow{\nu z'.out(e,z')} \nu s_1.\nu s_2.(\{^{s_2}/_{z}\} \mid \{^{s_2}/_{z'}\})$, but $\nu s_1.\nu s_2.(P_2(s_2) \mid \{^{s_2}/_{z}\}) \not\xrightarrow{\nu z'.out(e,z')}$. This trace inequivalence has also been shown using APTE.

To show that $\approx^{\mathsf{c}}_{r} \not\subseteq \approx^{\mathsf{p}}_{r}$ for $r \in \{\ell, t, m\}$ we show that there exist processes A and B such that $A \approx^{\mathsf{c}}_{\ell} B$ and $A \not\approx^{\mathsf{p}}_{m} B$. As in the first part of the proof, note that, as $\approx^{s}_{\ell} \subset \approx^{s}_{t} \subseteq \approx^{s}_{m}$ for $s \in \{\mathsf{c}, \mathsf{p}\}$ these processes demonstrate that $\approx^{\mathsf{c}}_{\ell} \not\subseteq \approx^{\mathsf{p}}_{\ell}$, $\approx^{\mathsf{c}}_{t} \not\subseteq \approx^{\mathsf{p}}_{t}$ and $\approx^{\mathsf{c}}_{m} \not\subseteq \approx^{\mathsf{p}}_{m}$.

Consider the processes A and B defined in Figure 5. The proof crucially relies on the fact that B may perform an internal communication in the classical semantics to mimic A, which becomes visible in the attacker in the private semantics. To see that $A \approx^{\mathsf{c}}_{\ell} B$ we first observe that the only first possible action from A or B is an input. In particular, given a term t, there is a unique B' such that $B \xrightarrow{in(c,t)} B'$ where $B' = \nu s.(\mathsf{out}^{\mathsf{ho}}(c,s).\mathsf{out}^{\mathsf{ho}}(d,a) \mid \mathsf{in}^{\mathsf{ho}}(c,y).P(y))$. However, if $A \xrightarrow{in(c,t)} A'$ then

$$A \mathrel{\hat{=}} \nu s.(\mathsf{in}^{\mathsf{ho}}(c,x).\mathsf{out}^{\mathsf{ho}}(c,s).\mathsf{out}^{\mathsf{ho}}(d,a) \mid \mathsf{in}^{\mathsf{ho}}(c,y).P(y))$$
$$B \mathrel{\hat{=}} \nu s.(\mathsf{in}^{\mathsf{ho}}(c,x).(\mathsf{out}^{\mathsf{ho}}(c,s).\mathsf{out}^{\mathsf{ho}}(d,a) \mid \mathsf{in}^{\mathsf{ho}}(c,y).P(y)))$$

where

$$P(y) \mathrel{\hat{=}} \mathsf{if}\ y = s\ \mathsf{then}\ \mathsf{in}^{\mathsf{ho}}(c,z).\mathsf{out}^{\mathsf{ho}}(c,s).\mathsf{out}^{\mathsf{ho}}(d,a)\ \mathsf{else}\ \mathsf{out}^{\mathsf{ho}}(d,a)$$

In the private semantics, a trace of A starting with the execution of $\mathsf{in}^{\mathsf{ho}}(c,y)$ can only be matched on B by executing $\mathsf{in}^{\mathsf{ho}}(c,x)$. B could then emit on channel c, which is not the case for A, hence yielding non equivalence. In the classic semantics, an internal communication between $\mathsf{out}^{\mathsf{ho}}(c,s)$ and $\mathsf{in}^{\mathsf{ho}}(c,y)$ allows to *hide* the fact that B can emit on c.

Fig. 5. Processes A and B such that $A \approx_\ell^{\mathsf{c}} B$ and $A \not\approx_m^{\mathsf{p}} B$.

either $A' = B'$ or $A' = A''$ with $A'' \mathrel{\hat{=}} \nu s.(\mathsf{in}^{\mathsf{ho}}(c,x).\mathsf{out}^{\mathsf{ho}}(c,s).\mathsf{out}^{\mathsf{ho}}(d,a) \mid P(t))$. Therefore, to complete the proof, we only need to find B'' such that $B \stackrel{in(c,t)}{\Longrightarrow} B''$ and $A'' \approx_\ell^{\mathsf{c}} B''$. Such process can be obtain by applying an internal communication on B', *i.e.* $B \xrightarrow{in(c,t)}_{\mathsf{c}} B' \xrightarrow{\tau} \nu s.(\mathsf{out}^{\mathsf{ho}}(d,a) \mid P(s))$. Note that $t \neq s$ since s is bound, meaning that $P(t) \approx_\ell^{\mathsf{c}} \mathsf{out}^{\mathsf{ho}}(d,a)$. Moreover, $P(s) \approx_\ell^{\mathsf{c}} \mathsf{in}^{\mathsf{ho}}(c,x).\mathsf{out}^{\mathsf{ho}}(c,s).\mathsf{out}^{\mathsf{ho}}(d,a)$. This allows us to conlude that $\nu s.(\mathsf{out}^{\mathsf{ho}}(d,a) \mid P(s)) \approx_\ell^{\mathsf{c}} A''$.

Again, as A and B are image-finite may and trace equivalence coincide. To see that $A \not\approx_t^{\mathsf{p}} B$ we first observe that A may perform the following transition sequence:

$$A \xrightarrow{in(c,t)}_{\mathsf{p}} A'' \xrightarrow{\tau}_{\mathsf{p}} \nu s.(\mathsf{in}^{\mathsf{ho}}(c,x).\mathsf{out}^{\mathsf{ho}}(c,s).\mathsf{out}^{\mathsf{ho}}(d,a) \mid \mathsf{out}^{\mathsf{ho}}(d,a))$$
$$\xrightarrow{\nu z.out(d,z)}_{\mathsf{p}} \nu s.(\mathsf{in}^{\mathsf{ho}}(c,x).\mathsf{out}^{\mathsf{ho}}(c,s).\mathsf{out}^{\mathsf{ho}}(d,a) \mid \{^{a}/_{z}\})$$

We conclude as $B \xrightarrow{in(c,t)}_{\mathsf{p}} B'$ but $B' \not\xrightarrow{\nu z.out(d,z)}_{\mathsf{p}}$. This trace disequivalence has also been shown using APTE. □

One may also note that the counter-example witnessing that equivalences in the private semantics do not imply equivalences in the classical semantics is *minimal*: it does not use function symbols, equational reasoning, private channels, replication nor else branches. The second part of the proof relies on the use of else branches. We can however refine this result in the case of labeled bisimulation to processes without else branches, the counter-example being the same processes A and B described in the proof but where we replace each $\mathsf{out}^{\mathsf{ho}}(d,a)$ by 0. In the case of trace equivalence, we can also produce a counter-example without else branches witnessing that trace equivalences in the classical semantics do no imply trace equivalences in the private semantics but provided that we rely on a function symbol h. In the appendix of the technical report [7],we describe in more details these processes and give the proofs of them being counter-examples.

Next, we show that the eavesdropping semantics yields strictly stronger bisimulations and trace equivalences: the eavesdropping semantics is actually strictly included in the intersection of the classic and private semantics.

Theorem 4. $\approx_\ell^{\mathsf{e}} \subsetneq \approx_\ell^{\mathsf{p}} \cap \approx_\ell^{\mathsf{c}}$.

Proof (Sketch).

$$A \hat{=}\ \nu s_1.\nu s_2.((\mathsf{out}^{\mathsf{ho}}(c,s_1).\mathsf{in}^{\mathsf{ho}}(c,x).P_1(x)) \mid (\mathsf{in}^{\mathsf{ho}}(c,y).P_2(y)))$$
$$B \hat{=}\ \nu s_1.\nu s_2.((\mathsf{out}^{\mathsf{ho}}(c,s_1).\mathsf{in}^{\mathsf{ho}}(c,x).P_2(x)) \mid (\mathsf{in}^{\mathsf{ho}}(c,y).P_1(y)))$$

where

$$P_1(x) \hat{=}\ (\mathsf{if}\ x = s_1\ \mathsf{then}\ \mathsf{in}^{\mathsf{ho}}(d,z).\mathsf{if}\ z = s_1\ \mathsf{then}\ \mathsf{out}^{\mathsf{ho}}(d,s_2)) \mid (\mathsf{if}\ x = s_2\ \mathsf{then}\ \mathsf{out}^{\mathsf{ho}}(e,x))$$
$$P_2(x) \hat{=}\ (\mathsf{if}\ x = s_1\ \mathsf{then}\ \mathsf{in}^{\mathsf{ho}}(d,z).\mathsf{if}\ z = s_1\ \mathsf{then}\ \mathsf{out}^{\mathsf{ho}}(d,s_2))$$

To emit on channel e, processes A and B must execute $P_2(s_1)$ by inputing twice s_1 followed by $P_1(s_2)$. In the classical semantics, an internal communication on A between $\mathsf{out}^{\mathsf{ho}}(c,s_1)$ and $\mathsf{in}^{\mathsf{ho}}(c,y)$ forces B to execute $P_1(s_1)$ but *hides* s_1, preventing a second input of s_1 by A. However, in the eavesdropping semantics, the internal communication *reveals* s_1 allowing A to emit on e but not B.

Fig. 6. Processes A and B such that $A \approx^{\mathsf{c}}_{\ell} B$, $A \approx^{\mathsf{p}}_{\ell} B$ but $A \not\approx^{\mathsf{e}}_{t} B$.

1. We first show that $\approx^{\mathsf{e}}_{\ell} \subseteq \approx^{\mathsf{p}}_{\ell}$. Suppose $A \approx^{\mathsf{e}}_{\ell} B$ and let $\mathcal{R}$ be the relation witnessing this equivalence. We will show that $\mathcal{R}$ is also a labelled bisimulation in the private semantics. Suppose $A\mathcal{R}B$.
   - as $A \approx^{\mathsf{e}}_{\ell} B$, we have that $\phi(A) \sim \phi(B)$.
   - if $A \xrightarrow{\tau}_{\mathsf{p}} A'$ then, as $\xrightarrow{\tau}_{\mathsf{p}} \subset \xrightarrow{\tau}_{\mathsf{e}}$, $A \xrightarrow{\tau}_{\mathsf{e}} A'$. As $A \approx^{\mathsf{e}}_{\ell} B$ there exists B' such that $B \stackrel{\epsilon}{\Rightarrow}_{\mathsf{e}} B'$ and $A'\mathcal{R}B'$. As B is a honest process no COMM-EAV transition is possible, and hence $B \stackrel{\epsilon}{\Rightarrow}_{\mathsf{p}} B'$.
   - if $A \xrightarrow{\ell}_{\mathsf{p}} A'$ and $bn(\ell) \cap fn(B) = \emptyset$ then we also have that $A \xrightarrow{\ell}_{\mathsf{e}} A'$ (as $\xrightarrow{\ell}_{\mathsf{p}} \subset \xrightarrow{\ell}_{\mathsf{e}}$ and there exists B' such that $B \stackrel{\ell}{\Rightarrow}_{\mathsf{e}} B'$ and $A'\mathcal{R}B'$. As no COMM-EAV are possible and ℓ is not of the form $eav(c,d)$ nor $\nu y.eav(c,y)$ we have that $B \stackrel{\ell}{\Rightarrow}_{\mathsf{p}} B'$.
2. We next show that $A \approx^{\mathsf{e}}_{\ell} B$ implies $A \approx^{\mathsf{c}}_{\ell} B$ for any A, B. We will show that $\approx^{\mathsf{e}}_{\ell}$ is also a labelled bisimulation in the classical semantics. The proof relies on similar arguments as in Item 2 of the proof of Theorem 5 and the facts that
   - $\nu\tilde{n}.(A' \mid \{^t/_x\}) \approx^{\mathsf{e}}_{\ell} \nu\tilde{n}.(B' \mid \{^u/_x\})$ implies $\nu\tilde{n}.A' \approx^{\mathsf{e}}_{\ell} \nu\tilde{n}.B'$,
   - $A' \approx^{\mathsf{e}}_{\ell} B'$ implies $\nu c.A' \approx^{\mathsf{e}}_{\ell} \nu c.B'$

   The first property is needed when an internal communication of a term or public channel is replaced by an eavesdrop action and an input. The second property handles the case when we replace the internal communication of a private channel by an application of the EAV-OCH rule and an input.
3. We now show that the implication $\approx^{\mathsf{e}}_{\ell} \subsetneq \approx^{\mathsf{c}}_{\ell} \cap \approx^{\mathsf{c}}_{t}$ is strict, i.e., there exist A and B such that $A \approx^{\mathsf{c}}_{\ell} B$, $A \approx^{\mathsf{p}}_{\ell} B$ but $A \not\approx^{\mathsf{e}}_{t} B$ (which implies $A \not\approx^{\mathsf{e}}_{\ell} B$).
   Consider the processes A and B defined in Figure 6. This example is a variant of the one given in Figure 4. The difference is the addition of "$\mathsf{in}^{\mathsf{ho}}(d,z).\mathsf{if}\ z = s_1\ \mathsf{then}$ " in processes $P_1(x)$ and $P_2(x)$: this additional check is used to verify whether the adversary learned s_1 or not. The proofs that $A \approx^{\mathsf{c}}_{\ell} B$ and $A \approx^{\mathsf{p}}_{\ell} B$ follow the same lines as in Theorem 3. We just additionally observe that $\nu s_1.(\mathsf{in}^{\mathsf{ho}}(d,z).\mathsf{if}\ z = s_1\ \mathsf{then}\ \mathsf{out}^{\mathsf{ho}}(d,s_2)) \approx^{s}_{\ell} \nu s_1.\ (\mathsf{in}^{\mathsf{ho}}(d,z).0)$ for $s \in \{\mathsf{c},\mathsf{p}\}$.
   The trace witnessing that $A \not\approx^{\mathsf{e}}_{t} B$ (which implies $A \not\approx^{\mathsf{e}}_{\ell} B$) is again similar to the one in Theorem 3, but starting with an eavesdrop transition which allows the

attacker to learn s_1, which in turn allows him to learn s_2 and distinguish $P_1(s_2)$ from $P_2(s_2)$. We have verified $A \not\approx_t^{\mathsf{e}} B$ using APTE which implies $A \not\approx_\ell^{\mathsf{e}} B$. □

Again we note that the implications are strict, even for processes containing only public channels.

Theorem 5. $\approx_t^{\mathsf{e}} \subsetneq \approx_t^{\mathsf{p}} \cap \approx_t^{\mathsf{c}}$.

Proof (Sketch).

1. We first prove that $\approx_t^{\mathsf{e}} \subseteq \approx_t^{\mathsf{p}}$. Suppose that $A \approx_t^{\mathsf{e}} B$. We need to show that for any A' such that $A \stackrel{tr}{\Longrightarrow}_{\mathsf{p}} A'$ there exists B' such that $B \stackrel{tr}{\Longrightarrow}_{\mathsf{p}} B'$. It follows from the definition of the semantics that whenever $A \stackrel{\mathsf{tr}}{\Longrightarrow}_{\mathsf{p}} A'$ then we also have $A \stackrel{\mathsf{tr}}{\Longrightarrow}_{\mathsf{e}} A'$ as $\stackrel{\ell}{\rightarrow}_{\mathsf{p}} \subset \stackrel{\ell}{\rightarrow}_{\mathsf{e}}$. As $A \approx_t^{\mathsf{e}} B$, we have that there exists B', such that $B \stackrel{\mathsf{tr}}{\Longrightarrow}_{\mathsf{e}} B'$ and $\phi(A') \sim \phi(B')$. As tr does not contain labels of the form $eav(c,d)$ nor $\nu y.eav(c,y)$ and as no COMM-EAV are possible (A and B are honest processes) we also have that $B \stackrel{\mathsf{tr}}{\Longrightarrow}_{\mathsf{p}} B'$. Hence $A \approx_t^{\mathsf{p}} B$.
2. We next prove that $\approx_t^{\mathsf{e}} \subseteq \approx_t^{\mathsf{c}}$. Similar to Item 1 we suppose that $A \approx_t^{\mathsf{e}} B$ and $A \stackrel{tr_{\mathsf{c}}}{\Longrightarrow}_{\mathsf{c}} A'_{\mathsf{c}}$. From the semantics, we obtain that $A \stackrel{tr_{\mathsf{e}}}{\Longrightarrow}_{\mathsf{e}} A'_{\mathsf{e}}$, where
   - $\phi(A'_{\mathsf{c}}) \subseteq \phi(A'_{\mathsf{e}})$, i.e., $dom(\phi(A'_{\mathsf{c}})) \subseteq dom(\phi(A'_{\mathsf{e}}))$ and the frames coincide on the common domain.
   - tr_{e} is constructed from tr by replacing any τ action resulting from the COMM rule by an application of an eavesdrop rule (EAV-T, EAV-CH, or EAV-OCH).

   The proof is done by induction on the length of tr and the proof tree of each transition. As $A \approx_t^{\mathsf{e}} B$ we also have that $B \stackrel{tr_{\mathsf{e}}}{\Longrightarrow}_{\mathsf{e}} B'_{\mathsf{e}}$ and $A'_{\mathsf{e}} \sim B'_{\mathsf{e}}$. We show by the definition of the semantics that $B \stackrel{tr_{\mathsf{c}}}{\Longrightarrow}_{\mathsf{c}} B'_{\mathsf{c}}$ and $\phi(B'_{\mathsf{c}}) \subseteq \phi(B'_{\mathsf{e}})$ (replacing each eavesdrop action by an internal communication). Due to the inclusions of the frames and $A'_{\mathsf{e}} \sim B'_{\mathsf{e}}$ we also have that $A'_{\mathsf{c}} \sim B'_{\mathsf{c}}$.
3. To show that the implication $\approx_t^{\mathsf{e}} \subsetneq \approx_t^{\mathsf{p}} \cap \approx_t^{\mathsf{c}}$ is strict, i.e., there exist processes A and B such that $A \approx_t^{\mathsf{c}} B$, $A \approx_t^{\mathsf{p}} B$ but $A \not\approx_t^{\mathsf{e}} B$. The processes defined in Figure 6 witness this fact (cf the discussion of these processes in the proof of Theorem 4). These trace (in)equivalences have also been verified using APTE.

We note from the processes defined in Figure 6 that the implications are strict even for processes that do not communicate on private channels, do not use replication, nor else branches and terms are simply names (no function symbols nor equational theories).

Theorem 6. $\approx_m^{\mathsf{e}} \subsetneq \approx_m^{\mathsf{p}} \cap \approx_m^{\mathsf{c}}$.

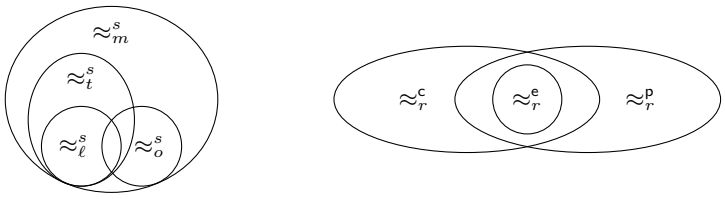
Proof (Sketch).

1. We first prove that $\approx_m^{\mathsf{e}} \subseteq \approx_m^{\mathsf{p}}$. Suppose that $A \approx_m^{\mathsf{e}} B$. Suppose that $A \approx_m^{\mathsf{e}} B$. We need to show that for all channel c, for all $C[\_]$ attacker evaluation contexts p-closing for A and B, $C[A] \Downarrow_c^{\mathsf{p}}$ is equivalent to $C[B] \Downarrow_c^{\mathsf{p}}$. It follows from the definition of the private semantics that any process $\mathsf{eav}(c,x).P$ in $C[\_]$ has the same behaviour as the process 0. Hence, we generate a context $C^1[\_]$ by replacing in $C[\_]$

any instance of $\mathsf{eav}(c,x).P$ by 0, and thus obtaining $C[A] \Downarrow_c^{\mathsf{p}} \Leftrightarrow C'[A] \Downarrow_c^{\mathsf{p}}$ and $C[B] \Downarrow_c^{\mathsf{p}} \Leftrightarrow C'[B] \Downarrow_c^{\mathsf{p}}$. Notice that the definition of semantics gives us $\rightarrow_{\mathsf{p}} \subseteq \rightarrow_{\mathsf{e}}$. Hence, $C'[A] \Downarrow_c^{\mathsf{p}}$ implies $C'[A] \Downarrow_c^{\mathsf{e}}$ and $C'[B] \Downarrow_c^{\mathsf{p}}$ implies $C'[B] \Downarrow_c^{\mathsf{e}}$. Furthermore, since we built $C'[\_]$ to not contain any process of the form $\mathsf{eav}(c,x).P$, we deduce that rules C-EAV and C-OEAV can never be applied in a derivation of $C'[A]$ or $C'[B]$. It implies that $C'[A] \Downarrow_c^{\mathsf{p}} \Leftrightarrow C'[A] \Downarrow_c^{\mathsf{e}}$ and $C'[B] \Downarrow_c^{\mathsf{p}} \Leftrightarrow C'[B] \Downarrow_c^{\mathsf{e}}$. Thanks to $A \approx_m^{\mathsf{e}} B$, we know that $C'[A] \Downarrow_c^{\mathsf{e}} \Leftrightarrow C'[B] \Downarrow_c^{\mathsf{e}}$ and so we conclude that $C[A] \Downarrow_c^{\mathsf{p}} \Leftrightarrow C[B] \Downarrow_c^{\mathsf{p}}$.

2. We next prove that $\approx_m^{\mathsf{e}} \subseteq \approx_m^{\mathsf{c}}$. Similarly to Item 1, we consider a channel c and an attacker evaluation context $C[\_]$ that is c-closing for A and B. The main difficulty of this proof is to match the application of the rule COMM in the classical semantics with the rules C-EAV and C-OEAC. However, $C[\_]$ does not necessarily contain eavesdrop process $\mathsf{eav}(d,x) \mid \omega c$. Moreover, as mentioned in Item 1, a process $\mathsf{eav}(d,x).P$ has the same behavior as 0 in the classical semantics but can have a completely different behaviour in the eavesdropping semantics if P is not 0. Thus, we remove from $C[\_]$ the eavesdrop processes, obtaining $C'[\_]$. Then, we define a new context $C''[\_]$ based on $C'[\_]$ where will add harmless eavesdrop process $\mathsf{eav}(d,y).0$. We first add in parallel the processes $!\mathsf{eav}(a,y) \mid \omega a$ for all free channels a in $C'[\_], A$ and B. Moreover, since private channels can be opened, we also replace any process $\nu d.P$, $\mathsf{in}^{\mathsf{at}}(c,x).P$ where d,x are of channel type with $\nu d.(P \mid !\mathsf{eav}(d,y))$ and $\mathsf{in}^{\mathsf{at}}(c,x).(P \mid !\mathsf{eav}(x,y))$. By induction of the derivations, we can show that $C[A] \Downarrow_c^{\mathsf{c}} \Leftrightarrow C''[A] \Downarrow_c^{\mathsf{e}}$ and $C[B] \Downarrow_c^{\mathsf{c}} \Leftrightarrow C''[B] \Downarrow_c^{\mathsf{e}}$. Since $A \approx_m^{\mathsf{e}} B$, we deduce that $C''[A] \Downarrow_c^{\mathsf{e}} \Leftrightarrow C''[B] \Downarrow_c^{\mathsf{e}}$ and so $C[A] \Downarrow_c^{\mathsf{c}} \Leftrightarrow C[B] \Downarrow_c^{\mathsf{c}}$.

3. To show that the implication $\approx_m^{\mathsf{e}} \subsetneq \approx_m^{\mathsf{p}} \cap \approx_m^{\mathsf{c}}$ is strict, i.e., there exist processes A and B such that $A \approx_m^{\mathsf{c}} B$, $A \approx_m^{\mathsf{p}} B$ but $A \not\approx_m^{\mathsf{e}} B$. The processes defined in Figure 6 witness this fact. They already were witness of the strict inclusion $\approx_t^{\mathsf{e}} \subsetneq \approx_t^{\mathsf{p}} \cap \approx_t^{\mathsf{c}}$ (see proof of Theorem 5) and since A and B are image finit, we know from Theorem 1 that may and trace equivalences between A and B coincide. □



for all $s \in \{\mathsf{c}, \mathsf{p}, \mathsf{e}\}$
for image finite processes $\approx_t^s = \approx_m^s$
if $s = \mathsf{c}$ then $\approx_\ell^s = \approx_o^s$ (conjectured for $s \in \{\mathsf{p}, \mathsf{e}\}$)

for all $r \in \{m, t, \ell\}$

Fig. 7. Overview of the results.

4 Subclasses of processes for which the semantics coincide

4.1 Simple processes

The class of simple processes was defined in [12]. It was shown that for these processes observational and may testing equivalences coincide. Intuitively, these processes are composed of parallel basic processes. Each basic process is a sequence of input, test on the input and output actions. Moreover, importantly, each basic process has a distinct channel for communication.

Definition 7 (basic process). *The set $\mathcal{B}(c, \mathcal{V})$ of* basic processes *built on $c \in \mathcal{C}h$ and $\mathcal{V} \subseteq \mathcal{X}$ (variables of base type) is the least set of processes that contains* 0 *and such that*

- *if $B_1, B_2 \in \mathcal{B}(c, \mathcal{V})$, $M, N \in \mathcal{T}(\mathcal{F}, \mathcal{N}, \mathcal{V})$, then* $\mathsf{if}\ M = N\ \mathsf{then}\ B_1\ \mathsf{else}\ B_2\ \in\ \mathcal{B}(c, \mathcal{V})$.
- *if $B \in \mathcal{B}(c, \mathcal{V})$, $u \in \mathcal{T}(\mathcal{F}, \mathcal{N}, \mathcal{V})$, then* $\mathsf{out}^{\mathsf{ho}}(c, u).B\ \in\ \mathcal{B}(c, \mathcal{V})$.
- *if $B \in \mathcal{B}(c, \mathcal{V} \uplus \{x\})$, x of base type ($x \notin \mathcal{V}$), then* $\mathsf{in}^{\mathsf{ho}}(c, x).B\ \in\ \mathcal{B}(c, \mathcal{V})$.

Definition 8 (simple process). *A* simple process *is obtained by composing and replicating basic processes and frames, hiding some names:*

$$\nu\tilde{n}.\ (\ \nu\tilde{n}_1.(B_1 \mid \sigma_1)\ \mid\ !(\nu c'_1, \tilde{m}_1.\mathsf{out}^{\mathsf{ho}}(p_1, c'_1).B'_1) \\ \vdots \qquad\qquad\qquad \vdots \\ \nu\tilde{n}_k.(B_k \mid \sigma_k)\ \mid !(\nu c'_n, \tilde{m}_n.\mathsf{out}^{\mathsf{ho}}(p_n, c'_n).B'_n)\ )$$

where $B_j \in \mathcal{B}(c_j, \emptyset)$, $B'_j \in \mathcal{B}(c'_j, \emptyset)$ and c_j are channel names that are pairwise distinct. The names $p_1, \ldots, p_n$ are distinct channel names that do not appear elsewhere and $\sigma_1, \ldots, \sigma_k$ are frames without restricted names (i.e. substitutions).

We have that for simple processes, all equivalences and semantics coincide.

Theorem 7. *When restricted to simple processes, we have that $\approx_{r_1}^{s_1} = \approx_{r_2}^{s_2}$ for $r_1, r_2 \in \{\ell, o, m, t\}$ and $s_1, s_2 \in \{\mathsf{c}, \mathsf{p}, \mathsf{e}\}$.*

Proof. The result when $s_1 = s_2 = \mathsf{c}$ was shown in [12]. As for simple processes, all parallel processes have distinct channels, the internal communication rule may never be triggered, and therefore it is easy to show that the three semantics coincide.

4.2 I/O-unambiguous processes

Restricting processes to simple processes is often too restrictive. For instance, when verifying unlinkability and anonymity properties, two outputs by different parties should not be distinguishable due to the channel name. We therefore introduce another class of processes, that we call io-unambiguous for which we also show that the different semantics (although not the different equivalences) do coincide.

Intuitively, an io-unambiguous process forbids an output and input on the same public channel to follow each other directly (or possibly with only conditionals in between). For instance, we forbid processes of the form $\mathsf{out}^{\theta}(c, t).\mathsf{in}^{\theta}(c, x).P$, $\mathsf{out}^{\theta}(c, t).(\mathsf{in}^{\theta}(c, x).P \mid Q)$ as well as $\mathsf{out}^{\theta}(c, t).\mathsf{if}\ \ t_1 = t_2\ \ \mathsf{then}\ P\ \mathsf{else}\ \mathsf{in}^{\theta}(c, x).Q$. We however allow inputs and outputs on the same channel in parallel.

Definition 9. *We define an honest extended process A to be I/O-unambiguous when* $\mathsf{ioua}(A, \_) = \top$ *where*

$$\mathsf{ioua}(0, c) = \top \qquad \mathsf{ioua}(\{^u/_x\}, c) = \top \qquad \mathsf{ioua}(!P, c) = \mathsf{ioua}(P, c)$$
$$\mathsf{ioua}(A \mid B, c) = \mathsf{ioua}(A, c) \wedge \mathsf{ioua}(B, c) \qquad \mathsf{ioua}(\nu x.A, c) = \mathsf{ioua}(A, c)$$
$$\mathsf{ioua}(\nu n.A, c) = \begin{cases} \bot & \text{if } n \in \mathcal{C}h \\ \mathsf{ioua}(A, c) & \text{otherwise} \end{cases}$$
$$\mathsf{ioua}(\mathsf{if}\ u = v\ \mathsf{then}\ P\ \mathsf{else}\ Q, c) = \mathsf{ioua}(P, c) \wedge \mathsf{ioua}(Q, c)$$
$$\mathsf{ioua}(\mathsf{out}^\theta(d, u).P, c) = \begin{cases} \bot & \text{if } u \text{ is of channel type} \\ \mathsf{ioua}(P, d) & \text{otherwise} \end{cases}$$
$$\mathsf{ioua}(\mathsf{in}^\theta(d, x).P, c) = \begin{cases} \bot & \text{if } x \text{ is of channel type or } d = c \\ \mathsf{ioua}(P, \_) & \text{otherwise} \end{cases}$$

Note that an I/O-unambiguous process does not contain private channels and always input/output base-type terms. We also note that a simple way to enforce that processes are I/O-unambiguous is to use disjoint channel names for inputs and outputs (at least in the same parallel thread).

Theorem 8. *When restricted to I/O-unambiguous processes, we have that* $\approx_r^{\mathsf{p}} = \approx_r^{\mathsf{e}}$ *but* $\approx_r^{\mathsf{e}} \subsetneq \approx_r^{\mathsf{c}}$ *for* $r \in \{\ell, t\}$.

Proof. From Theorems 4 and 5, we already know that $\approx_r^{\mathsf{e}} \subseteq \approx_r^{\mathsf{p}}$ and $\approx_r^{\mathsf{e}} \subseteq \approx_r^{\mathsf{c}}$. Hence, we only need to show that $\approx_r^{\mathsf{p}} \subseteq \approx_r^{\mathsf{e}}$ and $\approx_r^{\mathsf{p}} \subsetneq \approx_r^{\mathsf{c}}$. The latter is easily shown by noticing that the processes A and B in Figure 5 are I/O-unambiguous. Thus, we focus on $\approx_r^{\mathsf{p}} \subseteq \approx_r^{\mathsf{e}}$.

We start by proving that for all I/O-unambiguous processes A, for all $A \stackrel{\mathsf{tr}}{\Rightarrow} A'$, we have that A' is I/O-unambiguous. Note that structural equivalence preserves I/O-unambiguity, *i.e.* for all extended processes A, B, for all channel name c, $A \equiv B$ implies $\mathsf{ioua}(A, c) = \mathsf{ioua}(B, c)$. Hence, we assume w.l.o.g. that a name is bound at most once and the set of bound and free names are disjoint.

Second, we show that for all I/O-unambiguous processes A, for all $A \xrightarrow{\nu z.out(c,z).in(c,z)}_{\mathsf{p}} A'$, we have that $\xRightarrow{\nu z.eav(c,z)}_{\mathsf{e}} A'$. To prove this property, denoted $\mathcal{P}$, let us assume w.l.o.g. that $A \xrightarrow{\nu z.out(c,z)}_{\mathsf{p}} A_1 \rightarrow_{\mathsf{p}}^* A_2 \xrightarrow{in(c,z)}_{\mathsf{p}} A'$. The transition $A \xrightarrow{\nu z.out(c,z)}_{\mathsf{p}} A_1$ indicates that $A \equiv \nu\tilde{n}.(\mathsf{out}^{\mathsf{ho}}(c, u).P \mid Q)$ and $A_1 \equiv \tilde{n}.(P \mid Q \mid \{^u/_z\})$ for some $P, Q, \tilde{n}, c, u$. Note that A is I/O-unambiguous, and hence $\mathsf{ioua}(P, c) = \top$.

As A is I/O-unambiguous implies that A does not contain private channels, we have that the rule applied in $A_1 \rightarrow_{\mathsf{p}}^* A_2$ is either the rule THEN or ELSE. Therefore, there exists P' and Q' such that $P \rightarrow_{\mathsf{p}}^* P'$, $Q \rightarrow_{\mathsf{p}}^* Q'$, $A_n \equiv \nu\tilde{n}.(P' \mid Q' \mid \{^u/_x\})$ and $\mathsf{ioua}(P', c) = \top$. Hence, we deduce that there exists Q_1, Q_2 such that $Q' \equiv \nu\tilde{m}.(\mathsf{in}(c, x)Q_1 \mid Q_2)$ and $A' \equiv \nu\tilde{n}.\nu\tilde{m}.(P' \mid Q_1\{^u/_x\} \mid Q_2)$. We conclude the proof of this property by noticing that we can first apply on A the reduction rules of $Q \rightarrow_{\mathsf{p}}^* Q'$, then apply the rule C-EAV and finally apply the rules of $P \rightarrow_{\mathsf{p}}^* P'$.

1. To prove $\approx_t^{\mathsf{p}} \subseteq \approx_t^{\mathsf{e}}$, we assume that A,B are two closed honest extended processes such that $A \approx_t^{\mathsf{p}} B$. For all $A \stackrel{\mathsf{tr}}{\Rightarrow}_{\mathsf{e}} A'$, it follows from the semantics that $A \xRightarrow{\mathsf{tr}_{\mathsf{p}}}_{\mathsf{p}} A'$

where tr_p is obtained by replacing in tr each $\nu z.eav(c,z)$ by $\nu z.out(c,z).in(c,z)$. Since $A \approx_t^{\mathsf{p}} B$, there exists B' such that $B \stackrel{\mathsf{tr}_p}{\Longrightarrow}_{\mathsf{p}} B'$ and $\phi(A') \sim \phi(B')$. Thanks to the property $\mathcal{P}$, we conclude that $B \stackrel{\mathsf{tr}}{\Rightarrow}_{\mathsf{e}} B'$.

2. To prove $\approx_\ell^{\mathsf{p}} \subseteq \approx_\ell^{\mathsf{e}}$, we assume that A,B are two closed honest extended processes such that $A \approx_\ell^{\mathsf{p}} B$ and let $\mathcal{R}$ be the relation witnessing this equivalence. We will show that $\mathcal{R}$ is also a labelled bisimulation in the eavesdropping semantics. Suppose $A\mathcal{R}B$.
   - as $A \approx_\ell^{\mathsf{p}} B$, we have that $\phi(A) \sim \phi(B)$.
   - if $A \stackrel{\tau}{\rightarrow}_{\mathsf{e}} A'$ then, as A is honest, $A \stackrel{\tau}{\rightarrow}_{\mathsf{p}} A'$. As $A \approx_\ell^{\mathsf{p}} B$ there exists B' such that $B \stackrel{\epsilon}{\Rightarrow}_{\mathsf{p}} B'$ and $A'\mathcal{R}B'$. As $\stackrel{\tau}{\rightarrow}_{\mathsf{p}} \subset \stackrel{\tau}{\rightarrow}_{\mathsf{e}}$, $B \stackrel{\epsilon}{\Rightarrow}_{\mathsf{e}} B'$
   - if $A \stackrel{\ell}{\rightarrow}_{\mathsf{e}} A'$ then, as A is I/O-unambiguous, $A \stackrel{\mathsf{tr}}{\Rightarrow}_{\mathsf{e}} A'$ where $\mathsf{tr} = \nu z.out(c,z).in(c,z)$ when $\ell = \nu z.eav(c,z)$ else $\mathsf{tr} = \ell$. As $A \approx_\ell^{\mathsf{p}} B$, there exists B' such that $B \stackrel{\mathsf{tr}}{\Rightarrow}_{\mathsf{p}} B'$ and $A'\mathcal{R}B'$. When $\mathsf{tr} = \ell$, the definition of the semantics directly gives us $B \stackrel{\ell}{\Rightarrow}_{\mathsf{e}} B'$. When $\mathsf{tr} = \nu z.out(c,z).in(c,z)$, the property $\mathcal{P}$ gives us $B \stackrel{\ell}{\Rightarrow}_{\mathsf{e}} B'$. □

5 Different semantics in practice

As we have seen, in general, the three proposed semantics may yield different results. A conservative approach would consist in verifying always the eavesdropping semantics which is stronger than the two other ones, as shown before. However, this semantics seems also to be the least efficient one to verify.

We have implemented the three different semantics in the APTE tool, for processes with static channels, i.e. inputs and outputs may only have names in the channel position and not variables. This allowed us to investigate the difference in results and performance between the semantics.

In our experiments we considered several examples from APTE's repository:

- the Private Authentication protocol proposed by Abadi and Fournet [2];
- the passive authentication protocol implemented in the European Passport protocol [16, 4];
- the French and UK versions of the Basic Access Protocol (BAC) implemented in the European passport [16, 5].

For all these examples we found that the results, i.e., whether trace equivalence holds or not, was unchanged, independent of the semantics. However, as expected, performance of the private semantics was generally better. The existing protocol encodings generally used a single public channel. To enforce I/O-unambiguity, we introduced different channels and, surprisingly, noted that distinct channels significantly enhance the tool's performance. (The model using different channels in the case of RFID protocols such as the electronic passport is certainly questionable.)

The results are summarised in the following table. For each protocol we considered the original encoding, and a slightly changed one which enforces I/O-unambiguity. In the results column we mark an attack by a cross ($\times$) and a successful verification with

a check mark (✓). In case of an attack we generally considered the minimal number of sessions needed to find the attack. In case of a successful verification we consider more sessions, which is the reason for the much higher verification times.

Protocol	# sessions	Property	Time			Result
			$\approx_t^{\mathsf{e}}$	$\approx_t^{\mathsf{c}}$	$\approx_t^{\mathsf{p}}$	
Private Authentication	1	Anonymity	1s	1s	1s	✓
	2		53h 53m 20s	47h 46m 40s	46h 56m 40s	
I/O unambiguous	1		1s	1s	1s	
	2		31m 39s	21m 2s	19m 39s	
Passive Authentication	2	Anonymity	4s	3s	3s	✓
I/O unambiguous	2		4s	4s	3s	
	3		6h 38m 34s	6h 29m 24s	6h 36m 40s	
Passive Authentication	2	Unlinkability	4s	4s	3s	✓
I/O unambiguous	2		3s	3s	3s	
	3		7h 43m 2s	6h 39m 14s	4h 27m 47s	
FR BAC protocol	2	Unlinkability	1s	1m 29s	1s	×
I/O unambiguous	2		1s	1s	1s	
UK BAC protocol	2	Unlinkability	1h 2m 35s	?	6h 39m 14s	×
I/O unambiguous	2		4s	53s	2s	

6 Conclusion

In this paper we investigated two families of Dolev-Yao models, depending on how the hypothesis that the *attacker controls the network* is reflected. While the two semantics coincide for reachability properties, they yield incomparable notions of behavioral equivalences, which have recently been extensively used to model privacy properties. The fact that forcing all communication to be routed through the attacker may diminish his distinguishing power may at first seem counter-intuitive. We also propose a third semantics, where internal communication among honest participants is permitted but leaks the message to the attacker. This new communication semantics entails strictly stronger equivalences than the two classical ones. We also identify two subclasses of protocols for which (some) semantics coincide. Finally, we implemented the three semantics in the APTE tool. Our experiments showed that the three semantics provide the same result on the case studies in the APTE example repository. However, the private semantics is slightly more efficient, as less interleavings have to be considered. Our results illustrate that behavioral equivalences are much more subtle than reachability properties and the need to carefully choose the precise attacker model.

Acknowledgments. We would like to thank Catherine Meadows and Stéphanie Delaune for interesting discussions, as well as the anonymous reviewers for their comments. This work has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation program (grant agreement No 645865-SPOOC) and the ANR project SEQUOIA ANR-14-CE28-0030-01.

References

1. M. Abadi and C. Fournet. Mobile values, new names, and secure communication. In H. R. Nielson, editor, *28th Symposium on Principles of Programming Languages (POPL'01)*, pages 104–115, London, UK, Jan. 2001. ACM.
2. M. Abadi and C. Fournet. Private authentication. *Theor. Comput. Sci.*, 322(3):427–476, Sept. 2004.
3. M. Abadi and A. D. Gordon. A calculus for cryptographic protocols: The spi calculus. *Inf. Comput.*, 148(1):1–70, 1999.
4. M. Arapinis, V. Cheval, and S. Delaune. Verifying privacy-type properties in a modular way. In V. Cortier and S. Zdancewic, editors, *Proceedings of the 25th IEEE Computer Security Foundations Symposium (CSF'12)*, pages 95–109, Cambridge Massachusetts, USA, June 2012. IEEE Computer Society Press.
5. M. Arapinis, T. Chothia, E. Ritter, and M. Ryan. Analysing unlinkability and anonymity using the applied pi calculus. In *Proc. 23rd Computer Security Foundations Symposium (CSF'10)*, pages 107–121. IEEE Computer Society Press, 2010.
6. A. Armando, D. A. Basin, Y. Boichut, Y. Chevalier, L. Compagna, J. Cuéllar, P. H. Drielsma, P.-C. Héam, O. Kouchnarenko, J. Mantovani, S. Mödersheim, D. von Oheimb, M. Rusinowitch, J. Santiago, M. Turuani, L. Viganò, and L. Vigneron. The AVISPA tool for the automated validation of internet security protocols and applications. In *Proc. 17th International Conference on Computer Aided Verification (CAV'05)*, Lecture Notes in Computer Science, pages 281–285. Springer, 2005.
7. K. Babel, V. Cheval, and S. Kremer. On communication models when verifying equivalence properties. Technical report, HAL, 2017.
8. B. Blanchet. Automatic verification of correspondences for security protocols. *Journal of Computer Security*, 17(4):363–434, 2009.
9. B. Blanchet, M. Abadi, and C. Fournet. Automated verification of selected equivalences for security protocols. *Journal of Logic and Algebraic Programming*, 75(1):3–51, 2008.
10. R. Chadha, V. Cheval, Ş. Ciobâcă, and S. Kremer. Automated verification of equivalence properties of cryptographic protocol. *ACM Transactions on Computational Logic*, 2016. To appear.
11. V. Cheval, H. Comon-Lundh, and S. Delaune. Trace equivalence decision: Negative tests and non-determinism. In *Proc. 18th ACM Conference on Computer and Communications Security (CCS'11)*. ACM, Oct. 2011.
12. V. Cheval, V. Cortier, and S. Delaune. Deciding equivalence-based properties using constraint solving. *Theoretical Computer Science*, 492:1–39, June 2013.
13. C. J. Cremers. The Scyther Tool: Verification, falsification, and analysis of security protocols. In *Proc. 20th International Conference on Computer Aided Verification (CAV'08)*, volume 5123 of *Lecture Notes in Computer Science*, pages 414–418. Springer, 2008.
14. S. Delaune, S. Kremer, and M. D. Ryan. Verifying privacy-type properties of electronic voting protocols. *Journal of Computer Security*, 17(4):435–487, July 2009.
15. N. Dong, H. Jonker, and J. Pang. Analysis of a receipt-free auction protocol in the applied pi calculus. In S. Etalle and J. Guttman, editors, *Proc. International Workshop on Formal Aspects in Security and Trust (FAST'10)*, Pisa, Italy, 2010. To appear.
16. P. T. Force. PKI for machine readable travel documents offering ICC read-only access. Technical report, International Civil Aviation Organization, 2004.
17. J. K. Millen and V. Shmatikov. Constraint solving for bounded-process cryptographic protocol analysis. In *Proc. 8th Conference on Computer and Communications Security*, pages 166–175. ACM Press, 2001.

18. L. C. Paulson. The inductive approach to verifying cryptographic protocols. *Journal of Computer Security*, 6(1/2):85–128, 1998.
19. P. Ryan, S. Schneider, M. Goldsmith, G. Lowe, and A. Roscoe. *Modelling and Analysis of Security Protocols*. Addison Wesley, 2000.
20. B. Schmidt, S. Meier, C. Cremers, and D. Basin. The tamarin prover for the symbolic analysis of security protocols. In *Proc. 25th International Conference on Computer Aided Verification (CAV'13)*, volume 8044 of *Lecture Notes in Computer Science*, pages 696–701. Springer, 2013.
21. F. J. Thayer Fabrega, J. C. Herzog, and J. D. Guttman. Strand spaces: Proving security protocols correct. *Journal of Computer Security*, 7(2/3):191–230, 1999.
22. A. Tiu and J. E. Dawson. Automating open bisimulation checking for the spi calculus. In *Proc. 23rd Computer Security Foundations Symp. (CSF'10)*, pages 307–321. IEEE Comp. Soc., 2010.