D1.1 - A taxonomy of process equivalences: the communication model revisited

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

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

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 cryptgraphy 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 [4], Scyther [9], Tamarin [15], Millen and Shmatikov's constraint solver [12], and the model used in Paulson's inductive approach [13].

Some other models, such as those based on process algebras, e.g. work based on CSP [14], the Spi [2] and applied pi calculus [1], but also the strand space model [16], 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 [10, 3, 11]. The number of tools able to verify such properties is also increasing [5, 17, 7, 6]. We have noted that for instance the AKISS tool [6] does not allow any direct communication on public channels, while the APTE tool [7] 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).

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. The results are summarised in Figure 3.

1 Model

The *applied pi calculus* [1] is a well known 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.

1.1 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 denote by $u =_{\mathsf{E}} v$ 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 protocol 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.

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

where u and v are base type terms, n is a name, x is a variable and c is a term of channel type, *i.e.* a name or a variable, 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 be instantiate the variable x. The process $\mathsf{eav}(c, x)$ represents 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 $u = v$. Finally, the substitution $\{^u/_x\}$ is an active substitution that replaces the variables x with the term u. 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 process. 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. 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 eavesdroping process and without the tags correspond exactly to the syntax of the original applied pi calculus.

Lastly, we consider the notion of *frame* that are extended processes build from 0, parallel composition, name and variable restrictions and active substitutions. 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 $dom(\phi(A))$.

1.2 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:

$$\begin{array}{ccccc} A \equiv A \mid 0 & & !P \equiv !P \mid P & & \nu n.0 \equiv 0 \\ \nu i.\nu j.A \equiv \nu j.\nu i.A & & \nu x.\{^u/_x\} \equiv 0 & & \{^u/_x\} \mid A \equiv \{^u/_x\} \mid A\{^u/_x\} \\ A \mid \nu i.B \equiv \nu i.(A \mid B) & \text{when } i \notin fv(A) \cup fn(A) & & & \\ \{^u/_x\} \equiv \{^v/_x\} & \text{when } u =_{\mathsf{E}} v & & & \end{array}$$

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 equiv-

alence and such that:

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

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

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

$$\nu c.(\mathsf{out}^{\theta}(c, u).P \mid \mathsf{in}^{\theta'}(c, x).Q \mid R) \xrightarrow{\tau}_s \nu c.(P \mid Q\{^{u}/_{x}\} \mid R) \qquad \text{where } s \in \{\mathsf{p}, \mathsf{e}\} \quad \textsc{Comm-Priv}$$

$$\mathsf{out}^{\theta}(c, u).P \mid \mathsf{in}^{\theta'}(c, x).Q \xrightarrow{\tau}_s P \mid Q\{^{u}/_{x}\} \qquad \text{where } \mathsf{at} \in \{\theta, \theta'\} \text{ and } s \in \{\mathsf{p}, \mathsf{e}\} \quad \textsc{Comm-Env}$$

$$\mathsf{out}^{\mathsf{ho}}(c, u).P \mid \mathsf{in}^{\mathsf{ho}}(c, x).Q \mid \mathsf{eav}(c, y).R \xrightarrow{\tau}_{\mathsf{e}} P \mid Q\{^{u}/_{x}\} \mid R\{^{u}/_{y}\} \quad \textsc{Comm-Eav}$$

We denote by $\Rightarrow_s$ the reflexive, transitive closure of $\xrightarrow{\tau}_s$ for $s \in \{\mathsf{c}, \mathsf{p}, \mathsf{e}\}$.

1.3 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 encodes that a process cannot reach some bad state. Equivalences define the fact that no attacker can see the difference between two processes. This was originally defined by the *(may)-testing equivalence* [2] in the context of 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 $A \Downarrow_c^s$ to hold when $A \Rightarrow_s C[\mathsf{out}^{\theta}(c, t).P]$ for some evaluation context C that does not bind c, some θ, some term t and some plain process P. For example the secrecy of s in the process $\nu s.A$ can be encoded by checking

$$\nu s.(A \mid \mathsf{in}^{\mathsf{ho}}(c, x).\mathsf{if}\ x = s\ \mathsf{then}\ \mathsf{out}^{\mathsf{ho}}(\mathsf{bad}, s)) \Downarrow_{\mathsf{bad}}^s$$

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

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 context $C[\_]$ closing for A and B, for all channel c, $C[A] \Downarrow_c^s$ is equivalent to $C[B] \Downarrow_c^s$.*

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* $A' \approx_m^s B'$ *for some* B'*;*
– $C[A] \approx_m^s C[B]$ *for all attacker evaluation contexts* $C[\_]$ *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. $\approx_o^s \subsetneq \approx_m^s$ *for* $s \in \{\mathsf{c}, \mathsf{e}, \mathsf{p}\}$.

1.4 Labelled semantics

The internal reduction semantics introduced in the previous section requires to reason about arbitrary contexts, Similarly 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 2.

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

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

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

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

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

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

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

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

Fig. 2. 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 $\mathsf{tr} \in (\mathcal{A} \backslash \{\tau\})^*$,

we say that $A \stackrel{\mathsf{tr}}{\Rightarrow}_s A'$ when there exists $w \in \mathcal{A}^*$ such that tr is the word w where we remove all τ actions and $A \xrightarrow{w}_s A'$.

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 $A \stackrel{\mathsf{tr}}{\Rightarrow} A'$ and $out(c,t) \in \mathsf{tr}$ for some $\mathsf{tr} \in (\mathcal{A} \setminus \{\tau\})^*$, term t and extended process A'.

Proposition 2. *$A \downarrow\downarrow_c^s$ iff $\exists C^s[\_].\ c \notin fn(C)$ and $C^s[A] \Downarrow_c^s$ for $s \in \{\mathsf{c}, \mathsf{e}, \mathsf{p}\}$.*

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 $(u =_{\mathsf{E}} v)\phi_2$.*

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:*

- *$\phi(A) \sim \phi(B)$*
- *$A \xrightarrow{\tau}_s A'$ implies $B \stackrel{\epsilon}{\Rightarrow}_s B'$ and $A' \approx_\ell^s B'$ for some B',*
- *$A \xrightarrow{\ell}_s A'$ 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 [8], 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 traces that may be reached for any given trace 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 [8] and is easily adapted to the other semantics. To see that the implication is strict, we give an example of two processes A and B such that $A \approx_m^s B$, but $A \not\approx_t^s B$ (for $s \in \{\mathsf{c}, \mathsf{e}, \mathsf{p}\}$):

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

Process A computes a value to be output on channel c. 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 to it. Hence, the only possible traces are of the form $A \xRightarrow{\nu x.out(c,x)}_s A'$ where $\phi(A') = \{{}^{h^n(a)}/_x\}$ for $n \in \mathbb{N}$, where $h^n(a)$ denotes n applications of h and $h^0(a) = a$. Process B may either behave as A or output the fresh name s. 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 $x = h^n(a)$. To prove that $A \approx_m^s B$, we observe that process B can easily mimic process A, as $B \stackrel{\tau}{\Rightarrow}_s A$ (performing a private communication). Hence, it remains to show that for any context C closing for A and B, if $C[B] \Downarrow_{c_I}^s$ for some channel c_I then $C[A] \Downarrow_{c_I}^s$. Note that $C[B] \Downarrow_{c_I}^s$ indicates that there is a finite derivation from $C[B]$ to a process that can emit on c_I. The interesting derivations are the ones that output the fresh name s on c. Since the derivation is finite, there exists an integer n such that the equality and disequality tests performed along the derivation would be satisfied by replacing s with $h^{n+1}(a)$ (n can be computed as the number of occurrences of h in the tests). Therefore, in $C[A]$, for all the tests of the derivation to be satisfied we only need to trigger $n+1$ internal communications allowing $h^{n+1}(a)$ to be emitted on the channel c. 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.

2 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 any ground, honest extended process A, for all channel d, we have that $A \Downarrow_d^{\mathsf{p}}$ iff $A \Downarrow_d^{\mathsf{c}}$ iff $A \Downarrow_d^{\mathsf{e}}$.*

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 appear to be incomparable, and the labelled bisimulations as well.

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

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

$$\begin{aligned} A &\triangleq \nu s_1.\nu s_2.((\mathsf{out}^{\mathsf{ho}}(c, s_1).\mathsf{in}^{\mathsf{ho}}(c, x).P_1(x)) \mid (\mathsf{in}^{\mathsf{ho}}(c, y).P_2(y))) \\ B &\triangleq \nu s_1.\nu s_2.((\mathsf{out}^{\mathsf{ho}}(c, s_1).\mathsf{in}^{\mathsf{ho}}(c, x).P_2(x)) \mid (\mathsf{in}^{\mathsf{ho}}(c, y).P_1(y))) \end{aligned}$$

where

$$\begin{aligned} P_1(x) &\triangleq (\mathsf{if}\ x = s_1\ \mathsf{then}\ \mathsf{out}^{\mathsf{ho}}(d, s_2)) \mid (\mathsf{if}\ x = s_2\ \mathsf{then}\ \mathsf{out}^{\mathsf{ho}}(e, x)) \\ P_2(x) &\triangleq (\mathsf{if}\ x = s_1\ \mathsf{then}\ \mathsf{out}^{\mathsf{ho}}(d, s_2)) \end{aligned}$$

To see that $A \approx_\ell^{\mathsf{p}} B$ we first observe that if $A \xrightarrow{\nu z.out(c,z)}_{\mathsf{p}} A'$ 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 $P_1(t) \approx_\ell^{\mathsf{p}} 0 \approx_\ell^{\mathsf{p}} P_2(t)$. Finally, if $t \neq s_2$ we also have that $P_1(t) \approx_\ell^{\mathsf{p}} P_2(t)$ as in particular $P_1(s_1) \approx_\ell^{\mathsf{p}} P_2(s_1)$. Therefore,

$$\begin{array}{c} \nu s_1.\nu s_2.(\mathsf{out}^{\mathsf{ho}}(c, s_1).\mathsf{in}^{\mathsf{ho}}(c, x).P_1(x)) \\ \approx_\ell^{\mathsf{p}} \\ \nu s_1.\nu s_2.(\mathsf{out}^{\mathsf{ho}}(c, s_1).\mathsf{in}^{\mathsf{ho}}(c, x).P_2(x)) \end{array}$$

which allows us to conclude.

To see that $A \not\approx_t^{\mathsf{c}} B$ 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 $\approx_r^{\mathsf{c}} \not\subseteq \approx_r^{\mathsf{p}}$ for $r \in \{\ell, t\}$ we show that there exist processes A and B such that $A \approx_\ell^{\mathsf{c}} B$ and $A \not\approx_t^{\mathsf{p}} B$. As in the first part of the proof, note

that, as $\approx_\ell^s \subset \approx_t^s$ for $s \in \{\mathsf{c}, \mathsf{p}\}$ these processes demonstrate both that $\approx_\ell^{\mathsf{c}} \not\subseteq \approx_\ell^{\mathsf{p}}$ and $\approx_t^{\mathsf{c}} \not\subseteq \approx_t^{\mathsf{p}}$. We define the processes

$$\begin{aligned} A &\triangleq \nu s.(\mathsf{in}^{\mathsf{ho}}(c,x).\mathsf{out}^{\mathsf{ho}}(c,s).\mathsf{out}^{\mathsf{ho}}(d,a) \mid \mathsf{in}^{\mathsf{ho}}(c,y).P(y)) \\ B &\triangleq \nu s.(\mathsf{in}^{\mathsf{ho}}(c,x).(\mathsf{out}^{\mathsf{ho}}(c,s).\mathsf{out}^{\mathsf{ho}}(d,a) \mid \mathsf{in}^{\mathsf{ho}}(c,y).P(y))) \end{aligned}$$

where

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

To see that $A \approx_\ell^{\mathsf{c}} B$ 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 either $A' = B'$ or $A' = A''$ with $A'' \triangleq \nu s.(\mathsf{in}^{\mathsf{ho}}(c,x).\mathsf{out}^{\mathsf{ho}}(c,s).\mathsf{out}^{\mathsf{ho}}(d,a) \mid P(t))$. Therefore, to complete the proof, we only need to find B'' such that $B \xRightarrow{in(c,t)} B''$ 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 $\nu s.(\mathsf{out}^{\mathsf{ho}}(d,a) \mid P(s)) \approx_\ell^{\mathsf{c}} A''$.

To see that $A \not\approx_t^{\mathsf{p}} B$ we first observe that A may perform the following transition sequence:

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

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, we describe in more details these processes and give the proofs of them being counter-example.

Next, we show that the eavesdropping semantics yields strictly stronger bisimulations and trace equivalences.

Theorem 4. *$\approx_t^{\mathsf{e}} \subsetneq \approx_t^{\mathsf{p}}$ and $\approx_t^{\mathsf{e}} \subsetneq \approx_t^{\mathsf{c}}$.*

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. We can now show that the inclusion $\approx_t^{\mathsf{e}} \subsetneq \approx_t^{\mathsf{p}}$ is strict. Suppose, by contradiction, that $\approx_t^{\mathsf{p}} \subseteq \approx_t^{\mathsf{e}}$. By Item 2 and transitivity we obtain that $\approx_t^{\mathsf{p}} \subset \approx_t^{\mathsf{c}}$ contradicting Theorem 3
4. Finally, we show that the inclusion $\approx_t^{\mathsf{e}} \subsetneq \approx_t^{\mathsf{c}}$ is strict. We proceed similarly as in the previous item, but relying on Item 1, instead of Item 2. □

We note from the poof that the implications are strict even for processes containing only public channels.

Theorem 5. *$\approx_\ell^{\mathsf{e}} \subsetneq \approx_\ell^{\mathsf{p}}$ and $\approx_\ell^{\mathsf{e}} \subsetneq \approx_\ell^{\mathsf{c}}$.*

Proof (Sketch).

1. We first show that $\approx_\ell^{\mathsf{e}} \subseteq \approx_\ell^{\mathsf{p}}$. Suppose $A \approx_\ell^{\mathsf{e}} B$ 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 $A \approx_\ell^{\mathsf{e}} B$, we have that $\phi(A) \sim \phi(B)$.
   - if $A \stackrel{\tau}{\rightarrow}_{\mathsf{p}} A'$ then, as $\stackrel{\tau}{\rightarrow}_{\mathsf{p}} \subset \stackrel{\tau}{\rightarrow}_{\mathsf{e}}$, $A \stackrel{\tau}{\rightarrow}_{\mathsf{e}} A'$. As $A \approx_\ell^{\mathsf{e}} B$ 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 $A \stackrel{\ell}{\rightarrow}_{\mathsf{p}} A'$ and $bn(\ell) \cap fn(B) = \emptyset$ then we also have that $A \stackrel{\ell}{\rightarrow}_{\mathsf{e}} A'$ (as $\stackrel{\ell}{\rightarrow}_{\mathsf{p}} \subset \stackrel{\ell}{\rightarrow}_{\mathsf{e}}$ 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 $A \approx_\ell^{\mathsf{e}} B$ implies $A \approx_\ell^{\mathsf{c}} B$ for any A, B. We will show that $\approx_\ell^{\mathsf{e}}$ is also a labelled bisimulation in the classical semantics. The proof relies on similar arguments as in Item 2 of the proof of Theorem 4 and the facts that
   - $\nu\tilde{n}.(A' \mid \{^t/_x\}) \approx_\ell^{\mathsf{e}} \nu\tilde{n}.(B' \mid \{^u/_x\})$ implies $\nu\tilde{n}.A' \approx_\ell^{\mathsf{e}} \nu\tilde{n}.B'$,
   - $A' \approx_\ell^{\mathsf{e}} B'$ implies $\nu c.A' \approx_\ell^{\mathsf{e}} \nu c.B'$

   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 $\approx_\ell^{\mathsf{e}} \subsetneq \approx_\ell^{\mathsf{c}}$ is strict, i.e., there exist A and B such that $A \approx_\ell^{\mathsf{c}} B$, but $A \not\approx_t^{\mathsf{e}} B$ (which implies $A \not\approx_\ell^{\mathsf{e}} B$). We define the processes

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

   where

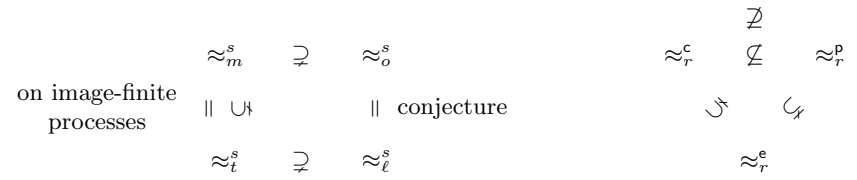
$$\begin{aligned} P_1(x) &\mathrel{\hat{=}} (\mathsf{if}\ x = s_1\ \mathsf{then}\ \mathsf{in}^{\mathsf{ho}}(d, z).\mathsf{if}\ z = s_1\ \mathsf{then}\ \mathsf{out}^{\mathsf{ho}}(d, s_2)) \mid \\ &\qquad (\mathsf{if}\ x = s_2\ \mathsf{then}\ \mathsf{out}^{\mathsf{ho}}(e, x)) \\ P_2(x) &\mathrel{\hat{=}}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

   This example is a variant of the one given in the proof of Theorem 3. 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 $P_2(x)$: tis additional check is used to verify whether the adversary learned s_1 or not. The proof that $A \approx_\ell^{\mathsf{c}} B$ follows the same lines as in Theorem 3. We just additionally observe that $\nu s_1.(\mathsf{in}^{\mathsf{ho}}(d, z).\mathsf{if}\ z = s_1\ \mathsf{then}\ \mathsf{out}^{\mathsf{ho}}(d, s_2)) \approx_\ell^{\mathsf{c}} \nu s_1.\ (\mathsf{in}^{\mathsf{ho}}(d, z).0)$.
   The trace witnessing that $A \not\approx_\ell^{\mathsf{e}} B$ 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 $P_2(s_2)$.
4. Finally we show that the implication $\approx_\ell^{\mathsf{e}} \subseteq \approx_\ell^{\mathsf{p}}$ is strict. Suppose, by contradiction, that $\approx_\ell^{\mathsf{p}} \subseteq \approx_\ell^{\mathsf{e}}$. We have shown above that $\approx_\ell^{\mathsf{e}} \subset \approx_\ell^{\mathsf{c}}$. By transitivity we obtain that $\approx_\ell^{\mathsf{p}} \subset \approx_\ell^{\mathsf{c}}$ contradicting Theorem 3. □

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

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 A. D. Gordon. A calculus for cryptographic protocols: The spi calculus. *Inf. Comput.*, 148(1):1–70, 1999.



for all $s \in \{\mathsf{c}, \mathsf{p}, \mathsf{e}\}$ and for all $r \in \{m, t, \ell\}$.

Fig. 3. Overview of the results.

3. M. Arapinis, T. Chothia, E. Ritter, and M. D. Ryan. Analysing unlinkability and anonymity using the applied pi calculus. In *Proc. 23rd Computer Security Foundations Symposium (CSF'10)*, pages 107–121. IEEE Comp. Soc. Press, 2010.
4. 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.
5. 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.
6. 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.
7. 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.
8. V. Cheval, V. Cortier, and S. Delaune. Deciding equivalence-based properties using constraint solving. *Theoretical Computer Science*, 492:1–39, June 2013.
9. 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.
10. 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.
11. 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.
12. 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.
13. L. C. Paulson. The inductive approach to verifying cryptographic protocols. *Journal of Computer Security*, 6(1/2):85–128, 1998.
14. P. Ryan, S. Schneider, M. Goldsmith, G. Lowe, and A. Roscoe. *Modelling and Analysis of Security Protocols*. Addison Wesley, 2000.

15. 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.
16. 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.
17. 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.

Theorem 6. *When restricted to processes without else branches, we have that* $\approx_r^{\mathsf{p}} \not\subseteq \approx_r^{\mathsf{c}}$ *and* $\approx_r^{\mathsf{c}} \not\subseteq \approx_r^{\mathsf{p}}$ *for* $r \in \{\ell, t\}$.

Proof. The fact that $\approx_r^{\mathsf{p}} \not\subseteq \approx_r^{\mathsf{c}}$ for $r \in \{\ell, t\}$ has already been shown in the proof of Theorem 3 as the processes A, B witnessing the result did not have else branches.

To show that $\approx_\ell^{\mathsf{c}} \not\subseteq \approx_\ell^{\mathsf{p}}$ we show that there exist processes A and B without else branches such that $A \approx_\ell^{\mathsf{c}} B$ and $A \not\approx_\ell^{\mathsf{p}} B$. We define the processes

$$\begin{array}{rcl} A & \hat{=} & \nu s.(\mathsf{in}^{\mathsf{ho}}(c,x).\mathsf{out}^{\mathsf{ho}}(c,s) \mid \mathsf{in}^{\mathsf{ho}}(c,y).P(y)) \\ B & \hat{=} & \nu s.(\mathsf{in}^{\mathsf{ho}}(c,x).(\mathsf{out}^{\mathsf{ho}}(c,s) \mid \mathsf{in}^{\mathsf{ho}}(c,y).P(y))) \end{array}$$

where

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

To see that $A \approx_\ell^{\mathsf{c}} B$ 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 $B' = \nu s.(\mathsf{out}^{\mathsf{ho}}(c,s) \mid \mathsf{in}^{\mathsf{ho}}(c,y).P(y))$. On the other hand, if $A \xrightarrow{in(c,M)} A'$ then either $A' = B'$ or $A' = A''$ where $A'' \hat{=} \nu s.(\mathsf{in}^{\mathsf{ho}}(c,x).\mathsf{out}^{\mathsf{ho}}(c,s) \mid P(t))$. Therefore, to complete the proof, we only need to find B'' such that $B \xRightarrow{in(c,t)} B''$ and $A'' \approx_\ell^{\mathsf{c}} B''$. Such process can be obtain by applying an internal communication on B', *i.e.* $B \xrightarrow{in(c,t)}_{\mathsf{c}} B' \xrightarrow{\tau} \nu s.P(s)$. Note that $t \neq s$ since s is bound, meaning that $P(t) \approx_\ell^{\mathsf{c}} 0$. Moreover, $P(s) \approx_\ell^{\mathsf{c}} \mathsf{in}^{\mathsf{ho}}(c,x).\mathsf{out}^{\mathsf{ho}}(c,s)$. This allows us to conclude that $\nu s.P(s) \approx_\ell^{\mathsf{c}} A''$.

To see that $A \not\approx_\ell^{\mathsf{p}} B$ we first observe that when $A \xrightarrow{in(c,t)}_{\mathsf{p}} A''$, B can only mimic A by preforming the transition $B \xrightarrow{in(c,t)} B'$. We conclude as $B' \xrightarrow{\nu z.out(c,z)}_{\mathsf{p}} \nu s.(\mathsf{in}^{\mathsf{ho}}(c,y).P(y) \mid \{^s/_z\})$ and $A'' \not\xrightarrow{\nu z.out(c,z)}_{\mathsf{p}}$.

We next show that there also exist A_1 and A_2 such that $A_1 \approx_t^{\mathsf{c}} A_2$, but $A_1 \not\approx_t^{\mathsf{p}} A_2$. We define the processes

$$\begin{array}{rcl} A_i & \hat{=} & \nu s_1.\nu s_2.(\ \mathsf{out}^{\mathsf{ho}}(c,h(s_1)) \mid \mathsf{out}^{\mathsf{ho}}(c,h(s_2)) \mid \\ & & \qquad \mathsf{in}^{\mathsf{ho}}(d,x).(\mathsf{if}\ x = h(s_1)\ \mathsf{then}\ Q_i \mid \mathsf{if}\ x = h(s_2)\ \mathsf{then}\ P_2) \end{array}$$

where $Q_1 \hat{=} P_1$, $Q_2 \hat{=} P_2$ and

$$\begin{array}{rcl} P_1 & \hat{=} & \mathsf{out}^{\mathsf{ho}}(e,a) \\ P_2 & \hat{=} & \mathsf{out}^{\mathsf{ho}}(f,a).\mathsf{out}^{\mathsf{ho}}(e,a) \mid \mathsf{in}^{\mathsf{ho}}(f,x) \end{array}$$

Using the APTE tool we have shown that indeed $A_1 \approx_t^{\mathsf{c}} A_2$ and $A_1 \not\approx_t^{\mathsf{p}} A_2$. The main argument why the result holds is that P_1 is trace included in P_2 in the classical semantics (as the output on channel f can be made silent through an internal communication) while this is not the case in the private semantics. □