V. Cortier

Observational equivalence and trace equivalence in an extension of Spi–calculus.

Application to cryptographic protocols analysis

Research Report LSV–02–3, Mar. 2002

Laboratoire Spécification et Vérification

Ecole Normale Supérieure de Cachan
61, avenue du Président Wilson
94235 Cachan Cedex France

http://www.lsv.ens-cachan.fr/Publis/
Research Report LSV-02-3, Lab. Spécification et Vérification, CNRS & ENS de Cachan, France, Mar. 2002

Observational equivalence and trace equivalence in an extension of Spi-calculus. Application to cryptographic protocols analysis. Extended version

Véronique Cortier
Laboratoire Spécification et Vérification, CNRS and ENS de Cachan
cortier@lsv.ens-cachan.fr

Abstract. Relating the spi-calculus with a trace based model is a key if we want to automate security proofs in these models. In this paper, our contribution is to propose an extension of the model allowing an arbitrary set of function symbols (either one-way or invertible, either public or private) and predicates: agents may recognize some type of messages. We extend the results of [7] to this case and provide a simple decision procedure for static equivalence between environments. We illustrate our results, providing a proof of the secrecy of Needham-Schroeder-Lowe protocol including replication. This opens the way to a deduction system for cryptographic properties expressed in the spi-calculus without bounding the number of process replications.

1 Introduction

Cryptographic protocols aim at achieving goals like authentication and secrecy. They become more and more important with the development of communications via public networks. Such protocols are notoriously difficult to verify and some of them have very serious flaws. Consequently, there has been a growing interest in applying formal methods for verifying cryptographic protocols.

Lots of models have been developped for cryptographic protocols, for instance strand spaces [22], process calculi [21, 20, 19, 4, 5] and state-transition systems inspired by Paulson [18], and developped by J. Millen and H. Ruess in [15]. These models are associated with automated deduction tools (see e.g. [6, 13]) but they often assume additional hypotheses, e.g. the absence of compound keys [22, 18, 15] and the absence of replication [20, 4]. Moreover, these models do not allow to express arbitrary security properties: they focus on secrecy and authentication.

On the other hand, in the spi-calculus [2], security properties are expressed using observational equivalence, which is more appropriate. The attacker is also more powerful in this model: for instance he can compare messages, which is a realistic capability. Unfortunately, proving properties in spi-calculus is hard; currently there is no tool allowing to automate these proofs. Boreale *et al* [7] and Abadi, Fournet [1] bridge the gap between spi-calculus and trace-based models: they develop a labelled transition system where transitions are constrained by the knowledge of the environment. Then they define a trace equivalence and

a weak bisimilarity equivalence that avoid quantification over contexts. Weak bisimilarity is shown to be equivalent to "barbed equivalence". This leads to more direct proof methods. Unfortunately, [7] still assumes severe restrictions: for instance public keys and compound keys are not allowed. In [1], there is no distinction between public and private symbols which may be hard to encode and moreover, there is no effective method to decide the static equivalence.

In this paper, we establish results similar to [7] in a more general framework:

- we allow public key encryption, which is an important feature of many protocols [9].
- we allow compound keys, which are used in many actual protocols like SSL [16]. Adding such a possibility is not innocent: this increases considerably the difficulty of decision techniques (see [12]).
- we add an arbitray set of function symbols, together with their properties (either one-way or invertible, either public or private).
- we parametrize the ability of agents to recognize any regular set of messages. Typically, we can choose to give the ability to recognize that a message is a key, a nonce, an agent name, a hashed message or only a part of this. Again, this is not innocent: by adding or removing such an ability, we may switch from decidability to decidability (see [11]).

This generalization relies on another characterization of environment equivalence, based on tree automata techniques. Our technique is more general than in [7] and , we think, simpler: this is our second main contribution.

In some respects, our equivalence result between weeak bisimilarity and barbed equivalence is less general than in [1]: we do not allow arbitrary equational theories for the messages. This is the price to pay for decidability of static equivalence between environments.

Our results open the way to a deduction system for cryptographic properties expressed as observational equivalences in spi-calculus . We illustrate this application providing a proof of the secrecy of Needham-Schroeder-Lowe procotol. This example is more general than those given in [7] in the sense that it allows replication of processes and public keys. A proof of equivalence between protocols with replication has already been provided in [2] but we believe that our proof technique is more intuitive.

The paper is structured as follows: Section 2 presents an extension of the language defined in [7]. In Section 3, we extend the definition of bisimulation semantics and give a new characterization of the equivalence between environments. In Section 4, we define the characteristic process, which allows to prove soundness and completeness of bisimulation with respect to barbed equivalence. In section 5, we list some properties on bisimulation semantics and we apply them to the proof of Needham-Schroeder-Lowe protocol. Concluding remarks are presented in Section 6.

2 The language

2.1 Syntax

The syntax of the calculus is summarized in Figure 1. The major difference with the spi-calculus of [7] is the description of the messages: they are built out of a set of function symbols $\mathcal{F}$, which is split into:

$a, b, \ldots, h, k, \ldots, x, y, z, \ldots$			names $\mathcal{N}$
$M, N, M_1, \ldots, M_n$	$::=$	$a \mid \{M\}_N \mid f(M_1, \ldots, M_n)$	messages $\mathcal{M}$ $\forall f \in \mathcal{IF} \uplus \mathcal{OF}$
$\zeta, \eta, \zeta_1, \ldots \zeta_n$	$::=$	$a \mid f(\zeta_1, \ldots, \zeta_n)$ $\mid \mathsf{dec}_\eta(\zeta) \mid g_i^{-1}(\zeta)$	expressions $\mathcal{Z}$ $\forall f \in \mathcal{VF}$, $\forall g \in \mathcal{IF}, \forall i \leq arity(g)$
ϕ, ψ	$::=$	$tt \mid [\zeta = \eta] \mid \mathsf{let}\ z = \zeta\ \mathsf{in}\ \phi$ $\mid V_1(\zeta) \mid \ldots \mid V_n(\zeta)$ $\mid \phi \wedge \psi \mid \phi \vee \psi \mid \neg\phi$	formulae Φ $V_i(\zeta)$ predicates
P, Q	$::=$		processes $\mathcal{P}$
		$\mathbf{0}$	(null)
	$\mid$	$\eta(x).P$	(input prefix)
	$\mid$	$\overline{\eta}(\zeta).P$	(output prefix)
	$\mid$	$P + Q$	(non-deterministic choice)
	$\mid$	$P \mid Q$	(parallel composition)
	$\mid$	$(\nu a)P$	(restriction)
	$\mid$	$!P$	(replication)
	$\mid$	ϕP	(boolean guard)
	$\mid$	$\mathsf{let}\ z = \zeta\ \mathsf{in}\ P$	(assignment)

Fig. 1. Syntax of the calculus

names : an infinite set $\mathcal{N}$ of constants standing for names.

invertible symbols : a set $\mathcal{IF}$ corresponding to constructions whose components are computable by an intruder. Typically, the *pairing* functions belong to $\mathcal{IF}$.

one way symbols : a set $\mathcal{OF}$ corresponding to constructions whose components cannot be computed by an intruder. Typically, *hash functions* belong to $\mathcal{OF}$. We assume that there are two unary function symbols $\mathsf{pub}()$ and $\mathsf{prv}()$ which correspond to public and private keys.

partially invertible symbols which correspond to constructions whose components can be computed by an intruder, subject to some knowledge of the

intruder. The only partially invertible symbol considered here is encryption. Such a binary function, written $\{u\}_k$, verifies that the intruder can build $\{u\}_k$ when he knows u and k and can retrieve u when he knows $\{u\}_k$ and the inverse key k^{-1}. Every term k can be considered as a symmetric key: $k^{-1} = k$, except $\mathsf{pub}(a)$ and $\mathsf{prv}(a)$ which are inverse of each other.

To summarize: $\mathcal{F} = \mathcal{N} \uplus \mathcal{IF} \uplus \mathcal{OF} \uplus \{\{_\}_\}$ such that $\mathsf{pub}(), \mathsf{prv}() \in \mathcal{OF}$.

In addition, some of the function symbols are publicly available, while others are invisible to all agents. We split $\mathcal{F}$ into two sets (this is orthogonal to the above decomposition): $\mathcal{PF}$ is the subset of $\mathcal{F}$ of *private symbols* and $\mathcal{VF}$ is the subset of *visible (or public) symbols*. For instance $\mathsf{prv}(), \mathsf{pub}()$ belong to $\mathcal{PF}$, while pairing, encryption and hashing are in $\mathcal{VF}$.

Compared to [7], we restore the possibility for keys to be compound messages as in Abadi and Gordon's spi-calculus [2]. In addition, we introduce public and private keys and one-way functions like hashing.

Abadi and Fournet encode also asymmetric encryption in [1] but in a slightly different way. Since they do not distinguish between public and private function, they need to introduce fresh nonces in the functions *pub* and *prv* and send explicitly every public key. This requires at least to be more careful and precise while using asymmetric encryption: instead of having the hypotheses "the intruder knows every public key and the intruder cannot compute the private keys", one should verify that the encoding of asymmetric functions verifies these hypotheses.

Expressions are modified consequently, e.g. using $f(\zeta_1, \ldots, \zeta_n)$ when $f \in \mathcal{VF}$ and $g^{-1}(\zeta)$ when $g \in \mathcal{IF}$. The result of evaluating $\mathsf{dec}_\eta(\zeta)$ is obtained by decrypting the ciphertext ζ using the value of η as key.

Logical *formulae* are also extended: we add any number of predicates V_i among predicates V whose interpretation S_V is either a set of names $S_P \subset \mathcal{N}$ or a set $S_f = \{f(m_1, \ldots, m_n) \mid m_i \in \mathcal{M}\}$ for some $f \in \mathcal{F}$. This amounts to give the ability for agents (and intruders) to recognize some types messages such as agent names, keys or hashed messages. In [1], there is no predicate and in [7], there is only one predicate *name* which distinguishes atomic names from compound message: in this case, $S_{name} = \mathcal{N}$. We emphasize here that our processes are parametrized by the set of predicates and our results hold for any such set of predicates. This leaves more flexibility in the specification of intruder and principals capabilities.
Processes, as recalled in figure 1, are built exactly as in [7].
Protocols are built exactly as processes except that expressions in protocols may use private functions. More formally, *private expressions* are defined by:

$$\zeta, \eta, \zeta_1, \ldots, \zeta_n ::= a \mid f(\zeta_1, \ldots, \zeta_n) \mid \mathsf{dec}_\eta(\zeta) \mid g_i^{-1}(\zeta) \qquad \forall f \in \mathcal{F}, \forall g \in \mathcal{IF}, \forall i \leq \mathit{arity}(g).$$

2.2 Operational semantics

The operational semantic defined in [7] is modified to take into account the extensions of the model such as encryption with private, public and compound keys.

We redefine the evaluation function for expressions, $\widehat{\ }: \mathcal{Z} \to \mathcal{M} \uplus \{\bot\}$, by induction on ζ as follows:

$$\widehat{a} = a$$

$$\widehat{f(\zeta_1, \ldots, \zeta_n)} = \begin{cases} f(M_1, \ldots, M_n) \text{ if } \widehat{\zeta_1} = M_1, \ldots, \widehat{\zeta_n} = M_n,\ f \in \mathcal{VF} \\ \bot \qquad\qquad\qquad \text{otherwise.} \end{cases}$$

$$\widehat{\mathsf{dec}_{\zeta_2}(\zeta_1)} = \begin{cases} M \text{ if } \widehat{\zeta_1} = \{M\}_k \text{ and } \widehat{\zeta_2} = k^{-1}, \\ \bot \text{ otherwise.} \end{cases}$$

$$\widehat{f_i^{-1}(\zeta)} = \begin{cases} M_i \text{ if } \widehat{\zeta} = f(M_1, \ldots, M_n),\ f \in \mathcal{IF} \\ \bot \text{ otherwise.} \end{cases}$$

Compared to [1], we fix here the underlying equational theory:
$\mathsf{dec}_{\mathsf{pub}(x)}(\{y\}_{\mathsf{prv}(x)}) = y$, $f_i^{-1} f(x_1, \ldots, x_n) = x_i$, ...
However, since we allow an arbitrary number of function symbols (one-way or invertible), we think that it is sufficient to encode the common destructor-constructor rules of cryptography. Of course, we cannot encode functions like XOR.

The evaluation function for formulae, $[\![\cdot]\!] : \Phi \to \{tt, ff\}$ is defined by induction on ϕ. The only non-standard clauses are $V_i(\zeta)$ and $\mathsf{let}\, z = \zeta \,\mathsf{in}\, \phi$:

$$[\![V_i(\zeta)]\!] = \begin{cases} tt & \text{if } \widehat{\zeta} \in S_{V_i}, \\ ff & \text{otherwise (in particular if } \widehat{\zeta} = \bot). \end{cases}$$

$$[\![\mathsf{let}\, z = \zeta \,\mathsf{in}\, \phi]\!] = \begin{cases} [\![\phi(\zeta/z)]\!] & \text{if } \widehat{\zeta} \neq \bot, \\ ff & \text{otherwise.} \end{cases}$$

For any substitution σ, $\sigma \models \phi$ means $[\![\phi\sigma]\!] = tt$.
We sometimes write $[\zeta \neq \bot]$ instead of $\mathsf{let}\, z = \zeta \,\mathsf{in}\, tt$ and $[\zeta = \bot]$ instead of $\neg(\mathsf{let}\, z = \zeta \,\mathsf{in}\, tt)$. $fn(\zeta)$, $fn(\phi)$ denote the free variables of ζ and ϕ.
The operational semantics defined by the inference rules of Figure 2 (Table 2 of [7]) describes the process actions.

Exactly as [7], process *actions* (i.e. labels of the transition system), typically written $\mu, \lambda, \ldots$ can be of three forms: τ (internal action), aM (input at a where message M is received) and $\nu(\tilde{b})\bar{a} < M >$ (output at a where message M containing the fresh, private names $\tilde{b}$ is sent). Input and output actions will be called *visible*. We use s to range over sequences of visible actions (traces), and write $\Rightarrow$ or $\stackrel{\epsilon}{\Rightarrow}$ to denote the reflexive and transitive closure of $\stackrel{\tau}{\rightarrow}$ and, inductively, $\stackrel{s}{\Rightarrow}$ for $\Rightarrow\stackrel{\mu}{\rightarrow}\stackrel{s'}{\Rightarrow}$ when $s = \mu \cdot s'$.

Example 1. We consider

$$A(s) \stackrel{\text{def}}{=} \bar{p}(< a, b >) \,|\, p(x).\, \mathsf{let}\, z = \mathsf{dec}_{\mathsf{prv}(a)}(x) \,\mathsf{in}\, \bar{p}(\{s\}_z).\mathbf{0}, \quad B \stackrel{\text{def}}{=} (\nu k)\bar{p}(\{k\}_{\mathsf{pub}(a)}).\mathbf{0}$$

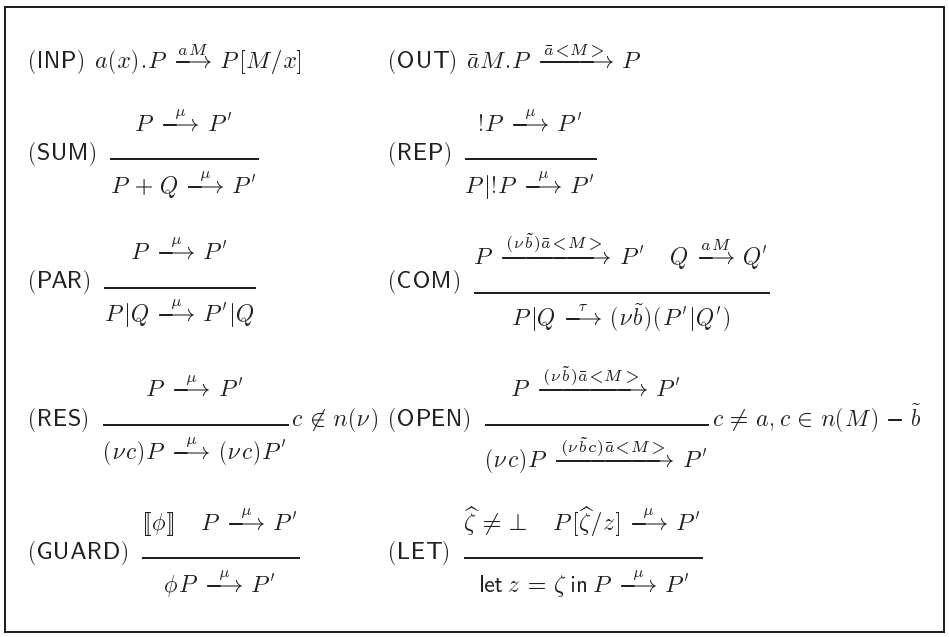

Fig. 2. Operational semantics (symmetric versions of (SUM), (PAR) and (COM) omitted)

and $P(s) \stackrel{\text{def}}{=} A|B$. Intuitively, a sends her name and b's name to signify her intention to talk to b. Then b sends to a a new fresh key k encrypted with the public key of a (second step) and a answers with a secret s encrypted with the key k of b, that she has just received (last step). Then, a possible sequence of actions for P is

$$P(s) \xrightarrow{\bar{p}<a,b>} P'(s) \xrightarrow{\tau} P''(s) \xrightarrow{\bar{p}(\{s\}_k)} \mathbf{0}$$

where $P'(s) \stackrel{\text{def}}{=} p(x).\,\mathsf{let}\ z = \mathsf{dec}_{\mathsf{prv}(a)}(x)\ \mathsf{in}\ \bar{p}(\{s\}_z).\mathbf{0} \mid (\nu k)\bar{p}(\{k\}_{\mathsf{pub}(a)}).\mathbf{0}$
and $P''(s) \stackrel{\text{def}}{=} (\nu k)\,\mathsf{let}\ z = \mathsf{dec}_{\mathsf{prv}(a)}(\{k\}_{\mathsf{pub}(a)})\ \mathsf{in}\ \bar{p}(\{s\}_z).\mathbf{0}$.

2.3 Barbed equivalence

We recall here the definition of *barbed equivalence* as defined in [2] or [7].

A process P *commits to* a, written $P \downarrow a$ if $P \xrightarrow{aM} P'$ or $P \xrightarrow{(\nu\tilde{b})\bar{a}<M>} P'$, for some M, $\tilde{b}$ and P'. We also write $P \Downarrow a$ if $P \Longrightarrow P'$ and $P' \downarrow a$, for some P'. $P \Downarrow a$ if P may output or input on a, where a is called a *barb*.
Observers, typically written $R, R', \ldots$ are processes that can perform a distinct "success" action w. For example, the observer $R \stackrel{\text{def}}{=} p(x).[x = n]w$ listens to channel p and tests the ability of the observed process to send n along the channel p. If one interprets "committing to a" as "revealing a piece of information" then two

processes P and Q are barbed equivalent if they reveal the same informations and have the same behavior, even in parallel with any observer R.

Definition 1 (barbed equivalence). *A symmetric relation $\mathcal{S} \subseteq \mathcal{P} \times \mathcal{P}$ is a* barbed bisimulation *if whenever $P\mathcal{S}Q$ then:*

1. *for each P', if $P \xrightarrow{\tau} P'$ then there is Q' such that $Q \Longrightarrow Q'$ and $P'\mathcal{S}Q'$, and*
2. *for each a, if $P \downarrow a$ then $Q \Downarrow a$.*

Barbed bisimilarity*, written $\dot{\cong}$, is the largest barbed bisimulation relation. Two processes P and Q are* barbed equivalent*, if for all R we have $P|R \dot{\cong} Q|R$.*

Barbed equivalence is used to define secrecy. From now on, a process P is *secret* for the variable z if for all messages M, M', $P(M/z) \cong P(M'/z)$. Other notions of secrecy can be defined with other notions of congruence between processes, see [2] for more details.
Remark : If there is no private function, we are back to the barbed equivalence defined in [1, 2, 7].

Example 2. Let s, s' be two terms and consider $P(s), P(s')$ as defined in Example 1. Then it can be shown that $P(s) \dot{\cong} P(s')$. This means intuitively that a passive attacker cannot tell the difference between $P(s)$ and $P(s')$. But $P(s) \not\cong P(s')$. Indeed, if an active attacker sends to a his own private key k' encrypted with a's public key, then a will respond with the secret s encrypted with the intruder key k' (instead of using the key sent by b) and the intruder will be able to tell the difference between $P(s)$ and $P(s')$. More formally, let $R = p(x).\bar{p}(\{k'\}_{\mathsf{pub}(x)}).p(y).\,\mathsf{let}\, z = \mathsf{dec}_{k'}(y)\,\mathsf{in}\,[z = s].\bar{k'}(0).\mathbf{0}$, then $P(s)|R$ can commit to k' after some internal moves while $P(s')|R$ cannot. Thus P is not secret according to our notion of secrecy.

In general, proving that two processes are barbed equivalent is difficult because of the quantification over the contexts R. That is why we associate with spi-calculus a trace model such that barbed equivalence can be expressed as an equivalence on the trace model.

3 Trace and Bisimulation Semantics

3.1 Transitions

States are *configurations* of the form $\sigma \triangleright P$, where P is a process and σ is a substitution which represents the environment. Transitions take the form

$$\sigma \triangleright P \overset{\mu}{\underset{\delta}{\longmapsto}} \sigma' \triangleright P'$$

and represent atomic interactions between the process P and the environment σ. Note that σ is not necessarily finite. μ is the process action (input, output or

τ) and δ is the complementary *environment action*. The latter can be of three forms, output, input and "no action":

$$\delta ::= (\nu\tilde{b})\overline{\eta}(\zeta) \mid \eta(\zeta) \mid -$$

The inference rules for the relation $\stackrel{\mu}{\underset{\delta}{\longmapsto}}$ are displayed in Figure 3 (Table 3 of [7]). Intuitively, μ denotes the action viewed by the process P and δ describes the action viewed by the environment σ. In the following, we write $\Longrightarrow$ to denote the reflexive and transitive closure of $\stackrel{\tau}{\underset{-}{\longmapsto}}$ and, inductively, write $\stackrel{s}{\underset{u}{\Longrightarrow}}$ for $\Longrightarrow\stackrel{\mu}{\underset{\delta}{\longmapsto}}\stackrel{s'}{\underset{u'}{\Longrightarrow}}$ when $s = \mu \cdot s'$ and $u = \delta \cdot u'$.

It is assumed that $n(\eta) \subseteq dom(\sigma)$ and that names in $\tilde{b}$ are fresh for σ and P.

$$\mathsf{E-OUT}\ \frac{P \xrightarrow{(\nu\tilde{b})\bar{a}<M>} P' \quad \widehat{\eta\sigma} = a}{\sigma \triangleright P \stackrel{(\nu\tilde{b})\bar{a}<M>}{\underset{\eta(x)}{\longmapsto}} \sigma[M/x] \triangleright P'} \qquad \mathsf{E-TAU}\ \frac{P \xrightarrow{\tau} P'}{\sigma \triangleright P \stackrel{\tau}{\underset{-}{\longmapsto}} \sigma \triangleright P'}$$

$$\mathsf{E-INP}\ \frac{P \xrightarrow{a\,M} P' \quad \widehat{\eta\sigma} = a \quad M = \widehat{\zeta\sigma} \quad \tilde{b} \stackrel{\text{def}}{=} (n(\zeta) - dom(\sigma))}{\sigma \triangleright P \stackrel{a\,M}{\underset{(\nu\tilde{b})\bar{\eta}(\zeta)}{\longmapsto}} \sigma[\tilde{b}/\tilde{b}] \triangleright P'}$$

Fig. 3. Rules for environment-sensitive lts

If $\sigma = [M_i/x_i]_{i\in J}$, $range(\sigma)$ denotes $\{M_i \mid i \in J\}$ and $dom(\sigma)$ denotes the domain of σ: $\{x_i \mid i \in J\}$. In addition, we sometimes write $\sigma = [M_1, \ldots, M_n]$ instead of $[M_i/x_i]_{1\leq i\leq n}$ and $M \in \sigma$ instead of $M \in range(\sigma)$.

Example 3. Consider $\sigma \stackrel{\text{def}}{=} [p/x_1, s/x_2]$ and $P(s)$ defined in Example 1. Note that putting s in σ does not mean that σ really "knows" the secret s but means that σ is now able to recognize s if s appears again. The same attack than in Example 2 can be played here: $P(s)$ sends the names of a and b that σ learns (first step). Then σ sends its new key k encrypted with a's public key (second step). Then the process is ready to send the secret encrypted with k. This is formally described by the following transitions:

$$\sigma \triangleright P(s) \stackrel{\bar{p}<a,b>}{\underset{x_1(x_3)}{\longmapsto}} \sigma[< a, b > /x_3] \triangleright P'(s) \stackrel{p(\{k\}_{\mathsf{pub}(a)})}{\underset{(\nu k)\bar{x_1}(\{k\}_{\mathsf{pub}(<>_1^{-1}(x_3))})}{\longmapsto}} \sigma[< a, b > /x_3, k/k] \triangleright P''(s)$$

where $P''(s) \stackrel{\text{def}}{=} \mathsf{let}\ z = \mathsf{dec}_{\mathsf{prv}(a)}(\{k\}_{\mathsf{pub}(a)})\ \mathsf{in}\ \bar{p}(\{s\}_z).\mathbf{0} \mid (\nu k)\bar{p}(\{k\}_{\mathsf{pub}(a)}).\mathbf{0}$.

3.2 Environment

Definition 2. *[7] Two substitutions σ and σ' are* equivalent, *written $\sigma \sim \sigma'$, if $dom(\sigma) = dom(\sigma')$ and for every formula ϕ with $fn(\phi) \subseteq dom(\sigma)$, $\sigma \models \phi$ if and only if $\sigma' \models \phi$.*

A difficulty is to effectively decide the equivalence $\sim$ since in this (observational) definition, there is a universal quantification over ϕ. In [1], $\sim$ is not decidable due to the arbitrary equational theory. In [7], they give a characterization of $\sim$ using an effective relation $\sim'$. Here, we cannot use $\sim'$ because of our additional constructions. We are going instead to characterize $\sim$ using a simpler and more general approach. Let us first define the knowledge of σ, i.e., the set of terms that the environment can build.

Definition 3. *Let W be a set of messages. The knowledge closure of W, written $kn(W)$, is the smallest set of messages such that:*

1. $W \subseteq kn(W)$
2. $\forall f \in \mathcal{VF}$, *if* $M_1, \ldots, M_n \in kn(W)$, *then* $f(M_1, \ldots, M_n) \in kn(W)$,
3. $\forall f \in \mathcal{IF}$, $\forall i \leq arity(f)$, *if* $f(M_1, \ldots, M_n) \in kn(W)$, *then* $M_i \in kn(W)$,
4. *if* $\{M\}_k \in kn(W)$ *and* $k^{-1} \in kn(W)$, *then* $M \in kn(W)$,

If σ is a substitution, $kn(\sigma) = kn(range(\sigma))$. $kn(W)$ is infinite but we can handle this set with tree automata:

Proposition 1. *Given a set of messages W, such that W is recognizable by a finite tree automaton, then $kn(W)$ is recognizable by a finite tree automaton.*

proof sketch: A set W is recognizable by a tree automaton if and only if there exists a definite set constraint $\mathcal{C}$ such that $W = \sigma(S)$, where σ is the minimal solution of $\mathcal{C}$ (see [8]). From $\mathcal{C}$, we construct a definite set constraint $\mathcal{C}'$ containing $\mathcal{C}$ and the inclusions given in Figure 4. Then $kn(W) = \sigma'(R)$ where σ' is the minimal solution of $\mathcal{C}'$. This is a simple extension of results already presented in [14] and [17].

$$
\begin{array}{ll}
S \subseteq R & \{\}_1^{-1}\left(\{Msg\}_{R \cap f(Msg,\ldots,Msg)}\right) \subseteq R \\
\{\}_1^{-1}\left(\{Msg\}_{\mathsf{pub}(\mathsf{prv}^{-1}(R))}\right) \subseteq R & \{\}_1^{-1}\left(\{Msg\}_{\mathsf{prv}(\mathsf{pub}^{-1}(R))}\right) \subseteq R \\
h(R, \ldots, R) \subseteq R & g_i^{-1}(R) \subseteq R \\
\multicolumn{2}{c}{\forall f \in \mathcal{F} \quad f \neq \mathsf{pub}, f \neq \mathsf{prv} \quad \forall h \in \mathcal{VF}, \quad \forall g \in \mathcal{IF}\ \forall i \leq arity(g)}
\end{array}
$$

Fig. 4. Set constraints for the knowledge of the intruder

We can now give an alternative definition of equivalence on environments. For every message M, $\overline{M}^{\sigma}$ is the message in which every non-decomposable part is replaced by a variable. A similar decomposition was developped in [3] in a less general context : no compound keys, no comparison, ... The ability of the environment to compare messages (even if it does not know the message in clear) is reflected by assigning the same variable to different occurrences of the same term.
We first define the set $path(t)$ of paths of a term t as usual, except that we skip the symbol $()^{-1}$.

Definition 4. *The set of* paths *of a term t, denoted by $path(t)$, is defined inductively by: $path(a) = \{1\}$ if a is a constant, $path(f(t_1, \ldots, t_n)) = \{\epsilon\} \cup 1.path(t_1) \cup \ldots \cup n.path(t_n)$ if $f \in \mathcal{F}$, and $path(t^{-1}) = path(t)$.*
Let $\sigma = [M_i/x_i]_{1 \leq i \leq n}$ be substitution. We define $path(\sigma) = \cup_{1 \leq i \leq n} i.path(M_i)$.
$t|_p$ is defined inductively by: $t|_\epsilon = t$ and $f(t_1, \ldots, t_n)|_{i.p} = t_i|_p$.

$t|_p$ is the subterm of t at p. $leaves(t)$ is the set of maximal paths in $path(t)$. The sequence of symbols encountered along the path p is written $s_t(p)$. Formally, $s_t(p)$ is defined inductively by $s_t(\epsilon) = \epsilon$ and $s_{f(t_1,\ldots,t_n)}(i.p') = f.s_{t_i}(p')$.

Definition 5. *Let σ be a substitution. Var_σ is the set of variables X_t such that t is a subterm of a message M in σ. We define $\overline{\cdot}^{\sigma} : \mathcal{T}(\mathcal{F}) \to \mathcal{T}(\mathcal{F} \cup Var_\sigma)$ by induction as follows: (we write $\overline{t}$ when σ is clear from the context.)*

$$\overline{a} = X_a \text{ if } a \text{ is a constant symbol} \tag{1}$$

$$\overline{f(t_1, \ldots, t_n)} = f(\overline{t_1}, \ldots, \overline{t_n}) \text{ if } f \in \mathcal{IF} \tag{2}$$

$$\overline{g(t_1, \ldots, t_n)} = \begin{cases} g(\overline{t_1}, \ldots, \overline{t_n}) \text{ if } g \in \mathcal{VF} \text{ and } t_1, \ldots, t_n \in kn(\sigma) \\ X_{g(t_1,\ldots,t_n)} \quad \text{otherwise} \end{cases} \tag{3}$$

$$\overline{\{t_1\}_{t_2}} = \begin{cases} \{\overline{t_1}\}_{\overline{t^{-1}}} \text{ if } t_2 = t^{-1} \text{ such that } t \in kn(\sigma) \\ \{\overline{t_1}\}_{\overline{t_2}} \quad \text{if both } t_1, t_2 \text{ are in } kn(\sigma) \text{ (as in 3)} \\ X_{\{t_1\}_{t_2}} \quad \text{otherwise} \end{cases} \tag{4}$$

$$\overline{t} = X_t \text{ otherwise} \tag{5}$$

Let $\mathcal{P}$ be the set of paths of $\overline{\sigma}^{\sigma}$. For every predicate V_i, we define the set
$A^i_\sigma \stackrel{def}{=} \{\zeta \in \mathcal{T}(\mathcal{VF} \cup \{x_p | p \in \mathcal{P}\}) \mid [\![V_i(\zeta[\sigma|_p/x_p]_{p \in \mathcal{P}})]\!] = tt\}$.
Let $\sigma = [M_i/x_i]_{i \in I}$ and $\sigma' = [M'_i/x_i]_{i \in I}$ be two substitutions and $\overline{\sigma}^{\sigma} \stackrel{def}{=} < \overline{M_i} >_{i \in I}$.
$\sigma \sim' \sigma'$ iff

1. *there exists a renaming $\theta : Var_\sigma \to Var_{\sigma'}$ such that $\overline{\sigma}^{\sigma}\theta = \overline{\sigma'}^{\sigma'}$ and θ* preserves the key types, *i.e., if $\theta(X_t) = X_{t'}$ then $t^{-1} = t$ iff $t'^{-1} = t'$.*
2. *and if for every $1 \leq i \leq n$, $A^i_\sigma = A^i_{\sigma'}$.*

Remark: If σ is finite, $\overline{\sigma}^{\sigma}$ is effectively computable using Proposition 1. Moreover, if the interpretation of V_i is a regular language, then A^i_σ is also a regular language (proof in A), thus A^i_σ is effectively computable.

Example 4. Let $\sigma_1 = [\{s\}_k, \{s\}_{\{s\}_k}, s]$ and $\sigma_2 = [\{s\}_k, \{s\}_{\{s\}_k}, s']$.
Then $\overline{\sigma_1}^{\sigma_1} =< X_{\{s\}_k} >, \{X_s\}_{X_{\{s\}_k}}, X_s >$ and $\overline{\sigma_2}^{\sigma_2} =< X_{\{s\}_k} >, \{X_s\}_{X_{\{s\}_k}}, X_{s'} >$.
Thus $\sigma_1 \not\sim' \sigma_2$. Intuitively, one can detect that for σ_1, the third message encrypted with the first one is equal to the second one, which is not true for σ_2.

The following theorem extends one of the main result of [7]:

Theorem 1 (coincidence of σ and σ'). *For any two substitutions σ and σ', $\sigma \sim \sigma'$ if and only if $\sigma \sim' \sigma'$. Hence $\sim$ is decidable.*

proof sketch (full proof in Appendix A): $\sim' \subseteq \sim$ is proved by induction on the length of formula ϕ which verifies $\sigma \models \phi$.

The main idea to prove $\sim \subseteq \sim'$ is the construction, for each path p of $\overline{\sigma}^{\sigma}$, of finitely many formulas ϕ_p^i such that if $\sigma \not\sim \sigma'$, then for at least one $p \in path(\overline{\sigma})$ and one i, $\sigma \models \phi_p^i$ and $\sigma' \not\models \phi_p^i$. For instance:

Proposition 2. *Let $\sigma = [t_1/x_1, \ldots, t_n/x_n]$ be a substitution. If p is a path in $\overline{\sigma}^{\sigma}$ then there exist ϕ_p and ζ_p such that $fn(\phi_p), fn(\zeta_p) \subset dom(\sigma)$, $\widehat{\zeta\sigma_p} = \sigma|_p$ and $\sigma \models \phi_p$. In addition, ϕ_p and ζ_p* characterize *p in the sense that for every substitution σ', if $dom(\sigma) = dom(\sigma')$ then*
$\sigma' \models \phi_p$ and $\widehat{\zeta_p\sigma'} \neq \bot \Rightarrow \widehat{\zeta_p\sigma'} = \sigma'|_p$, $s_\sigma(p) = s_{\sigma'}(p)$ and p is a path of $\overline{\sigma'}^{\sigma'}$.

3.3 Trace-based Equivalences

We are now ready to define trace-based equivalences as in [7].
In what follows $\sigma \triangleright P \stackrel{\widehat{\mu}}{\underset{\delta}{\Longrightarrow}} \sigma' \triangleright P'$ stands for $\sigma \triangleright P \stackrel{\mu}{\underset{\delta}{\Longrightarrow}} \sigma' \triangleright P'$ if $\mu \neq \tau$, and for $\sigma \triangleright P \stackrel{\mu}{\underset{\delta}{\Longrightarrow}} \sigma' \triangleright P'$ if $\mu = \tau$ (the $\widehat{\cdot}$ defined here has of course nothing to do with the evaluation function on expressions defined in the previous section).
A relation $\mathcal{R}$ is *compatible* if it only contains pairs of configurations $(\sigma_1 \triangleright P, \sigma_2 \triangleright Q)$ with $\sigma_1 \sim \sigma_2$. Given a binary relation $\mathcal{R}$, we write $(\sigma_1, \sigma_2) \vdash P\mathcal{R}Q$ if $(\sigma_1 \triangleright P, \sigma_2 \triangleright Q) \in \mathcal{R}$.

Definition 6 (weak bisimulation). *Let $\mathcal{R}$ be a compatible relation. $\mathcal{R}$ is a* weak bisimulation *if whenever $(\sigma_1, \sigma_2) \vdash P\mathcal{R}Q$ and $\sigma_1 \triangleright P \stackrel{\mu}{\underset{\delta}{\longmapsto}} \sigma_1' \triangleright P'$ then there are μ', σ_2' and Q' such that $\sigma_2 \triangleright Q \stackrel{\widehat{\mu'}}{\underset{\delta}{\Longrightarrow}} \sigma_2' \triangleright Q'$ and $(\sigma_1', \sigma_2') \vdash P'\mathcal{R}Q'$ and symmetrically exchanging Q and P.* Bisimilarity*, written $\approx$, is the largest weak bisimulation relation.*

See [7] for more details and examples.

4 Soundness and Completeness

In this section, we show that, on traces, the barbed equivalence coincides with bisimilarity, opening a way to a deduction system for the observational equivalence of spi-calculus processes.

The proof of soundness ($\approx \subseteq \cong$) is exactly the same as in [7]. For completeness, we have to extend the notion of characteristic formula of an environment. Indeed, in [7], the main ingredient in the proof is the notion of characteristic formula ϕ_σ which verifies

$$\forall \sigma' \quad \sigma' \models \phi_\sigma \Leftrightarrow \sigma \sim \sigma'. \qquad (*)$$

With our syntax extension, it is no longer possible to get a single formula ϕ_σ. Intuitively, an arbitrary long formula can be needed to distinguish two environments.

Thus, we define in Section 4.1 the *characteristic process* of an environment and complete the completeness proof in Section 4.2.

4.1 Characteristic Processes

Even if we no longer have a unique formula ϕ_σ satisfying the property $(*)$, we can construct a formula ϕ_σ which recognizes every σ', extending σ: for each path p of $\overline{\sigma}$, let ϕ_p and ζ_p as in Proposition 2 and let

$$\phi_\sigma \stackrel{\text{def}}{=} \bigwedge_{p \in leaves(\overline{\sigma}^\sigma)} \phi_p \wedge [f^{-1}(\zeta_p) = \bot] \wedge \epsilon_\sigma[\mathsf{dec}_{\zeta_p}(\{\zeta_p\}_{\zeta_p}) = \bot] \wedge \bigwedge_{p' \in leaves(\overline{\sigma}^\sigma)} \epsilon_\sigma[\zeta_p = \zeta_{p'}].$$

The formula $\epsilon_\sigma \phi$ stands for ϕ if $\sigma \models \phi$ and $\neg\phi$ otherwise.

Informally, ϕ_σ is satisfied by every substitution σ' whose accessible parts have the same structure than in σ and such that σ' satisfies the same equalities, predicates and key types than σ.

The following proposition (whose proof can be found in Appendix B.1) says that ϕ_σ *represents* σ: if $\sigma' \models \phi$ then $\overline{\sigma'}$ extends of $\overline{\sigma}$.

Proposition 3. *Let σ and σ' be two finite substitutions such that $dom(\sigma) = dom(\sigma')$. If $\sigma' \models \phi_\sigma$ then there exists a injective substitution $\theta : Var_\sigma \to \mathcal{T}(\mathcal{F} \cup Var_{\sigma'})$ such that*

1. $\overline{\sigma'}^{\sigma'} = \overline{\sigma}^\sigma \theta$,
2. *that θ preserves the key types: if $\theta(X_t) = X_{t'}$ then $t = t^{-1}$ iff $t' = t'^{-1}$, if $\theta(X_t) = f(\ldots)$ then $t = t^{-1}$ iff $f \neq \mathsf{pub}$ and $f \neq \mathsf{prv}$.*

What do we need to obtain a unique formula ϕ_σ which satisfies $(*)$? Intuitively, when checking $\sigma' \models \phi_\sigma$, we test if σ' verifies all the positive properties of σ: same term structure, same equalities between subterms, but we don't express some negative properties. we also need to check that σ' satisfies the same predicates than σ.

Example 5. Assume there is no predicate V_i in our syntax and consider $\sigma_1 = [k/x_1, k'/x_2]$, $\sigma_2 = [k/x_1.\{k'\}_{k^n}]$ where k^n, $n \geq 2$ is defined by $k^2 =< k, k >$ and $k^{n+1} =< k, k^n >$. Then $\phi_{\sigma_1} = \neg[x_1 = x_2]$ and $\sigma_2 \models \phi_{\sigma_1}$ but clearly $\sigma_1 \not\sim \sigma_2$. To characterize every σ' such that $\sigma_1 \sim \sigma'$, we should express that the terms in x_1 and x_2 can not be analyzed with the knowledge of σ_1. For example, we should check that $[\mathsf{dec}_{x_1}(x_2) = \bot]$, $[\mathsf{dec}_{<x_1,x_1>}(x_2) = \bot]$, ...

This example illustrates that testing that a term cannot be analyzed, may require arbitrary large expressions. That's why we cannot use a unique formula, but we use instead a process R_σ:

Theorem 2. *Let σ be a finite substitution, there exist a process R^f_σ and a formula ϕ_σ such that for every substitution σ' such that $dom(\sigma') = dom(\sigma)$, $\sigma \sim \sigma'$ if and only if $\sigma' \models \phi_\sigma$ and $R^f_\sigma \sigma' \not\Downarrow f$.*

To prove Theorem 2, we introduce two new predicates $vis(\sigma, \zeta)$ and $enc(\sigma, \zeta)$ which do not belong to our syntax. The predicate $vis(\sigma, \zeta)$ holds if one can detect that the head symbol of $\widehat{\zeta\sigma}$ is a visible symbol. In the same way, the predicate $enc(\sigma, \zeta)$ holds if one can detect that $\widehat{\zeta\sigma}$ is an encrypted message by computing explicitly the inverse of the key.

$$[\![vis(\sigma,\zeta)]\!] = \begin{cases} tt \text{ if there exist } f \in \mathcal{VF}, \zeta_1, \ldots, \zeta_n \\ \quad \text{such that } \widehat{\zeta\sigma} = f(\widehat{\zeta_1\sigma}, \ldots, \widehat{\zeta_n\sigma}), \\ ff \text{ otherwise.} \end{cases}$$

$$[\![enc(\sigma,\zeta)]\!] = \begin{cases} tt \text{ if there exist } \zeta' \text{ such that } \widehat{\mathsf{dec}_{\zeta'}(\zeta)} \neq \bot, \\ ff \text{ otherwise.} \end{cases}$$

Example 6. Consider $\sigma = [k, \{n\}_k]$ and $\sigma' = [k, \{n\}_{k'}]$, then $[\![enc(\sigma, x_2)]\!] = tt$, $[\![enc(\sigma', x_2)]\!] = ff$ and $[\![enc([\{n\}_k/x_2], x_2)]\!] = ff$.

With these two predicates, we are able to define

$$\phi^{char}_\sigma \stackrel{\text{def}}{=} \phi_\sigma \wedge \bigwedge_{p \in leaves(\overline{\sigma}^\sigma)} \neg vis(\sigma, \zeta_p) \wedge \neg enc(\sigma, \zeta_p).$$

Using Proposition 3, ϕ^{char}_σ satisfies the required property $(*)$.

Proposition 4. *Let σ, σ' be two finite substitutions such that $dom(\sigma) = dom(\sigma')$. We have that $\sigma \sim \sigma'$ if and only if $\sigma' \models \phi^{char}_\sigma$ and for every $1 \leq i \leq n$, $A^i_\sigma = A^i_{\sigma'}$.*

Since these two predicates do not belong to our syntax, we construct instead a process $R_{\zeta,\sigma}$:

$$\begin{array}{rll} R_{\zeta,\sigma} \stackrel{\text{def}}{=} & (\nu a)(!\bar{a}(x_1) \mid \ldots \mid !\bar{a}(x_n) & \\ & \mid a(y_1).\ldots.a(y_m).!\bar{a}(f(y_1,\ldots,y_m)) & \forall f \in \mathcal{VF} \\ & \mid a(x).a(y).!\bar{a}(\mathsf{dec}_x(y)) & \\ & \mid a(x).!\bar{a}(g_i^{-1}(x)) & \forall g \in \mathcal{IF}, \forall i \leq arity(g) \\ & \mid a(y_1).\ldots.a(y_m).[f(y_1,\ldots,y_m) = \zeta]\bar{f}(\zeta) & \forall f \in \mathcal{VF} \\ & \mid a(x).a(y).[\mathsf{dec}_x(y) = \zeta]\bar{f}(\zeta)) & \end{array}$$

Then $R_{\zeta,\sigma}$ commits to f if and only if the predicate $vis(\zeta, \sigma)$ or $enc(\zeta, \sigma)$ holds. Intuitively, $R_{\zeta,\sigma}$ generates arbitrarily large expressions $\zeta_1, \ldots, \zeta_n$ and checks the equalities $\zeta = f(\zeta_1, \ldots, \zeta_n)$ and $\zeta = \mathsf{dec}_{\zeta_1}(\zeta_2)$.

It remains to build a process R^i_σ such that $R^i_\sigma\sigma'$ commits to f if and only if $A^i_\sigma \neq A^i_{\sigma'}$: using the rules of the tree automata recognizing A^i_σ, we construct a process $R^{i,1}_\sigma$ such that $R^{i,1}_\sigma$ commits to f if and only if there exists $\zeta \in A^i_\sigma$ such that $\zeta \notin A^i_{\sigma'}$. Conversely, using the rules of the tree automata recognizing $(A^i_\sigma)^c$, we construct a process $R^{i,2}_\sigma$ such that $R^{i,2}_\sigma$ commits to f if and only if there exists $\zeta \notin A^i_\sigma$ such that $\zeta \in A^i_{\sigma'}$. Then, R_σ defined in Theorem 2 is simply the composition of the $R_{\zeta|_p,\sigma}$ for p leaf of $\overline{\sigma}$ and the $R^{i,1}_\sigma | R^{i,2}_\sigma$ for every $1 \leq i \leq n$. A detailed proof can be found in Appendix B.1.

Remark: let us return to the syntax defined in [7]. In this case, since there is only one visible and non-invertible symbol (the encryption symbol) and there is no compound keys, the two predicates *vis* and *enc* can be encoded in the syntax itself; in this particular case:

$$enc(\sigma,\zeta) \equiv vis(\sigma,\zeta) \equiv \bigwedge_{p \in leaves(\overline{\sigma}^\sigma)} [\mathsf{dec}_{\zeta_p}(\zeta) = \bot]$$

4.2 Barbed Equivalence

We are now ready to prove the equivalence between barbed equivalence and bisimilarity. We first generalize the definition of barbed equivalence between processes to barbed equivalence between processes in an environment like Boreale *et al.* in [7].

Definition 7. *Let $\sigma_1 \sim \sigma_2$. We construct $leaves(\overline{\sigma_1}^{\sigma_1})$ (= $leaves(\overline{\sigma_2}^{\sigma_2})$) the set of paths leading to leaves in $\overline{\sigma_1}^{\sigma_1}$. A binary relation S of processes is a (σ_1,σ_2)-*barbed bisimulation *if whenever $P\mathcal{S}Q$ then:*
- *for each P', if $P \xrightarrow{\tau} P'$ then there is Q' such that $Q \Longrightarrow Q'$ and $P'\mathcal{S}Q'$,*
- *for each $p \in leaves(\overline{\sigma_1}^{\sigma_1})$, if $P \downarrow \sigma_1|_p$, then $Q \downarrow \sigma_2|_p$*

and symmetrically exchanging Q and P.

*Two processes P and Q are (σ_1,σ_2)-*barbed bisimilar*, written $(\sigma_1,\sigma_2) \vdash P \dot{\cong} Q$, if (P,Q) belongs to the largest (σ_1,σ_2)-barbed bisimulation.*

*Two processes P and Q are (σ_1,σ_2)-*barbed equivalent*, written $(\sigma_1,\sigma_2) \vdash P \cong Q$, if for all R with $fn(R) \subseteq dom(\sigma_1)$ we have that $(\sigma_1,\sigma_2) \vdash P|R\sigma_1 \dot{\cong} Q|R\sigma_2$.*

Barbs are now checked relatively to the names that are known to the environments. We still have that $P \cong Q$ holds iff $(\epsilon_V,\epsilon_V) \vdash P \cong Q$, for some V containing $fn(P,Q)$, where ϵ_V simply denotes the substitution $[v/v]_{v\in V}$.

Theorem 3 (soundness of weak bisimilarity). *Let P and Q be processes and σ_1 and σ_2 be equivalent substitutions. If $(\sigma_1,\sigma_2) \vdash P \approx Q$, then $(\sigma_1,\sigma_2) \vdash P \cong Q$.*

The up-to techniques presented in [7] still work, thus the proof of Theorem 3 can be applied to our extension exactly on the same way.

As in [7], for completeness, we restrict ourself to a class of processes that have an image-finiteness property. Formally, a process P is *structurally image-finite* if

for each visible trace s, the set of equivalence classes $\{P' \mid P \stackrel{s}{\Longrightarrow} P'\}/\equiv$ is finite. The relevancy of image-finiteness is illustrated in the next section in which we reduce a process to a process without replication.

Theorem 4 (completeness for barbed equivalence). *Let P and Q be structurally image-finite processes and σ_1 and σ_2 be equivalent, finite substitutions. If $(\sigma_1, \sigma_2) \vdash P \cong Q$, then $(\sigma_1, \sigma_2) \vdash P \approx Q$.*

The main ingredient is to replace the characteristic formula ϕ_σ used in the proof of [7] by our characteristic process (detailed proof in Appendix B.2).
The interest of this equivalence result between weak bisimilarity and barbed equivalence is that the quantification over all contexts is replaced by a single environment which can be better controled.

5 Applications

We first present some properties on weak bisimilarity which are useful when reasoning on cryptographic processes and then use them to sketch the proof of the secrecy of Needham-Schroeder-Lowe protocol.

Notation: $\tilde{k}$ denotes the sequence $(k_i)_{i\in J}$ for some J and $[\tilde{M}/\tilde{x}]$ denotes the substitution $[M_i/x_i]_{i\in J}$ provided $\tilde{M}$ and $\tilde{x}$ have the same index J.

Several useful properties of $\approx$ can be proved using up-to techniques. For instance, *Weakening* means that if P and Q are bisimilar in some contexts then they are also bisimilar in weaker contexts. *Restriction* means that if P and Q are bisimilar in some contexts then the restricted process $(\nu\tilde{k})P$ and $(\nu\tilde{k})Q$ are also bisimilar in the same contexts, for some free variables $\tilde{h}$ and $\tilde{k}$ that do not appear in the contexts.

Proposition 5.
- (Weakening) *Suppose that* $(\sigma_1[M_i/x_i]_{i\in J}, \sigma_2[N_i/x_i]_{i\in J}) \vdash P \approx Q$ *then* $(\sigma_1, \sigma_2) \vdash P \approx Q$.
- (Contraction) *Suppose that* $(\sigma_1, \sigma_2) \vdash P \approx Q$ *and consider any* ζ *such that* $n(\zeta) \subseteq dom(\sigma_1)$ *and* $\widehat{\zeta\sigma_1} \neq \bot$. *Then* $(\sigma_1[\widehat{\zeta\sigma_1}/x], \sigma_2[\widehat{\zeta\sigma_2}/x]) \vdash P \approx Q$
- (Parallel) *Suppose that* $fn(R) \subseteq dom(\sigma_1)$ *and* $(\sigma_1, \sigma_2) \vdash P \approx Q$. *Then* $(\sigma_1, \sigma_2) \vdash P|R\sigma_1 \approx Q|R\sigma_2$.
- (Restriction) *Suppose that* $(\sigma_1, \sigma_2) \vdash P \approx Q$, *that* $\tilde{k} \cap n(\sigma_1) = \emptyset$ *and that* $\tilde{h} \cap n(\sigma_2) = \emptyset$. *Then* $(\sigma_1, \sigma_2) \vdash (\nu\tilde{k})P \approx (\nu\tilde{h})Q$.

In order to prove the secrecy of a protocol P, we construct an environment σ such that the run of P cannot increase the knowledge of σ. For this, we need at least to prevent the environment to increase its knowledge by creating new nonces. Thus we define a new bisimilarity $\approx'$ between processes such that the environment is no longer allowed to create new names: $\approx'$ is defined exactly like $\approx$ except that actions $\sigma_1 \triangleright P \stackrel{\mu}{\underset{\delta}{\longmapsto}} \sigma_1' \triangleright P'$ are of the form $(\mathsf{E}-\mathsf{OUT})$, $(\mathsf{E}-\mathsf{TAU})$ and $(\mathsf{E}-\mathsf{INP}')$ where $(\mathsf{E}-\mathsf{INP}')$ is an action $(\mathsf{E}-\mathsf{INP})$ such that the environment do not create nonce. Then $\approx$ is equivalent to $\approx'$ provided arbitrary many fresh names can be added to the environments.

Definition 8. *Let $\approx'$ be the largest binary compatible relation $\mathcal{R}$ of configurations which verifies whenever $(\sigma_1, \sigma_2) \vdash P \mathcal{R} Q$ and $\sigma_1 \triangleright P \xmapsto[\delta]{\mu} \sigma_1' \triangleright P'$ then there are μ', σ_2' and Q' such that $\sigma_2 \triangleright Q \xmapsto[\delta]{\widehat{\mu'}} \sigma_2' \triangleright Q'$ and $(\sigma_1', \sigma_2') \vdash P' \mathcal{R} Q'$ and the converse on the transitions of Q and P, where the actions $\xmapsto[\delta]{\mu}$ are (E-OUT) or (E-TAU) (defined Figure 3) or (E-INP'):*

$$(E - INP') \frac{P \xrightarrow{aM} P' \quad \widehat{\eta\sigma} \quad M = \widehat{\zeta\sigma} \quad n(\zeta) \subset dom(\sigma)}{\sigma \triangleright P \xrightarrow[\overline{\eta}(\zeta)]{aM} \sigma \triangleright P'}$$

Proposition 6. *Let σ_1, σ_2 be two equivalent substitutions, let P, Q be two processes then $(\sigma_1, \sigma_2) \vdash P \approx Q \Leftrightarrow \forall \tilde{h}\, \tilde{k}$ such that $|\tilde{h}| = |\tilde{k}|$, $\tilde{h} \cap dom(\sigma_1) = \tilde{k} \cap dom(\sigma_2) = \emptyset$ then $(\sigma_1[\tilde{h}/\tilde{h}], \sigma_2[\tilde{k}/\tilde{k}]) \vdash P \approx' Q$.*

In addition, Weakening still hold for $\approx'$ (using again the same up-to techniques than in [7]).

Following [7], we define a process R to be σ-*safe* if for each s, if $\sigma \triangleright R \xmapsto[u]{s} \sigma' \triangleright R'$ and $R' \xrightarrow{(\nu \tilde{b})\overline{a}<M>}$, then $a, M \in kn(\sigma)$. We define also R to be σ-*safe2* if R is σ-safe except that actions s are of the form $(\mathsf{E-OUT})$ or $(\mathsf{E-TAU})$ or $(\mathsf{E-INP'})$. Intuitively, R is σ-safe if R cannot increase the knowledge of σ. σ-safety gives a sufficient condition to compose processes (Propositions 7 and 8). Both properties are proved by introducing a new relation between processes and showing that it is a weak bisimulation. Similar properties hold for $\approx'$ (proofs in Appendix C.1).

Proposition 7. *Suppose that $(\sigma_1, \sigma_2) \vdash Q_1 \approx Q_2$ (resp. $(\sigma_1, \sigma_2) \vdash Q_1 \approx' Q_2$) and that $(\sigma_1, \sigma_2) \vdash R_1 \approx R_2$ (resp. $(\sigma_1, \sigma_2) \vdash R_1 \approx R_2$). Suppose that, for $i = 1, 2$, Q_i and R_i are σ_i-safe. (resp. σ_i-safe2). Then $(\sigma_1, \sigma_2) \vdash Q_1 | R_1 \approx Q_2 | R_2$ (resp. $(\sigma_1, \sigma_2) \vdash Q_1 | R_1 \approx' Q_2 | R_2$).*

Proposition 8. *Let σ and σ' be two equivalent substitutions such that the process P is σ and σ'-safe (resp. σ and σ'-safe2). Then $(\sigma, \sigma') \vdash P \approx P$ (resp. $(\sigma, \sigma') \vdash P \approx' P$).*

Needham-Schroeder-Lowe Protocol

We consider a version of Needham-Schroeder-Lowe with two participants a and b and an intruder i. We think it is not an real restriction since, for state-transitions systems, it has been shown that secrecy with an arbitrary number of participants is equivalent to secrecy with one honest participant and one dishonest participant (see [10]). In addition, the proof of secrecy sketched below seems to be easily generalizable to an arbitrary number of participants. Let us

define

$$I_{A,B}(n) = \overline{p}(\{n, A\}_{\mathsf{pub}(B)}).p(x).\,\mathsf{let}\,(x_n, x_{n_2}, x_2) = \mathsf{dec}_{\mathsf{prv}(A)}(x)\,\mathsf{in}\\ [x_n = n][x_2 = B]\,\overline{p}(\{x_{n_2}\}_{\mathsf{pub}(B)}).\mathbf{0}$$

$$R_{B,A}(n) = p(x).\,\mathsf{let}\,(x_{n_1}, x_1) = \mathsf{dec}_{\mathsf{prv}(B)}(x)\,\mathsf{in}\,[x_1 = A]\,\overline{p}(\{x_{n_1}, n, B\}_{\mathsf{pub}(A)}).\mathbf{0}$$

$$P(n_b) = R_{b,a}(n_b)\,|!(\nu n_1^{ab})I_{a,b}(n_1^{ab})\,|!(\nu n_1^{ai})I_{a,i}(n_1^{ai})|!(\nu n_1^{ba})I_{b,a}(n_1^{ba})\,|!(\nu n_1^{bi})I_{b,i}(n_1^{bi})\\ |!(\nu n_2^{ab})R_{a,b}(n_2^{ab})\,|!(\nu n_2^{ai})R_{a,i}(n_2^{ai})|!(\nu n_2^{ba})R_{b,a}(n_2^{ba})\,|!(\nu n_2^{bi})R_{b,i}(n_2^{bi})\,|\overline{p}(\mathsf{prv}(i)).\mathbf{0}$$

$I_{a,i}(n_1^{ai})$ describes the initiator a talking to i: a sends a message $\{n_1^{ai}, a\}_{\mathsf{pub}(i)}$, waits for a message of the form $\{n_1^{ai}, m, i\}_{\mathsf{pub}(a)}$ and answers with $\{m\}_{\mathsf{pub}(i)}$ (symmetrically for $I_{a,b}(n_1^{ab})$, $I_{b,a}(n_1^{ba})$ and $I_{b,i}(n_1^{bi})$). $R_{b,a}(n_2^{ba})$ waits for a message of the form $\{m, a\}_{\mathsf{pub}(b)}$ and answers with the message $\{m, n_2^{ba}, b\}_{\mathsf{pub}(a)}$ (and symmetrically for the other processes R).

The process $P(n_b)$ models the NSL protocol with two participants a and b and an intruder i (the environment can know the private key of i) playing arbitrary many sessions concurrently or sequentially. We want to prove the secrecy of P, i.e., that $\forall M, M'$, $P(M) \cong P(M')$. We first get rid of replications: it is sufficient to prove (by a straightforward property) that for all q, $(\epsilon_V, \epsilon_V) \vdash P_q(M) \cong P_q(M')$ where P_q represents q replications of the NSL protocol. We add a subscript j for the data used in the j^{th} replication.

$$P_q(M) = R_{ba}(M)\,|\,\overline{p}(\mathsf{pub}(i)).\mathbf{0}\,|\,\overline{p}(\mathsf{prv}(i)).\mathbf{0}\\ |(\nu n_{1,1}^{ab})I_{a,b}(n_{1,1}^{ab})|\cdots|(\nu n_{1,q}^{ab})I_{a,b}(n_{1,q}^{ab})|(\nu n_{1,1}^{ai})I_{a,i}(n_{1,1}^{ai})|\cdots|(\nu n_{1,q}^{ai})I_{a,i}(n_{1,q}^{ai})\\ |(\nu n_{1,1}^{ba})I_{b,a}(n_{1,1}^{ba})|\cdots|(\nu n_{1,q}^{ba})I_{b,a}(n_{1,q}^{ba})|(\nu n_{1,1}^{bi})I_{b,i}(n_{1,1}^{bi})|\cdots|(\nu n_{1,q}^{bi})I_{b,i}(n_{1,q}^{bi})\\ |(\nu n_{2,1}^{ab})R_{a,b}(n_{2,1}^{ab})|\cdots|(\nu n_{2,q}^{ab})R_{a,b}(n_{2,q}^{ab})|(\nu n_{2,1}^{ai})R_{a,i}(n_{2,1}^{ai})|\cdots|(\nu n_{2,q}^{ai})R_{a,i}(n_{2,q}^{ai})\\ |(\nu n_{2,1}^{ba})R_{b,a}(n_{2,1}^{ba})|\cdots|(\nu n_{2,q}^{ba})R_{b,a}(n_{2,q}^{ba})|(\nu n_{2,1}^{bi})R_{b,i}(n_{2,1}^{bi})|\cdots|(\nu n_{2,q}^{bi})R_{b,i}(n_{2,q}^{bi})$$

Then, applying our fundamental Theorem 1, it is sufficient to prove that for all q, $(\epsilon_V, \epsilon_V) \vdash P_q(M) \approx P_q(M')$.

The main step of the proof is to find a substitution σ, such that every component of $P_q(M)$ is σ-safe so that we can apply Proposition 7. Let $\sigma_{M,\tilde{h}} = \epsilon_{S_{M,\tilde{h}}}$ where

$$S_{M,\tilde{h}} = V \cup \cup\,\tilde{h} \quad \bigcup_{\substack{1 \le m \le q,\\ (A,B) \in \{(a,b),(b,a)\}}} \left\{n_{1,m}^{Ai}, n_{2,m}^{Ai}, \{n_{1,m}^{AB}, A\}_{\mathsf{pub}(B)}, \{n_{2,m}^{BA}\}_{\mathsf{pub}(B)}, \{M\}_{\mathsf{pub}(B)}\right\}\\ \cup \quad \bigcup_{\substack{1 \le m, m' \le q,\\ (A,B) \in \{(a,b),(b,a)\}}} \left\{n_{1,m}^{AB}, n_{2,m'}^{BA}, B\}_{\mathsf{pub}(A)}, \{n_{1,m}^{Ai}, n_{2,m'}^{BA}, B\}_{\mathsf{pub}(A)}\right\}\\ \cup \quad \bigcup_{\substack{1 \le m \le q,\\ (A,B) \in \{(a,b),(b,a)\}}} \left\{\{n_{1,m}^{AB}, M, B\}_{\mathsf{pub}(A)}, \{n_{1,m}^{Ai}, M, B\}_{\mathsf{pub}(A)}\right\}$$

Such a substitution is not big enough. Indeed, whatever the message t is, if the environment sends the message $\{t,a\}_{\mathsf{pub}(b)}$, then b will blindly respond with $\{t,n^{ab}_m,b\}_{\mathsf{pub}(a)}$ for some $m \leq q$, which increases intruder's knowledge. Thus, we define F_M to be the function on sets of messages:

$$F_M(S) = \bigcup_{1\leq m\leq q} \Big\{\big\{\{t,n^{ba}_{2,m},b\}_{\mathsf{pub}(a)},\{t,M,b\}_{\mathsf{pub}(a)} | t\in kn(S) \text{ or } \{t,a\}_{\mathsf{pub}(b)}\in kn(S)\big\} \\ \cup \big\{\{t,n^{ab}_{2,m},a\}_{\mathsf{pub}(b)},\{t,M,a\}_{\mathsf{pub}(b)} | t\in kn(S) \text{ or } \{t,b\}_{\mathsf{pub}(a)}\in kn(S)\big\}\Big\}$$

and $\overline{F_M}(S) = \cup_{i\geq 0} F^i_m(S)$. Let $\overline{F_M}(\sigma)$ be an infinite substitution, defined by $\overline{F_M}(\sigma) = \sigma[M_i/x_i]_{i\in J}$ where the $\tilde{x_i}$ are new variables and $range(\sigma') = \overline{F_M}(range(S))$. Note that every M_i is of the form $M_i = F_M(t_i)$ for some t_i.
Let $\sigma'_{M,\tilde{h}} = \overline{F_M}(\sigma_{M,\tilde{h}})$ and $\sigma'_{M',\tilde{k}} = \overline{F_{M'}}(\sigma_{M',\tilde{k}})$. The following lemmas are proved by a precise analysis of the atomic processes of $P_q(M)$ (detailed proofs in C.2).

Lemma 1. *$\forall 1\leq m\leq q, \forall A,B\in\{a,b\}, A\neq B$,*
$I_{A,B}(n^{AB}_{1,m})$,$I_{A,i}(n^{Ai}_{1,m})$, $R_{A,B}(n^{AB}_{2,m})$,$R_{A,i}(n^{Ai}_{2,m})$ are $\sigma'_{M,\tilde{h}}$ and $\sigma'_{M',\tilde{k}}$-safe.
Moreover, $R_{b,a}(M)$ is $\sigma'_{M,\tilde{h}}$-safe and $R_{b,a}(M')$ is $\sigma'_{M',\tilde{k}}$-safe.

Lemma 2. $\sigma'_{M,\tilde{h}} \sim \sigma'_{M',\tilde{k}}$.

Lemma 3. $(\sigma'_{M,\tilde{h}},\sigma'_{M',\tilde{k}}) \vdash R_{b,a}(M) \approx' R_{b,a}(M')$.

Using σ-safety and the above lemmas, we can re-compose our process $P_q(M)$:

$$(\sigma'_{M,\tilde{h}},\sigma'_{M',\tilde{k}}) \vdash R_{b,a}(M)|_{\substack{1\leq m\leq q\\(A,B)=(a,b)\\(A,B)=(b,a)}} I_{A,B}(n^{AB}_{1,m})|I_{A,i}(n^{Ai}_{1,m})|R_{A,B}(n^{AB}_{2,m})|R_{A,i}(n^{Ai}_{2,m}) \\ \approx' R_{b,a}(M')|_{\substack{1\leq m\leq q\\(A,B)=(a,b)\\(A,B)=(b,a)}} I_{A,B}(n^{AB}_{1,m})|I_{A,i}(n^{Ai}_{1,m})|R_{A,B}(n^{AB}_{2,m})|R_{A,i}(n^{Ai}_{2,m}).$$

By weakening

$$(\epsilon_V[\tilde{h}/\tilde{h}],\epsilon_V[\tilde{k}/\tilde{k}]) \vdash R_{b,a}(M)|_{\substack{1\leq m\leq q\\(A,B)=(a,b)\\(A,B)=(b,a)}} I_{A,B}(n^{AB}_{1,m})|I_{A,i}(n^{Ai}_{1,m})|R_{A,B}(n^{AB}_{2,m})|R_{A,i}(n^{Ai}_{2,m}) \\ \approx' R_{b,a}(M')|_{\substack{1\leq m\leq q\\(A,B)=(a,b)\\(A,B)=(b,a)}} I_{A,B}(n^{AB}_{1,m})|I_{A,i}(n^{Ai}_{1,m})|R_{A,B}(n^{AB}_{2,m})|R_{A,i}(n^{Ai}_{2,m}).$$

Applying Proposition 6

$$(\epsilon_V,\epsilon_V) \vdash R_{b,a}(M)|_{\substack{1\leq m\leq q\\(A,B)=(a,b)\\(A,B)=(b,a)}} I_{A,B}(n^{AB}_{1,m})|I_{A,i}(n^{Ai}_{1,m})|R_{A,B}(n^{AB}_{2,m})|R_{A,i}(n^{Ai}_{2,m}) \\ \approx R_{b,a}(M')|_{\substack{1\leq m\leq q\\(A,B)=(a,b)\\(A,B)=(b,a)}} I_{A,B}(n^{AB}_{1,m})|I_{A,i}(n^{Ai}_{1,m})|R_{A,B}(n^{AB}_{2,m})|R_{A,i}(n^{Ai}_{2,m}).$$

and the restriction property, we get $(\epsilon_V, \epsilon_V) \vdash P_q(M) \approx P_q(M')$. QED

We emphasize that the main step of this proof generating the "good" invariant σ such that each component of the protocol is σ-safe. This is a problem on traces, which is simpler than the original process equivalence question.

6 Conclusion

We have presented a relationship between spi-calculus and trace semantics in a general framework, which is adequate for the security verification of cryptographic protocols: it allows to express most of the protocols described in [9].

As illustrated in Section 5, a proof of secrecy of the form $P(M) \cong P(M')$ can be reduced to the generation of an invariant like our $\bar{F}(\sigma_{M,\bar{h}})$. This is more amenable to automation.

Our extended formalism gives the ability for an intruder to distinguish by comparison the message $\{n\}_{\mathsf{pub}(a)}$ from $\{n'\}_{\mathsf{pub}(a)}$ knowing n, n' and $\mathsf{pub}(a)$, even without knowing the private key $\mathsf{prv}(a)$ that "opens" these messages. This is an important feature if we want to express properties like anonymity. Consider, for instance, the very simple protocol where a sends to b the message $\{a\}_{\mathsf{pub}(b)}$ to notify to b her intention to start a communication. Then, even if the intruder is not able to decrypt the message $\{a\}_{\mathsf{pub}(b)}$, he can tell that a is starting a communication with b. Thus, this naive protocol is not anonymous.

Boreale *et al* proved soundness and completeness of trace and weak bisimulation equivalence with respect to may-testing and barbed equivalence, respectively. Due to lack of space and time, we have only considered the extension of the equivalence between weak bisimilarity and barbed equivalence but we are quite confident that the same work could be done for trace bisimulation and may-testing.

Acknowledgements: I am particularly grateful to Hubert Comon, Martin Abadi, Cédric Fournet and Vitaly Shmatikov for their enriching discussions that contributed to this work.

References

1. M. Abadi and Cédric Fournet. Mobile values, new names, and secure communication. In *Proceedings of the 28th ACM Symposium on Principles of Programming Languages (POPL'01)*, pages 104–115, January 2001.
2. M. Abadi and A. D. Gordon. A calculus for cryptographic protocols: The spi calculus. In *Fourth ACM Conference on Computer and Communications Security*, pages 36–47. ACM Press, 1997.
3. M. Abadi and P. Rogaway. Reconciling two views of cryptography. In *IFIP International Conference on Theoretical Science (IFIP TCS2000), Sendai, Japan*, August 2000.
4. R. Amadio and D. Lugiez. On the reachability problem in cryptographic protocols. In *Proc. CONCUR, vol. 1877 of Lecture Notes in Computer Science, pages 380-394*, 2000.
5. R. Amadio, D. Lugiez, and V. Vanackère. On the symbolic reduction of processes with cryptographic functions. In *Technical Report 4147, INRIA*, March 2001.

6. B. Blanchet. An efficient cryptographic protocol verifier based on prolog rules. In *CSFW: Proceedings of The 14th Computer Security Foundations Workshop*. IEEE Computer Society Press, 2001.
7. Michele Boreale, Rocco De Nicola, and Rosario Pugliese. Proof techniques for cryptographic processes. In *Logic in Computer Science*, pages 157–166, 1999. Extended version to appear in SIAM Journal of Computing, available at `http://citeseer.nj.nec.com/boreale99proof.html`.
8. Witold Charatonik and Andreas Podelski. Set constraints with intersection. In *Logic in Computer Science*, pages 362–372, 1997.
9. J. Clark and J. Jacob. A survey of authentication protocol literature. `http://www.cs.york.ac.uk/~jac/ papers/drareviewps.ps`, 1997.
10. H. Comon and V. Cortier. Tree automata with one memory, set constraints and cryptographic protocols. In *Research Report LSV-01-13*, december 2001.
11. H. Comon, V. Cortier, and J. Mitchell. Tree automata with one memory, set constraints and ping-pong protocols. In *Proc. 28th Int. Coll. Automata, Languages, and Programming (ICALP'01), Crete, Greece, July 2001*, volume 2076, pages 682–693. Springer, 2001.
12. H. Comon and V. Shmatikov. Is it possible to decide wheter a cryptographic protocol is secure or not? In *Journal of Telecommunications and Information Technology, to appear*, 2001.
13. V. Cortier, J. Millen, and Harald Rueß. Proving secrecy is easy enough. In *14th IEEE Computer Security Foundations Workshop*, pages 97–108. IEEE Computer Society, 2001.
14. J. Goubault-Larrecq. A method for automatic cryptographic protocol verification. In *Proc. 15 IPDPS 2000 Workshops, Cancun, Mexico, May 2000, volume 1800 of Lecture Notes in Computer Science, pages 977-984. Springer.*
15. J. Millen and H. Rueß. Protocol-independent secrecy. In *RSP: 21th IEEE Computer Society Symposium on Research in Security and Privacy*, 2000.
16. John C. Mitchell, V. Shmatikov, and U. Ster. Finite-state analysis of ssl 3.0. In *Computer Aided Verification*, pages 71–76, 1998.
17. D. Monniaux. Abstracting cryptographic protocols with tree automata. In *Proc. 6th International Static Analysis Symposium (SAS'99), vol. 1694 of Lecture Notes in Computer Science*. Springer-Verlag, 1999.
18. L. Paulson. The inductive approach to verifying cryptographic protocols. *Journal of Computer Security*, 6(1):85–128, 1998.
19. M. Rusinowitch and M. Turuani. Protocol insecurity with finite number of sessions is np-complete. In *14th IEEE Computer Security Foundations Workshop*, pages 174–190. IEEE Computer Society, 2001.
20. Schneider. Verifying authentication protocols with CSP. In *PCSFW: Proceedings of The 10th Computer Security Foundations Workshop*. IEEE Computer Society Press, 1997.
21. Schneider and Sidiropoulos. CSP and anonymity. In *ESORICS: European Symposium on Research in Computer Security*. LNCS, Springer-Verlag, 1996.
22. J. Thayer, J. Herzog, and J. Guttman. Strand spaces: proving security protocols correct. In *Journal of Computer Security, Vol. 7*, pages 191–230. IEEE Computer Society, 1999.

A Environment Equivalence

To prove Theorem 1, we first prove some useful lemmas.

Proposition 9. *Let σ be a substitution. $t \in kn(\sigma)$ if and only if there exists an expression ζ such that $n(\zeta) \subset dom(\sigma)$ and $\widehat{\zeta\sigma} = t$.*

$\boxed{\Rightarrow}$ Let $P(t)$ a predicate defined to hold if there exists an expression ζ such that $n(\zeta) \subset dom(\sigma)$ and $\widehat{\zeta\sigma} = t$. Clearly, for all $t_i/x_i \in \sigma$, $P(t_i)$ holds (take $\zeta = x_i$). We can easily verify that P verifies the induction rules of the definition of $kn(W)$ (Definition 3). Thus, $\forall t \in kn(\sigma)$, $P(t)$ holds.
$\boxed{\Leftarrow}$ Conversely, we prove by induction on the size of ζ that if there exists an expression ζ such that $n(\zeta) \subset dom(\sigma)$ and $\widehat{\zeta\sigma} = t$ then $t \in kn(\sigma)$.

Proposition 10 (Proposition 2). *Let $\sigma = [t_1/x_1, \ldots, t_n/x_n]$ be a substitution, $[t/x] \in \sigma$. If p is a path in $\overline{\sigma}^{\sigma}$ then there exist ϕ_p and ζ_p such that $dom(\phi_p), dom(\zeta_p) \subset dom(\sigma)$, $\widehat{\zeta_p\sigma} = \sigma|_p$ (thus $\sigma|_p \in kn(\sigma)$) and $\sigma \models \phi_p$. In addition, ϕ_p and ζ_p* characterize *p in the sense that for all substitution σ', if $dom(\sigma) = dom(\sigma')$ then*

$$\sigma' \models \phi_p \text{ and } \widehat{\zeta_p\sigma'} \neq \bot \Rightarrow \widehat{\zeta_p\sigma'} = \sigma'|_p,\ s_\sigma(p) = s_{\sigma'}(p) \text{ and } p \text{ is a path of } \overline{\sigma'}^{\sigma'}.$$

We prove it by induction on the length of p.
If $p = i \in \mathbb{N}$, $\sigma|_i = t_i$, let $\zeta = x_i$, $\phi = \#$, then ζ and ϕ characterize i.
If the property is true for $|p| = n$, let $p' = p.i$ such that $|p| = n$ and p' is a path in $\overline{\sigma}^{\sigma}$. By induction, there exist ϕ, ζ such that $\sigma \models \phi$, $\widehat{\zeta\sigma} = \sigma|_p$ and ϕ and ζ characterize p.

1. Either $\sigma|_p = f(t_1, \ldots, t_n)$, $f \in \mathcal{IF}$ and $t|_{p'} = t_i$. Let $\zeta' = f_i^{-1}(\zeta)$ and $\phi' = \phi$. Let σ' be a substitution such that $\phi' \models \sigma'$ and $\widehat{\zeta'\sigma'} = t \neq \bot$. Thus $\widehat{\zeta\sigma'} = f(t_1, \ldots, t, \ldots, t_n) = \sigma'|_p$ by hypothesis. Thus $t = \sigma'|_{p.i}$, i.e $\widehat{\zeta'\sigma'} = \sigma'|_{p'}$, $s_\sigma(p') = s_{\sigma'}(p')$ and p' is path in $\overline{\sigma'}$.
2. Or $\sigma|_p = g(t_1, \ldots, t_n)$, $g \in \mathcal{VF}$, $t|_{p'} = t_i$. Since p' is a path in $\overline{\sigma}$, we have $\overline{\sigma}|_p = g(\overline{t_1}, \ldots, \overline{t_n})$, thus and $t_1, \ldots, t_n \in kn(\sigma)$ (by definition of $\overline{\cdot}$). Applying Proposition 9, for all $1 \leq j \leq n$, there exists ζ_j such that $\widehat{\zeta_j\sigma} = t_j$. Let $\zeta' = \zeta_i$ and $\phi' = \phi \wedge [\zeta = g(\zeta_1, \ldots, \zeta_n)]$. Clearly, $\sigma \models \phi'$ and $\widehat{\zeta'\sigma} = \sigma|_{p'}$.
   Let σ' be a substitution such that $\sigma' \models \phi'$. In particular, $\widehat{\zeta\sigma'} = t \neq \bot$. Since $\sigma \models \phi'$ we have $\widehat{\zeta\sigma'} = \sigma'|_p$ by the induction hypothesis. Since $\sigma' \models [\zeta = g(\zeta_1, \ldots, \zeta_n)]$, we have $\sigma'|_p = g(t_1, \ldots, \widehat{\zeta'\sigma'}, \ldots, t_n)$ thus $\widehat{\zeta'\sigma'} = \sigma'|_{p'}$, $s_\sigma(p') = s_{\sigma'}(p')$ and p' is path in $\overline{\sigma'}$.
3. Or $t|_p = \{t_1\}_{t_2}$ and $t_2 = t^{-1}$ such that $t \in kn(\sigma)$.
   Applying Proposition 9, there exists ζ_1 such that $\widehat{\zeta_1\sigma} = t$.
   If $t|_{p'} = t_1$, let $\zeta' = \mathsf{dec}_{\zeta_1}(\zeta)$, $\phi' = \phi$. They satisfy the required conditions.
   If $t|_{p'} = t$, let $\zeta = \zeta_1$ and $\phi' = \phi \wedge [\mathsf{dec}_{\zeta_1}(\zeta) \neq \bot]$, they satisfy the required conditions.
4. Or (last case) $t|_p = \{t_1\}_{t_2}$, $t|_{p'} = t_1$ or t_2 and $t_1, t_2 \in kn(\sigma)$. This case is solved like Case 2.

Lemma 4. *Let σ be a substitution. We extend the evaluation function of expressions to expressions with variables of Var_σ: for every $X_t \in Var_\sigma$, let $(X_t)^{-1} =$*

$X_{(t)^{-1}}$. *Let* ζ *an expression such that* $dom(\zeta) \subset dom(\sigma)$, *then the following equality holds:*

$$\overline{\widehat{\zeta\sigma}}^{\sigma} = \widehat{\zeta\overline{\sigma}^{\sigma}}$$

Corollary 1. *Let* σ *and* σ' *be two equivalent substitutions, let* θ *a renaming function such that* $\overline{\sigma}^{\sigma}\theta = \overline{\sigma'}^{\sigma'}$. *Then* $\overline{\widehat{\zeta\sigma}}^{\sigma}\theta = \overline{\widehat{\zeta\sigma'}}^{\sigma'}$.

We prove Lemma 4 by induction on ζ.

$\zeta = x$ then $\widehat{\zeta\sigma} = x\sigma$, thus $\overline{\widehat{\zeta\sigma}} = \overline{x\sigma} = x\overline{\sigma} = \widehat{\zeta\overline{\sigma}}$.

$\zeta = f(\zeta_1, \ldots, \zeta_n)$ for some f in $\mathcal{VF}$. If for some i, $\widehat{\zeta_i\sigma} = \bot$, then $\overline{\widehat{\zeta_i\sigma}} = \bot = \widehat{\zeta_i\overline{\sigma}}$ by induction. Thus the equality $\overline{\widehat{\zeta\sigma}}^{\sigma} = \widehat{\zeta\overline{\sigma}}$ holds. If for every i, $\widehat{\zeta_i\sigma} \neq \bot$, then $\widehat{\zeta\sigma} = f(\widehat{\zeta_1\sigma}, \ldots, \widehat{\zeta_n\sigma})$ and $\widehat{\zeta_i\sigma} \in kn(\sigma)$ by Proposition 9. Thus $\overline{\widehat{\zeta\sigma}} = f(\overline{\widehat{\zeta_1\sigma}}, \ldots, \overline{\widehat{\zeta_n\sigma}}) = f(\widehat{\zeta_1\overline{\sigma}}, \ldots, \widehat{\zeta_n\overline{\sigma}}) = \widehat{\zeta\overline{\sigma}}$.

$\zeta = \mathsf{dec}_{\eta}(\zeta')$. Either $\widehat{\eta\sigma} = \bot$ or $\widehat{\zeta'\sigma} = \bot$, then by induction $\widehat{\eta\overline{\sigma}} = \bot$ or $\widehat{\zeta'\overline{\sigma}} = \bot$ thus the equality $\overline{\widehat{\zeta\sigma}}^{\sigma} = \widehat{\zeta\overline{\sigma}}$ holds. Or $\widehat{\eta\sigma} \neq \bot$ and $\widehat{\zeta'\sigma} \neq \bot$. Assume $\widehat{\zeta\sigma} = \bot$. Either $\widehat{\zeta'\sigma}$ is not an encrypted text, then $\widehat{\zeta'\overline{\sigma}} = \overline{\widehat{\zeta'\sigma}}$ is also not an encrypted text; or $\widehat{\zeta'\sigma} = \{M\}_k$ and $\widehat{\eta\sigma} = k'$ where $k' \neq k^{-1}$. In this case $\widehat{\zeta'\overline{\sigma}} = \overline{\widehat{\zeta'\sigma}} = X_{\{M\}_k}$ or $\{\overline{M}\}_{\overline{k}}$. By induction $\widehat{\eta\overline{\sigma}} = \overline{\widehat{\eta\sigma}} = \overline{k'} \neq (\overline{k})^{-1}$, thus $\widehat{\zeta\overline{\sigma}} = \widehat{\mathsf{dec}_{\eta}(\zeta')\overline{\sigma}} = \bot$ and the equality $\overline{\widehat{\zeta\sigma}}^{\sigma} = \widehat{\zeta\overline{\sigma}}$ holds. Assume $\widehat{\zeta\sigma} \neq \bot$. Then $\widehat{\zeta'\sigma} = \{M\}_k$ and $\widehat{\eta\sigma} = k'$ where $k' = k^{-1}$. Applying Proposition 9, $k' \in \sigma$, $\overline{\{M\}_k} = \{\overline{M}\}_{(\overline{k'})^{-1}}$. Applying the induction hypothesis, $\widehat{\zeta\overline{\sigma}} = \overline{M} = \overline{\widehat{\zeta\sigma}}$.

$\zeta = g_i^{-1}(\zeta')$ for some $g \in \mathcal{IF}$. Same case study as above leads to the equality $\overline{\widehat{\zeta\sigma}}^{\sigma} = \widehat{\zeta\overline{\sigma}}$.

The proof of the Corollary uses that $(\theta(X_t))^{-1} = \theta(X_{t^{-1}})$, which is ensured by the fact that θ respects the type (symmetric or not) of the keys.

We can now prove Theorem 1:

Theorem 5. *(Theorem 1) For any two substitutions* σ *and* σ', *it holds that* $\sigma \sim \sigma'$ *if and only if* $\sigma \sim' \sigma'$.

Let $\sigma = [M_i/x_i]_{i\in J}$ and $\sigma' = [M'_i/x_i]_{i\in J}$ be two substitutions.

$\boxed{\Rightarrow}$ Assume $\sigma \not\sim' \sigma'$. Let us show that there exists a formula ϕ such that $\sigma \models \phi$ and $\sigma' \not\models \phi$ (or $\sigma' \models \phi$ and $\sigma \not\models \phi$).

Assume first that there exists i such that $A^i_{\sigma} \neq A^i_{\sigma'}$, i.e., there exists $\zeta \in A^i_{\sigma}$ such that $\zeta \notin A^i_{\sigma'}$ (or conversely), i.e. $[\![P_i(\widehat{\zeta[\sigma_p/x_p]})]\!] = tt$. Using Proposition 2, we construct for every path $p \in \mathcal{P}$, ϕ_p and ζ_p which characterize p. Let $\zeta' = \zeta[\zeta_p/x_p]$ and $\phi = \bigwedge_{p\in\mathcal{P}} \phi_p \wedge P_i(\zeta)$ Then $\sigma \models \phi$ and $\sigma' \not\models \phi$ since $\zeta \notin A^i_{\sigma'}$.

We define $I_v = \{p \mid \overline{\sigma}|_p \in Var_{\sigma} \text{ and } \overline{\sigma'}|_p \in Var_{\sigma'}\}$, the set of paths p such that both $\sigma|_p$ and $\sigma'|_p$ are variables. For all $X \in Var_{\sigma}$, let $P_X = \{p \in I_v \mid \overline{\sigma}|_p = X\}$

and for all $X \in Var_{\sigma'}$, let $P'_X = \{p \in I_v \mid \overline{\sigma'}|_p = X\}$. Clearly,

$$I_v = \biguplus_{X \in Var_\sigma} P_X = \biguplus_{X \in Var_{\sigma'}} P'_X.$$

Assume $\biguplus_{X \in Var_\sigma} P_X$ and $\biguplus_{X \in Var_{\sigma'}} P'_X$ do not define the same partition of I_v, i.e. there exists $X \in Var_\sigma$ such that, for all $X' \in Var_{\sigma'}$, $P_X \neq P_{X'}$. Since σ and σ' play symmetric roles, we can assume that for all $X' \in Var_{\sigma'}$, $P_X \not\subseteq P_{X'}$.

Let $\{p_1, \ldots, p_n\} = P_X$. Since p_i is a path in $\overline{\sigma}$, let ϕ_i and ζ_i defined according to Proposition 2: $\sigma \models \phi_i$ and $\widehat{\zeta_i \sigma} = \sigma|_{p_i}$. Let $\phi \stackrel{\text{def}}{=} \bigwedge_{1 \leq i \leq n} \phi_i \wedge [\zeta_1 = \zeta_2][\zeta_2 = \zeta_3] \ldots [\zeta_{n-1} = \zeta_n]$. X is of the form X_t and for all $1 \leq i \leq n$, $\sigma|_{p_i} = t$, thus $\sigma \models \phi$. Conversely, assume $\sigma' \models \phi$. There exists t' such that $\forall 1 \leq i \leq n$, $\widehat{\zeta_i \sigma'} = t'$. By definition of ϕ_i and ζ_i, since $\phi_i \not\models \sigma'$ and $\widehat{\zeta_i \sigma'} \neq \bot$, we have $\widehat{\zeta_i \sigma'} = \sigma'|_{p_i} = t'$ By definition of P_X, $p_i \in P_X \Rightarrow \overline{\sigma'}|_{p_i} \in Var_{\sigma'}$, thus $\overline{\sigma'}|_{p_i} = X_{t'}$, i.e. $P_X \subset P'_{X_{t'}}$, contradiction.

Assume $\biguplus_{X \in Var_\sigma} P_X$ and $\biguplus_{X \in Var_{\sigma'}} P'_X$ define the same partition but do not respect the keys, i.e., there exist $X_t, X_{t'}$ such that $P_{X_t} = P_{X_{t'}}$ but t is symmetric while t' is not (or conversely). Then. let $\{p_1, \ldots, p_n\} = P_{X_t}$ and let ϕ_1 and ζ_1 defined according to Proposition 2: $\sigma \models \phi_1$ and $\widehat{\zeta_1 \sigma} = \sigma|_{p_1}$. Let $\phi = \phi_1 \wedge [\mathsf{dec}_{\zeta_1}(\{\zeta_1\}_{\zeta_1}) = \zeta_1]$ then $\sigma \models \phi$ and $\sigma' \not\models \phi$.

Thus, we can assume that $\biguplus_{X \in Var_\sigma} P_X$ and $\biguplus_{X \in Var_{\sigma'}} P'_X$ define the same partition of I_v and respect the predicates and the keys. Let θ be a substitution such that $\forall X \in Var_\sigma, \forall X' \in Var_{\sigma'}$, $P_X = P'_{X'} \Rightarrow X\theta = X'$ and if $\theta(X_t) = X_{t'}$ then t is a constant iff t' is a constant. Thus if $\overline{\sigma}|_p$ and $\overline{\sigma'}|_p$ are both variable than $\overline{\sigma}|_p \theta = \overline{\sigma'}|_p$. Since $\sigma \not\sim' \sigma'$, $\overline{\sigma}^\sigma \theta \neq \overline{\sigma'}^{\sigma'}$. Let i and p such that $\overline{M_i|_p}^\sigma \theta \neq \overline{M'_i|_p}^{\sigma'}$ and the head symbols of $\overline{M_i|_p}^\sigma \theta$ and of $\overline{M'_i|_p}^{\sigma'}$ are distinct. We have just seen that $\overline{M_i|_p}^\sigma \theta$ and $\overline{M'_i|_p}^{\sigma'}$ can not be both variable. Assume (for example) that $\overline{M_i|_p}^\sigma \theta$ is not a variable. We have to consider three cases.

Either $\overline{M_i|_p} = f(\overline{t_1}, \ldots, \overline{t_n})$, where $f \in \mathcal{VF}$. Since $p_j = i.p.j$ $(1 \leq j \leq n)$ is also a path in $\overline{\sigma}$, we can define ϕ_j and ζ_j according to Lemma 2: $t_j = \widehat{\zeta_j \sigma}$, $\sigma \models \phi_j$. Let ϕ, ζ which characterize $p' = i.p$ according to Lemma 2 and let

$$\phi' = \phi \wedge \bigwedge_{1 \leq j \leq n} \phi_i \wedge [\zeta = f(\zeta_1, \ldots, \zeta_n)].$$

Clearly $\sigma \models \phi$. Let us show that $\sigma' \not\models \phi$.
Assume $\sigma' \models \phi$, than by construction, $\sigma'|_{p'} = M'_i|_p = \widehat{\zeta \sigma'} = f(\widehat{\zeta_1 \sigma'}, \ldots, \widehat{\zeta_n \sigma'})$, thus $\widehat{\zeta_1 \sigma'}, \ldots, \widehat{\zeta_n \sigma'} \in kn(\sigma')$, thus in any case, $\overline{\sigma'|_{p'}}^{\sigma'} = \overline{M'_i|_p}^{\sigma'} = f(\overline{\cdot}, \ldots, \overline{\cdot})$, i.e., $\overline{M_i|_p \theta}^\sigma \theta$ and of $\overline{M'_i|_p}^{\sigma'}$ have the same head symbols, contradiction.

Or $\overline{M_i|_p} = \{\overline{t_1}\}_{(\overline{t})^{-1}}$ and $M_i|_p = \{t_1\}_{t_2}$ such that $t_2 = t^{-1}$. Applying Lemma 2, we define $\zeta, \phi, \zeta_1, \phi_1, \zeta_2, \phi_2$ such that $\widehat{\zeta\sigma} = M_i|_p$, $\widehat{\zeta_1\sigma} = t_1$, $\widehat{\zeta_2\sigma} = t$ and $\sigma \models \phi_1, \phi_2$. Let

$$\phi = \phi_1 \wedge \phi_2 \wedge [\zeta_1 = \mathsf{dec}_{\zeta_2}(\zeta) = \zeta_1] \wedge \neg[\zeta = \{\zeta_1\}_{\zeta_2}]$$

ϕ satisfies $\sigma \models \phi$ and $\sigma' \not\models \phi$.

Or (last case) $\overline{M_i|_p} = f(\overline{t_1}, \ldots, \overline{t_n})$, where $f \in \mathcal{IF}$. Since $p_j = i.p.j$ $(1 \leq j \leq n)$ is also a path in $\overline{\sigma}$, we can define ϕ_j and ζ_j according to Proposition 2: $t_j = \widehat{\zeta_j\sigma}$, $\sigma \models \phi_j$. Let ϕ, ζ which characterize $p' = i.p$ according to Proposition 2 and let

$$\phi' = \phi \wedge \bigwedge_{1 \leq j \leq n} \phi_i \wedge [f_1^{-1}(\zeta) = \zeta_1] \wedge \ldots \wedge [f_n^{-1}(\zeta) = \zeta_n].$$

ϕ satisfies $\sigma \models \phi$ and $\sigma' \not\models \phi$. QED

$\boxed{\Leftarrow}$ We prove by induction on the length of ϕ that given two substitutions σ and σ', $\sigma \sim' \sigma \Rightarrow [\sigma \models \phi \Leftrightarrow \sigma' \models \phi]$. Let θ be a renaming function such that θ respects the key types and $\overline{\sigma}^{\sigma}\theta = \overline{\sigma'}^{\sigma'}$.

$|\phi| = 0$ If $\phi = tt$, OK.

If $\phi = P_i(\zeta)$ and $\sigma \models \phi$, then $\widehat{\zeta\sigma} = t$ and $P_i(t)$ holds. Let us show by induction on ζ that

If $\widehat{\zeta\sigma} \neq \bot$ *then* $\exists \zeta'$ *s.t.* $\widehat{\zeta\sigma} = \zeta'[\widehat{\sigma_p/x_p}]_{p\in\mathcal{P}}$ *where* $\zeta' \in \mathcal{T}(\mathcal{VF} \cup \{x_p | p \in \mathcal{P}\})$.

If $\zeta = \mathsf{dec}_\eta(\zeta'')$, then either $\zeta'' = \{\eta_1\}_{\eta_2}$ and $\widehat{\zeta\sigma} = \widehat{\eta_1\sigma}$, thus by induction hypothesis, there exists ζ' such that $\widehat{\eta_1\sigma} = \zeta'[\widehat{\sigma_p/x_p}]_{p\in\mathcal{P}}$. Or $\zeta'' = x_i \in dom(\sigma)$ and $\widehat{\zeta\sigma} = \sigma|_{i\cdot 1}$. Thus $i \cdot 1$ is a path of $\overline{\sigma}$, thus $\zeta' = x_{i\cdot 1}$ works. The other cases for ζ are straightforward.

If $\phi = [\zeta = \eta]$ and $\sigma \models \phi$, i.e. $\widehat{\zeta\sigma} = \widehat{\eta\sigma}$. Thus $\overline{\widehat{\zeta\sigma}} = \overline{\widehat{\eta\sigma}}$, thus $\overline{\widehat{\zeta\sigma}}\theta = \overline{\widehat{\eta\sigma}}\theta$, thus, by Lemma 4, $\overline{\widehat{\zeta\sigma'}} = \overline{\widehat{\eta\sigma'}}$. By construction of $\bar{\cdot}^{\sigma}$, $\bar{t}^{\sigma} = \overline{t}^{\sigma}$ iff $t = t'$, thus $\widehat{\zeta\sigma'} = \widehat{\eta\sigma'}$, i.e. $\sigma' \models \phi$.

$|\phi| = n + 1$ The only non obvious case is when $\phi = \mathsf{let}\ z = \zeta\ \mathsf{in}\ \phi'$, $\sigma \models \phi$. Let $\sigma_1 = \sigma[\widehat{\zeta\sigma}/z]$ and $\sigma_2 = \sigma'[\widehat{\zeta\sigma'}/z]$, $\sigma \models \phi$ iff $\sigma_1 \models \phi'$ and $\sigma' \models \phi$ iff $\sigma_2 \models \phi'$. Using the induction hypothesis, it remains us to show that $\sigma_1 \sim' \sigma_2$. Since, $kn(\sigma_1) = kn(\sigma)$ and $kn(\sigma_2) = kn(\sigma')$, it is sufficient to prove that $\overline{\widehat{\zeta\sigma}}^{\sigma}\theta = \overline{\widehat{\zeta\sigma'}}^{\sigma'}$, which is ensured by Lemma 4.

For decidability of $\sim$, it remains us to prove that $A^i_\sigma = A^i_{\sigma'}$ is decidable:

Proposition 11. *Let σ be a substitution, let $\mathcal{P}$ be the set of the paths of $\overline{\sigma}$. Let P be a predicate such that its interpretation is a regular language (i.e. recognized by a tree automata $\mathcal{A}_P$), then $A_\sigma \stackrel{def}{=} \{\zeta \in \mathcal{T}(\mathcal{VF} \cup \{x_p | p \in \mathcal{P}\}) \mid [\![P(\zeta[\sigma|_p/x_p]_{p\in\mathcal{P}})]\!] = tt\}$ is a regular language.*

Proof. We construct the automata $\mathcal{A}$ recognizing A_σ on the following way:
- the rules of $\mathcal{A}_P$ are rules of $\mathcal{A}$,
- for each $p \in P$, if $\sigma|_p$ is accepted in a state q in $\mathcal{A}_P$, then $x_p \to q$ is a rule of $\mathcal{A}_\sigma$.

It is easy to verify that the language $\mathcal{L}(\mathcal{A})$ recognized by $\mathcal{A}$ is equal to A_σ.

B Completeness of weak bisimilarity

B.1 Characteristic Processes

We have constructed in Section 4.1 a formula ϕ_σ:

$$\phi_\sigma \stackrel{\text{def}}{=} \bigwedge_{p \in leaves(\overline{\sigma}^\sigma)} \phi_p \wedge [f^{-1}(\zeta_p) = \bot] \wedge \epsilon_\sigma[\mathsf{dec}_{\zeta_p}(\{\zeta_p\}_{\zeta_p}) = \bot] \wedge \bigwedge_{p' \in leaves(\overline{\sigma}^\sigma)} \epsilon_\sigma^{p,p'}[\zeta_p = \zeta_{p'}].$$

Proposition 12 (Proposition 3). *Let σ and σ' be two finite substitutions such that $dom(\sigma) = dom(\sigma')$. If $\sigma' \models \phi_\sigma$ than there exists a substitution $\theta : Var_\sigma \to \mathcal{T}(\mathcal{F} \cup Var_{\sigma'})$ such that*

1. *$\overline{\sigma'}^{\sigma'} = \overline{\sigma}^\sigma \theta$,*
2. *that θ respects the symmetry: if $\theta(X_t) = X_{t'}$ then t is symmetric iff t' is symmetric, if $\theta(X_t) = f(\ldots)$ then t is symmetric iff $f \neq \mathsf{pub}$ and $f \neq \mathsf{prv}$.*

Let σ and σ' be two substitutions such that $dom(\sigma) = dom(\sigma')$ and $\sigma' \models \phi_\sigma$. $\sigma' \models \phi_\sigma$, thus, for every path p leading to a leaf in $\overline{\sigma}$, $\sigma' \models \phi_p$. By construction of ϕ_p and ζ_p (constructed according to Proposition 2), it holds that p is a path in $\overline{\sigma'}^{\sigma'}$ and $s_\sigma(p) = s_{\sigma'}(p)$. Let θ such that $\theta(\overline{\sigma}|_p) = \overline{\sigma'}|_p$ for all path p leading to a leaf in $\overline{\sigma}$. Since $\sigma' \models \phi_\sigma$, we have $\sigma|_p = \sigma|_{p'} \Rightarrow \sigma'|_p = \sigma'|_{p'}$, which ensures that θ is a function. Conversely, $\sigma|_p \neq \sigma|_{p'} \Rightarrow \sigma'|_p \neq \sigma'|_{p'}$, which ensures that θ is a substitution: $\overline{\sigma'}^{\sigma'} = \overline{\sigma}^\sigma \theta$.
Let us show that θ preserves the symmetry: If $\theta(X_t) = X_{t'}$ ($t = \sigma|_p = \widehat{\zeta_p \sigma}$ and $t' = \sigma'|_p = \widehat{\zeta_p \sigma'}$ for some p) then t is symmetric if and only if $[\![[\mathsf{dec}_{\zeta_p}(\{\zeta_p\}_{\zeta_p}) = \bot]\sigma]\!] = \mathit{ff}$, thus t is symmetric iff t' is symmetric. Same thing if $\theta(X_t) = f(\ldots)$. QED.

Using Proposition 3, it is easy to verify that ϕ_σ^{char} satisfies the following property:

Proposition 13 (Proposition 4). *Let σ, σ' be two finite substitutions such that $dom(\sigma) = dom(\sigma')$. We have that $\sigma \sim \sigma'$ if and only if $\sigma' \models \phi_\sigma^{char}$ and for every $1 \leq i \leq n$, $A_\sigma^i = A_{\sigma'}^i$.*

We are now ready to prove Theorem 2.

Theorem 6 (Theorem 2). *Let σ be a finite substitution, there exist R_σ^f and ϕ_σ such that for every substitution σ' such that $dom(\sigma') = dom(\sigma)$, $\sigma \sim \sigma'$ if and only if $\sigma' \models \phi_\sigma$ and $R_\sigma^f \sigma' \not\Downarrow f$.*

Let $\sigma = [M_i/x_i]_{i\leq n}$ be a finite substitution.
First step : we first show that, for each p leading to a leaf in $\overline{\sigma}$, there exists a process $R^f_{\zeta_p}$ such that for every substitution σ' such that $dom(\sigma') = dom(\sigma)$, $R^f_{\zeta_p}\sigma'$ commits to f if and only if the predicate $vis(\sigma', \zeta_p)$ or $enc(\sigma', \zeta_p)$ holds. Let

$$\begin{array}{rll} R^f_\zeta \stackrel{\text{def}}{=} & (!\bar{a}(x_1) \mid \ldots \mid !\bar{a}(x_n) & \\ & \mid a(y_1).\ldots.a(y_m).!\bar{a}(f(y_1,\ldots,y_m)) & \forall f \in \mathcal{VF} \\ & \mid a(x).a(y).!\bar{a}(\mathsf{dec}_x(y)) & \\ & \mid a(x).!\bar{a}(g_i^{-1}(x)) & \forall g \in \mathcal{IF}, \forall i \leq arity(g) \\ & \mid a(y_1).\ldots.a(y_m).[f(y_1,\ldots,y_m) = \zeta]\bar{f}(\zeta) & \forall f \in \mathcal{VF} \\ & \mid a(x).a(y).[\mathsf{dec}_x(y) = \zeta]\bar{f}(\zeta)) & \end{array}$$

Intuitively, $R^f_\zeta\sigma'$ sends every term of σ' arbitrary many times (the term M_i such that $M_i/x_i \in \sigma'$). Then $R^f_\zeta\sigma'$ computes every term M such that $M = \widehat{\zeta'\sigma'}$ for some ζ'. Eventually, whenever $R^f_\zeta\sigma$ finds $M_1,\ldots,M_m$ ($M_i = \widehat{\zeta_i\sigma'}$) such that $\widehat{\zeta\sigma'} = f(M_1,\ldots,M_m)$ or $\widehat{\zeta\sigma'} = \mathsf{dec}_{M_1}(M_2)$, then $R^f_\zeta\sigma'$ commits to f. More formally:

Lemma 5. *Let σ' be a substitution such that $dom(\sigma') = dom(\sigma)$. Then:*

- *there exists μ such that $R^f_\zeta \stackrel{\mu}{\Rightarrow} R' \xrightarrow{\bar{a}<M>} R''$ if and only if there exists ζ such that $\widehat{\zeta\sigma} = M$.*
- *$R^f_\zeta \stackrel{\mu}{\Rightarrow} R' \xrightarrow{\bar{f}<M>} R''$ for some M and μ if and only if one of the predicates $vis(\sigma', \zeta_p)$ or $enc(\sigma', \zeta_p)$ holds.*

An induction on ζ on one hand and an induction of the length of μ on the other hand prove the first part of this lemma. The second part is then straightforward.
Second step : It remains to build a process R^i_σ such that $R^i_\sigma\sigma'$ commits to f if and only if $A^i_\sigma \neq A^i_{\sigma'}$. We assume that for each path p of $\overline{\sigma}$, we have constructed ζ_p and ϕ_p following Proposition 2. Using the rules of the tree automata recognizing A^i_σ, we construct a process $R^{i,1}_\sigma$ such that $R^{i,1}_\sigma$ commits to f if and only if there exists $\zeta \in A^i_\sigma$ such that $\zeta \notin A^i_{\sigma'}$: Let $Q^{i,1}_\sigma$ be the set of states of the automata recognizing A^i_σ and $T^{i,1}_\sigma$ its set of rules. For each $p \in \mathcal{P}$, for each q state of the automata recognizing A^i_σ such that σ_p is accepted in q, let $P_{p,q} = !(\bar{a} < \zeta_p, q >)$. For each rule $rl : f(q_1,\ldots,q_n) \to q$ of A^i_σ, let

$$P_{rl} \stackrel{\text{def}}{=} !(a(x_1).\,\mathsf{let}\, x_1 =< x'_1, q_1 > \,\mathsf{in}\, (a(x_2).\,\mathsf{let}\, x_2 =< x'_2, q_2 > \,\mathsf{in}\, \cdots \bar{a} < f(x_1,\ldots,x_n), q >)$$

where $\mathsf{let}\, x_1 =< x'_1, q_1 > \,\mathsf{in}\, R$ stands for $[q_1 =<>_2^{-1} x_1]\,\mathsf{let}\, x'_1 =<>_1^{-1} x_1 \,\mathsf{in}\, R$. P_{rl} simulates the rule rl of the automata. It remains to check if the terms accepted in q_f satisfy the predicate P_i: let

$$P_{q_f} \stackrel{\text{def}}{=} !(a(x).\,\mathsf{let}\, x =< x', q_f > \,\mathsf{in}\, \neg P_i(x')\bar{f}).$$

Then

$$R^{i,1}_\sigma \stackrel{\text{def}}{=} (\nu a)\left(\Big|_{\substack{p\in\mathcal{P}\\ q\in Q^{i,1}_\sigma}} P_{p,q}\,|_{rl\in T^{i,1}_\sigma} P_{rl} \,|\, \left(\bigwedge_{p\in\mathcal{P}} \phi_p\right) P_{q_f}\right).$$

Conversely, using the rules of the tree automata recognizing $(A^i_\sigma)^c$, we construct a process $R^{i,2}_\sigma$ such that $R^{i,2}_\sigma$ commits to f if and only if there exists $\zeta \notin A^i_\sigma$ such that $\zeta \in A^i_{\sigma'}$: Let $Q^{i,2}_\sigma$ be the set of states of the automata recognizing $(A^i_\sigma)^c$ and $T^{i,2}_{\sigma'}$ its set of rules. We only have to change the process which simulates the final rule: let

$$P'_{q_f} \stackrel{\text{def}}{=} !(a(x).\,\mathsf{let}\, x =< x', q_f > \,\mathsf{in}\, P_i(x')\bar{f}).$$

Then

$$R^{i,2}_\sigma \stackrel{\text{def}}{=} (\nu a)\left(\Big|_{\substack{p\in\mathcal{P}\\ q\in Q^{i,2}_\sigma}} P_{p,q}\,|_{rl\in T^{i,2}_\sigma} P_{rl} \,|\, \left(\bigwedge_{p\in\mathcal{P}} \phi_p\right) P'_{q_f}\right).$$

Then, one can verify (by induction) that $R^{i,1}_\sigma$ and $R^{i,2}_\sigma$ verify the required property:

Lemma 6. *For every substitution σ',*

- *$R^{i,1}_\sigma\sigma' \Downarrow f$ if and only if there exists $\zeta \in A^i_\sigma$ such that $P_i(\zeta[\zeta_p/x_p]_{p\in\mathcal{P}}\sigma') = ff$ and*
- *$R^{i,2}_\sigma\sigma' \Downarrow f$ if and only if there exists $\zeta \notin A^i_\sigma$ such that $P_i(\zeta[\zeta_p/x_p]_{p\in\mathcal{P}}\sigma') = tt$.*

Thus $A^i_\sigma = A^i_{\sigma'}$ if and only if $R^{i,1}_\sigma\sigma' \not\Downarrow f$ and $R^{i,2}_\sigma\sigma' \not\Downarrow f$.

Let $R^f_\sigma \stackrel{\text{def}}{=} |_{p\in leaves(\overline{\sigma})} (\nu a) R^f_{\zeta_p} |_{1\le i\le n} R^{i,1}_\sigma | R^{i,2}_\sigma$. Then, using Proposition 4 and Lemmas 5 and 6, R^f_σ satisfies the requirements of Theorem 2.

B.2 Completeness

Theorem 7 (Theorem 4). *Let P and Q be structurally image-finite processes and σ_1 and σ_2 be equivalent, finite substitutions. If $(\sigma_1, \sigma_2) \vdash P \cong Q$, then $(\sigma_1, \sigma_2) \vdash P \approx Q$.*

We rewrite here the proof given in [7], adapted to our extended syntax. In general, we won't rewrite any definition or notation but those which change compared to [7]. Consequently, this proof won't be really understandable without the proof given in [7].

We use the chain of relations $\approx_i$, $i \ge 0$ defined in [7] to approximate bisimilarity $\approx$ over configurations and we define $rk(\sigma) = |\phi_\sigma|$, where $|E|$ denotes the syntactic size of some term E. We let $\approx_w \stackrel{\text{def}}{=} \cap_{i\ge 0} \approx_i$.

Lemmas 4.20 and 4.22 of [7] still holds:

Lemma 7 (Lemma 4.20). *Let P and Q be structurally image-finite processes. Then $(\sigma_1, \sigma_2) \vdash P \approx Q$ if and only if $(\sigma_1, \sigma_2) \vdash P \approx_w Q$.*

Lemma 8 (Lemma 4.22). *If $\sigma \sim \sigma'$ then $rk(\sigma) = rk(\sigma')$.*

To prove Lemma 8, we prove a stronger property: if $\sigma \sim \sigma'$ then $\phi_\sigma = \phi_{\sigma'}$, which is true by construction of ϕ_σ.

We show now that $\cong$ implies $\approx_w$, from which completeness of $\approx$ for structurally image-finite processes will follow. We redefine a class of canonical contexts $R_{i,\sigma}$, depending on some $i \geq 0$ and σ, which can be used to configurations are related by $\approx_i$.

Definition 9 (canonical contexts). *Define $R_{i,\sigma}$, for $i \geq 0$, by induction on i as follows. $R_{0,\sigma} \stackrel{def}{=} \mathbf{0}$ and, for $i > 0$:*
$R_{n,\sigma} \stackrel{def}{=} \sum\{R_\eta^{inp} + R_\eta^{out} : n(\eta) \subseteq dom(\sigma), \widehat{\eta\sigma} \in \mathcal{N} \text{ and } |\eta| < i\} + R_\epsilon + \bar{e}_i$, *where:*
$R_\eta^{inp} \stackrel{def}{=} \eta(x).\sum\{\phi_{\sigma'}(R_{\sigma'}^{h_{\eta,\sigma',i}} \mid \bar{f}_{\eta,\phi_{\sigma'},i} + \tau.R_{i-1,\sigma'}) :$
there is M s.t. $\sigma' = \sigma[M/x]$, and $rk(\sigma') \geq i\}$
$R_\eta^{out} \stackrel{def}{=} \sum\{(\nu\tilde{b})\overline{\eta}(\zeta).(\bar{g}_{\eta,\zeta,i} + \tau.R_{i-1,\sigma'}) : fn((\nu\tilde{b})\overline{\eta}(\zeta)) \subseteq dom(\sigma), \sigma' = \sigma[\tilde{b}/\tilde{b}] \text{ and } |\zeta| < i\}$
$R_\epsilon \stackrel{def}{=} \tau.(\bar{h}_i) + \tau.R_{i-1,\sigma}$

Note, that, exactly like in [7], the sum R_η^{inp} is finite, because, for fixed $\tilde{x}$ and i, there are finitely many phi_σ and $R_{i,\sigma}$ s.t. $dom(\sigma) = \tilde{x}$ and $rk(\sigma) \leq i$ (this is formally proven by induction on i). The sum R_η^{out} is finite as well, because actions $(\nu\tilde{b})\overline{\eta}(\zeta)$ are considered up to alpha-equivalence. Note also that $fn(R_{i,\sigma}) \subseteq dom(\sigma) \cup \tilde{l}$, where $\tilde{l}$ is the set of all names $e_j, h_{\eta,\sigma',j}, f_{\eta,\phi_{\sigma'},j}, g_{\eta,\zeta,j}$ and h_j $(0 \leq j \leq i)$ occurring in $R_{i,\sigma}$.

Assume $(\sigma_1, \sigma_2) \vdash P \cong Q$, let us prove that $(\sigma_1, \sigma_2) \vdash P \approx Q$. By virtue of Lemma 4.20, it is sufficient to prove that for each $i \geq 0$ it holds that $(\sigma_1, \sigma_2) \vdash P \approx_i Q$. Consider R_{i,σ_1} and let $\rho_1 = \sigma_1[\tilde{l}/\tilde{l}]$, where $\tilde{l}$ is the set of fresh names $e_j, h_{\eta,\sigma',j}, f_{\eta,\phi_{\sigma'},j}, g_{\eta,\zeta,j}$ and h_j $(0 \leq j \leq i)$ occurring in R_{i,σ_1}. We prove that if $(\rho_1, \rho_2) \vdash P|R_{i,\sigma_1}\sigma_1 \dot{\cong} Q|R_{i,\sigma_1}\sigma_2$, then $(\rho_1, \rho_2) \vdash P \approx_i Q$.

We proceed by induction on i. The case $i = 0$ is trivial, thus suppose $i > 0$. We only consider the case of a (C-INP) transition, that is:

$$\sigma_1 \triangleright P \stackrel{(\nu\bar{b})\bar{a}<M>}{\underset{\eta(x)}{\longmapsto}} \sigma_1' \triangleright P' \tag{6}$$

with $|\eta(x)| \leq i, \sigma_1' = \sigma_1[M/x]$ and $rk(\sigma 1') \leq i$, as the other cases are similar or easier. We show the existence of a transition of $\sigma_2 \triangleright Q$ that matches this one. From (6) above, we can infer that:

$$P \mid R_{i,\sigma_1}\sigma_1 \xrightarrow{\tau} (\nu\tilde{b})(P' \mid \phi_{\sigma_1'}(R_{\sigma_1'}^{h_{\eta,\sigma_1',i}} \mid \bar{f}_{\eta,\phi_{\sigma_1'},i} + \tau.R_{i-1,\sigma_1'})\sigma_1') \stackrel{\text{def}}{=} A.$$

Since $(\rho_1, \rho_2) \vdash P|R_{i,\sigma_1}\sigma_1 \dot{\cong} Q|R_{i,\sigma_1}\sigma_2$, and $A \downarrow f_{\eta,\phi_{\sigma'_1},i}$ we deduce the existence of a transition

$$Q \,|\, R_{i,\sigma_1}\sigma_2 \Rightarrow (\nu\tilde{b})(Q' \,|\, \phi_{\sigma'_1}(R_{\sigma'_1}^{h_{\eta,\sigma'^{1},i}}\sigma'_2 \,|\, \bar{f}_{\eta,\phi_{\sigma'_1},i} + \tau.R_{i-1,\sigma'_1})\sigma'_2) \stackrel{\text{def}}{=} B.$$

with $(\rho_1, \rho_2) \vdash A \dot{\cong} B$ and $B \downarrow f_{\eta,\phi_{\sigma'_1},i}$, where $\sigma'_2 = \sigma_2[M'/x]$ s.t. $Q \xRightarrow{(\nu\tilde{b'})\overline{a'}<M'>} Q'$ $(a' = \widehat{\eta\sigma_2})$. Hence

$$\sigma_2 \triangleright Q \xRightarrow[\eta(x)]{(\nu\tilde{b'})\overline{a'}<M'>} \sigma'_2 \triangleright Q'.$$

Moreover, since $B \downarrow f_{\eta,\phi_{\sigma'_1},i}$, it holds that $\sigma'_2 \models \phi_{\sigma'_1}$. In addition, since $R_{\sigma'_1}^{h_{\eta,\sigma'^{1},i}}\sigma'_1 \not\Downarrow h_{\eta,\sigma'_1,i}$ and $(\rho_1, \rho_2) \vdash A \dot{\cong} B$, we have $R_{\sigma'_1}^{h_{\eta,\sigma'^{1},i}}\sigma'_2 \not\Downarrow h_{\eta,\sigma'_1,i}$ thus, applying Theorem 2, $\sigma'_1 \sim \sigma'_2$.

The proof ends exactly like in [7].

C Applications

C.1 Properties on Weak Bisimilarity

We prove here the two properties regarding σ-safety.

Proposition 14 (Proposition 7). *Suppose that $(\sigma_1, \sigma_2) \vdash Q_1 \approx Q_2$ (resp. $(\sigma_1, \sigma_2) \vdash Q_1 \approx' Q_2$) and that $(\sigma_1, \sigma_2) \vdash R_1 \approx R_2$ (resp. $(\sigma_1, \sigma_2) \vdash R_1 \approx R_2$). Suppose that, for $i = 1, 2$, Q_i and R_i are σ_i-safe (resp. σ_i-safe2). Then $(\sigma_1, \sigma_2) \vdash Q_1|R_1 \approx Q_2|R_2$ (resp. $(\sigma_1, \sigma_2) \vdash Q_1|R_1 \approx' Q_2|R_2$).*

We prove Proposition 7 only for $\approx$, since the proof is exactly the same for $\approx'$. This proof is quite close to the one given in [7].

Consider the relation $\mathcal{R}$ defined by: $(\sigma_1, \sigma_2) \vdash (Q_1 \,|\, R_1)\mathcal{R}(Q_2 \,|\, R_2)$ if:

1. $(\sigma_1, \sigma_2) \vdash Q_1 \approx Q_2$ and $(\sigma_1, \sigma_2) \vdash R_1 \approx R_2$, and
2. for $i = 1, 2$, Q_i and R_i are σ_i-safe.

One shows that $\mathcal{R}$ is a weak bisimulation. Suppose that $(\sigma_1, \sigma_2) \vdash (Q_1 \,|\, R_1)\mathcal{R}(Q_2 \,|\, R_2)$. We only examine the most delicate case, that is when Q_1 and R_1 interact with one another. Thus, suppose that

$$\sigma_1 \triangleright Q_1 \,|\, R_1 \xrightarrow[-]{\tau} \sigma_1 \triangleright (\nu\tilde{h})(Q'_1 \,|\, R'_1) \tag{7}$$

where $Q_1 \xrightarrow{(\nu\tilde{h})\bar{a}<M>} Q'_1$ and $R_1 \xrightarrow{aM} R'_1$ (the others cases follow by symmetry). Since Q_1 is σ_1-safe, $a, M \in kn(\sigma_1)$, thus $a = \widehat{\eta\sigma_1}$ for some η (Lemma 9). Thus $\tilde{h}$ and $\sigma_1 \triangleright Q_1 \xrightarrow[\eta(x)]{(\nu\tilde{h})\bar{a}<M>} \sigma'_1 \triangleright Q'_1$. Since $(\sigma_1, \sigma_2) \vdash Q_1 \approx Q_2$, we get that $\sigma_2 \triangleright Q_2 \xrightarrow[\eta(x)]{(\nu\tilde{k})\bar{a'}<M'>} \sigma'_2 \triangleright Q'_2$, for some a' $(a = \widehat{\eta\sigma_2})$, M' and Q'_2, such that $(\sigma'_1, \sigma'_2) \vdash Q'_1 \approx Q'_2$ (here $\sigma'_1 \stackrel{\text{def}}{=} \sigma_1[M/x]$) and $\sigma'_2 \stackrel{\text{def}}{=} \sigma_2[M'/x]$, for a fresh x). Also in this case, $\tilde{k} = \emptyset$ since $M' \in kn(\sigma_2)$.

Now, since $M \in kn(\sigma_1)$, there is ζ s. t. $n(\zeta) \subseteq dom(\sigma_1)$ and $M = \widehat{\zeta\sigma_1}$ (Proposition 9). Since $M \in kn(\sigma_1)$, we get $kn(\sigma_1') = kn(\sigma_1)$, thus the application $_^{\sigma_1'}$ is equal to $_^{\sigma_1}$. Similarly, $_^{\sigma_2'} = _^{\sigma_2}$. Since $\sigma_1' \sim \sigma_2'$, we have $\overline{M}^{\sigma_1'}\theta = \overline{M'}^{\sigma_2'}$ for some substitution θ. Thus $\overline{M}^{\sigma_1}\theta(= \overline{M}^{\sigma_1'}\theta) = \overline{M'}^{\sigma_2}$. In the other hand, applying Corollary 1, $\overline{M}^{\sigma_1}\theta = \widehat{\zeta\sigma_1}^{\sigma_1}\theta = \widehat{\zeta\sigma_2}^{\sigma_2}$. Since the application $_^{\sigma_2}$ is injective, we get $M' = \widehat{\zeta\sigma_2}$.

Therefore, from $R_1 \xrightarrow{aM} R_1'$, we deduce $\sigma_1 \triangleright R_1 \xmapsto[\overline{\eta}\zeta]{aM} \sigma_1 \triangleright R_1'$; hence, from $(\sigma_1, \sigma_2) \vdash R_1 \approx R_2$, we get $\sigma_2 \triangleright R_2 \xmapsto[\overline{\eta}\zeta]{a'M'} \sigma_2 \triangleright R_2'$, where $(\sigma_1, \sigma_2) \vdash R_1' \approx R_2'$. Combining $Q_2 \xRightarrow{\bar{a'}<M'>} Q_2'$ and $R_2 \xRightarrow{a'M'} R_2'$, we obtain $Q_2 \,|\, R_2 \Rightarrow Q_2' \,|\, R_2'$, hence

$$\sigma_2 \triangleright Q_2 \,|\, R_2 \xmapsto[-]{\tau} \sigma_2 \triangleright (Q_2' \,|\, R_2').$$

To see that the above weak transition matches 7, let us check conditions 1 and 2 of the definition of $\mathcal{R}$. Indeed, $(\sigma_1, \sigma_2) \vdash Q_1' \approx Q_2'$ (by $(\sigma_1', \sigma_2') \vdash Q_1' \approx Q_2'$ and weakening) and $(\sigma_1, \sigma_2) \vdash R_1' \approx R_2'$ imply condition 1; moreover, for $i = 1, 2$, R_i' and Q_i' are σ_i-safe, because R_i and Q_i are, thus condition 2 is true.

Remark: With the definition of σ-safety given in [7], this Proposition do not hold: let us recall the definition of σ-safety given in [7].

A process P is σ-safe if for each s, if $\sigma \triangleright R \xRightarrow[u]{s} \sigma' \triangleright R' \xmapsto[\eta(x)]{(\nu \bar{b}\bar{a}<M>)}$, then $M \in kn(\sigma)$.

Intuitively, a process R is σ-safe for this new definition, if whenever a process can send a message M on a channel known by the intruder, then M is already in the knowledge of the intruder. But this definition does not require that a process P can only commit to channel a such that a is already in the knowledge of the intruder.

Let us show that this definition is not sufficient to ensure Proposition 7.

Let $\sigma_1 \stackrel{\text{def}}{=} \sigma_2 \stackrel{\text{def}}{=} [p/p]$. ($\sigma_1$ and σ_2 only know the public channel p.)

Let $Q_1 \stackrel{\text{def}}{=} \bar{a} < a > .\bar{p} < a > .\mathbf{0}$: Q_1 sends first something on channel a and then sends a secret s on the public channel p. Let $Q_2 \stackrel{\text{def}}{=} \mathbf{0}$, $R_1 \stackrel{\text{def}}{=} R_2 \stackrel{\text{def}}{=} a(x).\mathbf{0}$.

Then, clearly, R_1 is σ_1-safe, Q_2, R_2 are σ_2-safe and $(\sigma_1, \sigma_2) \vdash R_1 \approx R_2$. In addition there is no possible moves for $\sigma_1 \triangleright P_1$, thus Q_1 is σ_1-safe and $(\sigma_1, \sigma_2) \vdash P_1 \approx P_2$.

But, unfortunately, we have not that $(\sigma_1, \sigma_2) \vdash Q_1|R_1 \approx Q_2|R_2$.

Proposition 15 (Proposition 8). *Let σ and σ' be two equivalent substitutions such that the process P is σ and σ'-safe (resp. σ and σ'-safe2). Then $(\sigma, \sigma') \vdash P \approx P$ (resp. $(\sigma, \sigma') \vdash P \approx' P$).*

Again, we just prove this proposition for $\approx$ since the proof is exactly the same for $\approx'$.

Consider the relation $\mathcal{R}$ defined by: $(\sigma_1, \sigma_2) \vdash P \mathcal{R} Q$ if and only if $\sigma_1 \sim \sigma_2$ and there exists a process A s. t. A is σ_1 and σ_2-safe and $P = A[\widehat{\zeta_i\sigma_1}/x_i]_{i \in J}$ and $Q = A[\widehat{\zeta_i\sigma_1}/x_i]_{i \in J}$.

Let us show that $\mathcal{R}$ is a weak bisimulation. This relies on the following lifting lemma:

Lemma 9. *Let σ_1 and σ_2 be two equivalent substitutions and let A be a process σ_1 and σ_2-safe. Then property $(*)$ holds: whenever $A[\widehat{\zeta_i\sigma_1}/x_i]_{i\in J} \xrightarrow{\mu\sigma_1} A'$ for some μ s. t. $fn(\mu) \subseteq fn(A) \cup \tilde{x_i}$, then there exist A'' s. t. $A' = A''[\widehat{\zeta_i\sigma_1}/x_i]_{i\in J}$ and $A[\widehat{\zeta_i\sigma_2}/x_i]_{i\in J} \xrightarrow{\mu\sigma_2} A''[\widehat{\zeta_i\sigma_2}/x_i]_{i\in J}$*

Let $\sigma'_1 = [\widehat{\zeta_i\sigma_1}/x_i]_{i\in J}$ and $\sigma'_2 = [\widehat{\zeta_i\sigma_2}/x_i]_{i\in J}$. We prove property $(*)$ by induction on the operational semantics.
(INP): $A = \eta(x).A'$ and $A\sigma'_1 \xrightarrow{aM} A'\sigma'_1$ such that $a = \widehat{\eta\sigma'_1}$ $(= \widehat{\eta'\sigma_1}$ by construction of $\sigma'_1)$. By induction hypothesis, there exists ζ such that $M = \widehat{\zeta\sigma_1}$. Let $a' = \widehat{\eta\sigma'_2}$ $(= \widehat{\eta'\sigma_2})$, $a' \neq \bot$ since $\sigma_1 \sim \sigma_2$ and let $M' = \zeta\sigma_2$. Then $A\sigma'_2 \xrightarrow{a'M'} A'\sigma'_2$, thus property $(*)$ is verified.
(OUT) is an easier case.
Let us sketch that it is preserved by the other operational semantics. The only non trivial cases are (GUARD) and (LET).
Case (GUARD) uses $[\![\phi\sigma_1]\!] = tt$ iff $[\![\phi\sigma_2]\!] = tt$ since σ_1 and σ_2 are equivalent.
Case (LET) uses that $\widehat{\zeta\sigma_1} \neq \bot$ if and only if $\widehat{\zeta\sigma_2} \neq \bot$ since σ_1 and σ_2 are equivalent.

C.2 Proofs of Needham-Schroeder-Lowe Protocol

We prove here the technical lemmas used for the proof of the secrecy of Needham-Schroeder-Lowe protocol.
Consider the two environments $\sigma'_{M,\tilde{h}}$ and $\sigma'_{M',\tilde{k}}$. Then the private keys of a and b ($\mathsf{prv}(a)$ and $\mathsf{prv}(b)$) are not in the knowledge of $\sigma'_{M,\tilde{h}}$ and $\sigma'_{M',\tilde{k}}$. Indeed, $\mathsf{prv}(a)$ and $\mathsf{prv}(b)$ do not even appear as subterm of $\sigma'_{M,\tilde{h}}$ or $\sigma'_{M',\tilde{k}}$ and the function symbol prv is a private function symbol. Using that $\mathsf{prv}(a)$ and $\mathsf{prv}(b)$ are not in $\sigma'_{M,\tilde{h}}$ or $\sigma'_{M',\tilde{k}}$, we prove by induction on the definition of the knowledge closure that if t is in the knowledge of $\sigma'_{M,\tilde{h}}$ and if $M, n^{ab}_{1,m}, n^{ba}_{1,m}, n^{ab}_{2,m}$ or $n^{ba}_{2,m}$ appears as a subterm of t then $M, n^{ab}_{1,m}, n^{ba}_{1,m}, n^{ab}_{2,m}$ or $n^{ba}_{2,m}$ is encrypted by $\mathsf{pub}(a)$ or $\mathsf{pub}(b)$. More formally:

Proposition 16. *If there exists $t \in kn(\sigma'_{M,\tilde{h}})$ such that $t|_p \in \{M, n^{ab}_{1,m}, n^{ba}_{1,m}, n^{ab}_{2,m}, n^{ba}_{2,m}\}$, then $p = p'.1.i$ and $t_{p'} = \{N\}_{\mathsf{pub}(a)}$ or $t_{p'} = \{N\}_{\mathsf{pub}(b)}$ for some path p' and message N. The corresponding property holds for $\sigma'_{M',\tilde{k}}$.*

Consequently, $M, n^{ab}_{1,m}, n^{ba}_{1,m}, n^{ab}_{2,m}$ and $n^{ba}_{2,m}$ are neither in $kn(\sigma'_{M,\tilde{h}})$ nor in $kn(\sigma'_{M,\tilde{h}})$.
Let $\sigma'_{M,\tilde{h}} \stackrel{\text{def}}{=} [M_i/x_i]_{i\in J}$ and $\sigma'_{M',\tilde{h}} \stackrel{\text{def}}{=} [M'_i/x_i]_{i\in J}$. We rename M_i and M'_i such that for every i, there exists t_i such that $M_i = t_i(M)$ and $M'_i = t_i(M')$ or $M_i = h_j$ and $M'_i = k_j$. We deduce from above that $\overline{\sigma'_{M,\tilde{h}}}^{\sigma'_{M,\tilde{h}}} = [X_{M_i}/x_i]_{i\in J}$ and $\overline{\sigma'_{M',\tilde{k}}}^{\sigma'_{M',\tilde{k}}} = [X_{M'_i}/x_i]_{i\in J}$. Lemma 2 is now straightforward.

Lemma 10 (Lemma 2). $\sigma'_{M,\bar{h}} \sim \sigma'_{M',\bar{k}}$.

Indeed, the renaming function θ defined by $\theta(X_{t_i(M)}) = X_{t_i(M')}$ and $\theta(X_{h_i}) = X_{k_i}$ satisfies the requirements of Definition 5.

The following property will be used to prove Lemma 1:

Proposition 17. *Let $\{M\}_k$ be an encrypted message such that $\{M\}_k \in kn(\sigma)$, where $\sigma = \sigma'_{M,\bar{h}}$ or $\sigma'_{M',\bar{k}}$ (by Lemma 4), then either $\{M\}_k \in \sigma$ or $M \in kn(\sigma)$.*

Indeed, since $\{M\}_k \in kn(\sigma)$, $\{M\}_k = \widehat{\zeta\sigma}$ for some ζ, thus $\overline{\{M\}_k} = \overline{\widehat{\zeta\sigma}} = \widehat{\zeta\overline{\sigma}}$. The case $\overline{\{M\}_k} = X_{\{M\}_k}$ corresponds to $\{M\}_k \in kn(\sigma)$ and case $\overline{\{M\}_k} = \{\overline{M}\}_{\overline{k}}$ or $\overline{\{M\}_k} = \{\overline{M}\}_{(\overline{k'})^{-1}}$ corresponds to $M \in kn(\sigma)$.

Lemma 11 (Lemma 1). *$\forall 1 \leq m \leq q, \forall A, B \in \{a, b\}, A \neq B$,*
$I_{A,B}(n^{AB}_{1,m})$, $I_{A,i}(n^{Ai}_{1,m})$, $R_{A,B}(n^{AB}_{2,m})$, $R_{A,i}(n^{Ai}_{2,m})$ are $\sigma'_{M,\bar{h}}$ and $\sigma'_{M',\bar{k}}$-safe.
Moreover, $R_{b,a}(M)$ is $\sigma'_{M,\bar{h}}$-safe and $R_{b,a}(M')$ is $\sigma'_{M',\bar{k}}$-safe.

We only prove that $I_{a,b}(n^{ab}_{1,m})$ and $I_{a,i}(n^{ai}_{1,m})$ are $\sigma'_{M,\bar{h}}$-safe since the other cases are similar or easier.
Suppose $\sigma'_{M,\bar{h}} \triangleright I_{a,b}(n^{ab}_{1,m}) \Longmapsto \xrightarrow{\bar{p}<N>}$. Let us show that $N \in kn(\sigma'_{M,\bar{h}})$. Either $N = \{n^{ab}_{1,m}, a\}_{\mathsf{pub}(b)}$, thus $N \in \sigma_{M,\bar{h}}$, which implies $N \in kn(\sigma'_{M,\bar{h}})$. Or N is of the form $N = \{N'\}_{\mathsf{pub}(b)}$ such that $\{n^{ab}_{1,m}, N', b\}_{\mathsf{pub}(a)} \in kn(\sigma'_{M,\bar{h}})$. Applying Proposition 17, either $\{n^{ab}_{1,m}, N', b\}_{\mathsf{pub}(a)} \in \sigma'_{M,\bar{h}}$ or $N' \in kn(\sigma'_{M,\bar{h}})$.

- $\{n^{ab}_{1,m}, N', b\}_{\mathsf{pub}(a)} \in \sigma'_{M,\bar{h}}$. Then, by construction of $\sigma'_{M,\bar{h}}$, $N' = n^{ba}_{2,m'}$ for some $1 \leq m' \leq q$. Since $\{n^{ba}_{2,m'}\}_{\mathsf{pub}(b)} \in \sigma_{M,\bar{h}}$, we get $N \in kn(\sigma'_{M,\bar{h}})$.
- $N' \in kn(\sigma'_{M,\bar{h}})$. Then $N = \{N'\}_{\mathsf{pub}(b)} \in kn(\sigma'_{M,\bar{h}})$.

Conclusion, in any case, $I_{a,b}(n^{ab}_{1,m})$ is $\sigma'_{M,\bar{h}}$-safe.

Suppose $\sigma'_{M,\bar{h}} \triangleright I_{a,i}(n^{ai}_{1,m}) \Longmapsto \xrightarrow{\bar{p}<N>}$. Let us show that $N \in kn(\sigma'_{M,\bar{h}})$. Either $N = \{n^{ai}_{1,m}, a\}_{\mathsf{pub}(i)}$, thus $N \in \sigma_{M,\bar{h}}$, which implies $N \in kn(\sigma'_{M,\bar{h}})$. Or N is of the form $N = \{N'\}_{\mathsf{pub}(i)}$ such that $\{n^{ai}_{1,m}, N', i\}_{\mathsf{pub}(a)} \in kn(\sigma'_{M,\bar{h}})$. Applying Proposition 17, either $\{n^{ai}_{1,m}, N', i\}_{\mathsf{pub}(a)} \in \sigma'_{M,\bar{h}}$ or $N' \in kn(\sigma'_{M,\bar{h}})$.

- $\{n^{ai}_{1,m}, N', i\}_{\mathsf{pub}(a)} \in \sigma'_{M,\bar{h}}$, Impossible by construction of $\sigma'_{M,\bar{h}}$.
- $N' \in kn(\sigma'_{M,\bar{h}})$. Then $N = \{N'\}_{\mathsf{pub}(i)} \in kn(\sigma'_{M,\bar{h}})$.

Conclusion, in any case, $I_{a,i}(n^{ai}_{1,m})$ is $\sigma'_{M,\bar{h}}$-safe.

It remains us to prove that $R_{b,a}(M)$ and $R_{b,a}(M')$ are bisimular resp. in $\sigma'_{M,\bar{h}}$ and $\sigma'_{M',\bar{k}}$:

Lemma 12 (Lemma 3). $(\sigma'_{M,\bar{h}}, \sigma'_{M',\bar{k}}) \vdash R_{b,a}(M) \approx' R_{b,a}(M')$.

The only possible moves for $\sigma'_{M,\bar{h}} \triangleright R_{b,a}(M)$ is

$$\sigma'_{M,\bar{h}} \triangleright R_{b,a}(M) \xrightarrow[\bar{p}<\zeta>]{p(N_1)} \sigma'_{M,\bar{h}} \triangleright R'_{b,a}(M)[N_1/x]$$

where $N_1 = \widehat{\zeta\sigma'_{M,\tilde{h}}}$ and
$R'_{b,a}(M) \stackrel{\text{def}}{=} \mathsf{let}\,(x_{n_a}, x_a) = \mathsf{dec}_{\mathsf{prv}(b)}(x)\,\mathsf{in}\,[x_a = a]\,\overline{p}(\{x_{n_a}, M, b\}_{\mathsf{pub}(a)}).\mathbf{0}$. Then

$$\sigma'_{M',\tilde{k}} \triangleright R_{b,a}(M') \xmapsto[\bar{p}<\zeta>]{p(N_2)} \sigma'_{M',\tilde{k}} \triangleright R'_{b,a}(M')[N_2/x]$$

where $N_2 = \widehat{\zeta\sigma'_{M',\tilde{k}}}$. Let us show that $(\sigma'_{M,\tilde{h}}, \sigma'_{M',\tilde{k}}) \vdash R'_{b,a}(M)[N_1/x] \approx' R'_{b,a}(M')[N_2/x]$. The possible moves for $\sigma'_{M,\tilde{h}} \triangleright R'_{b,a}(M)[N_1/x]$ is

$$\sigma'_{M,\tilde{h}} \triangleright R'_{b,a}(M)[N_1/x] \xmapsto[p(z)]{\bar{p}<\{N'_1,M,b\}_{\mathsf{pub}(a)}>} \sigma'_{M,\tilde{h}}[\{N'_1, M, b\}_{\mathsf{pub}(a)}/z] \triangleright \mathbf{0}$$

for some N'_1 such that $N_1 = \{N'_1, a\}_{\mathsf{pub}(b)}$. Let us consider $\overline{N_1}^{\sigma'_{M,\tilde{h}}}$. Since $N_1 = \widehat{\zeta\sigma'_{M,\tilde{h}}}$, applying Corollary 1, $\overline{N_1}^{\sigma'_{M,\tilde{h}}}\theta = \overline{N_2}^{\sigma'_{M',\tilde{k}}}$. We have to consider two cases:

- either $\overline{N_1}^{\sigma'_{M,\tilde{h}}} = X_{N_1}$, then $N_1 \in \sigma'_{M,\tilde{h}}$ thus $N'_1 = n^{ab}_{1,m}$ for some $1 \leq m \leq q$. We have $\overline{N_2}^{\sigma'_{M',\tilde{k}}} = X_{N_1}\theta = X_{\{n^{ab}_{1,m},a\}_{\mathsf{pub}(b)}}\theta = X_{\{n^{ab}_{1,m},a\}_{\mathsf{pub}(b)}}$. Thus $N_2 = \{n^{ab}_{1,m}, a\}_{\mathsf{pub}(b)}$ and

  $$\sigma'_{M',\tilde{k}} \triangleright R'_{b,a}(M')[N_2/x] \xmapsto[p(z)]{\bar{p}<\{n^{ab}_{1,m},M,b\}_{\mathsf{pub}(a)}>} \sigma'_{M',\tilde{k}}[\{n^{ab}_{1,m}, M, b\}_{\mathsf{pub}(a)}/z] \triangleright \mathbf{0}.$$

  Since $\sigma'_{M,\tilde{h}}[\{n^{ab}_{1,m}, M, b\}_{\mathsf{pub}(a)}/z] \sim \sigma'_{M',\tilde{k}}[\{n^{ab}_{1,m}, M, b\}_{\mathsf{pub}(a)}/z]$, $(\sigma'_{M,\tilde{h}}[\{n^{ab}_{1,m}, M, b\}_{\mathsf{pub}(a)}/z], \sigma'_{M',\tilde{k}}[\{n^{ab}_{1,m}, M, b\}_{\mathsf{pub}(a)}/z]) \vdash \mathbf{0} \approx' \mathbf{0}$ is obviously verified.
- or $\overline{N_1}^{\sigma'_{M,\tilde{h}}} = \{\overline{N'_1}, a\}_{\mathsf{pub}(a)}$. In this case $\overline{N_2}^{\sigma'_{M',\tilde{k}}} = \{\overline{N'_1}\theta, a\}_{\mathsf{pub}(b)}$, thus $N_2 = \{N'_2, a\}_{\mathsf{pub}(b)}$ for some N'_2 such that $\overline{N'_2} = \overline{N'_1}\theta$. Thus

  $$\sigma'_{M',\tilde{k}} \triangleright R'_{b,a}(M')[N_2/x] \xmapsto[p(z)]{\bar{p}<\{N'_2,M,b\}_{\mathsf{pub}(a)}>} \sigma'_{M',\tilde{k}}[\{N'_2, M, b\}_{\mathsf{pub}(a)}/z] \triangleright \mathbf{0}.$$

  Clearly $\sigma'_{M,\tilde{h}}[\{N'_1, M, b\}_{\mathsf{pub}(a)}/z] \sim \sigma'_{M',\tilde{k}}[\{N'_2, M, b\}_{\mathsf{pub}(a)}/z]$, thus $(\sigma'_{M,\tilde{h}}[\{N'_1, M, b\}_{\mathsf{pub}(a)}/z], \sigma'_{M',\tilde{k}}[\{N'_2, M, b\}_{\mathsf{pub}(a)}/z]) \vdash \mathbf{0} \approx' \mathbf{0}$ is obviously verified.