Bisimulation and Unwinding for Verifying Possibilistic Security Properties *

Annalisa Bossi, Riccardo Focardi, Carla Piazza, and Sabina Rossi

Dipartimento di Informatica, Università Ca' Foscari di Venezia
{bossi,focardi,piazza,srossi}@dsi.unive.it

Abstract. We study bisimulation-based information flow security properties which are *persistent*, in the sense that if a system is secure, then all states reachable from it are secure too. We show that such properties can be characterized in terms of *bisimulation-like equivalence relations* between the system and the system itself prevented from performing confidential actions. Moreover, we provide a characterization of such properties in terms of *unwinding conditions* which demand properties of individual actions. These two different characterizations naturally lead to efficient methods for the verification and construction of secure systems. We also prove several *compositionality* results and discuss a sufficient condition to define *refinement* operators preserving security.

1 Introduction

Non-interference was introduced by Goguen and Meseguer [11, 12] as a concept for formalizing security within deterministic systems. Given a system in which *confidential* (i.e., high level) and *public* (i.e., low level) information may coexist, *non-interference* requires that confidential inputs never affect the output on the public interface of the system, i.e., never interfere with the low level users. If such a property holds, one can conclude that no information flow is ever possible from high to low level.

A possibilistic security property can be regarded as an extension of non-interference to non-deterministic systems. Starting with Sutherland [34], various such extensions have been proposed, e.g., [4, 9, 16, 21–24, 28, 33, 35]. Most of these properties are based on *traces*, i.e., the behavior of a system that may possibly be observed is the set of its execution sequences. Examples are non-inference [28], generalized non-interference [21], restrictiveness [21], and the perfect security property [35].

In [4], Focardi and Gorrieri express the concept of non-interference in the *Security Process Algebra* (*SPA*, for short) language in terms of bisimulation semantics. In particular, they introduce the notion of *Bisimulation-based non Deducibility on Compositions* (*BNDC*, for short): a system E is *BNDC* if what

* This work has been partially supported by MURST projects "Interpretazione astratta, type systems e analisi control-flow" and "Modelli formali per la sicurezza" and the EU project MyThS (IST-2001-32617).

a low level user sees of the system is not modified (in the sense of the bisimulation semantics) by composing any high level process Π with E. The main advantage of *BNDC* with respect to trace-based properties is that it is powerful enough to detect information flows due to the possibility for a high level malicious process to block or unblock a system (see [4, 6] for more detail). As a matter of fact, although Martinelli [20] has shown that *BNDC* is decidable over finite state processes, the problem of verifying *BNDC* is still open. The main difficulty consists of getting rid of the universal quantification on high level processes Π. A way to overcome this problems is to adopt sufficient conditions for *BNDC*. We recall from [6, 8] two of them, named *Strong BNDC* (*SBNDC*, for short) and *Persistent_BNDC* (*P_BNDC*, for short) [1]. In particular, *P_BNDC* has been shown to be suitable for analysing systems in dynamic contexts [8].

In this paper we consider *P_BNDC* and *SBNDC* and for both these properties we study two different characterizations that allow to exploit different verification techniques. The first kind of characterization is based on *bisimulation-like equivalence relation* between the system E to be analysed and the low level view of the system itself, denoted by $E \setminus H$ (the system E prevented from performing confidential actions). These bisimulation-based characterizations allow to exploit very efficient techniques for verifying the properties over finite-state processes using existing algorithms for the verification of strong bisimulation. The second kind of characterization is given in terms of *unwinding conditions* which demand properties of individual actions. Unwinding conditions aim at "distilling" the local effect of performing high level actions and are useful to define both proof systems (see, e.g., [2]) and *refinement* operators that preserve security properties, as done in [17]. Proof systems allow to incrementally build systems which are secure by construction. Similarly refinement operators are useful in a stepwise development process as properties which have been already investigated in some phase need not to be re-investigated in later phases.

In particular, we start by considering the two characterizations above, given in [2] for *P_BNDC* . By studying the relation between such two characterizations, we are able to give a new bisimulation-based characterization for *SBNDC*, which was originally defined through unwinding conditions. As a next step we investigate the compositionality of *P_BNDC* and *SBNDC*. Compositionality is useful for both verification and synthesis: if a property is preserved when systems are composed, then the analysis may be performed on subsystems and, in case of success, the system as a whole can be proved to satisfy the desired property. We notice that both *P_BNDC* and *SBNDC* are compositional with respect to the parallel operator, but they are not *fully* compositional, since they are not compositional with respect to the non-deterministic choice operator, which allows us to built a system that may choose to behave as one of two specified subsystems. It would be intuitive to require that a choice between two secure processes is still secure as observed in [10]. To this aim we introduce a new security property, named *Compositional P_BNDC* (*CP_BNDC*, for short), properly included in *P_BNDC*, which is fully compositional, i.e., it is compositional also with re-

[1] In [8], *P_BNDC* has been shown to be equivalent to the *SBSNNI* property of [6].

spect to the non-deterministic choice. *CP_BNDC* can be equivalently expressed through both a bisimulation-like equivalence and unwinding conditions.

We show that the bisimulation-based characterizations of our persistent security properties allow us to perform the verification task for finite state processes in polynomial time with respect to the number of states of the system, also improving on the polynomial time complexity required by the Compositional Security Checker Cosec presented in [5]. Finally, we provide a sufficient condition to define refinement operators preserving all our security properties.

The paper is organized as follows. In Section 2 we introduce some basic notions on the *SPA* language and the security properties *BNDC* and *P_BNDC*. In Section 3 we study the property *SBNDC* and provide a bisimulation-based characterization of it. In Section 4 we introduce the class of *CP_BNDC* processes and prove that it is fully compositional. Section 5 is devoted to complexity results for the bisimulation-based characterizations of the three properties. In Section 6 we propose a sufficient condition to define refinement operators for SPA processes preserving security. Finally, in Section 7 we discuss related works and draw some conclusions. All the proofs of propositions and theorems can be found in [1].

2 Basic Notions

In this section we report the syntax and semantics of the *Security Process Algebra* (*SPA*, for short) [6] and the definition of the security properties *BNDC* [4] and *P_BNDC* [8] together with some main results [2].

The SPA Language. The *Security Process Algebra* [6] is a variation of Milner's CCS [27], where the set of visible actions is partitioned into high level actions and low level ones in order to specify multilevel systems. SPA syntax is based on the same elements as CCS that is: a set $\mathcal{L}$ of *visible* actions such that $\mathcal{L} = I \cup O$ where $I = \{a, b, \ldots\}$ is a set of *input* actions and $O = \{\bar{a}, \bar{b}, \ldots\}$ is a set of *output* actions; a special action τ which models internal computations, i.e., not visible outside the system; a complementation function $\bar{\cdot} : \mathcal{L} \to \mathcal{L}$, such that $\bar{\bar{a}} = a$, for all $a \in \mathcal{L}$. Function $\bar{\cdot}$ is extended to *Act* by defining $\bar{\tau} = \tau$. $Act = \mathcal{L} \cup \{\tau\}$ is the set of all *actions*. The set of visible actions is partitioned into two sets, H and L, of high and low actions such that $\overline{H} = H$ and $\overline{L} = L$. The syntax of SPA *terms* (or *processes*) is defined as follows:

$$E ::= \mathbf{0} \mid a.E \mid E + E \mid E|E \mid E \setminus v \mid E[f] \mid Z$$

where $a \in Act$, $v \subseteq \mathcal{L}$, $f : Act \to Act$ is such that $f(\bar{\alpha}) = \overline{f(\alpha)}$, $f(\tau) = \tau$, $f(H) \subseteq H \cup \{\tau\}$, and $f(L) \subseteq L \cup \{\tau\}$, and Z is a constant that must be associated with a definition $Z \stackrel{\text{def}}{=} E$.

We denote by $\mathcal{E}$ the set of all SPA processes and by $\mathcal{E}_H$ the set of all high level processes, i.e., those constructed only using actions in $H \cup \{\tau\}$. The operational semantics of SPA agents is given in terms of *Labelled Transition Systems* (*LTS*, for short) as defined in [6].

The concept of *observation equivalence* is used to establish equalities among processes and it is based on the idea that two systems have the same semantics if and only if they cannot be distinguished by an external observer. This is obtained by defining an equivalence relation over $\mathcal{E}$. The *weak bisimulation* relation [27] equates two processes if they are able to mutually simulate their behavior step by step. Weak bisimulation does not care about internal τ actions.

We will use the following auxiliary notations. If $t = a_1 \cdots a_n \in Act^*$ and $E \stackrel{a_1}{\to} \cdots \stackrel{a_n}{\to} E'$, then we write $E \stackrel{t}{\to} E'$. We also write $E \stackrel{t}{\Longrightarrow} E'$ if $E(\stackrel{\tau}{\to})^* \stackrel{a_1}{\to} (\stackrel{\tau}{\to})^* \cdots (\stackrel{\tau}{\to})^* \stackrel{a_n}{\to} (\stackrel{\tau}{\to})^* E'$ where $(\stackrel{\tau}{\to})^*$ denotes a (possibly empty) sequence of τ labelled transitions. If $t \in Act^*$, then $\hat{t} \in \mathcal{L}^*$ is the sequence gained by deleting all occurrences of τ from t. As a consequence, $E \stackrel{\hat{a}}{\Longrightarrow} E'$ stands for $E \stackrel{a}{\Longrightarrow} E'$ if $a \in \mathcal{L}$, and for $E(\stackrel{\tau}{\to})^* E'$ if $a = \tau$ (note that $\stackrel{\tau}{\Longrightarrow}$ requires at least one τ labelled transition while $\stackrel{\hat{\tau}}{\Longrightarrow}$ means zero or more τ labelled transitions).

Definition 1 (Weak Bisimulation). *A binary relation $\mathcal{R} \subseteq \mathcal{E} \times \mathcal{E}$ over agents is a* weak bisimulation *if $(E, F) \in \mathcal{R}$ implies, for all $a \in Act$,*

- *if $E \stackrel{a}{\to} E'$, then there exists F' such that $F \stackrel{\hat{a}}{\Longrightarrow} F'$ and $(E', F') \in \mathcal{R}$;*
- *if $F \stackrel{a}{\to} F'$, then there exists E' such that $E \stackrel{\hat{a}}{\Longrightarrow} E'$ and $(E', F') \in \mathcal{R}$.*

Two agents $E, F \in \mathcal{E}$ are weakly bisimilar*, denoted by $E \approx F$, if there exists a weak bisimulation $\mathcal{R}$ containing the pair (E, F).*

The relation $\approx$ is the largest weak bisimulation and is an equivalence relation [27].

Security Properties. The *BNDC* [4] security property aims at guaranteeing that no information flow from the high to the low level is possible, even in the presence of malicious processes. The main motivation is to protect a system also from internal attacks, which could be performed by the so called *Trojan Horse* programs, i.e., programs that are apparently honest but hide inside some malicious code. Property *BNDC* is based on the idea of checking the system against all high level potential interactions, representing every possible high level malicious program. In particular, a system E is *BNDC* if for every high level process Π a low level user cannot distinguish E from $(E|\Pi)$, i.e., if Π cannot interfere with the low level execution of the system E.

Definition 2 (BNDC). *Let $E \in \mathcal{E}$.*

$$E \in BNDC \;\; \text{iff} \;\; \forall\, \Pi \in \mathcal{E}_H,\; E \setminus H \approx (E|\Pi) \setminus H.$$

Example 1. The *BNDC* property is powerful enough to detect information flows due to the possibility for a high level malicious process to block or unblock a system. Let $H = \{h\}$, $L = \{l, j\}$ and $E_1 = l.h.j.\mathbf{0} + l.j.\mathbf{0}$. Consider the process $\Pi = \bar{h}.\mathbf{0}$. We have that $(E_1|\Pi) \setminus H \approx l.j.\mathbf{0}$, while $E_1 \setminus H \approx l.\mathbf{0} + l.j.\mathbf{0}$. Note that the latter may (nondeterministically) block after the l input. Having many instances of this process, a low level user could deduce if $\bar{h}$ is executed by observing whether the system always performs j or not. Process E_1 may be "repaired", by including the possibility of choosing to execute j or not inside the process. Indeed, process $E_2 = l.h.j.\mathbf{0} + l.(\tau.j.\mathbf{0} + \tau.\mathbf{0})$ is *BNDC*.

In [8], it is introduced a security property called *Persistent_BNDC* (*P_BNDC*, for short), which is suitable for analysing systems in dynamic execution environments. Intuitively, a system E is *P_BNDC* if it never reaches insecure states.

Definition 3 (P_BNDC). *Let* $E \in \mathcal{E}$.

$$E \in P\_BNDC \ \text{ iff } \ \forall\ E' \text{ reachable from } E,\ E' \in BNDC.$$

Example 2. Consider the process E_2 of Example 1, i.e., $E_2 = l.h.j.\mathbf{0} + l.(\tau.j.\mathbf{0} + \tau.\mathbf{0})$ where $l, j \in L$ and $h \in H$. Suppose now that E_2 is moved in the middle of a computation. This might happen when it find itself in the state $h.j.\mathbf{0}$ (after the first l is executed). Now it is clear that this process is not secure, as a direct causality between h and j is present. In particular $h.j.\mathbf{0}$ is not *BNDC* and this gives evidence that E_2 is not *P_BNDC*. The process may be "repaired" as follows: $E_3 = l.(h.j.\mathbf{0}+\tau.j.\mathbf{0}+\tau.\mathbf{0})+l.(\tau.j.\mathbf{0}+\tau.\mathbf{0})$. It may be proved that E_3 is *P_BNDC*. Note that, from this example it follows that *P_BNDC* $\subset$ *BNDC*.

In [8] it has been shown that even if the definition of *P_BNDC* introduces an universal quantification over all the possible reachable states, this can be avoided by including the idea of "being secure in every state" inside the bisimulation equivalence notion. This is done by defining an equivalence notion which just focus on observable actions which do not belong to H. More in details, it is defined an observation equivalence, named *weak bisimulation up to H* where actions from H are allowed to be ignored, i.e., they are allowed to be matched by zero or more τ actions. To this aim, the following transition relation is used.

Definition 4. *Let* $a \in Act$. *We define the transition relation* $\stackrel{\hat{a}}{\Longrightarrow}_{\setminus H}$ *as follows:*

$$\stackrel{\hat{a}}{\Longrightarrow}_{\setminus H} = \begin{cases} \stackrel{\hat{a}}{\Longrightarrow} & \text{if } a \notin H \\ \stackrel{a}{\Longrightarrow} \ \text{ or } \ \stackrel{\hat{\tau}}{\Longrightarrow} & \text{if } a \in H \end{cases}$$

Note that the relation $\stackrel{\hat{a}}{\Longrightarrow}_{\setminus H}$ is a generalization of the relation $\stackrel{\hat{a}}{\Longrightarrow}$ used in the definition of weak bisimulation [27]. In fact, if $H = \emptyset$, then for all $a \in Act$, $E \stackrel{\hat{a}}{\Longrightarrow}_{\setminus H} E'$ coincides with $E \stackrel{\hat{a}}{\Longrightarrow} E'$.

Definition 5 (Weak Bisimulation up to H). *A binary relation* $\mathcal{R} \subseteq \mathcal{E} \times \mathcal{E}$ *over agents is a* weak bisimulation up to H *if* $(E, F) \in \mathcal{R}$ *implies, for all* $a \in Act$,

- *if* $E \stackrel{a}{\to} E'$, *then there exists* F' *such that* $F \stackrel{\hat{a}}{\Longrightarrow}_{\setminus H} F'$ *and* $(E', F') \in \mathcal{R}$;
- *if* $F \stackrel{a}{\to} F'$, *then there exists* E' *such that* $E \stackrel{\hat{a}}{\Longrightarrow}_{\setminus H} E'$ *and* $(E', F') \in \mathcal{R}$.

Two agents $E, F \in \mathcal{E}$ *are* weakly bisimilar up to H, *written* $E \approx_{\setminus H} F$, *if* $(E, F) \in \mathcal{R}$ *for some weak bisimulation* $\mathcal{R}$ *up to* H.

The relation $\approx_{\setminus H}$ is the largest weak bisimulation up to H and it is an equivalence relation. In [8] *P_BNDC* has been characterized in terms of $\approx_{\setminus H}$.

Theorem 1 (P_BNDC - Bisimulation). *Let* $E \in \mathcal{E}$. $E \in P\_BNDC$ *iff* $E \approx_{\setminus H} E \setminus H$.

In [2] we give a further characterization of *P_BNDC* processes in terms of *unwinding conditions*. This new characterization provides a better understanding of the operational semantics of *P_BNDC* processes. In practice, whenever a state E' of a *P_BNDC* process may execute a high level action moving to a state E'', then E' should be also able to simulate such high move through a τ sequence moving to a state E''' which is equivalent to E'' for a low level user.

Theorem 2 (P_BNDC - Unwinding). *Let $E \in \mathcal{E}$ be a process. $E \in$ P_BNDC iff for all E' reachable from E, if $E' \xrightarrow{h} E''$, then $E' \stackrel{\hat{\tau}}{\Longrightarrow} E'''$ and $E'' \setminus H \approx E''' \setminus H$.*

Here we observe that there is a strict relation between the bisimulation-based characterization of *P_BNDC* given in Theorem 1 and the unwinding condition of Theorem 2: the equivalence $\approx_{\setminus H}$ between E and $E \setminus H$ in Theorem 1 states that high level actions of E are simulated by zero or more τ actions of $E \setminus H$, while the unwinding condition in Theorem 2 say that for every high level action there must exists a path of zero or more τ actions leading to equivalent states from the low level view. This suggests us that consistent changes in the way of dealing with high level actions in $\approx_{\setminus H}$ and in the corresponding unwinding condition, may lead to different bisimulation-like and unwinding characterizations of novel information flow security properties.

This idea will be exploited in the next sections when we study the properties *SBNDC* and *CP_BNDC*.

In [8] it is also proved that *P_BNDC* is compositional with respect to the parallel composition, restriction and low level prefix operators. Unfortunately, *P_BNDC* is not compositional with respect to the nondeterministic choice operator as illustrated in Example 4 in the next section.

3 Strong BNDC

The property *Strong BNDC* (*SBNDC*, for short) has been introduced in [4] as a sufficient condition for verifying *BNDC*. It just requires that before and after every high step, the system appears to be the same, from a low level perspective. It has been defined through unwinding conditions as follows.

Definition 6 (SBNDC - Unwinding). *Let $E \in \mathcal{E}$. $E \in$ SBNDC iff for all E' reachable from E, if $E' \xrightarrow{h} E''$, then $E' \setminus H \approx E'' \setminus H$.*

SBNDC is *persistent* in the sense that if a process E is *SBNDC* then all processes E' reachable from E are *SBNDC*, i.e., every state reachable from a secure system is still secure. From Theorem 2 it is easy to prove the following:

Corollary 1. *SBNDC* $\subseteq$ *P_BNDC* $\subseteq$ *BNDC*.

By exploiting the relationships between the unwinding and the bisimulation characterizations discussed for the property *P_BNDC* in the previous section, we show that we can avoid the universal quantification over all the possible reachable states in the definition of *SBNDC* by defining a suitable bisimulation equivalence notion. Note that Definition 6 requires that high level actions of E are simulated by no moves, i.e. by zero τ actions, thus we define an observation equivalence, named *weak bisimulation up to H with zero* τ, where actions from H are allowed to be totally ignored, i.e., they are allowed to be matched by zero actions. To this aim, we use the following transition relation which does not take care of internal actions and may totally ignore actions from H.

Definition 7. *Let $a \in Act$. We define the transition relation $\stackrel{\hat{a}}{\Longrightarrow}{}^0_{\setminus H}$ as follows:*

$$\stackrel{\hat{a}}{\Longrightarrow}{}^0_{\setminus H} = \begin{cases} \stackrel{\hat{a}}{\Longrightarrow} & \text{if } a \notin H \\ \stackrel{a}{\Longrightarrow} \;\; or \;\; \stackrel{-}{\rightarrow} & \text{if } a \in H \end{cases}$$

where $\stackrel{-}{\rightarrow}$ denotes a sequence of zero actions [2].

Note that relation $\stackrel{\hat{a}}{\Longrightarrow}{}^0_{\setminus H}$ is included into $\stackrel{\hat{a}}{\Longrightarrow}_{\setminus H}$, introduced in Definition 4, since the empty sequence is a particular sequence of τ actions.

The concept of *weak bisimulation up to H with zero* τ is defined as follows.

Definition 8 (Weak Bisimulation up to H with zero τ). *A* weak bisimulation up to H with zero τ *is a weak bisimulation where the transition relation $\stackrel{\hat{a}}{\Longrightarrow}$ is replaced by $\stackrel{\hat{a}}{\Longrightarrow}{}^0_{\setminus H}$. Two agents $E, F \in \mathcal{E}$ are* weakly bisimilar up to H with zero τ, *written $E \approx^0_{\setminus H} F$, if $(E, F) \in \mathcal{R}$ for some weak bisimulation $\mathcal{R}$ up to H with zero τ.*

The relation $\approx^0_{\setminus H}$ is the largest weak bisimulation up to H with zero τ and it is an equivalence relation.

SBNDC processes can be characterized in terms of $\approx^0_{\setminus H}$ as follows.

Theorem 3 (SBNDC - Bisimulation). *Let $E \in \mathcal{E}$. $E \in SBNDC$ iff $E \approx^0_{\setminus H} E \setminus H$.*

Example 3. Let us consider the process depicted below, modelling the use of a shared resource by a low level *producer* and an high level *consumer*, i.e., *produce* $\in L$ and *consume* $\in H$.

$$\begin{aligned} R_0 &= produce.R_1 \\ R_i &= produce.R_{i+1} + \overline{consume}.R_{i-1} \qquad \text{for } i \in [1, n-1] \\ R_n &= produce.R_n + \overline{consume}.R_{n-1} \end{aligned}$$

Note that the resource has a maximum capacity of n and the low level *produce* action is ignored when such a limit is reached. This non-intuitive behavior is needed in order to avoid a potential flow from high to low level. In particular, if the low level producer could observe when the resource is full, this will be exploited to deduce how many high level *consume* actions have been performed.

[2] If $E \stackrel{-}{\rightarrow} E'$ then E coincides with E'.

It is easy to see that this process is *SBNDC* by directly applying Definition 6. In fact all the R_j states are equivalent when restricted on high level actions, as they may only perform a *produce* action moving to another restricted $R_{j'}$.

In [6] (see Theorem 4) it is proved that *SBNDC* is compositional with respect to the parallel and restriction operators. It is easy to extend the compositionality result by showing that *SBNDC* is also compositional with respect to low level prefix and relabelling.

Proposition 1. *Let $E, F \in \mathcal{E}$. If $E, F \in$ SBNDC, then*

- *$a.E \in$ SBNDC, for all $a \in L \cup \{\tau\}$;*
- *$(E|F) \in$ SBNDC;*
- *$E \setminus v \in$ SBNDC, for all $v \subseteq \mathcal{L}$;*
- *$E[f] \in$ SBNDC.*

As *P_BNDC* also *SBNDC* is not compositional with respect to the nondeterministic choice operator. The following example concerns *SBNDC*, but a similar reasoning can be done for *P_BNDC*.

Example 4. Consider the processes $E_4 = h.\mathbf{0}$ with $h \in H$ and $E_5 = l.\mathbf{0}$ with $l \in L$. It is easy to see that both E_4 and E_5 are *SBNDC* but $E_4 + E_5$ is not *SBNDC*. In fact $E_4 + E_5 \xrightarrow{h} \mathbf{0}$ while $E_4 + E_5 \xrightarrow{\hat{\tau}} E_4 + E_5 = h.\mathbf{0} + l.\mathbf{0}$, but $(h.\mathbf{0} + l.\mathbf{0}) \setminus H \not\approx \mathbf{0}$. The problem lies in the fact that while the high level action in E_4 is safely simulated by a sequence of zero τ in $E_4 \setminus H$, the same high level action in $E_4 + E_5$ is not safely simulated by a sequence of zero τ in $(E_4 + E_5) \setminus H$ due to the presence of the additional component E_5. This problem would not arise if h were be simulated by at least one τ action. This observation will be exploited in the next section to define a fully compositional security property.

4 Compositional P_BNDC

It is well-known that security properties are, in general, not preserved under composition [21]. We have seen in the previous sections that *P_BNDC* and *SBNDC* are both non-compositional with respect to the nondeterministic choice operator. However, compositionality results are crucial for making the development of large and complex systems feasible [23, 25, 19]. In this section we show how the notion of *P_BNDC* can be slightly restricted in order to obtain a class of processes which is *fully compositional* (i.e., it is compositional also with respect to the nondeterministic choice). We call such a class *Compositional P_BNDC* (*CP_BNDC*, for short). We also show that this class can be equivalently characterized in terms of a bisimulation-like relation and unwinding conditions.

We start by modifying the way of dealing with high level actions in the first characterization of *P_BNDC* given in terms of $\approx_{\setminus H}$. The idea is that of defining an observation equivalence, named *weak bisimulation up to H with at least one* τ, where actions from H are allowed to be matched by one or more τ actions, but not zero τ. To this aim, we use the following transition relation which generalizes the relation $\stackrel{\hat{a}}{\Longrightarrow}$. As in Definition 4, a high level move can be simulated by a sequence of τ moves, but now we require that the sequence is not empty.

Definition 9. *Let $a \in Act$. We define the transition relation $\stackrel{\hat{a}}{\Longrightarrow}{}^{+}_{\setminus H}$ as follows:*

$$\stackrel{\hat{a}}{\Longrightarrow}{}^{+}_{\setminus H} = \begin{cases} \stackrel{\hat{a}}{\Longrightarrow} & \textit{if } a \notin H \\ \stackrel{a}{\Longrightarrow} \ \textit{or} \ \stackrel{\tau}{\Longrightarrow} & \textit{if } a \in H \end{cases}$$

The concept of *weak bisimulation up to H with at least one τ* is as follows.

Definition 10 (Weak Bisimulation up to H with at least one τ). *A* weak bisimulation up to H with zero τ *is a weak bisimulation where the transition relation $\stackrel{\hat{a}}{\Longrightarrow}$ is replaced by $\stackrel{\hat{a}}{\Longrightarrow}{}^{+}_{\setminus H}$. Two agents $E, F \in \mathcal{E}$ are* weakly bisimilar up to H with at least one τ, *written $E \approx^{+}_{\setminus H} F$, if $(E, F) \in \mathcal{R}$ for some weak bisimulation $\mathcal{R}$ up to H with at least one τ.*

The relation $\approx^{+}_{\setminus H}$ is the largest weak bisimulation up to H with at least one τ and it is an equivalence relation. The relation $\stackrel{\hat{a}}{\Longrightarrow}{}^{+}_{\setminus H}$ is included in $\stackrel{\hat{a}}{\Longrightarrow}_{\setminus H}$.

The class of *CP_BNDC* processes is defined in terms of $\approx^{+}_{\setminus H}$ as follows.

Definition 11 (CP_BNDC - Bisimulation). *Let $E \in \mathcal{E}$.*

$$E \in \textit{CP\_BNDC} \quad \textit{iff} \quad E \approx^{+}_{\setminus H} E \setminus H.$$

CP_BNDC can be characterized in terms of unwinding conditions.

Theorem 4 (CP_BNDC - Unwinding). *Let $E \in \mathcal{E}$. $E \in$ CP_BNDC iff for all E' reachable from E, if $E' \stackrel{h}{\to} E''$ then $E' \stackrel{\tau}{\Longrightarrow} E'''$ and $E'' \setminus H \approx E''' \setminus H$.*

Corollary 2. *CP_BNDC $\subseteq$ P_BNDC $\subseteq$ BNDC.*

Notice that neither *SBNDC* implies *CP_BNDC* nor *CP_BNDC* implies *SBNDC*. For example, process $h.\mathbf{0}$ is *SBNDC* but it is not *CP_BNDC*, as no τ transitions simulate the high level h. On the other side, process $h.\mathbf{0} + l.\mathbf{0} + \tau.\mathbf{0}$ is *CP_BNDC* but not *SBNDC*, as, after performing h, the low level action l is no longer executable. However, there are processes which are both *SBNDC* and *CP_BNDC*, e.g., processes which perform only low level actions. The situation is summarized in Fig. 1. Notice that all the inclusions are strict.

Example 5. Consider the process C (channel) described through a value-passing extension of SPA by:

$$C = in(x).(\overline{out}(x).C + \tau.C).$$

C may accept a value x at the left-hand port, labelled in. When it holds a value, it either delivers it at the right-hand port, labelled $\overline{out}$, or resets itself performing an internal transition.

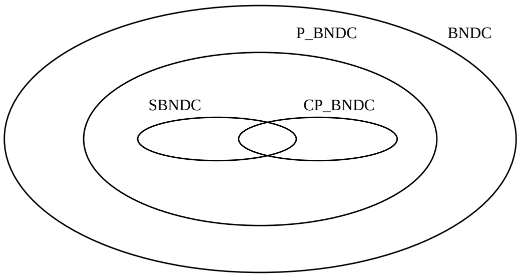

Fig. 1. Security Properties.

If the domain of x is $\{0, 1\}$, then the channel C can be translated into SPA in a standard way by following [27] as:

$$C = in_0.(\overline{out}_0.C + \tau.C) + in_1.(\overline{out}_1.C + \tau.C).$$

Let us assume that C is used as communication channel from low to high level. This can be expressed as $in_0, in_1 \in L$ and $\overline{out}_0, \overline{out}_1 \in H$. Since, in correspondence of each high level action $(\overline{out}_0, \overline{out}_1)$ there is a τ transition leading to the same state, by Theorem 4 we can conclude that C is *CP_BNDC*. The τ transitions basically makes the channel a lossy one, as high level outputs may be non-deterministically lost. However, note that non-determinism is used to abstract away implementation details. For example, such τ's could correspond, at implementation time, to time-outs for the high level output actions, i.e., events that empty the channel and allow a new low level input, whenever high outputs are not accepted within a certain amount of time. Analogously, it is possible to see that C is also *SBNDC*. Note that process $C' = in(x).\overline{out}(x).C'$ with no τ's is neither *CP_BNDC* nor *SBNDC*. Indeed, a high level user may block and unblock C' in order to transmit information to low level user. $\square$

Exploiting the unwinding characterization we are now ready to prove that *CP_BNDC* is compositional with respect to the nondeterministic choice operator.

Proposition 2. *Let $E, F \in \mathcal{E}$. If $E, F \in CP\_BNDC$, then*

- *$a.E \in CP\_BNDC$, for all $a \in L \cup \{\tau\}$;*
- *$(E + F) \in CP\_BNDC$;*
- *$(E|F) \in CP\_BNDC$;*
- *$E \setminus v \in P\_BNDC$, for all $v \subseteq \mathcal{L}$;*
- *$E[f] \in CP\_BNDC$.*

5 Verification Complexity

Let us denote with $\approx^*_{\setminus H}$ the relation $\approx_{\setminus H}$. By adopting this notation we have that a process E is *P_BNDC*, *SBNDC*, and *CP_BNDC* if and only if $E \approx^s_{\setminus H} E \setminus H$ for $s = *$, $s = 0$ and $s = +$, respectively.

The characterizations of properties in terms of bisimulation equivalences allow us to efficiently verify them. Let $n = |S_E|$ be the number of states in $LTS(E)$, for each $a \in Act$, let m_a be the number of $\stackrel{a}{\rightarrow}$ transitions in $LTS(E)$, and $m = \sum_{a \in Act} m_a$. Similarly, let $\hat{m}_a$ be the number of $\stackrel{\hat{a}}{\Longrightarrow}^{s}_{\backslash H}$ transitions, and $\hat{m} = \sum_{a \in Act} \hat{m}_a$.

Theorem 5. *Let $s \in \{0, *, +\}$. The test $E \approx^{s}_{\backslash H} E \setminus H$ can be performed in time $O(n\hat{m}_\tau + n^w + \hat{m}\log n)$ and space $O(n^2)$, where w denotes the exponent in the running time of the matrix multiplication algorithm used.*[3]

The proof of this complexity result follows exactly the lines of the proof presented in the case of *P_BNDC* in [7] paying some attention to modify the third point of the algorithm. In particular the time complexity depends on the fact that in all the cases it is necessary to compute the transitive closure of the τ-transitions. Notice that in the complexity result $\hat{m}\log n$ comes from the fact that we use the algorithm by Paige and Tarjan ([30]) to compute the maximum bisimulation.

6 Preserving Security under Refinement

In a stepwise development process, one usually starts with a very abstract specification of the desired system. The specification is then refined and decomposed until one arrives at a concrete specification that can be directly implemented. Naturally, one expects that a system which is formally developed in this way satisfies all properties that are satisfied by the abstract specification (plus possibly additional ones). While this holds for safety and liveness properties, it is not true for most information flow properties. This problem has been widely discussed in [14] and some progress toward a solution has been made in [13, 29, 31, 18]. In particular, in [18] Mantel shows how from unwinding conditions one can easily define refinement operators which preserve security.

A refinement for a process is defined in terms of a basic refinement operator $ref : \mathcal{E} \rightarrow \mathcal{E}$ that, given a process E, returns a process $ref(E)$ which is a refinement of E.

Following [18], we identify a sufficient condition to be satisfied by basic refinement operators in order to preserve the bisimulation-based possibilistic security properties studied in this paper.

Definition 12. *A basic refinement operator ref* preserves the low level observations *if for all $E, F \in \mathcal{E}$ if $E \setminus H \approx F \setminus H$, then $ref(E) \setminus H \approx ref(F) \setminus H$.*

Example 6. Let $v \subseteq \mathcal{L}$. The restriction operator $\backslash v$ is a basic refinement operator which preserves the low level observations. In fact, if $E \setminus H \approx F \setminus H$ then it is easy to prove that $(E \setminus v) \setminus H \approx (E \setminus v) \setminus H$.

[3] In the algorithm in [3], which is at the moment the fastest in literature, we have that $w = 2.376$.

Given a basic refinement operator *ref*, a refinement $\mathit{refine}(E, \mathit{ref}, S)$ for a complex system E is the process obtained by applying *ref* to all $E' \in S$ reachable from E. If E satisfies *P_BNDC* (or *CP_BNDC* or *SBNDC*) then we would like that also the resulting system satisfies it. However, by simply applying the *ref* operator to all the processes in S one may obtain a system which does not satisfy the desired property.

Example 7. Consider the process $E_6 = E_7 + h.E_8$, where $E_7 = l.h.\mathbf{0}$ and $E_8 = l.\mathbf{0}$, with $h \in H$ and $l \in L$. The process E_6 is *SBNDC*. If we consider the basic refinement operator $\backslash\{l\}$ and the set $S = \{E_8\}$ we obtain that $\mathit{refine}(E_6, \mathit{ref}, S) = l.h.\mathbf{0} + h.\mathbf{0}$ which is not *SBNDC*. The problem is due to the fact that by refining E_8 we loose the unwinding property: $\mathit{refine}(E_6, \mathit{ref}, S)$ does not contain any subprocess E' reachable with zero τ actions and such that $E' \setminus H \approx \mathit{ref}(E_8) \setminus H$. On the other hand, $\mathit{refine}(E_6, \mathit{ref}, \{E_7, E_8\}) = h.\mathbf{0}$ is *SBNDC*.

The above example suggests how to guarantee the unwinding conditions, and then our security properties, in refining a process: when we refine a subprocess E' we have to refine also all the subprocesses E'' such that $E' \setminus H \approx E'' \setminus H$.

Theorem 6. *Let $E \in \mathcal{E}$, ref be a basic refinement operator which preserves the low level observations. Let S be a set of states such that for all E', E'' reachable from E if $E' \in S$ and $E' \setminus H \approx E'' \setminus H$ then $E'' \in S$ too.*
If E satisfies P_BNDC (CP_BNDC, SBNDC) then $\mathit{refine}(E, \mathit{ref}, S)$ satisfies P_BNDC (CP_BNDC, SBNDC, respectively).

Proof. Immediate by the unwinding Theorems 2 and 4, and Definition 6.

Given an intended refinement $\mathit{refine}(E, \mathit{ref}, S)$ which does not satisfy the hypothesis on S of the above theorem, there are two natural ways for obtaining an approximation of it which preserves our security properties. We denote them by $\mathit{refine}^+(E, \mathit{ref}, S)$ and $\mathit{refine}^-(E, \mathit{ref}, S)$. While $\mathit{refine}^+(E, \mathit{ref}, S)$ refines through *ref* all the states which are in S (plus possibly states not in S), $\mathit{refine}^-(E, \mathit{ref}, S)$ only refines through *ref* states which are in S (but possibly not all states in S). The formal definition of $\mathit{refine}^+(E, \mathit{ref}, S)$ and $\mathit{refine}^-(E, \mathit{ref}, S)$ are as follows.

Definition 13 **(*refine*$^+$ and *refine*$^-$).** *Let $E \in \mathcal{E}$, let ref be a basic refinement operator which preserves the low level observations and let S be a set of states reachable from E.*

$\mathit{refine}^+(E, \mathit{ref}, S) = \mathit{refine}(E, \mathit{ref}, S \cup S')$ *where*
$\quad S' = \{E'' \text{ reachable from } E \mid \exists E' \in S \text{ and } E' \setminus H \approx E'' \setminus H\}$

$\mathit{refine}^-(E, \mathit{ref}, S) = \mathit{refine}(E, \mathit{ref}, S')$ *where*
$\quad S'$ *is the greatest subset of S such that if $E' \in S'$ and E'' is reachable from E and $E' \setminus H \approx E'' \setminus H$ then $E'' \in S$.*

If a state $E' \in S$ is refined through *ref* then $\mathit{refine}^+(E, \mathit{ref}, S)$ refines also all states E'' which are equivalent to E' from the low level view. On the other hand, $\mathit{refine}^-(E, \mathit{ref}, S)$ refines through *ref* a state $E' \in S$ only if all states E'' which are equivalent to E' from the low level view belong to S.

Corollary 3. *Let $E \in \mathcal{E}$, ref be a basic refinement operator which preserves the low level observations, and S be a set of states reachable from E. If E satisfies P_BNDC (CP_BNDC, SBNDC) then* $\mathit{refine}^{+}(E, \mathit{ref}, S)$ *and* $\mathit{refine}^{-}(E, \mathit{ref}, S)$ *both satisfy P_BNDC (CP_BNDC, SBNDC, respectively).*

7 Related Works and Conclusions

In this paper we study three persistent information flow security properties based on the bisimulation semantics model. For these properties we provide two characterizations: one in terms of a bisimulation-like equivalence relation and another one in terms of unwinding conditions.

The first characterization allows us to perform the verification of the properties for finite state processes in polynomial time with respect to the number of states of the system, also improving on the polynomial time complexity required by the Compositional Security Checker Cosec presented in [5].

The second characterization is based on unwinding conditions. This kind of conditions for possibilistic security properties have been previously proposed in many papers, see, e.g., [13, 32, 26, 17]. All such conditions have been proposed for traces-based models and are, in most cases, only sufficient for the respective security properties. Here we propose new necessary and sufficient unwinding conditions for bisimulation-based properties.

In [2] we show how unwinding conditions can be exploited for defining a proof system which provides a very efficient technique for the verification and the development of *P_BNDC* secure processes. Indeed, the proof system allows us to verify whether a process is secure just by inspecting its syntax, and thus avoiding the state-explosion problem. In particular, it allows us to deal with recursive processes which may perform unbounded sequences of actions, possibly reaching an infinite number of states. Moreover, the system offers a mean to built processes which are *P_BNDC* by construction in an incremental way. Such a proof system could be easily adapted to deal with the *CP_BNDC* and *SBNDC* properties studied in this paper.

We show that *P_BNDC* and *SBNDC* are compositional with respect to all the operators of SPA, except the non-deterministic choice. Moreover, we prove that the new property named *CP_BNDC* is fully compositional. Compositionality of possibilistic security properties has been widely studied in the literature. There are several information flow properties based on the traces model which have been proved to be fully compositional like, e.g., restrictiveness [21], forward correctability [15] or separability [23]. In [23, 25] it has been studied how to restrict composition in order to preserve certain security properties which are not preserved by (more general) composition. To the best of our knowledge, *CP_BNDC* is the only bisimulation-based security property in literature which is fully compositional.

Finally, we provide a sufficient condition to define refinement operators preserving our persistent security properties. The problem of finding refinements under which security is preserved has been widely discussed in [14] and some progress toward a solution has been made in [13, 29, 31, 18]. In particular, in [18] Mantel shows how one can easily define refinement operators which preserve security, starting from unwinding conditions. The approach we follow in this paper is indeed inspired by that work.

References

1. A. Bossi, R. Focardi, C. Piazza, and S. Rossi. Bisimulation and unwinding for verifying possibilistic security properties. Technical Report CS-2002-15, Dipartimento di Informatica, Università Ca' Foscari di Venezia, Italy, 2002. http://www.dsi.unive.it/ricerca/TR/index.htm.
2. A. Bossi, R. Focardi, C. Piazza, and S. Rossi. A Proof System for Information Flow Security. In M. Leuschel, editor, *Proc. of Int. Workshop on Logic Based Program Development and Transformation*, LNCS. Springer-Verlag, 2002. To appear.
3. D. Coppersmith and S. Winograd. Matrix multiplication via arithmetic progression. In *Proc. of the 19th Symposium on Theory of Computing*, pages 1–6, 1987.
4. R. Focardi and R. Gorrieri. A Classification of Security Properties for Process Algebras. *Journal of Computer Security*, 3(1):5–33, 1994/1995.
5. R. Focardi and R. Gorrieri. The Compositional Security Checker: A Tool for the Verification of Information Flow Security Properties. *IEEE Transactions on Software Engineering*, 23(9):550–571, 1997.
6. R. Focardi and R. Gorrieri. Classification of Security Properties (Part I: Information Flow). In R. Focardi and R. Gorrieri, editors, *Foundations of Security Analysis and Design*, volume 2171 of *LNCS*. Springer-Verlag, 2001.
7. R. Focardi, C. Piazza, and S. Rossi. Proof Methods for Bisimulation based Information Flow Security. In A. Cortesi, editor, *Proc. of Int. Workshop on Verification, Model Checking and Abstract Interpretation*, volume 2294 of *LNCS*, pages 16–31. Springer-Verlag, 2002.
8. R. Focardi and S. Rossi. Information Flow Security in Dynamic Contexts. In *Proc. of the IEEE Computer Security Foundations Workshop*, pages 307–319. IEEE Computer Society Press, 2002.
9. S. N. Foley. A Universal Theory of Information Flow. In *Proc. of the IEEE Symposium on Security and Privacy*, pages 116–122. IEEE Computer Society Press, 1987.
10. R. Forster. *Non-Interference Properties for Nondeterministic Processes*. PhD thesis, Oxford University Computing Laboratory, 1999.
11. J. A. Goguen and J. Meseguer. Security Policies and Security Models. In *Proc. of the IEEE Symposium on Security and Privacy*, pages 11–20. IEEE Computer Society Press, 1982.
12. J. A. Goguen and J. Meseguer. Inference Control and Unwinding. In *Proc. of the IEEE Symposium on Security and Privacy*, pages 75–86. IEEE Computer Society Press, 1984.
13. J. Graham-Cumming and J. W. Sanders. On the Refinement of Non-Interference. In *Proc. of the IEEE Computer Security Foundations Workshop*, pages 35–42. IEEE Computer Society Press, 1991.
14. J. Jacob. On the Derivation of Secure Components. In *Proc. of the IEEE Symposium on Security and Privacy*, pages 242–247. IEEE Computer Society Press, 1989.

15. D. M. Johnson and F. J. Thayer. Security and the Composition of Machines. In *Proc. of of the IEEE Computer Security Foundations Workshop*, pages 72–89. IEEE Computer Society Press, 1988.
16. H. Mantel. Possibilistic Definitions of Security - An Assebly Kit -. In *Proc. of the IEEE Symposium on Security and Privacy*, pages 185–199. IEEE Computer Society Press, 2000.
17. H. Mantel. Unwinding Possibilistic Security Properties. In *Proc. of the European Symposium on Research in Computer Security*, volume 2895 of *LNCS*, pages 238–254. Springer-Verlag, 2000.
18. H. Mantel. Preserving Information Flow Properties under Refinement. In *Proc. of the IEEE Symposium on Security and Privacy*, pages 78–91. IEEE Computer Society Press, 2001.
19. H. Mantel. On the Composition of Secure Systems. In *Proc. of the IEEE Symposium on Security and Privacy*, pages 88–101. IEEE Computer Society Press, 2002.
20. F. Martinelli. Partial Model Checking and Theorem Proving for Ensuring Security Properties. In *Proc. of the IEEE Computer Security Foundations Workshop*, pages 44–52. IEEE Computer Society Press, 1998.
21. D. McCullough. Specifications for Multi-Level Security and a Hook-Up Property. In *Proc. of the IEEE Symposium on Security and Privacy*, pages 161–166. IEEE Computer Society Press, 1987.
22. J. McLean. Security Models and Information Flow. In *Proc. of the IEEE Symposium on Security and Privacy*, pages 180–187. IEEE Computer Society Press, 1990.
23. J. McLean. A General Theory of Composition for Trace Sets Closed under Selective Interleaving Functions. In *Proc. of the IEEE Symposium on Security and Privacy*, pages 79–93. IEEE Computer Society Press, 1994.
24. J. McLean. Security Models. *Encyclopedia of Software Engineering*, 1994.
25. J. McLean. A General Theory of Composition for a Class of "Possibilistic" Security Properties. *IEEE Trabsactions on Software Engineering*, 22(1):53–67, 1996.
26. J. K. Millen. Unwinding Forward Correctability. In *Proc. of the IEEE Computer Security Foundations Workshop*, pages 2–10. IEEE Computer Society Press, 1994.
27. R. Milner. *Communication and Concurrency*. Prentice-Hall, 1989.
28. C. O'Halloran. A Calculus of Information Flow. In *Proc. of the European Symposium on Research in Security and Privacy*, pages 180–187. AFCET, 1990.
29. C. O'Halloran. Refinement and Confidentiality. In *Proc. of the 5th Refinement Workshop*, pages 119–139, 1992.
30. R. Paige and R. E. Tarjan. Three partition refinement algorithms. *SIAM Journal on Computing*, 16(6):973–989, 1987.
31. A. W. Roscoe, J. C. P. Woodcock, and L. Wulf. Non-Interference through Determinism. In *Proc. of the European Symposium on Research in Computer Security*, volume 875 of *LNCS*, pages 33–53. Springer-Verlag, 1994.
32. P. Y. A. Ryan. A CSP Formulation of Non-Interference and Unwinding. *Cipher*, pages 19–27, 1991.
33. S. Schneider. May Testing, Non-Interference, and Compositionality. *Electronic Notes in Theoretical Computer Science*, 40, 2000.
34. D. Sutherland. A Model of Information. In *Proc. of the 9th National Computer Security Conference*, pages 175–183, 1986.
35. A. Zakinthinos and E. S. Lee. A General Theory of Security Properties. In *Proc. of the IEEE Symposium on Security and Privacy*, pages 74–102. IEEE Computer Society Press, 1997.