Non-Interference Proof Techniques for the Analysis of Cryptographic Protocols [*]

Michele Bugliesi and Sabina Rossi

Dipartimento di Informatica,
Università Ca' Foscari di Venezia
via Torino 155, 30172 Venezia, Italy
{bugliesi,srossi}@dsi.unive.it

Abstract

Non-interference has been advocated by various authors as a uniform framework for the formal specification of security properties in cryptographic protocols. Unfortunately, specifications based on non-interference are often non-effective, as they require protocol analyses in the presence of *all* possible intruders.

This paper develops new characterizations of non-interference that rely on a finitary representation of intruders. These characterizations draw on equivalence relations built on top of labelled transition systems in which the presence of intruders is accounted for, indirectly, in terms of their (the intruders') knowledge of the protocols' initial data. The new characterizations apply uniformly to trace and bisimulation non-interference, yielding proof techniques for the analysis of various security properties. We demonstrate the effectiveness of such techniques in the analysis of different properties of a fair exchange protocol.

1 Introduction

Non-interference has been advocated by various authors [1, 14, 15] as a powerful method for the analysis of cryptographic protocols. In [14, 16], Focardi *et al.* propose a general schema for specifying security properties with a uniform and concise definition. The approach draws on earlier work by the same authors on characterizing information-flow security in terms of Non-Interference.

Informally, the idea is that a system is secure if what a low-level user sees of the system does not change when the system is composed with any high-level component. In [13] these ideas are formalized in the the Security Process Algebra (SPA, for short), a variant of CCS in which the set of actions is partitioned into two sets: L, for low, and

[*] This work has been partially supported by the MIUR project "Modelli formali per la sicurezza (MEFISTO)" and the EU project IST-2001-32617 "Models and types for security in mobile distributed systems (MyThS)".

H for high. In this context, a Non-Interference property NI for a process P is expressed as follows:

$$P \in \mathsf{NI} \text{ if } (P||\Pi) \setminus H \approx_{\mathsf{NI}} P \setminus H,\ \forall \Pi \in \mathcal{P}_H \qquad (1)$$

Here $\mathcal{P}_H$ is the class of high-level processes (i.e., built around actions in H), $\approx_{\mathsf{NI}}$ is an observation equivalence (parametric in the property NI), while $||$ and $\setminus$ denote parallel composition and restriction. The processes $P \setminus H$ and $(P||\Pi) \setminus H$ represent the low-level views of P and of $P||\Pi$, respectively, and property (1) above formalize the intuition that: "If no high-level process can change the low behavior of the system, then no flow of information from high to low is possible".

In [14] Non-Interference is employed to provide a general definition of security properties for cryptographic protocols described as terms of CryptoSPA, a process algebra that extends SPA with cryptographic primitives. Here the idea is to view the participants in a protocol as low-level processes, and to represent the possible external attackers as high-level processes. Then, Non-Interference implies that the attackers have no way to change the low (honest) behavior of the protocol.

There are two aspects of this idea that need to be addressed to formalize it in full. First, the intruder should be assumed to have complete control over the public components of the network. Consequently, any step in a protocol involving a public channel should be classified as a high-level action. However, since a protocol specification is determined by the exchange of messages over public channels, a characterization like (1) becomes trivial, as $(P||\Pi) \setminus H$ and $P \setminus H$ are simply the null processes. In [14] this is rectified by extending the CryptoSPA specification of a protocol with low-level actions associated with certain steps in the protocol execution, and by formulating the desired security property in terms of the observation of those actions.

A further problem in applying Non-Interference to protocol analysis arises from the need to formalize the import of the cryptographic primitives in protecting the protocol participants from the intruders (dually, this corresponds to formalizing the power of the intruders). In [14] this is achieved by making the definition of Non-Interference dependent (i) on the initial data in control of the attacker, and (ii) on an inference system which expresses the ability of the attacked to compute new data. Typically, the initial data include the attacker's private keys, as well as any piece of publicly available information, such as names of entities and public keys. Given the initial data, the power of the attacker is completely characterized by its ability to compute new data (messages and keys) by means of the rules of the inference system. If we denote the initial knowledge with ϕ, we can reformulate property (1) for a protocol P as follows:

$$P \in \mathsf{NI} \text{ if } (P||\Pi) \setminus H \approx_{\mathsf{NI}} P \setminus H,\ \forall \Pi \in \mathcal{P}_H^{\phi} \qquad (2)$$

where $\mathcal{P}_H^{\phi}$ is the set of the high-level processes Π which can perform only actions using the public channel names and whose messages (those syntactically appearing in Π) can be deduced from ϕ. The term $P \setminus H$ represents the secure specification of the protocol P running in isolation on perfectly secure channels: the visible behaviour of P is given by the low actions included in the specification to characterize the security property of interest. If $P \setminus H$ is equivalent to $(P||\Pi) \setminus H$ then we have a guarantee that Π is not able to modify in any way the observable execution of P, i.e., the security property holds.

This framework is very general, and lends itself to the characterization of various security properties, obtained by integrating the protocol specification with suitable low-level actions and instantiating the equivalence $\approx_{\mathsf{NI}}$ in the schema above. Instead, this framework is less effective as a proof method, due to the universal quantification over all the possible intruders Π in the class $\mathcal{P}_H^{\phi}$. In [14], the problem is circumvented by analyzing the protocol in presence of the "hardest attacker". However, this characterization is proved correct only for the class of relationships $\approx_{\mathsf{NI}}$ that are behavioral preorders on processes. In particular, the proof method is not applicable for equivalences based on bisimulation.

In this paper we partially rectify the problem by developing a technique which does not require us to exhibit an explicit attacker (nor, in particular, it requires the existence of a hardest attacker). Our approach draws on ideas from [6] to represent the attacker indirectly, in terms of a context-sensitive labelled transition system. In our approach, the labelled transitions take the form

$$\phi \triangleright P \xrightarrow{a} \phi' \triangleright P'$$

where ϕ represents the context's knowledge prior to the transition, and ϕ' is the new knowledge resulting from P performing the action a. Building on this labelled transition system, we provide quantification-free characterizations for different instantiations of (2), specifically when $\approx_{\mathsf{NI}}$ is instantiated to trace equivalence, and to weak bisimulation. This allows us to apply our technique to the analysis of safety properties, e.g., secrecy, authentication and integrity, as well as failure sensitive properties such as fairness and non-repudiation. We demonstrate the latter with a protocol of *fair exchange*: In particular, we apply our method to the ASW contract signing protocol [2], whose applications include home banking and electronic commerce.

Plan of the paper Section 2 gives a brief review of the process calculus CryptoSPA. Section 3 introduces context-sensitive labelled transition systems. Section 4 gives characterizations for various security properties. Section 5 develops our case study and and Section 6 draws some conclusions.

A preliminary version of this paper appeared in [9].

2 The Calculus CryptoSPA

The *Cryptographic Security Process Algebra* (CryptoSPA, for short) [16] is an extension of SPA [13] with cryptographic primitives and constructs for value passing. This section provides a brief overview of the syntax and semantics of the calculus, based on [16], to which we refer the interested reader for full details.

We presuppose a set *Const* of constants, ranged over by capital letters $A, B, \ldots$, and a set $\mathcal{C}$ of channels, partitioned into two sets H and L of high and low channels, respectively. A *sort* function *Msg* maps every channel into the set of messages that can legally be transmitted over that channel. We let $a, b, c, \ldots$ and $\overline{a}, \overline{b}, \overline{c}, \ldots$ range over input and output channels, respectively, and assume $Msg(c) = Msg(\overline{c})$ for all $c \in \mathcal{C}$.

Messages, ranged over by m, form a set $\mathcal{M}$ built around two further sets M, of basic messages and K of encryption keys, and closed under pairing (m,m') and encryption $\{m\}_k$. We also presuppose a function $\cdot^{-1}: K \longrightarrow K$ such that $(k^{-1})^{-1} = k$, for all $k \in K$.

The syntax of CryptoSPA *terms* (or *processes*) is defined as the following extension of value-passing CCS:

$$\begin{aligned} P ::= \quad & \mathbf{0} \mid c(x).P \mid \overline{c}m.P \mid \tau.P \mid P+P \mid P||P \mid P\setminus C \mid P[f] \mid \\ & \mid A(m_1,...,m_n) \mid [m=m']P;P \mid [\langle m_1...m_n\rangle \vdash_{rule} x]P;P \end{aligned}$$

Both $c(x).P$ and $[\langle m_1...m_n\rangle \vdash_{rule} x]P;P'$ bind the variable x in P. Constants are defined by equations of the form $A(x_1,...,x_n) \overset{def}{=} P$, where P is a process that may contain no free variables except $x_1,\ldots,x_n$, which must be pairwise distinct. We write $c.P$ and $\overline{c}.P$ in place of $c(x).P$ and $\overline{c}m.P$, respectively, whenever the messages exchanged on c is irrelevant.

The reading of most of the constructs is standard. $\mathbf{0}$ is the empty process, which does nothing; $c(x).P$ waits for input m on channel c, and then behaves as $P[m/x]$ (i.e., P with all the free occurrences of x substituted by m); $\overline{c}m.P$ outputs m on channel c and continues as P; $\tau.P$ performs the internal action τ and continues as P; P_1+P_2 represents the nondeterministic choice between P_1 and P_2; $P_1||P_2$ is parallel composition, where the executions of P_1 and P_2 are interleaved, possibly synchronized on complementary input/output actions, producing an internal action τ; $P\setminus C$ behaves like process P but the restriction $\setminus C$ makes P's exchanges over the channels in C invisible to the context; $P[f]$ is like P with every channel c relabelled into $f(c)$; $A(m_1,...,m_n)$ behaves like its defining process with the variables $x_1,\cdots,x_n$ substituted with messages $m_1,\cdots,m_n$; $[m=m']P_1;P_2$ behaves like P_1 if $m=m'$ and lie P_2 otherwise; finally, $[\langle m_1...m_n\rangle \vdash_{rule} x]P_1;P_2$ behaves like $P_1[z/x]$ for all z derivable from the set of messages $m_1...m_n$ by an application of rule $\vdash_{rule}$; If no such z exists, it behaves like P_2.

To ease the notation, we write $[m=m']P_1$ and $[\langle m_1...m_n\rangle \vdash_{rule} x]P_1$ instead of $[m=m']P_1;\mathbf{0}$ and $[\langle m_1...m_n\rangle \vdash_{rule} x]P_1;\mathbf{0}$, respectively. We denote $\mathcal{P}$ the set of all CryptoSPA processes and by $\mathcal{P}_H$ the set of all high-level processes, i.e., those constructed only using actions in $H\cup\{\tau\}$.

The operational semantics of CryptoSPA is defined in terms of labelled transitions of the form $P \xrightarrow{a} P'$ where P and P' are processes and a is an action in the set $Act = \mathcal{L}\cup\{\tau\}$. Here τ is the internal, (invisible, or silent) action, while $\mathcal{L}$ is the set of visible actions defined as $\mathcal{L} = \{c(m) \mid m \in Msg(c)\} \cup \{\overline{c}m \mid m \in Msg(c)\}$. We presuppose a function $chan(a)$ which returns c if a is either $c(m)$ or $\overline{c}m$ and the special channel *void* when $a=\tau$. By an abuse of notation, we write $c(m),\overline{c}m \in H$ whenever $c,\overline{c}\in H$, and similarly for L. Also, we often abbreviate $c(m)$ and $\overline{c}m$ to c and $\overline{c}$, respectively, when m can safely be disregarded.

The labelled transition system is defined in Figure 2. Most of the transitions are standard, and formalize the intuitive semantics of the process constructs discussed above. The two rules $(\vdash_i)$ connect the labelled transition system with the inference system in Fig. 1. As we mentioned in the Introduction, the inference system models the ability of the attacker to compute new information from its initial knowledge.

In particular, the system in Fig. 1, implements the so called "perfect cryptography" assumption, whereby an intruder may encrypt and decrypt messages, but only using cryptographic keys in control of the intruder itself. As a consequence, as in [14, 16], we disregard cryptographic attacks, based on the ability to guess, or break, secret keys. We say that m is *deducible* from a set of messages ϕ (and write $\phi \vdash m$) if m can be obtained from ϕ by applying the inference rules in Fig. 1.

$$\frac{m \quad m'}{(m,m')}\ (\vdash_{pair}) \qquad \frac{(m,m')}{m}\ (\vdash_{fst}) \qquad \frac{(m,m')}{m'}\ (\vdash_{snd})$$

$$\frac{m \quad k}{\{m\}_k}\ (\vdash_{enc}) \qquad \frac{\{m\}_k \quad k^{-1}}{m}\ (\vdash_{dec})$$

Figure 1: Inference system for messages: $m, m' \in \mathcal{M}$ and $k, k^{-1} \in K$

We complement the definition of the semantics with corresponding notions of *observation equivalence*, used to establish equalities among processes and based on the idea that two systems have the same semantics if and only if they cannot be distinguished by an external observer. The equivalences that are relevant to the present discussion are *trace equivalence*, denoted by $\simeq$, and *weak bisimulation*, denoted by $\approx$. We recall them below.

Let us first introduce the following auxiliary notations. We denote by $P \overset{a}{\Longrightarrow} P'$ the sequence of transitions $P(\xrightarrow{\tau})^* P_1 \xrightarrow{a} P_2(\xrightarrow{\tau})^* P'$. Moreover, let $\gamma = a_1 \dots a_n \in \mathcal{L}^*$ be a sequence of (non silent) actions; we write $P \overset{\gamma}{\Longrightarrow} P'$ if there are $P_1, P_2, \dots, P_{n-1} \in \mathcal{P}$ such that $P \overset{a_1}{\Longrightarrow} P_1 \overset{a_2}{\Longrightarrow} \dots \overset{a_{n-1}}{\Longrightarrow} P_{n-1} \overset{a_n}{\Longrightarrow} P'$. The notation $P \overset{\hat{a}}{\Longrightarrow} P'$ stands for $P \overset{a}{\Longrightarrow} P'$ if $a \in \mathcal{L}$ and for $P\ (\xrightarrow{\tau})^*\ P'$ if $a = \tau$ (note that $\overset{\tau}{\Longrightarrow}$ requires at least one τ labelled transition while $\overset{\hat{\tau}}{\Longrightarrow}$ means zero or more τ labelled transitions).

The relation of *trace equivalence* [11] equates two processes if they have the same sets of traces, disregarding the τ actions.

Definition 2.1 (Trace Equivalence).

- $T(P) = \{\gamma \in \mathcal{L}^* \mid \exists P' : P \overset{\gamma}{\Longrightarrow} P'\}$ is the set of *traces* associated with process P.
- Two processes $P, Q \in \mathcal{P}$ are *trace equivalent*, noted $P \simeq Q$, if $T(P) = T(Q)$.

$\square$

The *weak bisimulation* relation [20] equates two processes if they are able to mutually simulate each other's behavior step by step (modulo τ transitions).

Definition 2.2 (Weak Bisimilarity). A binary relation $\mathcal{R} \subseteq \mathcal{P} \times \mathcal{P}$ over processes is a *weak bisimulation* if $(P, Q) \in \mathcal{R}$ implies, for all $a \in Act$,

$$\frac{m \in Msg(c)}{c(x).P \xrightarrow{c(m)} P[m/x]}\ (input) \qquad \frac{m \in Msg(c)}{\overline{c}m.P \xrightarrow{\overline{c}(m)} P}\ (output)$$

$$\frac{}{\tau.P \xrightarrow{\tau} P}\ (tau) \qquad \frac{P \xrightarrow{\overline{c}(m)} P' \quad Q \xrightarrow{c(m)} Q'}{P||Q \xrightarrow{\tau} P'||Q',\ Q||P \xrightarrow{\tau} Q'||P'}\ (synch)$$

$$\frac{P \xrightarrow{a} P'}{P+Q \xrightarrow{a} P',\ Q+P \xrightarrow{a} P'}\ (+) \qquad \frac{P \xrightarrow{a} P'}{P||Q \xrightarrow{a} P'||Q,\ Q||P \xrightarrow{a} Q||P'}\ (||)$$

$$\frac{P_2 \xrightarrow{a} P_2' \quad m \neq m'}{[m=m']P_1;P_2 \xrightarrow{a} P_2'}\ (\neq) \qquad \frac{P_1 \xrightarrow{a} P_1'}{[m=m]P_1;P_2 \xrightarrow{a} P_1'}\ (=)$$

$$\frac{P \xrightarrow{a} P'}{P[f] \xrightarrow{f(a)} P'[f]}\ ([f]) \qquad \frac{P \xrightarrow{a} P' \quad chan(a) \notin C}{P \setminus C \xrightarrow{a} P' \setminus C}\ (\backslash C)$$

$$\frac{P[m_1/x_1,\ldots,m_n/x_n] \xrightarrow{a} P' \quad A(x_1,\ldots,x_n) \overset{def}{=} P}{A(m_1,\ldots,m_n) \xrightarrow{a} P'}\ (constant)$$

$$\frac{m_1,\ldots,m_n \vdash_{rule} m \quad P_1[m/x] \xrightarrow{a} P_1'}{[\langle m_1,\ldots,m_n\rangle \vdash_{rule} x]P_1;P_2 \xrightarrow{a} P_1'}\ (\vdash_1)$$

$$\frac{\nexists m : m_1,\ldots,m_n \vdash_{rule} m \quad P_2 \xrightarrow{a} P_2'}{[\langle m_1,\ldots,m_n\rangle \vdash_{rule} x]P_1;P_2 \xrightarrow{a} P_2'}\ (\vdash_2)$$

Figure 2: The operational rules for CryptoSPA

- if $P \xrightarrow{a} P'$, then there exists Q' such that $Q \xRightarrow{\hat{a}} Q'$ and $(P',Q') \in \mathcal{R}$;

- if $Q \xrightarrow{a} Q'$, then there exists P' such that $P \xRightarrow{\hat{a}} P'$ and $(P',Q') \in \mathcal{R}$.

Two processes $P,Q \in \mathcal{P}$ are *weakly bisimilar*, denoted by $P \approx Q$, if there exists a weak bisimulation $\mathcal{R}$ containing the pair (P,Q). Weak bisimilarity, noted $\approx$, is the largest weak bisimulation (and it is an equivalence relation). □

Trace equivalence is less demanding than weak bisimulation, hence if two processes are weak bisimilar, then they are also trace equivalent.

In the next section, we introduce coarser versions of these equivalences, denoted by $\simeq^{\phi}$ and $\approx^{\phi}$, which distinguish processes in contexts with initial knowledge ϕ.

$$\frac{P \xrightarrow{\overline{c}m} P' \quad \overline{c}m \in H}{\phi \triangleright P \xrightarrow{\overline{c}m} \phi \cup \{m\} \triangleright P'}\ (\mathit{output}) \qquad \frac{P \xrightarrow{c(m)} P' \quad \phi \vdash m \quad c(m) \in H}{\phi \triangleright P \xrightarrow{c(m)} \phi \triangleright P'}\ (\mathit{input})$$

$$\frac{P \xrightarrow{\tau} P'}{\phi \triangleright P \xrightarrow{\tau} \phi \triangleright P'}\ (\mathit{tau}) \qquad \frac{P \xrightarrow{a} P' \quad a \in L}{\phi \triangleright P \xrightarrow{a} \phi \triangleright P'}\ (\mathit{low})$$

Figure 3: Labelled transitions for configurations

3 Context-Sensitive Equivalences

Following [6], we characterize the behavior of processes in terms of a refined version of labelled transitions where each process transition depends on the knowledge of the context. To motivate, consider a process P that produces and sends a message $\{m\}_k$ reaching the state P', and assume that m and k are known to P but not to the context. Under these hypotheses, the context will never be able to reply the message m to P' or its continuation, unless, of course, they send m in clear. Hence, if P' waits for further input, we can safely leave any input transition involving m out of the LTS, as the P' will never receive m from the context.

The states of the new labelled transition system are *configurations* $\phi \triangleright P$, where P is a process and ϕ is the current knowledge of the context, represented as a set of messages. The transitions represent the possible interactions between the process and the context and now take the form $\phi \triangleright P \xrightarrow{a} \phi' \triangleright P'$, expressing the fact that the process P running in a context with a knowledge ϕ may execute an action a reaching a process P' and ϕ' is the new knowledge at disposal to the context for further interactions with P'.

The transitions between configurations, in Fig. 3, are defined rather directly from the corresponding transitions between processes. They formalize the intuitions we gave earlier on how to represent the possible interactions between the process and an attacker with knowledge ϕ. Specifically, in rule (*output*), the process performs a high-level output while the context performs an input; correspondingly, the context's knowledge is augmented with the information sent by the process. Dually, rule (*input*) assumes that the context performs an output action synchronizing with the input of the process. The message sent by the context must be deducible from the context's knowledge ϕ, otherwise the corresponding transition is impossible. The remaining rules, (*tau*) and (*low*), state that the internal actions of the protocol, and the low actions do not contribute to the knowledge of the context in any way.

In the rest of the presentation, we refer to the transition rules in Fig. 3 collectively as the *enriched LTS* (*ELTS*, for short). The notation for weak action and sequence of actions extends directly to configurations. In particular, we write $\phi \triangleright P \overset{a}{\Longrightarrow} \phi' \triangleright P'$ to denote the sequence of transitions $\phi \triangleright P\ (\xrightarrow{\tau})^* \ \phi \triangleright P_1 \ \xrightarrow{a}\ \phi' \triangleright P_2\ (\xrightarrow{\tau})^*\ \phi' \triangleright P'$, where, as expected, $\phi = \phi'$ if $\xrightarrow{a}$ is an input, low or silent action. Furthermore, let

$\gamma = a_1 \dots a_n \in \mathcal{L}^*$ be a sequence of (non silent) actions; then $\phi \rhd P \stackrel{\gamma}{\Longrightarrow} \phi' \rhd P'$ if there are $P_1, P_2, \dots, P_{n-1} \in \mathcal{P}$ and $\phi_1, \phi_2, \dots, \phi_{n-1}$ states such that $\phi \rhd P \stackrel{a_1}{\Longrightarrow} \phi_1 \rhd P_1 \stackrel{a_2}{\Longrightarrow} \dots \stackrel{a_{n-1}}{\Longrightarrow} \phi_{n-1} \rhd P_{n-1} \stackrel{a_n}{\Longrightarrow} \phi' \rhd P'$. The notation $\phi \rhd P \stackrel{\hat{a}}{\Longrightarrow} \phi' \rhd P'$ stands for $\phi \rhd P \stackrel{a}{\Longrightarrow} \phi' \rhd P'$ if $a \in \mathcal{L}$ and for $\phi \rhd P \ (\stackrel{\tau}{\longrightarrow})^* \ \phi' \rhd P'$ if $a = \tau$, as usual.

The notions of traces, trace equivalence and weak bisimilarity for configurations arise as expected:

Definition 3.1 (Trace Equivalence over configurations).

- $T(\phi \rhd P) = \{\gamma \in \mathcal{L}^* \mid \exists \phi', P' : \phi \rhd P \stackrel{\gamma}{\Longrightarrow} \phi' \rhd P'\}$ is the set of *traces* associated with the configuration $\phi \rhd P$.
- Two configurations are *trace equivalent*, denoted by $\phi_P \rhd P \simeq^c \phi_Q \rhd Q$, iff $T(\phi_P \rhd P) = T(\phi_Q \rhd Q)$.

□

Definition 3.2 (Weak Bisimilarity over configurations). A binary relation $\mathcal{R}$ over configurations is a weak bisimulation if, whenever $(\phi_P \rhd P, \phi_Q \rhd Q) \in \mathcal{R}$, one has, for all $a \in Act$:

- if $\phi_P \rhd P \stackrel{a}{\longrightarrow} \phi_{P'} \rhd P'$, then there exists a configuration $\phi_{Q'} \rhd Q'$ such that $\phi_Q \rhd Q \stackrel{\hat{a}}{\Longrightarrow} \phi_{Q'} \rhd Q'$ and $(\phi_{P'} \rhd P', \phi_{Q'} \rhd Q') \in \mathcal{R}$;
- if $\phi_Q \rhd Q \stackrel{a}{\longrightarrow} \phi_{Q'} \rhd Q'$ then there exists a configuration $\phi_{P'} \rhd P'$ such that $\phi_P \rhd P \stackrel{\hat{a}}{\Longrightarrow} \phi_{P'} \rhd P'$ and $(\phi_{P'} \rhd P', \phi_{Q'} \rhd Q') \in \mathcal{R}$.

Two configurations $\phi_P \rhd P$ and $\phi_Q \rhd Q$ are *weakly bisimilar*, written $\phi_P \rhd P \approx^c \phi_Q \rhd Q$, if there exists a weak bisimulation containing the pair $(\phi_P \rhd P, \phi_Q \rhd Q)$. □

By construction, $\approx^c$ is the largest weak bisimulation over configurations, and it is easy to prove that is an equivalence relation. As for trace equivalence, we can recover an equivalence relation on processes executing in a context with initial knowledge ϕ.

3.1 Equivalences under ϕ

Now we may define corresponding notions of process equivalence, over processes executing in an environment with initial knowledge ϕ.

Definition 3.3 (Trace equivalence under ϕ). Two processes P and Q are trace equivalent under ϕ, noted $P \simeq^\phi Q$, iff $\phi \rhd P \simeq^c \phi \rhd P$. □

Below, we show that $\simeq^\phi$ is strictly coarser than $\simeq$: intuitively, this follows by observing that whenever the initial contexts of P and Q share the same knowledge, then they evolve in the same way: the execution of any trace leads to contexts again equal. We first prove two useful simple lemmas. The first relates transitions and weak transitions over configurations, the second relate process transitions and configuration transitions.

Lemma 3.4. *Assume $\phi \rhd P \xrightarrow{a} \phi' \rhd P'$. If $\phi \rhd P \xRightarrow{a} \phi'' \rhd P''$ for some P'' then $\phi' = \phi''$.* □

Proof. The proof follows from the fact that $\phi \rhd P(\xrightarrow{\tau})^* \phi' \rhd P'$ implies $\phi = \phi'$, i.e. the knowledge component of configurations is invariant through τ-transitions. This follows directly by an inspection of the (*tau*) rule in Fig. 3 and by induction on the length of the derivation $\phi \rhd P(\xrightarrow{\tau})^* \phi' \rhd P'$. □

Lemma 3.5. *Assume $\phi \rhd P \xRightarrow{a} \phi' \rhd P'$ and $Q \xRightarrow{a} Q'$. Then also $\phi \rhd Q \xRightarrow{a} \phi' \rhd Q'$.* □

Proof. First observe that $\phi \rhd P \xrightarrow{a} \phi' \rhd P'$ and $Q \xrightarrow{a} Q'$ imply $\phi \rhd Q \xrightarrow{a} \phi' \rhd Q'$. This follows by a case analysis on the possible shapes of a, and an inspection of the transition rules in Fig. 3. Then the proof follows by Lemma 3.4. □

Proposition 3.6. *If $P \simeq Q$ then $P \simeq^{\phi} Q$.* □

Proof. Let $\gamma \in T(\phi \rhd P)$. By definition there exist ϕ' and P' such that $\phi \rhd P \xRightarrow{\gamma} \phi' \rhd P'$. An inspection of the transition rules in Fig. 3 shows that $P \xRightarrow{\gamma} P'$, hence $\gamma \in T(P)$. By the hypothesis $P \simeq Q$, we know that $\gamma \in T(Q)$. Thus, $Q \xRightarrow{\gamma} Q'$ for some Q'. We prove, by induction on the length of γ, that $\phi \rhd Q \xRightarrow{\gamma} \phi' \rhd Q'$ holds, i.e., $\gamma \in T(\phi \rhd Q)$.

- *Base.* If γ is the empty trace then the proof is trivial.
- *Induction step.* Let γ be a non-empty trace of the form $\gamma' a$. Then there exist P'', Q'' and ϕ'' such that $\phi \rhd P \xRightarrow{\gamma'} \phi'' \rhd P'' \xRightarrow{a} \phi' \rhd P'$, and also $P \xRightarrow{\gamma'} P'' \xRightarrow{a} P'$ and $Q \xRightarrow{\gamma'} Q'' \xRightarrow{a} Q'$. By the induction hypothesis, we know that $\phi \rhd Q \xRightarrow{\gamma'} \phi'' \rhd Q''$. To conclude, we need to show that $\phi'' \rhd Q'' \xRightarrow{a} \phi' \rhd Q'$, which follows by Lemma 3.5 from from $Q'' \xRightarrow{a} Q'$ and $\phi'' \rhd P'' \xRightarrow{a} \phi' \rhd P'$.

This shows that $T(\phi \rhd P) \subseteq T(\phi \rhd Q)$. The $T(\phi \rhd P) \supseteq T(\phi \rhd Q)$ inclusion is proved exactly in the same way. □

To see that $\simeq^{\phi}$ is strictly coarser than $\simeq$, consider $P \overset{def}{=} l_1.h(x).[x = k]l_2.\mathbf{0}$ and $Q \overset{def}{=} l_1.h(x).\mathbf{0}$, and observe that $P \simeq^{\phi} Q$ for all ϕ such that $\phi \not\vdash k$. In fact, the only transition from $\phi \rhd P$ is to the configuration $\phi \rhd h(x).[x = k]l_2.\mathbf{0}$. Now, since $\phi \not\vdash k$, all further transitions from the latter configurations lead to new configurations of the form $\phi \rhd [m = k]l_2.\mathbf{0}$ with $m \neq k$, which are deadlocked. Exactly the same transitions are available from $\phi \rhd Q$. On the other hand, if we disregard the initial knowledge ϕ, $l_1.h(k).l_2$ a trace in $T(P)$, which is not part of $T(Q)$.

Next, we introduce a knowledge dependent notion of labelled bisimilarity. The construction mimics the construction of $\simeq^{\phi}$ from $\simeq$.

Definition 3.7 (Weak bisimilarity under ϕ). Two processes P and Q are *weakly ϕ-bisimilar*, noted $P \approx^{\phi} Q$, if $\phi \rhd P \approx^{c} \phi \rhd Q$. □

Proposition 3.8. *If $P \approx Q$ then $P \approx^{\phi} Q$.* □

Proof. It is sufficient to show that

$$\mathcal{R} \ = \ \{(\phi \rhd P,\ \phi \rhd Q) \ \mid \ P \approx Q \text{ and } \phi \text{ is a set of messages}\}$$

is a weak bisimulation over configurations. Take $(\phi \rhd P,\ \phi \rhd Q) \in \mathcal{R}$ and let $\phi \rhd P \xrightarrow{a} \phi' \rhd P'$. Then $P \xrightarrow{a} P'$. Since $P \approx Q$, we know that there exists Q' such that $Q \overset{\hat{a}}{\Longrightarrow} Q'$ and $P' \approx Q'$. Then, by Lemma 3.5, we have $\phi \rhd Q \overset{\hat{a}}{\Longrightarrow} \phi' \rhd Q'$, which implies $(\phi' \rhd P',\ \phi' \rhd Q') \in \mathcal{R}$ as desired. □

For future reference, we note that for any process P which executes only low or internal actions, each ELTS of P coincides (up to isomorphism) with the unique LTS of P itself. Let then $\mathcal{P}_L = \{P \mid T(P) \subseteq L^*\}$ be the class of processes that only exhibit low or internal actions.

Proposition 3.9. *If $P \in \mathcal{P}_L$ then $T(P) = T(\phi \rhd P)$ for all ϕ.* □

As a consequence, the relations $\simeq$ and $\simeq^\phi$, and similarly the relations $\approx$ and $\approx^\phi$, coincide for the class of processes which only execute low or internal actions.

Proposition 3.10. *Let $P, Q \in \mathcal{P}_L$. For all ϕ, $P \simeq Q$ iff $P \simeq^\phi Q$ and $P \approx Q$ iff $P \approx^\phi Q$.* □

4 Non-Interference Proof Techniques

We show that the new definitions of behavioral equivalence may be used to construct effective proof methods for various security properties within the general schema proposed in [14, 16]. In particular, we show that making our equivalences dependent on the initial knowledge of the attacker provides us with security characterizations that are stated independently from the attacker itself.

We note $\mathcal{P}_H^\phi$ the (infinite) set of high-level processes build around messages that are deducible from ϕ. $\mathcal{P}_H^\phi$ represents the set of all possible attackers, which have only access to the public (high) channel names, and whose output messages may be formed starting from the initial data ϕ.

4.1 A characterization of *NDC*$^\phi$

The first property we study, known as NDC, results from instantiating $\approx_{\mathsf{NI}}$ in (2) (see the introduction) to the trace equivalence relation $\simeq$. As discussed in [14, 16], NDC is a generalization of the classical idea of Non-Interference to non-deterministic systems. Property NDC can readily be extended to account for the context's knowledge as suggested in [16], namely:

Definition 4.1 (NDC$^\phi$). $P \in NDC^\phi$ if $P \setminus H \simeq (P||\Pi) \setminus H,\ \forall\, \Pi \in \mathcal{P}_H^\phi$. □

A process P is NDC^ϕ if for every high-level process Π with initial knowledge ϕ a low-level user cannot distinguish P from $(P||\Pi)$, i.e., if Π cannot interfere with the low-level execution of the process P.

Focardi *et al.* in [16] show that when ϕ is finite it is possible to find a most general intruder Top^{ϕ} so that verifying NDC^{ϕ} reduces to checking $P \setminus H \simeq (P || Top^{\phi}) \setminus H$. Here we provide an alternative[1], quantification-free characterization of NDC^{ϕ}. This is based on the following notion of trace equivalence over configurations "up to high-level actions".

Definition 4.2 (Trace equivalence under ϕ up to H).

- $T(\phi \rhd P)/H = \{\gamma' \in \mathcal{L}^* \mid \exists \phi', P' : \phi \rhd P \overset{\gamma}{\Longrightarrow} \phi' \rhd P'$ and γ' is obtained from γ by deleting all high-level actions $\}$.
- Two configurations $\phi_P \rhd P$ and $\phi_Q \rhd Q$ are *trace equivalent up to H*, denoted by $\phi_P \rhd P \simeq^{c}_{/H} \phi_Q \rhd Q$, if $T(\phi_P \rhd P)/H = T(\phi_Q \rhd Q)/H$.

We then define a corresponding notion of process equivalence, for processes executing in an environment with initial knowledge ϕ. P and Q are trace equivalent under ϕ up to H, noted $P \simeq^{\phi}_{/H} Q$, if $\phi \rhd P \simeq^{c}_{/H} \phi \rhd Q$. □

Theorem 4.3 (NDC$^{\phi}$). *$P \in NDC^{\phi}$ if and only if $P \setminus H \simeq^{\phi}_{/H} P$.* □

Proof. We first prove that, for all ϕ,

$$T((P||\Pi) \setminus H) = T(\phi \rhd P)/H,\ \forall \Pi \in \mathcal{P}^{\phi}_{H} \quad \text{iff} \quad T(P \setminus H) = T(\phi \rhd P)/H \tag{3}$$

($\Rightarrow$) If $T((P||\Pi) \setminus H) = T(\phi \rhd P)/H$ for all $\Pi \in \mathcal{P}^{\phi}_{H}$, this holds in particular for $\Pi = \mathbf{0}$; hence, since $T((P||\mathbf{0}) \setminus H) = T(P \setminus H)$, we obtain that $T(P \setminus H) = T(\phi \rhd P)/H$.

($\Leftarrow$) Assume $T(P \setminus H) = T(\phi \rhd P)/H$ for all $\Pi \in \mathcal{P}^{\phi}_{H}$. Since $T(P \setminus H) \subseteq T((P||\Pi) \setminus H)$ for all such Π (see [13]), we obtain that $T(\phi \rhd P)/H \subseteq T((P||\Pi) \setminus H)$. For the reverse inclusion, we show that $T((P||\Pi) \setminus H) \subseteq T(\phi \rhd P)$ for all ϕ, and all $\Pi \in \mathcal{P}^{\phi}_{H}$. Let $\gamma \in T((P||\Pi) \setminus H)$. Then there exist P', Π' such that $(P||\Pi) \setminus H \overset{\gamma}{\Longrightarrow} (P'||\Pi') \setminus H$. The proof is by induction on the length l of the derivation from $(P||\Pi) \setminus H$ to $(P'||\Pi') \setminus H$.

Base. If $l = 0$ we are done, for γ is the empty trace.

Induction step. Let $l > 0$. Then there exist P'',Π'', $a \in Act$ and $\gamma' \in \mathcal{L}^*$ such that $(P||\Pi) \setminus H \xrightarrow{a} (P''||\Pi'') \setminus H \overset{\gamma'}{\Longrightarrow} (P'||\Pi') \setminus H$, and with $\Pi'' \in \mathcal{P}^{\phi''}_{H}$ for an appropriate ϕ''. By the induction hypothesis we know that $\gamma' \in T(\phi'' \rhd P'')$. To conclude, we need to show that $\phi \rhd P \xrightarrow{a} \phi'' \rhd P''$. The proof is by a case analysis on the shape of the transition $(P||\Pi) \setminus H \xrightarrow{a} (P''||\Pi'') \setminus H$. An inspection of the transition rules in Fig. 2 shows that a may only be a low action or the silent action τ. We analyze these two cases below.

If a is a low action, since Π is a high-level process, it must be the case that $P \xrightarrow{a} P''$ and that $\Pi'' = \Pi$. Hence in particular $\Pi'' \in \mathcal{P}^{\phi}$. Now, by rule (*low*) in Fig. 3 we derive $\phi \rhd P \xrightarrow{a} \phi \rhd P''$ as desired.

If instead $a = \tau$ we have four possible subcases.

- $P \xrightarrow{\tau} P''$ and $P = \Pi''$. Then $P'' \in \mathcal{P}^{\phi}_{H}$, and an inspection of the transition rules shows that $\phi \rhd P \xrightarrow{\tau} \phi \rhd P''$.

- $\Pi \xrightarrow{\tau} \Pi''$ and $P = P''$. Again, $\Pi'' \in \mathcal{E}_H^\phi$, and we conclude because $\phi \rhd P \Longrightarrow \phi \rhd P$.

- $P \xrightarrow{c(m)} P''$ and $\Pi \xrightarrow{\bar{c}m} \Pi''$. Again $\Pi'' \in \mathcal{P}_H^\phi$, with $m \in \phi$ by definition (for m occurs in Π which is a process in $\mathcal{P}_H^\phi$). Hence, in particular, $\phi \vdash m$, and then from $P \xrightarrow{c(m)} P''$ we derive $\phi \rhd P \xrightarrow{c(m)} \phi \rhd P''$ by rule (*input*) in Fig. 3.

- $P \xrightarrow{\bar{c}m} P''$ and $\Pi \xrightarrow{c(m)} \Pi''$. Here $\Pi'' \in \mathcal{P}_H^{\phi''}$ with $\phi'' = \phi \cup \{m\}$. Moreover, from $P \xrightarrow{\bar{c}m} P''$ one derives $\phi \rhd P \xrightarrow{\bar{c}m} \phi'' \rhd P''$ by (*output*) in Fig. 3.

From (3) we obtain that, for all ϕ

$$T((P||\Pi)\setminus H) = T(\phi \rhd P)/H,\ \forall \Pi \in \mathcal{P}_H^\phi \ \text{ iff } \ T((P||\Pi)\setminus H) = T(P \setminus H),\ \forall \Pi \in \mathcal{P}_H^\phi \quad (4)$$

Hence, from (3) and (4), it follows that

$$T((P||\Pi)\setminus H) = T(P \setminus H),\ \forall \Pi \in \mathcal{P}_H^\phi \ \text{ iff } \ T(P \setminus H) = T(\phi \rhd P)/H,\ \forall \Pi \in \mathcal{P}_H^\phi \quad (5)$$

In other words, $P \in NDC^\phi$ iff $T(P \setminus H) = T(\phi \rhd P)/H$, $\forall \Pi \in \mathcal{P}_H^\phi$. By Proposition 3.9 above, since $T(P \setminus H) \subseteq L^*$, $T(P \setminus H) = T(\phi \rhd P \setminus H) = T(\phi \rhd P \setminus H)/H$. Hence $P \in NDC^\phi$ iff $T(\phi \rhd P \setminus H)/H = T(\phi \rhd P)/H$, i.e, $P \in NDC^\phi$ iff $P \setminus H \simeq_{/H}^\phi P$. □

4.2 A characterization of $BNDC^\phi$

A second, more interesting, application of our approach is in characterizing the $BNDC^\phi$ property [14, 16], which results from instantiating (2) in the introduction with the equivalence $\approx$ as shown below. Again, the definition is due to [16].

Definition 4.4 (BNDC$^\phi$). $P \in BNDC^\phi$ if $P \setminus H \approx (P||\Pi) \setminus H$, $\forall \Pi \in \mathcal{P}_H^\phi$. □

As for NDC^ϕ, the definition falls short of being effective due to the universal quantification over Π. Here, however, the problem may not be circumvented by resorting to a hardest attacker, as the latter does not exist, being there no (known) preorder on processes corresponding to weak bisimilarity.

What we propose here is a partial solution that relies on providing a coinductive (and quantification free) characterization of a sound approximation of $BNDC^\phi$, based on the following *persistent* version of $BNDC^\phi$. We write $P \overset{*}{\Longrightarrow} P'$ (respectively $\phi \rhd P \overset{*}{\Longrightarrow} \phi' \rhd P'$) to state that P' ($\phi' \rhd P'$) is reachable from P ($\phi \rhd P$) by means of a sequence of transitions, irrespective of the trace involved in the sequence.

Definition 4.5 (P_BNDC$^\phi$). $P \in P\_BNDC^\phi$ if $P' \in BNDC^{\phi'}$ whenever $\phi \rhd P \overset{*}{\Longrightarrow} \phi' \rhd P'$. In particular, $P \in P\_BNDC^\phi$ if $P' \in P\_BNDC^{\phi'}$, for all $\phi' \rhd P'$ reachable from $\phi \rhd P$. □

[1] An analogous result has been recently presented by Gorrieri *et al.* in [18] for a *timed* extension of CryptoSPA. We discuss the relationships between our and their result in Section 6.

P_BNDC$^{\phi}$ is the context-sensitive version of the *P_BNDC* property studied in [17]. Following the technique in [17], one can show that *P_BNDC*$^{\phi}$ is a sound approximation of *BNDC*$^{\phi}$ which admits elegant quantification-free characterizations. Specifically, like *P_BNDC*, *P_BNDC*$^{\phi}$ can be characterized both in terms of a suitable weak bisimulation relation "up to high-level actions", noted $\approx^{\phi}_{/H}$, and in terms of unwinding conditions, as discussed next. We first need the following definition:

Definition 4.6. Let $a \in Act$. The transition relation $\stackrel{\hat{a}}{\Longrightarrow}_{/H}$ is defined as follows:

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

□

The transition relation $\stackrel{\hat{a}}{\Longrightarrow}_{/H}$ is defined as $\stackrel{\hat{a}}{\Longrightarrow}$, except that it treats H-level actions as silent actions. Now, weak bisimulations up to H over configurations are defined as weak bisimulations over configurations except that they allow a high action to be matched by zero or more high actions. Formally:

- A binary relation $\mathcal{R}$ over configurations is a *weak bisimulation up to H* if whenever $(\phi_P \rhd P, \phi_Q \rhd Q) \in \mathcal{R}$ one has, for all $a \in Act$,
  - if $\phi_P \rhd P \xrightarrow{a} \phi_{P'} \rhd P'$, then there exists a configuration $\phi_{Q'} \rhd Q'$ such that $\phi_Q \rhd Q \stackrel{\hat{a}}{\Longrightarrow}_{/H} \phi_{Q'} \rhd Q'$ and $(\phi_{P'} \rhd P', \phi_{Q'} \rhd Q') \in \mathcal{R}$;
  - if $\phi_Q \rhd Q \xrightarrow{a} \phi_{Q'} \rhd Q'$, then there exists a configuration $\phi_{P'} \rhd P'$ such that $\phi_P \rhd P \stackrel{\hat{a}}{\Longrightarrow}_{/H} \phi_{P'} \rhd P'$ and $(\phi_{P'} \rhd P', \phi_{Q'} \rhd Q') \in \mathcal{R}$.
- Two configurations $\phi_P \rhd P$ and $\phi_Q \rhd Q$ are *weakly bisimilar up to H*, denoted by $\phi_P \rhd P \approx^{c}_{/H} \phi_Q \rhd Q$, if there exists a weak bisimulation up to H containing the pair $(\phi_P \rhd P, \phi_Q \rhd Q)$.

The relation $\approx^{c}_{/H}$ may equivalently be defined as follows:

$$\approx^{c}_{/H} = \bigcup \{\mathcal{R} \mid \mathcal{R} \text{ is a weak bisimulation up to } H \text{ over configurations}\}.$$

Also, it is easy to prove that

- relation $\approx^{c}_{/H}$ is the largest weak bisimulation up to H over configurations
- relation $\approx^{c}_{/H}$ is an equivalence relation.

Finally, as for previous relations over configurations, we can recover an associated relation over processes in a context with initial knowledge ϕ.

Definition 4.7 (Weak bisimilarity under ϕ and H). $P \approx^{\phi}_{/H} Q$ iff $\phi \rhd P \approx^{c}_{/H} \phi \rhd Q$. □

Now we can state and prove the two characterizations of P_BNDC^{ϕ}. The former characterization is expressed in terms of $\approx^{\phi}_{/H}$ (with no quantification on the reachable states and on the high-level malicious processes). We first prove the following following lemma.

Lemma 4.8. *Let $P \in \mathcal{P}$ such that $P \setminus H \approx^{\phi}_{/H} P$. If $\phi \rhd P \overset{*}{\Longrightarrow} \phi' \rhd P'$, then there exists P'' such that $P \setminus H \overset{*}{\Longrightarrow} P'' \setminus H$ and $P'' \setminus H \approx^{\phi'}_{/H} P'$.*

□

Proof. Let $P \setminus H \approx^{\phi}_{/H} P$ and assume $\phi \rhd P \overset{*}{\Longrightarrow} \phi' \rhd P'$ with a sequence of l steps (noted $\phi \rhd P \overset{*}{\Longrightarrow}_l \phi' \rhd P'$). The proof follows by induction on l.

- *Base* ($l = 0$) In this case we can choose $P'' \setminus H$ equal to $P \setminus H$; then $phi' \rhd P' = \phi \rhd P$ and $P'' \setminus H = P \setminus H$ and we know that $P \setminus H \approx^{\phi}_{/H} P$.
- *Inductive step* ($l > 0$) Assume $\phi \rhd P \overset{*}{\Longrightarrow}_{l-1} \psi \rhd Q \xrightarrow{a} \phi' \rhd P'$. By the induction hypothesis, we find Q' with $P \setminus H \overset{*}{\Longrightarrow} Q' \setminus H$ and $Q' \setminus H \approx^{\psi}_{/H} Q$. We proceed by a case analysis on the possible shape of a.
  - $a \notin H$. From $Q' \setminus H \approx^{c\ \psi}_{/H} Q$ and $\psi \rhd Q \xrightarrow{a} \phi' \rhd P'$ we know that there exists P'' such that $\psi \rhd Q' \setminus H \xrightarrow{\hat{a}} \psi' \rhd P'' \setminus H$ and $\psi' \rhd P'' \setminus H \approx^{c}_{/H} \phi' \rhd P'$. In addition, since $a \notin H$, clearly $\psi = \phi' = \psi'$ and thus $P'' \setminus H \approx^{\phi'}_{/H} P'$.
  - $a = c(m) \in H$. Then $\psi \vdash m$ and $\psi = \phi'$. From From $Q' \setminus H \approx^{\psi}_{/H} Q$, and the observation that and $Q' \setminus H$ does not perform high-level actions, we find P'' and ψ' such that $\psi \rhd Q' \setminus H \overset{\hat{\tau}}{\Longrightarrow} \psi' \rhd P'' \setminus H$ and $\psi' \rhd P'' \setminus H \approx^{c}_{/H} \phi' \rhd P'$. Furthermore, since the knowledge component of configurations is invariant through τ, we have, $\psi = \psi'$ in the last weak transition, which implies $\psi' = \phi'$. Thus $P'' \setminus H \approx^{\phi'}_{/H} P'$ as desired.
  - $a = \overline{c}m \in H$. Then $\phi' = \psi \cup \{m\}$ and the reasoning is similar to the previous case. In fact, from $Q' \setminus H \approx^{\psi}_{/H} Q$ and the fact that $Q' \setminus H$ does not perform high-level actions, there exist P'' and ψ' such that $\psi \rhd Q' \setminus H \overset{\hat{\tau}}{\Longrightarrow} \psi' \rhd P'' \setminus H$ and $\psi' \rhd P'' \setminus H \approx^{c}\ _{/H} \phi' \rhd P'$. Again, $\psi = \psi'$ in the last weak transition, and since $P'' \setminus H$ does not perform any high-level action, $\psi' \rhd P'' \setminus H \approx^{c}_{/H} \phi' \rhd P'' \setminus H$ and thus $P'' \setminus H \approx^{\phi'}_{/H} P'$.

□

Theorem 4.9 (P_BNDC$^{\phi}$ 1). *$P \in P\_BNDC^{\phi}$ if and only if $P \setminus H \approx^{\phi}_{/H} P$.* □

Proof. ($\Rightarrow$) We first show that $P \in P\_BNDC^{\phi}$ implies $P \setminus H \approx^{\phi}_{/H} P$. To this end it is sufficient to prove that

$$\begin{array}{rcll} \mathcal{R} & = & \{(\phi \rhd P_1 \setminus H,\ \phi \rhd P_2) \mid & P_1 \setminus H \approx P_2 \setminus H,\ P_2 \in P\_BNDC^{\phi} \\ & & & \text{and } \phi \text{ is a set of messages}\} \end{array}$$

is a weak bisimulation up to H over configurations.

This follows from the following cases. Let $(\phi \rhd P_1 \setminus H,\ \phi \rhd P_2) \in \mathcal{R}$.

- $\phi \rhd P_1 \setminus H \xrightarrow{a} \phi' \rhd P_1' \setminus H$ with $a \notin H$. Hence, $P_1 \setminus H \xrightarrow{a} P_1' \setminus H$. By the hypothesis that $P_1 \setminus H \approx P_2 \setminus H$ there exists P_2' such that $P_2 \setminus H \stackrel{\hat{a}}{\Longrightarrow} P_2' \setminus H$ and $P_1' \setminus H \approx P_2' \setminus H$. Hence, since both internal and low actions do not depend on the context's knowledge, $\phi = \phi'$ and $\phi \rhd P_2 \stackrel{\hat{a}}{\Longrightarrow} \phi' \rhd P_2'$, i.e., $\phi \rhd P_2 \stackrel{\hat{a}}{\Longrightarrow}_{/H} \phi' \rhd P_2'$. Moreover, since $P_2 \in P\_BNDC^{\phi}$ it holds that $P_2' \in P\_BNDC^{\phi'}$, and thus, by definition of $\mathcal{R}$, $(\phi' \rhd P_1' \setminus H, \phi' \rhd P_2') \in \mathcal{R}$.

- $\phi \rhd P_2 \xrightarrow{a} \phi' \rhd P_2'$ with $a \notin H$. Hence, $P_2 \setminus H \xrightarrow{a} P_2' \setminus H$. Since $P_1 \setminus H \approx P_2 \setminus H$, there exists P_1' such that $P_1 \setminus H \stackrel{\hat{a}}{\Longrightarrow} P_1' \setminus H$ and $P_1' \setminus H \approx P_2' \setminus H$. Hence, since both internal and low actions do not depend on the context's knowledge, $\phi = \phi'$ and $\phi \rhd P_1 \setminus H \stackrel{\hat{a}}{\Longrightarrow} \phi' \rhd P_1' \setminus H$, i.e., $\phi \rhd P_1 \setminus H \stackrel{\hat{a}}{\Longrightarrow}_{/H} \phi' \rhd P_1' \setminus H$. Moreover, since $P_2 \in P\_BNDC^{\phi}$ it holds that $P_2' \in P\_BNDC^{\phi'}$, and thus, by definition of $\mathcal{R}$, $(\phi' \rhd P_1' \setminus H, \phi' \rhd P_2') \in \mathcal{R}$.

- $\phi \rhd P_2 \xrightarrow{a} \phi' \rhd P_2'$ with $a = c(m) \in H$ and $\phi \vdash m$. Since $P_2 \in P\_BNDC^{\phi}$ then $P_2 \in BNDC^{\phi}$. Let Π be the process $\overline{c}m$. We have that $\Pi \in \mathcal{P}_H^{\phi}$ and $P_2 \setminus H \approx (P_2 || \Pi) \setminus H$. Hence, $(P_2 || \Pi) \setminus H \xrightarrow{\tau} P_2' \setminus H$. Since $P_1 \setminus H \approx P_2 \setminus H \approx (P_2 || \Pi) \setminus H$, there exists P_1' such that $P_1 \setminus H \stackrel{\hat{\tau}}{\Longrightarrow} P_1' \setminus H$ and $P_1' \setminus H \approx P_2' \setminus H$. Hence, since internal actions do not depend on the context's knowledge, $\phi = \phi'$ and $\phi \rhd P_1 \setminus H \stackrel{\hat{\tau}}{\Longrightarrow} \phi' \rhd P_1' \setminus H$, i.e., $\phi \rhd P_1 \setminus H \stackrel{\hat{\tau}}{\Longrightarrow}_{/H} \phi' \rhd P_1' \setminus H$. Moreover, since $P_2 \in P\_BNDC^{\phi}$ it holds that $P_2' \in P\_BNDC^{\phi'}$, and thus, by definition of $\mathcal{R}$, $(\phi' \rhd P_1' \setminus H, \phi' \rhd P_2') \in \mathcal{R}$.

- $\phi \rhd P_2 \xrightarrow{a} \phi' \rhd P_2'$ with $a = \overline{c}m \in H$ and $\phi' = \phi \cup \{m\}$. Since $P_2 \in P\_BNDC^{\phi}$ then $P_2 \in BNDC^{\phi}$. Let Π be the process $c(x).\mathbf{0}$. We have that $\Pi \in \mathcal{P}_H^{\phi}$ and $P_2 \setminus H \approx (P_2 || \Pi) \setminus H$. Hence, $(P_2 || \Pi) \setminus H \xrightarrow{\tau} P_2' \setminus H$. Since $P_1 \setminus H \approx P_2 \setminus H \approx (P_2 || \Pi) \setminus H$, there exists P_1' such that $P_1 \setminus H \stackrel{\hat{\tau}}{\Longrightarrow} P_1' \setminus H$ and $P_1' \setminus H \approx P_2' \setminus H$. Hence, since internal actions do not depend on the context's knowledge, $\phi \rhd P_1 \setminus H \stackrel{\hat{\tau}}{\Longrightarrow} \phi' \rhd P_1' \setminus H$, i.e., $\phi \rhd P_1 \setminus H \stackrel{\hat{\tau}}{\Longrightarrow}_{/H} \phi' \rhd P_1' \setminus H$. Moreover, since $P_2 \in P\_BNDC^{\phi}$ it holds that $P_2' \in P\_BNDC^{\phi'}$, and thus, by definition of $\mathcal{R}$, $(\phi' \rhd P_1' \setminus H, \phi' \rhd P_2') \in \mathcal{R}$.

$\Leftarrow$ We now show that $P \setminus H \approx^{\phi}_{/H} P$ implies $P \in P\_BNDC^{\phi}$. In order to do it we prove that

$$\mathcal{R} \quad = \quad \{(P_1 \setminus H, (P_2 || \Pi) \setminus H) \mid \quad \Pi \in \mathcal{P}_H^{\phi} \text{ and } P_1 \setminus H \approx^{\phi}_{/H} P_2\}$$

is a weak bisimulation. This is sufficient to say that $P \in P\_BNDC^{\phi}$. In fact, by Lemma 4.8, for every $\phi' \rhd P'$ reachable from $\phi \rhd P$ there exists $P'' \setminus H$ reachable from $P \setminus H$ such that $P'' \setminus H \approx^{\phi'}_{/H} P'$. Hence, by definition of $\mathcal{R}$, we have that for all $\Pi \in \mathcal{P}_H^{\phi'}$, $(P'' \setminus H, (P' || \Pi) \setminus H) \in \mathcal{R}$. Since $\mathcal{R}$ is a weak bisimulation, we have that for all $\Pi \in \mathcal{P}_H^{\phi}$, $P'' \setminus H \approx (P'||\Pi) \setminus H$ and, in particular, $P'' \setminus H \approx P' \setminus H$. Since $\approx$ is an equivalence relation, by symmetry and transitivity, we have that for every $\phi' \rhd P'$ reachable from $\phi \rhd P$ and for all $\Pi \in \mathcal{P}_H^{\phi'}$, $P' \setminus H \approx (P'||\Pi) \setminus H$, i.e., $P' \in P\_BNDC^{\phi'}$. Thus, by definition of P_BNDC^{ϕ}, $P \in P\_BNDC^{\phi}$.

The fact that $\mathcal{R}$ is a weak bisimulation follows from the following cases.

Let $(P_1 \setminus H, (P_2||\Pi) \setminus H) \in \mathcal{R}$.

- $P_1 \setminus H \xrightarrow{a} P_1' \setminus H$ with $a \notin H$. Thus, $\phi \rhd P_1 \setminus H \xrightarrow{a} \phi' \rhd P_1' \setminus H$ with $\phi = \phi'$. By the hypothesis that $P_1 \setminus H \approx^{\phi}_{/H} P_2$, there exist P_2' and ϕ'' such that $\phi \rhd P_2 \overset{a}{\Longrightarrow} \phi'' \rhd P_2'$ and $\phi' \rhd P_1' \setminus H \approx_{/H} \phi'' \rhd P_2'$. Since both internal and low actions do not affect the context's knowledge, $\phi = \phi' = \phi''$ and $P_1' \setminus H \approx^{\phi}_{/H} P_2'$. In particular, $P_2 \overset{a}{\Longrightarrow} P_2'$ and thus $(P_2||\Pi) \setminus H \overset{\hat{a}}{\Longrightarrow} (P_2'||\Pi) \setminus H$, i.e., by definition of $\mathcal{R}$, $(P_1' \setminus H, (P_2'||\Pi) \setminus H) \in \mathcal{R}$.

- $(P_2||\Pi) \setminus H \xrightarrow{a} (P_2'||\Pi) \setminus H$ where also $P_2 \setminus H \xrightarrow{a} P_2' \setminus H$ and $a \notin H$. Thus, $\phi \rhd P_2 \xrightarrow{a} \phi' \rhd P_2'$ with $\phi = \phi'$. By the hypothesis that $P_1 \setminus H \approx^{\phi}_{/H} P_2$, there exist $P_1' \setminus H$ and ϕ'' such that $\phi \rhd P_1 \setminus H \overset{a}{\Longrightarrow} \phi'' \rhd P_1' \setminus H$ and $\phi'' \rhd P_1' \setminus H \approx_{/H} \phi' \rhd P_2'$. Since both internal and low actions do not affect the context's knowledge, $\phi = \phi' = \phi''$ and thus $P_1' \setminus H \approx^{\phi}_{/H} P_2'$. Moreover, since $a \notin H$, we have that $P_1 \setminus H \overset{\hat{a}}{\Longrightarrow} P_1' \setminus H$ and, by definition of $\mathcal{R}$, $(P_1' \setminus H, (P_2'||\Pi) \setminus H) \in \mathcal{R}$.

- $(P_2||\Pi) \setminus H \xrightarrow{\tau} (P_2||\Pi') \setminus H$ with $\Pi \xrightarrow{\tau} \Pi'$. If $\Pi \in \mathcal{P}_H^{\phi}$ then also $\Pi' \in \mathcal{P}_H^{\phi}$ and thus, by definition of $\mathcal{R}$, it trivially follows that $(P_1 \setminus H, (P_2||\Pi') \setminus H) \in \mathcal{R}$.

- $(P_2||\Pi) \setminus H \xrightarrow{\tau} (P_2'||\Pi') \setminus H$ where $P_2 \xrightarrow{c(m)} P_2'$, $\Pi \xrightarrow{\bar{c}m} \Pi'$, $\phi \vdash m$ and $c(m) \in H$. Thus, $\phi \rhd P_2 \xrightarrow{c(m)} \phi' \rhd P_2'$ with $\phi = \phi'$. By the hypothesis that $P_1 \setminus H \approx^{\phi}_{/H} P_2$ and the fact that $P_1 \setminus H$ does not perform high-level actions, there exist $P_1' \setminus H$ and ϕ'' such that $\phi \rhd P_1 \setminus H \overset{\hat{\tau}}{\Longrightarrow} \phi'' \rhd P_1' \setminus H$ and $\phi'' \rhd P_1' \setminus H \approx_{/H} \phi' \rhd P_2'$. Since internal actions do not affect the context's knowledge, $\phi = \phi' = \phi''$ and thus $P_1' \setminus H \approx^{\phi}_{/H} P_2'$. Moreover, $P_1 \setminus H \overset{\hat{\tau}}{\Longrightarrow} P_1' \setminus H$ and then, by definition of $\mathcal{R}$, $(P_1' \setminus H, (P_2'||\Pi) \setminus H) \in \mathcal{R}$.

- $(P_2||\Pi) \setminus H \xrightarrow{\tau} (P_2'||\Pi') \setminus H$ where $P_2 \xrightarrow{\bar{c}m} P_2'$, $\Pi \xrightarrow{c(m)} \Pi'$ and $\bar{c}m \in H$. Thus, $\phi \rhd P_2 \xrightarrow{\bar{c}m} \phi' \rhd P_2'$ with $\phi' = \phi \cup \{m\}$. By the hypothesis that $P_1 \setminus H \approx^{\phi}_{/H} P_2$ and the fact that $P_1 \setminus H$ does not perform high-level actions, there exist $P_1' \setminus H$ and ϕ'' such that $\phi \rhd P_1 \setminus H \overset{\hat{\tau}}{\Longrightarrow} \phi'' \rhd P_1' \setminus H$ and $\phi'' \rhd P_1' \setminus H \approx_{/H} \phi' \rhd P_2'$. Since internal actions do not affect the context's knowledge, $\phi = \phi''$. Moreover, since $P_1' \setminus H$ does not perform any high-level action, $\phi \rhd P_1' \setminus H \approx_{/H} \phi' \rhd P_1' \setminus H$, and thus $P_1' \setminus H \approx^{\phi'}_{/H} P_2'$. Moreover, $\Pi \in \mathcal{P}_H^{\phi'}$ and $P_1 \setminus H \overset{\hat{\tau}}{\Longrightarrow} P_1' \setminus H$. Hence, by definition of $\mathcal{R}$, $(P_1' \setminus H, (P_2'||\Pi) \setminus H) \in \mathcal{R}$.

□

The second characterization of P_BNDC^{ϕ} 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., [8]) and refinement operators that preserve security properties, as done in [19].

Theorem 4.10 ($\mathbf{P\_BNDC^{\phi}}$ 2). *$P \in P\_BNDC^{\phi}$ if and only if for all $\phi' \rhd P'$ reachable from $\phi \rhd P$, if $\phi' \rhd P' \xrightarrow{h} \psi \rhd Q$ for some $h \in H$, then $\phi' \rhd P' \overset{\hat{\tau}}{\Longrightarrow} \psi' \rhd Q'$ and $Q \setminus H \approx Q' \setminus H$.* □

Proof. $\Leftarrow$ Let P be a process and ϕ be a set of messages such that for all $\phi' \rhd P'$ reachable from $\phi \rhd P$ and $\phi' \rhd P' \xrightarrow{h} \psi \rhd Q$ with $h \in H$, there exist Q' and ψ' such that $\phi' \rhd P' \overset{\hat{\tau}}{\Longrightarrow} \psi' \rhd Q'$ and $Q \setminus H \approx Q' \setminus H$. Let

$$\mathcal{R} = \{(P' \setminus H, (P'||\Pi) \setminus H) | \ \Pi \in \mathcal{P}_H^{\phi'} \text{ and } \phi' \rhd P' \text{ is reachable from } \phi \rhd P\}.$$

We prove that $\mathcal{R}$ is a weak bisimulation up to $\approx$. We have to consider the following cases.

- $P' \setminus H \xrightarrow{a} Q \setminus H$ with $a \notin H$. Then, $\phi' \rhd P' \xrightarrow{a} \phi' \rhd Q$, i.e., $\phi' \rhd Q$ is reachable from $\phi \rhd P$. Moreover, $(P'||\Pi) \setminus H \xrightarrow{a} (Q||\Pi) \setminus H$ and then, by definition of $\mathcal{R}$, $(Q \setminus H, (Q||\Pi) \setminus H) \in \mathcal{R}$.

- $(P'||\Pi) \setminus H \xrightarrow{a} (Q||\Pi) \setminus H$, with $a \notin H$ and $P' \setminus H \xrightarrow{a} Q \setminus H$. Hence $\phi' \rhd P' \xrightarrow{a} \phi' \rhd Q$, i.e., $\phi' \rhd Q$ is reachable from $\phi \rhd P$. Thus, by definition of $\mathcal{R}$, $(Q \setminus H, (Q||\Pi) \setminus H) \in \mathcal{R}$.

- $(P'||\Pi) \setminus H \xrightarrow{\tau} (P'||\Pi') \setminus H$ where $\Pi \xrightarrow{\tau} \Pi'$. Since $P' \setminus H \overset{\hat{\tau}}{\Longrightarrow} P' \setminus H$ and $\Pi' \in \mathcal{P}_H^{\phi'}$, by definition of $\mathcal{R}$ we immediately have $(P' \setminus H, (P'|\Pi') \setminus H) \in \mathcal{R}$.

- $(P'||\Pi) \setminus H \xrightarrow{\tau} (Q||\Pi') \setminus H$ where $P' \xrightarrow{c(m)} Q$, $\Pi \xrightarrow{\overline{c}m} \Pi'$, $\phi' \vdash m$, $c(m) \in H$ and $\Pi' \in \mathcal{P}_H^{\phi'}$. Then, $\phi' \rhd P' \xrightarrow{c(m)} \phi' \rhd Q$, i.e., $\phi' \rhd Q$ is reachable from $\phi \rhd P$. By hypothesis, there exist Q' and ψ' such that $\phi' \rhd P' \overset{\hat{\tau}}{\Longrightarrow} \psi' \rhd Q'$ and $Q \setminus H \approx Q' \setminus H$. Hence, $P' \setminus H \overset{\hat{\tau}}{\Longrightarrow} Q' \setminus H$ and $Q' \setminus H \approx Q \setminus H \ \mathcal{R} \ (Q||\Pi') \setminus H)$.

- $(P'||\Pi) \setminus H \xrightarrow{\tau} (Q||\Pi') \setminus H$ where $P' \xrightarrow{\overline{c}m} Q$, $\Pi \xrightarrow{c(m)} \Pi'$, $\overline{c}m \in H$ and $\Pi' \in \mathcal{P}_H^{\phi' \cup \{m\}}$. Then, $\phi' \rhd P' \xrightarrow{c(m)} \phi' \cup \{m\} \rhd Q$, i.e., $\phi' \cup \{m\} \rhd Q$ is reachable from $\phi \rhd P$. By hypothesis, there exist Q' and ψ' such that $\phi' \rhd P' \overset{\hat{\tau}}{\Longrightarrow} \psi' \rhd Q'$ and $Q \setminus H \approx Q' \setminus H$. Hence, $P' \setminus H \overset{\hat{\tau}}{\Longrightarrow} Q' \setminus H$ and $Q' \setminus H \approx Q \setminus H \ \mathcal{R} \ (Q||\Pi') \setminus H)$.

$\Rightarrow$ Let P be P_BNDC^{ϕ}. Then, for all $\phi' \rhd P'$ reachable from $\phi \rhd P$, $P' \in BNDC^{\phi'}$. In particular, for all $\phi' \rhd P'$ reachable from $\phi \rhd P$ and for all $\Pi \in \mathcal{P}_H^{\phi'}$, $P' \setminus H \approx (P'||\Pi) \setminus H$. Suppose that $\phi' \rhd P' \xrightarrow{h} \psi \rhd Q$ for some $h \in H$. We distinguish two cases.

- $\phi' \rhd P' \xrightarrow{c(m)} \psi \rhd Q$ with $\phi' \vdash m$ and $\phi' = \psi$. Let $\Pi = \overline{c}m.\mathbf{0}$. Then $\Pi \in \mathcal{P}_H^{\phi'}$ and $(P'||\Pi) \setminus H \xrightarrow{\tau} Q \setminus H$. By the fact that $P' \setminus H \approx (P'||\Pi) \setminus H$ for all $\Pi \in \mathcal{P}_H^{\phi'}$, we have that there exists $Q' \setminus H$ such that $P' \setminus H \overset{\hat{\tau}}{\Longrightarrow} Q' \setminus H$ and $Q \setminus H \approx Q' \setminus H$. Hence, in particular, $\phi' \rhd P' \overset{\hat{\tau}}{\Longrightarrow} \phi' \rhd Q'$ and $Q \setminus H \approx Q' \setminus H$.

- $\phi' \rhd P' \xrightarrow{\overline{c}m} \psi \rhd Q$ with $\psi = \phi' \cup \{m\}$. Let $\Pi = c(x).\mathbf{0}$. Then $\Pi \in \mathcal{P}_H^{\phi'}$ and $(P'||\Pi) \setminus H \xrightarrow{\tau} Q \setminus H$. By the fact that $P' \setminus H \approx (P'||\Pi) \setminus H$ for all $\Pi \in \mathcal{P}_H^{\phi'}$, we have that there exists $Q' \setminus H$ such that $P' \setminus H \overset{\hat{\tau}}{\Longrightarrow} Q' \setminus H$ and $Q \setminus H \approx Q' \setminus H$. Hence, in particular, $\phi' \rhd P' \overset{\hat{\tau}}{\Longrightarrow} \phi' \rhd Q'$ and $Q \setminus H \approx Q' \setminus H$.

□

Both the characterizations can be used for verifying cryptographic protocols. A concrete example of a fair exchange protocol is illustrated in the next section.

5 The *Asokan-Shoup-Waidener* Fair Exchange Protocol

We illustrate the proof techniques developed in the previous section with a case study in which we show their use in the verification of different properties of a protocol of fair exchange. Fair exchange protocols are used extensively in applications such as online payment systems [10] contract signing [4, 2], certified electronic mail [3, 24, 12], and other purposes.

Our case study is a simplified version of the optimistic contract signing protocol by Asokan, Shoup, and Waidner [2], which we shall refer to as the ASW protocol. The ASW protocol enables two parties, named O (originator) and R (responder), to obtain each other's commitment on a previously agreed contractual text M.

The protocol consists of three independent sub-protocols: *Exchange*, *Abort* and *Resolve*. The parties initiates with the *Exchange* sub-protocol which is meant to provide for the fair exchange of the contract. The originator O has the option to request a trusted third party T to stop the exchange by running the *Abort* sub-protocol with T. Intuitively, an honest O might choose to do that if a response from R is not received after a reasonable waiting period. Finally, either O or R may individually request that T resolve the exchange and issue a replacement contract: the *Resolve* sub-protocol is designed for that purpose. The expected property of the ASW protocol is that at completion each party is guaranteed to end up with a valid contract or an abort token.

The original specification of the protocol uses digital signatures. Here we study a variant, based on an asymmetric cryptosystem, described by the informal narration in Fig. 4.

M is the contractual text on which we assume the two parties have agreed, while K_O, K_R and K_T are the private keys owned by O, R, and T respectively. The protocol description is as follows [2]:

Step 1. O commits to the contractual text by hashing a random number N_O, and signing a message that contains both $h(N_O)$ and M: $h(N_O)$ is used as a public com-

Exchange

$$\begin{array}{lll}
O \rightarrow R & : msg_1 & = \{M, h(N_O)\}_{K_O} \\
R \rightarrow O & : msg_2 & = \{msg_1, h(N_R)\}_{K_R} \\
O \rightarrow R & : msg_3 & = N_O \\
R \rightarrow O & : msg_4 & = N_R
\end{array}$$

Abort

$$\begin{array}{lll}
O \rightarrow T & : ma_1 & = \{aborted, \{M, h(N_O)\}_{K_O}\}_{K_O} \\
T \rightarrow O & : ma_2 & = resolved \ ? \ \{msg_1, msg_2\}_{K_T} \\
 & & \qquad\qquad : \ aborted := true, \ \{aborted, ma_1\}_{K_T}
\end{array}$$

Resolve

$$\begin{array}{lll}
O, R \rightarrow T & : mr_1 & = (msg_1, msg_2) \\
T \rightarrow R, O & : mr_2 & = aborted \ ? \ \{aborted, ma_1\}_{K_T} \\
 & & \qquad\qquad : \ resolved := true, \ \{msg_1, msg_2\}_{K_T}
\end{array}$$

Figure 4: The ASW protocol

mitment to the secret N_O, while N_O is the *contract authenticator* (or, the non-repudiation token) by O: once $h(N_O)$ is given to R, O may not change the token N_O. The inability to repudiate the token N_O is a consequence of the (standard) assumption that it is not computationally feasible for O to find a different number N'_O such that $h(N'_O) = h(N_O)$.

Step 2. If R decides to give up, it simply terminates (this may happen if R does not receive any message within a given time limit). Otherwise R verifies the signature of msg_1 (by decrypting msg_1 with $O's$ public key K_O^{-1}) and checks that the contents is formed correctly (namely, that M is indeed the contractual text that was agreed upon), and replies with its own public commitment $h(N_R)$.

Step 3. If O decides to give up, it invokes T by running the *Abort* protocol. Otherwise it sends its secret N_O to R.

Step 4. If R decides to give up, it invokes T by running the *Resolve* protocol. Otherwise it sends its secret N_R to O and completes

Step 5. If O decides to give up, it invokes T by running the *Resolve* protocol. Otherwise it completes.

At the completion of the above steps, both O and R should obtain a valid contract, i.e., either a standard contract $\{msg_1, N_O, msg_2, N_R\}$, including the text and the non-repudiation tokens, or a replacement contract $\{msg_1, msg_2\}_{K_T}$.

We say that the protocol guarantees *fairness* to the originator O on message M, if whenever the (possibly dishonest) responder R gets evidence that O has originated

M (i.e., R receives N_O), then O itself will eventually obtain the evidence that R has received M (i.e., O receives N_R). Dually, the protocol guarantees fairness to the responder R if the above holds with the roles of O and R exchanged.

5.1 Analysis of the *Exchange* sub-protocol

We start our analysis by disregarding all issues concerning time and/or the conditions governing the decision to abort or resolve the contract, and concentrate on the exchange sub-protocol instead. The analysis draws on a representation of the sub-protocol in CryptoSPA. The cryptoSPA specification, in Fig. 5, defines one instance of the protocol as the parallel composition of the originator and the responder.

$$\begin{aligned}
O(M,N_O) &\overset{def}{=} \overline{c}\, msg_1 .\ c(v).\ check_{msg_2}(v).\ \overline{c} N_O .\ c(j).\ check_{N_R}(j).\ \overline{done} \\
R(M,N_R) &\overset{def}{=} c(q).\ check_{msg_1}(q).\ \overline{c}\, msg_2 .\ c(u).\ check_{N_O}(u).\ \overline{done}.\ \overline{c} N_R \\
P &\overset{def}{=} O(M,N_O) \mid\mid R(M,N_R)
\end{aligned}$$

where

$$\begin{aligned}
\overline{c}\, msg_1 &\equiv [\langle N_O, k_h\rangle \vdash_{enc} n][\langle (M,n), K_O\rangle \vdash_{enc} p]\ \overline{c}\, p \\
\overline{c}\, msg_2 &\equiv [\langle N_R, k_h\rangle \vdash_{enc} r][\langle (q,r), K_R\rangle \vdash_{enc} t]\ \overline{c}\, t \\
check_{msg_2}(v) &\equiv [\langle v, K_R^{-1}\rangle \vdash_{dec} i][i \vdash_{fst} p'][i \vdash_{snd} r'][p' = p] \\
check_{N_R}(j) &\equiv [\langle j, k_h\rangle \vdash_{enc} r''][r'' = r'] \\
check_{msg_1}(q) &\equiv [\langle q, K_O^{-1}\rangle \vdash_{dec} s][s \vdash_{fst} m][s \vdash_{snd} n'][m = M] \\
check_{N_O}(u) &\equiv [\langle u, k_h\rangle \vdash_{enc} n''][n'' = n']
\end{aligned}$$

Figure 5: One instance of the *Exchange* sub-protocol in CryptoSPA

We use a high-level (hence public) channel c to circulate all messages between the parties, and represent the application of the hash function h by encryption under a corresponding key k_h. In addition, we include outputs on the low-level channel *done* to formalize the intended properties of the protocol: before writing on *done* the parties validate the messages they receive against the message they expect at the corresponding protocol step.

It is immediate to see that the exchange sub-protocol, by itself, does not provide the intended fairness guarantees. Indeed, for the sub-protocol to be *fair* one would at least need to make guarantees that both parties reach the completion of the protocol. Clearly, this is not the case, as even with no other knowledge ϕ than the channel c, an attacker can block the messages circulated on c and prevent either party to complete.

This is easily observed by noting that the process P, representing one instance of the protocol, is not P_BNDC^{ϕ} for any $\phi \supseteq \{c\}$. This follows by the unwinding characterization in Theorem 4.10. To see that, take the transition $\phi \rhd P \xrightarrow{\overline{c}\, msg_1} \phi \cup \{msg_1\} \rhd P'$ resulting from O sending its first message. It is a routine check to verify that there exists no configuration $\phi'' \rhd P''$ such that $\phi \rhd P \overset{\hat{\tau}}{\Longrightarrow} \phi'' \rhd P''$ and $P' \setminus H \approx P''_i \setminus H$. In fact, on easily shows that $P' \setminus H \approx \mathbf{0}$, while $P'' \setminus H \not\approx \mathbf{0}$ for all P'' and ϕ'' such that $\phi \rhd P \overset{\hat{\tau}}{\Longrightarrow} \phi'' \rhd P''$.

Notice, on the other hand, that the protocol is NDC^{ϕ}. This is not too surprising, as the property we are looking at is a *liveness* property: in fact, we are requiring that something "good" should happen, namely that both participants complete their run. Put differently, detecting and attack to the protocol requires the ability to observe failures, something that cannot be accomplished by means of properties based on trace-equivalence such as NDC^{ϕ}.

5.2 Analysis of the complete protocol

The fact that the exchange sub-protocol does not satisfy P_BNDC^{ϕ} does not represent a real attack, since the ASW protocol resolves such situations by inching the trusted party T. Indeed, as we mentioned above, in this case O might choose to request the trusted third party T to abort the exchange, leading to a *fair* completion of the protocol.

A more faithful representation of the originator and responder is discussed below, where we give an explicit account of the decisions to abort or resolve the protocol. To formalize the protocol in full, we need to address two further aspects, relative to the underlying communication model and to the way the decisions to abort/resolve are made. According to [2]

- all decisions to abort/resolve are made non-deterministically, based on (implicit) timeouts, by internal choices of the parties,

- no assumption should be made on the channels connecting O and R, while the channels between T and the two parties may be assumed to be *resilient*, i.e., they guarantee delivery within finite time bounds.

While non-deterministic choice is primitive in CryptoSPA, the presence of timeouts and the resilience of channels do not have a direct counterpart in our calculus. We therefore need to provide an explicit encoding. We represent timeouts by structuring the processes so that any output on the channel c, as in $\overline{c}\, m.\, P$, may have two transitions, namely: $\overline{c}\, m.\, P \xrightarrow{\overline{c}\, m} P$ or $\overline{c}\, m.\, P \xrightarrow{\tau} P'$. The first transition is standard, modeling the fact that the message m is sent and eventually received (by the intended recipient or by the intruder). The second transition models the fact P may decide to timeout and continue as P', irrespective of the reception of the message. The resulting specification is given in Fig. 6.

As we did earlier, we analyze one instance of the protocol given by the process $P = O(M, N_O) \mid\mid R(M, N_R)$. Notice that although we disregard T in our analysis, process P is a sound abstraction of (an instance of) a complete protocol, including T. In fact, given the assumption that the channel between the parties and T is resilient, we may simply assume that all messages sent to T will reach their destination. In Fig.

$$\begin{aligned}
O(M,N_O) &\overset{def}{=} \tau.\,\overline{abort} + \\
&\quad \overline{c}\;msg_1.\,(\;\tau.\,\overline{abort} + \\
&\qquad\qquad c(v).\,check_{msg_2}(v).(\;\tau.\,\overline{done}(resolve) + \\
&\qquad\qquad\qquad\qquad \overline{c}N_O.\,(\;\tau.\,\overline{done}(resolve) + \\
&\qquad\qquad\qquad\qquad\quad c(j).check_{N_R}(j).\,\overline{done}(v,j)\;)\;) \\
R(M,N_R) &\overset{def}{=} \tau + \\
&\quad c(q).\,check_{msg_1}(q).(\;\tau.\,\overline{done}(resolve) + \\
&\qquad\qquad\qquad \overline{c}\;msg_2.\,(\;\tau.\,\overline{done}(resolve) + \\
&\qquad\qquad\qquad\quad c(u).check_{N_O}(u).\;\overline{done}(q,u).\,\overline{c}N_R\;)\;)\;) \\
&\qquad\qquad ;\,\tau.\,R(M,N_R) \\
P &\overset{def}{=} O(M,N_O)\;||\;R(M,N_R)
\end{aligned}$$

Figure 6: An instance of the *exchange* sub-protocol, with abort/resolve

6 these exchanges are represented by the low-level outputs *abort* and $done(resolve)$, with the latter giving an abstract representation of the completion of the protocol with the replacement contract. Similarly, the outputs $done(v,j)$ and $done(q,u)$ give and abstract signal of the successful completion of the protocol with the participants having obtained a standard contract.

5.2.1 Honest participants.

We first assume that both parties are honest, i.e., they behave according to the specification and willing to complete the exchange. This corresponds to analyzing protocol runs starting with $\phi = \{c, K_0^{-1}, K_R^{-1}\}$, i.e. in a context with access to c and only informed on the public keys of the participants (being given no access to the participants private keys, the context may not simulate the behavior of a dishonest principal). Given this assumption, we can simplify the protocol by disregarding the messages output on *done* as shown below:

$$\begin{array}{lcl}
O(M,N_O) & \overset{def}{=} & \tau.\ \overline{abort} + \\
 & & \overline{c}\ msg_1.\ (\ \tau.\ \overline{abort} + \\
 & & \hspace{3em} c(v).\ check_{msg_2}(v).\ (\ \tau.\ \overline{done} + \\
 & & \hspace{9em} \overline{c}N_O.\ (\ \tau.\ \overline{done} + \\
 & & \hspace{12em} c(j).check_{N_R}(j).\ \ \overline{done}\ )\ )
\end{array}$$

$$\begin{array}{lcl}
R(M,N_R) & \overset{def}{=} & \tau + \\
 & & c(q).\ check_{msg_1}(q).\ (\ \tau.\ \overline{done} + \\
 & & \hspace{7em} \overline{c}\ msg_2.\ (\ \tau.\ \ \overline{done} + \\
 & & \hspace{10em} c(u).check_{N_O}(u).\ \ \overline{done}.\ \overline{c}N_R\ )\ )\ ) \\
 & & \hspace{6.5em} ;\ \tau.\ R(M,N_R)
\end{array}$$

To motivate, notice that in this case the intruder may not forge a valid contract (for it does not the private keys of the participants), and hence, all components of a standard contract are guaranteed to originate from the parties. Thus, for the purpose of fairness, we only need to make sure that the protocol either aborts, or completes with both parties receiving a valid contract (either standard or replacement): in particular, in the latter case, both parties are guaranteed to receive the same contract.

Based on the simplified specification, we may say that the protocol is fair if it exhibits either a single *abort*, or two *done*'s. In fact, it is not difficult to see that $P \setminus H$, the secure specification, is (weakly) bisimilar to $\tau.\ \overline{abort}\ + \tau.\ \overline{done}.\ \overline{done}$. Thus, to verify the correctness of the protocol we need to show that $(P||\Pi) \setminus H \approx \tau.\ \overline{abort}\ + \tau.\ \overline{done}.\ \overline{done}$ for all $\Pi \in \mathcal{P}_H^\phi$, i.e., that P is P_BNDC^ϕ. By our characterizations, this can be accomplished by either exhibiting a bisimulation to show that $P \approx^\phi_{/H} \tau.\ \overline{abort}\ + \tau.\ \overline{done}.\ \overline{done}$, or by checking that the unwinding conditions in Theorem 4.10 are verified. As it turns out, P is indeed P_BNDC^ϕ, which confirms the correctness theorem in [2].

5.2.2 Dishonest participants.

We conclude with an analysis in the case that one of the participants is corrupt. Clearly, fairness for the corrupt party cannot be guaranteed in this case. For instance, if the intruder is able to sign messages with *O*'s private key, it is then able to impersonate *O* in any exchange and convince *R* that *O* has committed to a contract. The real question is whether sharing its private key with the intruder allows the corrupt participant to gain an unfair advantage over the other party. Below, we study the case in which the dishonest party is the responder *R*.

A dishonest responder may be represented by assuming that *R* leaked its secretes (key and contract authenticator) to the intruder. In our framework, this corresponds to assuming that ϕ includes such bits of information relative to *R*. An analysis of the processes in Fig. 6 shows that P is not $BNDC^\phi$ for any such ϕ. To see that, take the trace

$$\gamma = \overline{done}\ (msg_1, N_O)\ \overline{done}\ (msg'_2, N'_R)$$

with $msg_2' \neq msg_2$ and $N_R' \neq N_R$, and note that $\gamma \notin T(P \setminus H)$ while $\gamma \in T(\phi \rhd P)/H$, which imply that $P \setminus H \not\simeq^{\phi}_{/H} P$.

Interestingly, the trace γ represents the same attack to the protocol as the ony discovered with Murφ in [23]. In this attack, the intruder, which knows *R*'s private key, computes a different message msg_2' in response to *O*'s initial message msg_1 using a different nonce N_R' and sends it out. Then, *O* obtains the contract $\{msg_1, N_O, msg_2', N_R'\}$ while *R* has the valid contract $\{msg_1, N_O, msg_2, N_R\}$ which is inconsistent with the one obtained by *O*. Clearly, this is a problem, since each party possesses a valid contract, but the two contracts are inconsistent. Even though the contractual texts in the two contracts are the same, the secrets and the public commitments are different, and it is unclear how the contracts should be enforced or interpreted, given that both are valid according to the protocol specification. As noted in [23], the original paper [2] does not say anything about how this situation should be handled.

6 Conclusions and Related Work

We have studied context-sensitive properties of Non-Interference, and we have given powerful, quantification-free, characterizations of such properties. Our characterizations apply uniformly to trace and bisimulation-based notions of Non-Interference, and provide effective proof techniques for the analysis of security protocols. We have illustrated such techniques with the analysis of a non-trivial protocol of fair exchange.

Failure-sensitive properties, such as fairness, have been addressed by other authors. In particular, in [22], Schneider develops a formal analysis of a non-repudiation protocol expressed as a process of CSP [21] whose fairness properties are formalized in terms of the process' refusals set. Thus, in that case, the analysis is based on failure semantics. Linear-time, failure-sensitive equivalences such as *must* test would have been appropriate for the analysis of the protocol we have studied here. On the other hand, must-test equivalences are notoriously difficult to deal with, and implied by bisimulation equivalences such as the ones we have investigated.

Other papers in the literature have investigated knowledge-sensitive characterizations of behavioral equivalence and applied them to the verification of cryptographic protocols. We briefly discuss the approaches closest to ours below.

In a recent paper Gorrieri *et al.* [18] prove results related to ours, for a real-time extension of CryptoSPA. In particular, they prove an equivalent of Theorem 4.3: however, while the results are equivalent, the underlying proof techniques are not. More precisely, instead of using context-sensitive LTS's, [18] introduces a special hiding operator $/^{\phi}$ and proves that

$$P \in NDC^{\phi} \text{ if and only if } P \setminus H \simeq P/^{\phi}H \qquad (6)$$

Indeed $T(P/^{\phi}H)$ coincides with $T(\phi \rhd P)/H$ and thus (6) is simply a different way to write our Theorem 4.3. However, the approach of [18] is still restricted to the class of observation equivalences that are behavioral preorders on processes and thus it does not extend to bisimulations.

As we pointed out since the outset, our approach is inspired by the work by Boreale *et al.* [6] on characterizing may test and barbed congruence in the spi calculus by means of trace and bisimulation equivalences built on top of context-sensitive LTS's. Based on the same technique, symbolic semantics and compositional proofs have been recently studied in [5, 7], to provide effective tools for the verification of cryptographic protocols. Such methods could be exploited to allow a finitary representaton of the context-sensitive labelled transition systems we have studied in the present paper. Future plans include work in that direction.

Acknowledgements The valuable comments of the anonymous referees are gratefully acknowledged: they helped improve the presentation substantially. Thanks to Ambra Ceccato for her contributions to earlier drafts of the paper.

References

[1] M. Abadi. Security Protocols and Specifications. In W. Thomas, editor, *Proc. of the Second International Conference on Foundations of Software Science and Computation Structure (FoSSaCS'99)*, volume 1578 of *LNCS*, pages 1–13. Springer-Verlag, 1999.

[2] N. Asokan, V. Shoup, and M. Waidener. Asynchronuous Protocols for Optimistic Fair Exchange. In *Proc. of the IEEE Symposium on Research in Security and Privacy*, pages 86–99. IEEE Computer Society Press, 1998.

[3] A. Bahreman and J.D. Tygar. Certified electronic mail. In *Proc. of the Internet Society Symposium on Network and Distributed System Security*, pages 3–19, 1994.

[4] M. Ben-Or, O. Goldreich, S. Micali, and R.L. Rivest. A fair protocol for signing contracts. *IEEE Transactions on Information Theory*, 36(1):40–46, 1990.

[5] M. Boreale and M. G. Buscemi. A Framework for the Analysis of Security Protocols. In *Proc. of the 13th International Conference on Concurrency Theory (CONCUR'02)*, volume 2421 of *LNCS*, pages 483–498. Springer-Verlag, 2002.

[6] M. Boreale, R. De Nicola, and R. Pugliese. Proof Tecniques for Cryptographic Processes. In *Proc. of the 14th IEEE Symposium on Logic in Computer Science (LICS'99)*, pages 157–166. IEEE Computer Society Press, 1999.

[7] M. Boreale and D. Gorla. On Compositional Reasoning in the spi-calculus. In *Proc. of the 5th International Conference on Foundations of Software Science and Computation Structures (FossaCS'02)*, volume 2303 of *LNCS*, pages 67–81. Springer-Verlag, 2002.

[8] 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, pages 199–218. Springer-Verlag, 2002.

[9] M. Bugliesi, A. Ceccato, and S. Rossi. Non Interference Proof Techniques for the Analysis of Cryptographic Protocols. *Workshop on Issues in the Theory of Security (WITS '03)*. An extended abstract is also available from *Proc. of the 14th International Symposium on Fundamentals of Computation Theory (FCT'03)*, volume 2751 of *LNCS* pages 364–375, Springer-Verlag, 2003.

[10] B. Cox, J. D. Tygar, and M. Sirbu. Netbill security and transaction protocol. In *Proc. of the First USENIX Workshop in Electronic Commerce*, pages 77–88, 1995.

[11] R. De Nicola and M Hennessy. Testing Equivalences for Processes. *Theoretical Computer Science*, 34:83–133, 1984.

[12] R. H. Deng, L. Gong, A. A. Lazar, and W. Wang. Practical protocols for certified electronic mail. *Journal of Network and Systems Management*, 4(3):279–297, 1996.

[13] 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.

[14] R. Focardi, R. Gorrieri, and F. Martinelli. Non Interference for the Analysis of Cryptographic Protocols. In U. Montanari, J.D.P. Rolim, and E. Welzl, editors, *Proc. of Int. Colloquium on Automata, Languages and Programming (ICALP'00)*, volume 1853 of *LNCS*, pages 744–755. Springer-Verlag, 2000.

[15] R. Focardi, R. Gorrieri, and F. Martinelli. A comparison of three authentication properties. *TCS*, 291(3):285–327, 2003.

[16] R. Focardi, R. Gorrieri, and F. Martinelli. Classification of Security Properties (Part II: Network Security). In R. Focardi and R. Gorrieri, editors, *Foundations of Security Analysis and Design II - Tutorial Lectures*, volume 2946 of *LNCS*. Springer-Verlag, 2004.

[17] R. Focardi and S. Rossi. Information Flow Security in Dynamic Contexts. In *Proc. of the 15th IEEE Computer Security Foundations Workshop*, pages 307–319. IEEE Computer Society Press, 2002.

[18] R. Gorrieri, E. Locatelli, and F. Martinelli. A Simple Language for Real-time Cryptographic Protocol Analysis. In *Proc. of 12th European Symposium on Programming Languages and Systems*, volume 2618 of *LNCS*, pages 114–128. Springer-Verlag, 2003.

[19] 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.

[20] R. Milner. *Communication and Concurrency*. Prentice-Hall, 1989.

[21] A. W. Roscoe. *The Theory and Practice of Concurrency*. Series in Computer Science. Prentice Hall, 1998.

[22] S. Schneider. Formal Analysis of a Non-Repudiation Protocol. In *Proc. of the IEEE Computer Security Foundations Workshop*, pages 54–65. IEEE Computer Society Press, 1998.

[23] V. Shmatikov and J. C. Mitchell. Analysis of a Fair Exchange Protocol. In *Proc. of 7th Annual Symposium on Network and Distributed System Security (NDSS 2000)*, pages 119–128. Internet Society, 2000.

[24] J. Zhou and D. Gollmann. A Fair Non-Repudiation Protocol. In *Proc. of the IEEE Symposium on Security and Privacy*, pages 55–61. IEEE Computer Society Press, 1996.